National College of Ireland

Efficacy of Deep Learning Model for Plant Disease Classification With Limited Data

MSc Research Project
Data Analytics

Saurabh Sharma
Student ID: x19239301

School of Computing
National College of Ireland

Supervisor: Dr. Bharathi Chakravarthi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Saurabh Sharma
Student ID:	x19239301
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Bharathi Chakravarthi
Submission Due Date:	31/01/2022
Project Title:	Efficacy of Deep Learning Model for Plant Disease Classification With Limited Data
Word Count:	7024
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Saurabh Sharma
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Efficacy of Deep Learning Model for Plant Disease Classification With Limited Data

Saurabh Sharma
x19239301

Abstract

Plants, like humans, are prone to a wide range of diseases, which have ramifications not just for human health but also for the country's economic development. Hence, it is essential to detect plant disease early to avoid such unfavourable effects. As per the research done in this field, deep learning models work well in classifying plant disease. However, it has a drawback: it needs a large amount of data to get trained. Plant leaf data collection is a tedious task and requires manual effort. This paper focuses on validating the efficacy of the deep learning model with a small amount of data set for plant classification. The plant disease dataset has three types of leaves: healthy, powdery and rust. It is seen from the research that a pre-trained MobileNetV2 model performs well with this limitation. The research uses ResNet50, a pre-trained MobileNetv2 and a hybrid CNN-RF for plant disease classification. Out of the three models, MobileNetV2 achieves an accuracy of over 95% and has a precision and recall value of 0.97. The model is also predicting unseen images correctly. The MobileNetV2 model is then converted as a TensorFlow Lite model, ready to be deployed in mobile for real-life prediction.

1 Introduction

Agriculture is one of the critical innovations that paved the way for sedentary human civilisation. It is the most significant step forward in human evolution. It is vital for moulding healthy diets and boosting human nutrition. As per (Max Roser and Ortiz-Ospina; 2013), the world population increases by 1.1% annually and hence, agriculture becomes one of the critical aspects to consider for fulfilling the ever-increasing demand for food. Plants, like humans, are subject to disease. For thousands of years, plant disease has significantly impacted food production and human social advancement. There have been countless examples of plant disease outbreaks in the past that have wreaked havoc on humanity. For instance, potato late blight and rice brown spot caused the Irish and Bengal famine in 1840 and 1943 (Bourke; 1964; Padmanabhan; 1973). Also, as the population grows, people are more focused on boosting food production and, hence, quality suffers. As a result, it is critical to diagnose plant disease as soon as possible to improve the crops' quality and volume and meet the demand for food.

Aside from that, agriculture adds to a country's economic prosperity. Agriculture, for example, accounts for over 19 per cent of India's total GDP and employs 60 per cent of the country's people directly or indirectly (Jethwani et al.; 2021). Plant disease, according to (Mutka and Bart; 2015), affects around 10% of global crop production. Plant disease

causes significant global loss, accounting for an average loss of 21.5% for wheat, 22.1% for maize, and 21.4% for soybean. It not only reduces agricultural production, but also causes the extinction of species' heterogeneity and harms human health. According to the United Nations, food production is expected to rise 60% to fuel the estimated 10 billion people by 2050 (Ristaino et al.; 2021). As a result, early plant disease identification becomes a critical goal in order to keep developing countries' GDP expanding and to meet the projected food demand in the near future.

Manual plant disease identification is a tedious and time-consuming task that requires the expertise of an agricultural expert. Thanks to improvements in artificial intelligence, several machine learning and deep learning algorithms have been developed to detect plant ailments early in the game.

A Deep Neural Network(DNN) is a Neural Network(ANN) that comprises several layers between the input and output layers and functions similar to human brains. Deep neural networks are widely employed in many diverse disciplines, such as computer vision, object detection, speech recognition, social network filtering, medical diagnosis, and self-driving cars. It is also utilized in agricultural areas to detect plant diseases early. Deep Learning methods such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are used to uncover hidden features in data. There are numerous advantages to using deep learning techniques for image prediction, including the fact that feature extraction is done automatically from untrained data; there is no need to include any specific methods for this. Secondly, the time required to process a massive amount of data is considerably shorter when compared to machine learning algorithms.

The major setback of the deep learning model is that it requires a large amount of data to train itself. In a real-life scenario, data collection of plant leaves is challenging and requires a lot of human effort. Most of the time, data provided for model building is not enough. What can be done in that case? Will deep learning be still effective with a small amount of data? This research focuses on finding a solution to this problem.

This research shows that deep learning can perform well even with a small dataset if transfer learning is used to train the data. In this research, a ResNet50 model is compared to a hybrid CNN-RF and a pre-trained MobileNetV2 model to show the effectiveness of transfer learning models. To the best of research done for this paper, the dataset used here is novel and has never been used in any study. A pre-trained MobileNetV2 is the clear winner among them. The model is then exported as TensorFlow lite and is ready to be deployed in mobile for real-life prediction. The TensorFlow lite model is compatible with both android and apple devices.

The remainder of the paper is carried out as follows. Related work on this issue is covered in Section 2, and Section 3 comprises the methodology, which illustrates all three models used in this research. The design specification, implementation, and results of each model are explained in Sections 4 and Section 5 respectively. Section 6.4 compares each model and evaluates their performances, and finally, the conclusion and future work come in Section 7.

1.1 Research Question

The research question for this paper is to verify how well the deep learning model performs with a limited set of data for plant disease classification. The dataset consist of three types of leaves; healthy, Powdery and Rust.

2 Related Work

Plant diseases have a negative impact on human health as well as the country's economic progress. Several attempts have been made to detect plant illnesses at the earliest by utilizing deep learning approaches such as Region-based Convolutional Neural Network (RCNN), Convolutional Neural Network (CNN), and others as technology has advanced. (Abbas et al.; 2021) has created a model to detect disease on tomato leaves for three different classes. The first class consists of five classes having four different categories of diseased leaves and healthy leaves of tomato. In second, it detects six types of diseased and healthy leaves. Finally, it is tested to distinguish between nine types of diseased and healthy leaves. A publicly available Plant Village dataset is used here that contains images of nine different types of diseased and healthy leaves. The data augmentation technique called Conditional Generative Adversarial Network (C-GAN) generates synthetic images from the dataset. Then DenseNet121 is used on both synthetic and real images to train the model. The model achieved more than 95% accuracy, and precision, recall, and F1-score were close to 1 for all three different classes. This proposed model has achieved high accuracy and has worked well with this dataset, but it needs to be tested with other datasets which are fresh and raw. Apart from DenseNet121 other models should also have been used with C-GAN and compared to see which model performs better.

(Bedi and Gole; 2021) has built a hybrid model using Convolutional Autoencoders (CAE) and Convolutional Neural Network (CNN) and used this on peach plant leaf dataset to detect Bacterial Spot disease. The publicly available plant village dataset is used, containing 4457 images of peach plants, out of which 2297 images are of bacterial spot disease, and the rest are healthy leaves. The model here employs the idea of applying dimensionality reduction using Convolutional Autoencoders (CAE), which decreases the parameters required for training. Then on this Convolutional Neural Network (CNN) is applied for disease detection. The model performed well and got an accuracy of 98.38% for the test dataset. The precision, recall and F1-score achieved is also good as it is close to 1. The parameters used for training this model are considerably less when compared with other models and use only 9914 training parameters. Hence, the training time required here is more petite as compared to other models. This model has used a novel approach for disease detection, but data augmentation techniques must have been used here to increase the number of images, and then the model could have been trained and compared with other models.

The research paper (Ashwinkumar et al.; 2021) aims at developing an automated model for detecting tomato plant disease using Optimal Mobile Network-based Convolutional Neural Network (OMNCNN). The model distinguishes five classes: four types of diseased leaves and healthy leaf. The tomato leaf dataset is taken, which is publicly available and contains around 5500 images, including healthy and other kinds of diseased leaves. Various techniques are applied at each stage to build the proposed model. For the preprocessing stage, bilateral filtering is employed. Image segmentation is done using Kapoor's thresholding to detect diseased areas on the leaf. Then to improve the rate of leaf disease identification, the MobileNet model is employed for feature extraction by tuning hyperparameters using Emperor Penguin Optimizer (EPO). Finally, classification of the leaves is done using the Extreme Learning Machine (ELM). With this dataset, this model had a greater accuracy of 98% and the same precision, recall, and F1-score values compared with other deep learning techniques. Despite its high accuracy, this dataset might have been tested with other datasets or validation datasets to see how it performs

on different datasets. Other methods, such as data augmentation to broaden the scope of training datasets and algorithms to detect overfitting and superior image segmentation techniques, should have been explored.

One more unique approach, as shown in paper (Shah et al.; 2021), adapts Teacher/Student architecture to classify disease on plant leaves. It proposes an advanced architecture that uses residual connections to do this job and calls it Residual Teacher/Student (ResTS) architecture. The Residual Teacher/Student (ResTS) uses Convolutional Neural Network (CNN) as its base and has a decoder as ResTeacher and ResStudent as classifiers. Both classifiers are trained reciprocally to recognize diseased areas on the leaf in order to classify the disease of plant leaves. The data set used here is the plant village dataset for 14 different plant leaves. This model had surpassed the former Teacher-student model and got an F1 score of 0.972. When VGG-16 was used for the Teacher-student model, training data was getting overfitted; hence, even after getting a good F1-score for the training dataset, it was performing poorly with validation data. The model uses batch normalization for standardizing the input and to increase the model's authenticity.

A relatively new approach is tried in the (Wang et al.; 2021) paper. It uses Trilinear Convolutional Neural Network (TCNN) to detect diseases in plants. The researcher has used Bilinear Convolutional Neural Network (BCNN) as its base and uses three convolutional layers; the job of the first layer is to detect the area in the image identification, which needs to be trained. The other two layers do feature extraction for crop and disease detection. The dataset used here is from two publicly available data sources with fourteen different types of crops, twenty categories of diseases, and one health category. TCNN uses InceptionV3, ResNeXt101 and VGG16 networks as the base because of their distinctive design and depth. All three varieties of CNN parameter sharing are employed in TCNN, namely partially shared, fully shared, and no share. The data is trained among these three sharing parameters and compared. Out of the three shared models, fully shared ResNeXt101 with TCNN reaches an accuracy of 99.7% and 99.8% for disease and crop identification. The model used the second dataset to check its reliability and got the best accuracy of 84% with the above fully shared parameter. The technique is good but has ample computational time compared with the amount of dataset. If it had focused on identifying relevant features, the model could have worked better with real-world data.

The shortcomings mentioned above is taken care of in the (Yogeshwari and Thailambal; 2021) paper. The paper uses a Deep Convolutional Neural Network (DCNN) to recognise diseases on plant leaves. Different techniques have been applied at the preprocessing layer to enhance disease detection's model capability. A technique called 2-Dimensional Adaptive Anisotropic Diffusion Filter (2D AADF) is used to remove noise in the images. These images are then magnified to focus more on diseased spot identification using Adaptive Mean Adjustment (AMA) method. Clustering and Thresholding are done for image segmentation using Improved Fast Fuzzy C Means (IFFCMC) and Adaptive Otsu (AO) methods. GLCM, i.e. Grey level Co-occurrence matrix, is used to extract crucial features from an enhanced image obtained from the above methods. PCA is used for dimensionality reduction. Finally, the DCNN model is used to train the dataset. The publicly available plant village dataset is used, which has data for fourteen different types of crops. The model is compared with other deep learning methods, and accuracy is used as the evaluation metrics. The accuracy achieved for this model is 0.997, which says the model is performing well. The model here is not tested with a validation dataset or real-world to prove its robustness.

The research paper (Kaur and Devendran; 2020) proposes a fresh approach using

the machine learning technique called Support Vector Machine (SVM) and by applying methods like optimising segmentation phase and doing feature extraction using law mask to classify plant disease category. The author used various steps to create a model that successfully recognises plant disease, and the primary step is to reduce noise in the image using Gaussian Distribution. The next step involves focusing on the area with a high probability of disease and enhancing these selection features using a clustering algorithm called Grey Wolves Optimisation (GWO). Further, the feature extraction is done using Law Mask Gray-level spatial dependence matrix methods. Finally, the classification is done through a Support Vector Machine (SVM) classifier. The dataset used in this research is the plant village dataset which contains images of pepper, potato and tomato leaves. The evaluation metrics used for these experiments are precision, recall, and accuracy, and the values of all these three metrics were close to 0.9. There is no augmentation involved with the dataset to train the model with more images; also, there is no fine-tuning done on the model to increase the value of metrics to build a robust model.

The research paper (Mohanty et al.; 2016) employs two deep learning models AlexNet and GoogleNet, to diagnose disease on plant leaves. It uses the plant village dataset, which is publicly available as mentioned above. Three different versions of the dataset version have been used to train the model: first, the regular colour version; the model is then trained on the grey-scaled version of the dataset; lastly used leaf segmented images to train the model. To verify the model's performance, both the models have been taught from scratch and transfer learning techniques. To avoid over-fitting dataset has been split into various ratios between test and train datasets. GoogleNet outperforms the competition in both training approaches, achieving accuracy and an F1 score of over 98%. Another paper, (Ferentinos; 2018), uses five different methods: AlexNet, AlexNetOWTBn, GoogleNet, OverFeat and VGG, for plant disease identification on the same plant village dataset. VGG is the winner among these models and achieved an accuracy of 99.48%. Both the papers are good, but data augmentation is missing in both the articles and also, they have not been tested with validation sets to verify their robustness.

The research paper (Sujatha et al.; 2021) compares the performances of various machine learning and deep learning models to assess which learning technique is best suitable for diagnosing the disease of the citrus plants. Support Vector Machine (SVM), Random Forest (RF), Stochastic Gradient Descent (SGD) machine learning models and Deep learning models" Inception-V3, VGG-19 and VGG-16 are used for comparison in this paper. Dataset is collected from Punjab, Pakistan, for this research and has 609 images divided into four categories of disease and one of healthy category. Deep learning models perform better than machine learning models and achieve an accuracy of almost 90%. Overall, the dataset size is small, and there is nothing mentioned about the data augmentation technique applied. Hence, there are chances that there must have been an over-fitting issue with this model, and it would not work well while predicting the real images.

The (Hernández and López; 2020) research study focuses on fine-tuning deep learning models to improve model performance prediction using Bayesian fine-tuning methods. According to the report, model predictions on unseen images are unreliable and not exceptionally high. As a result, model performance can be improved utilizing bayesian fine-tuning strategies to anticipate unseen images. To improve model accuracy, various fine-tuning methods such as Stochastic Gradient Descent (SGD), MC dropout (Monte

Carlo Sampling), and Stochastic Gradient Langevin Dynamics (SGLD) are used. This experiment makes use of a publicly available plant village dataset, as previously stated. Here, the VGG16 deep learning algorithm is utilized, and bayesian fine-tuning is given to it. SDG outperforms the other two fine-tuning methods and improves the model's prediction when metrics like accuracy, recall, and f1 score is compared to standard optimization techniques.

The (Mukti and Biswas; 2019) research paper uses a transfer learning approach to recognise plant disease. The author has developed various Convolutional Neural Network (CNN) based transfer learning approaches for disease diagnosis. The transfer learning architecture is built using several pre-trained models such as VGG16, VGG19, AlexNet, and ResNet50. After that, the models are compared using different evaluation criteria such as accuracy, precision, recall, and f1-score. Stochastic Gradient Descent (SGD) fine-tuning is applied in all the models to increase their accuracy. ResNet50 based transfer learning approach outperforms the rest and acquires an accuracy of 99.8%. The Dataset is taken from a publicly available GitHub repository and has 38 different categories of leaves disease, including healthy leaves. The approach used here is good and can be used as a foundation for other methods. A thorough check of Over-fitting could have been done by applying callbacks to monitor val_loss and early stopping the model when it has achieved the required accuracy. A similar approach is used in (Sagar and Jacob; 2021) but with many other pre-trained Convolutional Neural Network (CNN) architectures such as DensNet169, InceptionResNet, Inception V3, VGG16 and ResNet50. The performance of each transfer learning method is compared to select the best model out of all. Here also, ResNet50 shows good stability and gives an accuracy of 98.2% and precision, recall and f1 score of 0.94. The dataset used here is similar to the above paper. The author has not done any fine-tuning in this method to increase the accuracy of the model and has not tested it for unseen images. The papers conclude that ResNet50 is an excellent model to be used as a pre-trained model when building a transfer learning design.

The (Chen et al.; 2020) is another example of using transfer learning architecture for detecting plant disease. Here, the author uses VGG19 as the pre-trained model, and after this, the model uses a convolution and two inception modules to train the plant dataset. After this, Stochastic Gradient Descent (SGD) fine-tuning is applied to enhance the model's accuracy. A Chinese institute gave the dataset for this research containing 12 different rice, maize and wheat diseases in whole and healthy leaves. The model acquired an efficiency of 91.83%. The following paper, (Barbedo; 2018), focuses on the limitation of deep learning: it needs a large dataset for training to be accurate in predictions. The author proposes transfer learning as a saviour as it uses pre-trained Convolutional Neural Network (CNN) algorithms like VGG19, VGG16, AlexNet, ResNet50 etc. Pre-trained GooglNet is used here, and on that, the plant dataset is used for image detection, which has 12 different categories of diseases. The model obtained good accuracy for some species, but it was not that good for some. But both these paper shows some light on how transfer learning is effective for image classification problem.

3 Research Methodology

This section concentrates on delivering a synopsis of the proposed methodology; however, the implementation and design section covers its details. With proper research done on the existing papers for instance (Chen et al.; 2020) and (Barbedo; 2018), it is clear that

a large dataset is required to make predictions for image classification problems. Hence, this research is novel as it tries to predict leave's health status with limited data. Before moving on to the implementation and design section, it is crucial to understand the domain and other terminologies as explained in the following section.

3.1 Understanding the Project and Application domain

Before starting any research topic, the primary step is to understand the research domain. Diseases are a widespread phenomenon in plants. It not only impacts human health, but also hampers a country's economic growth. Early detection of disease in plants not only saves money and effort but also ensures excellent food quality. Hence, having domain knowledge about the research topic is an added advantage for this research. Various data models related to deep learning and machine learning are applied to complete this research.

3.2 Understanding the Data

The crucial phase of the research is to gather and collect data requirements, and it is essential to collect pertinent data for the study. The dataset for this project is open to the public and comes from Kaggle[1]. Understanding data is critical in this stage for choosing the proper preprocessing, augmentation techniques to decide models that can be applied to address the research question.

3.3 Data Pre-Processing and Transformation

The primary step before starting model building is to prepare the data for it, and it involves approaches like data preprocessing and augmentation. There are many advantages of applying the above methodologies; namely, it helps in reducing computational time, helps in better training the model by ensuring that the image size is the same across the dataset. Data augmentation is used to increase the dataset size to better train the model. Methods like normalizing the image size for the entire dataset and augmentation techniques like flip and rotation are applied before moving to the modelling phase.

3.4 Data Modelling

Data is divided into two sets: train and test for building a plant disease detection model. Three models, namely, ResNet50, a pre-trained MobilNetV2 and a hybrid model, are made by combining Convolutional Neural Network (CNN) and Random Forest (RF) classifier. These models are compared against various evaluation metrics to verify which model performs better with this small set of data. A summary of each model is given in the following sections, and a detailed explanation will appear in the implementation section.

3.4.1 MobilenetV2

MobileNetV2 efficiently uses depth-wise layers as building blocks. It is built on an inverted residual structure, with residual connections between bottleneck levels. As a source of

[1]https://www.kaggle.com/rashikrahmanpritom/plant-disease-recognition-dataset

non-linearity, the intermediate expansion layer filters features with lightweight depthwise convolutions. Compared to the previous version, it has two new features; in between the layers, it creates linear bottlenecks and shortcut connections between the bottlenecks. It has two blocks, with a stride equal to one residual block and a downsizing block with stride two as discussed in the (Sandler et al.; 2019) paper. In total, it has convolutional layers with 32 filters, and after that, it has 19 residual bottleneck layers. It can easily be converted into a TensorFlow Lite model, which can directly be used in any android or apple mobile. The reason behind selecting this model is that it takes less time and has higher accuracy, and works better on smartphones than any other model.

3.4.2 ResNet-50

ResNet50 is nothing but a residual network with 50 layers. Vanishing gradient is a fundamental problem for training any intense neural network. The residual network comes to the rescue of solving this issue by creating a skip connection that adds the original input to the convolutional block output. The idea of skip connection as discussed in (He et al.; 2015) is to make sure that the higher layer performance is at least on par with the lower layer and not worse. This project would use ResNet50 to train the dataset without a per-trained model to evaluate its performance with a limited set of data.

3.4.3 CNN-RF

The deep learning neural network is excellent for image recognition, but it has one drawback: training the model requires a vast dataset. This research focuses on developing a model that works well while dealing with a limited dataset. As a result, a hybrid model is developed and trained on a novel dataset. This model combines two of the most excellent algorithms: a classic Convolutional Neural Network (CNN), which excels at extracting features from images, and Random Forest (RF), a well-known image classification classifier.

3.5 Results and Evaluation

Once the model is complete, it is critical to assess each model's performance to determine which model yields the best results. Multiple assessment criteria such as f1score, recall, and accuracy are used to evaluate alternative models. The metrics above are compared among all three models in this study to see which one works best with a limited dataset.

3.6 Research Deployment

This research aims to verify the performance of various models against a limited dataset so that the best amongst them can be used in real-life for disease detection. In this last phase of the research, additional attention is given to evaluating each model's performance based on the earlier mentioned evaluation metrics.

4 Design Specification

A framework has been created to better understand each phase involved in the plant disease detection research. The first step is image acquisition, next comes the preprocessing

stage, where images are normalized by resizing the image to a standard size. After that, data augmentation is done to feed model with more data. The pre-final step is to train various models on this data. Finally, all the models are compared with various metrics to select the best model for plant disease detection with the limited dataset. The Figure 1 represents the design specification applied to complete this research successfully.

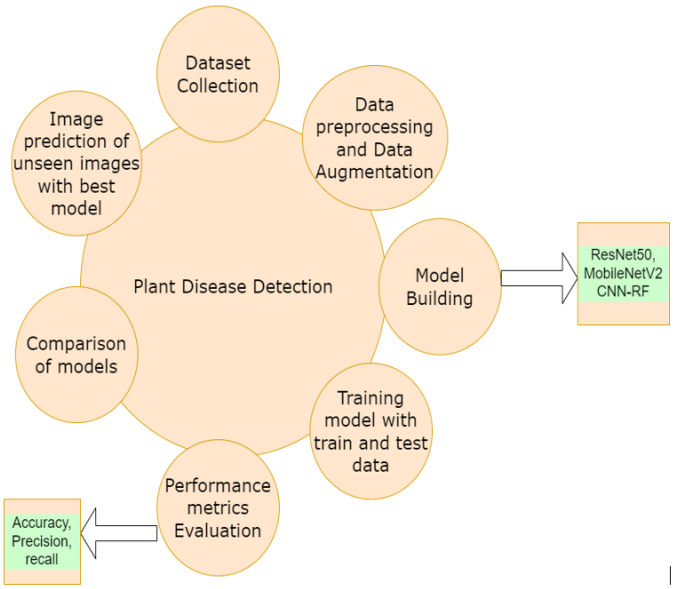

Figure 1: Design Approach

The data preparation stage or the phases involved before training the model and metric evaluation have been given specific importance in this design. The detailed illustration of which is in Section 5. The overview of the stages mentioned above can be seen in Table 1. To summarize the output of the research in the best possible way, the implementation is done iteratively.

Table 1: Business Logic Layer

Data Collection	**Pre-Processing**	**Modelling**	**Evaluation**
Plant Disease Recognition	Normalization	ResNet50	Accuracy, precision, recall
Plant Disease Recognition	Re scaling	MobileNetV2	Accuracy, precision, recall
Plant Disease Recognition	Augmentation	CNN-RF	Accuracy, precision, recall

The manual document submitted along with this research paper contains the configuration information related to software and hardware requirements for successfully carrying out this project. It also contains a step-by-step approach required to execute the implementation design mentioned above.

5 Implementation

This research aims to verify how well the deep learning model performs, compared to transfer learning and the hybrid CNN-RF model, with a limited data set. To evaluate

this research question, three different models are applied to the plant leaves dataset to classify them into three categories: healthy leaves and infected leaves into powdery and rust. The issues encountered throughout this implementation are as follows:

- The most prominent hurdle of this research is that it has a limited dataset. Hence, normalization and data augmentation was applied to the dataset to train the model.
- The images in the dataset have high quality; hence, resizing the image was required to upload the dataset into the cloud and reduce the computational time.
- There is a need for high RAM and GPU machines to execute the model with an increased number of epochs.

5.1 Data Collection

Using the plant disease recognition dataset supplied by the Kaggle([2]), the intended architecture of plant disease detection is created and assessed. This dataset is publicly available in Kaggle and was published in July 2021. This dataset is relatively novel, and no one has worked on this dataset based on the literature review done for this research. The dataset contains 1530 images divided into train, test and validation sets. Each set has three different classes of plant leaves: healthy, powdery and rust. The details of the dataset are explained in the Figure 2. After the dataset is collected successfully, the next step is to analyse the data and assign labels to each category of the leaves available in the dataset. Labels are assigned to each category of leaves: Healthy 0, Powdery 1 and Rust 2.

	File	DiseaseID	Disease Type
0	Powdery/ced36b93c8c498c3.jpg	1	Powdery
1	Powdery/8781dcd12a57aec5.jpg	1	Powdery
2	Healthy/8db3c8abe2745554.jpg	0	Healthy
3	Healthy/9dd96a6e60863497.jpg	0	Healthy
4	Healthy/81b331e1c1720fdf.jpg	0	Healthy

Figure 2: Sample Dataset

5.2 Data Preprocessing and augmentation

Data-preprocessing starts with dividing the dataset into three sets: train, test and validation. Each set contains three classes: healthy, powdery and rust. The next phase is resizing the image into a standard size. All the images in the dataset are resized to 224*224 pixels, as for implementing MobileNetV2 standard size required is 224 pixels.

[2]https://www.kaggle.com/rashikrahmanpritom/plant-disease-recognition-dataset

Since there are three different leaves classes, class_code is set to categorical, and shuffle is set to true to rearrange the order of the images yielded. After data preprocessing comes the data augmentation part. It is a very crucial state for this research and for any study where the dataset is limited. A python library called Keras is used to implement data augmentation. Several techniques like rescaling, rotating, zooming, channel_shift, and the vertical and horizontal flip are created to create a suitable dataset. The sample images in the dataset can be seen in Figure 3.

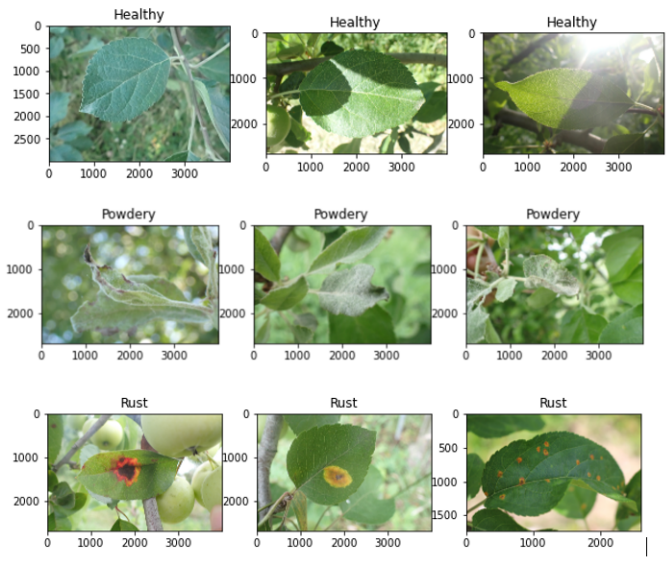

Figure 3: Sample Images

- **Why Image Augmentation is important?**

Performing data augmentation in a dataset has three main advantages: For better performance of deep learning model, it needs to be fed with a massive amount of data; secondly to avoid overfitting and thirdly to improve data relevancy.

5.3 Hyperparameters

Hyperparameters are the parameters used to control a model's learning process, and it helps the model learn the data set for prediction. Various parameters like adam optimizer with learning rate, activation function, drop out with probability, the number of layers and epochs are used to better model prediction. The details of each hyperparameter are described in the model building section.

5.4 Model Building

This section focuses on implementing three different models to assess their performance in plant disease detection on a limited dataset. The first is a hybrid model that combines Convolutional Neural Network (CNN) and Random Forest (RF). The second model employed for this research is the ResNet50 residual network from scratch, and finally, pretrained MobileNetV2 will be used on the plant disease dataset.

5.4.1 Convolutional Neural Network

The Convolutional Neural Network (CNN) builds a funnel network and has a fully-connected layer as output which helps in predictions as all neurons are connected to one another. CNN has five layers comprising of pooling layer, convolutional layer, rectified linear unit, flatten and a dense layer or fully connected layer. The convolutional layer is the first layer that tries to conserve vital features from the images in its all possible forms. A traditional CNN uses a 3*3 filter to move around each corner of the image to build an activation map. To increase the non-linearity in the network, a rectified linear unit is used. The next critical layer is the pooling layer, where the network learns to distinguish between images using spatial invariance. It also keeps only essential elements and clears all unwanted noise in the image to avoid over-fitting. These features are then flattened before passing them to the dense or fully connected layer. Finally, the classification is done in the dense layer or fully-connected layer.

5.4.2 Random Forest

Random Forest is a machine learning model used as a classifier in this research. It employs averaging to increase predicted accuracy and control over-fitting by equipping several decision tree classifiers on different sub-samples of the dataset.

5.4.3 Hybrid CNN-RF

The best of both worlds combined to predict plant diseases in the hybrid CNN-RF model. Random Forest classifier is used to indicate whether the plant leaves are healthy or powdery or rust disease based on the features extracted by Convolutional Neural Network (CNN) as its input. In this model, the accuracy of CNN, when solely used for prediction, is compared with the prediction accuracy of hybrid CNN-RF. It is visible that the hybrid model enhances the model's accuracy as RF comes with the benefit that it minimizes the over-fitting of data and has excellent tolerance to outliers. It amplifies the accuracy even when useful features cannot be pulled from CNN due to a limitation on the size of the dataset.

Both training and test datasets are pitched into the train and test variable of the type list. Since working with a list becomes complex, the train and test variables are converted into a NumPy array. The image size for both train and test is 64 pixels, and one hot encoding is used for the categorical variables. For CNN, it is critical to select an optimal size of convolution filters to get adequate performance. As discussed in Subsection 5.4.1 3*3 filter is used to reduce over-smoothing. This project uses three convolutional layers with 32, 64 and 128 filters, with max-pooling and batch normalization applied at each layer to standardize the input. Finally, the model is flattened to convert it into a single column. The job of all these above layers is to extract features from the images and act as an input to Random Forest (RF) classifier. The output of the feature extractor is also given as input to two dense layers of 128 and 64 dimensions, and a regularization technique dropout is applied with value 0.2 before passing it to the dense layer of 3 dimensions as we have three classes to be identified. The RF classifier surpasses CNN prediction in terms of accuracy. The details of all the evaluation metrics are explained in the results section. The architecture of the hybrid CNN-RF applied is shown in the Figure 4.

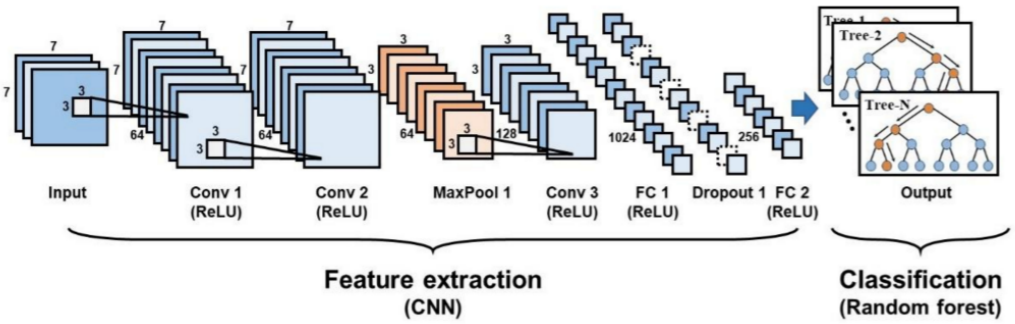

Figure 4: CNN-RF

5.4.4 ResNet50

ResNet is nothing but a Residual Neural Network that serves as the basis for many image classification tasks. Training a deep neural network is challenging due to the vanishing gradient problem. Residual Network resolves this issue as it allows to train a deep neural network with more than 150 layers. This research aims at using a smaller version of ResNet152, i.e. ResNet50, for image classification. ResNet outperforms other deep learning models because it uses skip connection to connect original input to the convolution block output. Hence, ResNet50 was selected in this research for plant disease detection.

Depending on whether the input and output dimensions are the same or not, the identity and convolution blocks are used. When input and output size is the same, an identity block is used, and when different convolution block is used. The basic architecture and layers for this dataset are shown in Figure 5.

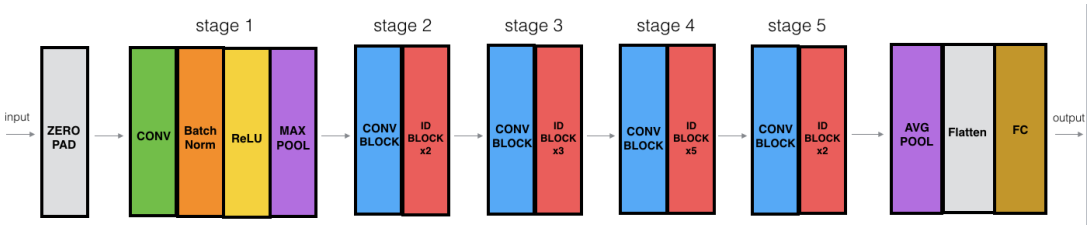

Figure 5: ResNet50

In total, there are five stages involved before the flatten layer. The crucial step before entering phase one is doing zero-padding of (3*3). In phase one, a convolution layer of 64 filters of shape (7,7) and stride of (2,2), batch normalization and max-pooling are used. Phase two involves convolution and identity block, as mentioned above. Both the blocks use three filters, namely 64, 64 and 256. Phase three, four and five also use convolution and identity blocks with three filters each, namely (128,128,512), (256, 256, 1024) and (512,512, 2048) respectively. After this, a 2D pooling average is applied. Before passing this to a fully connected layer, it is flattened to get a single column for prediction. While compiling the model, Adam optimizer is used with loss as categorical_crossentropy. The evaluation metric used here are accuracy, recall and precision.

5.4.5 MobileNetV2

MobileNetV2 is a CNN based architecture that works better on mobile devices. It uses an inverted residual structure with connections between bottleneck layers. As a source of non-linearity, the intermediate expansion layer filters features using lightweight depthwise convolutions. The MobileNetV2 architecture includes a full convolution layer with 32 filters and 19 residual bottleneck layers.

In this research, pre-trained MobileNetV2 will be used from a TensorFlow-hub python library to train the model with the limited dataset to check its performance. The initial requirement to build this model is to have an image size of (224,224). After this, a pre-trained model trained on the ImageNet dataset is selected from the TensorFlow-hub library. The trainable is set to false to freeze the variable so that training will only impact new classifier layers. The output of the feature extract is a 1280-long vector for every single image. Finally, the output of the feature extractor is connected to a dense or a fully connected layer using tf.Keras.Sequential. Adam optimizer is used along with loss as sparse_categorical_crossentropy to compile the model. Regularization technique like early stopping is used to avoid over-fitting. Evaluation metrics used in this model are accuracy, precision and recall.

6 Model Results and Evaluation

This research aims to verify whether the deep learning model can give good predictions with a limited data set. Hence, to validate this aim, the ResNet50 model is compared with the other two models, namely, pre-trained MobilenetV2 and hybrid CNN-RF models. All three models are applied to the plant disease dataset to classify the status of the plant leaves. This section evaluates all three models based on accuracy, recall, f1score and precision metrics. The batch size for MobileNetV2 and ResNet50 is 128, the epoch for ResNet50 and MobileNetV2 is kept as 100, and early stopping is applied to avoid over-fitting . For CNN-RF, epoch size is marked as 50. Each model works best when the learning rate is kept as 0.001. The accuracy achieved by all the models are shown in the table below.

Table 2: Accuracy Evaluation of Models

Model Employed	**Learning Rate**	**Metrics**	**Value**
ResNet50	0.001	Accuracy	67
CNN-RF	None	Accuracy	84
MobileNetV2	0.001	Accuracy	95.53

6.1 Experiment 1 - ResNet50

The ResNet50 performs reasonably well for this novel dataset and achieves a test accuracy of 70%. ResNet50 network is provided with an image size of 64 pixels, and a batch size of 128 is used for this model. Hyperparameters like epochs were frequently varied to obtain high accuracy. With an epoch of 100, ResNet50 achieved a testing accuracy of 71.88%. The precision, recall and f1 score achieved is very well when compared to the size of the dataset that was used for the training. The Figure 6 shows the confusion matrix of the model.

Classification Report

	precision	recall	f1-score	support
Healthy	0.70	0.56	0.62	50
Powdery	0.77	0.86	0.81	50
Rust	0.56	0.60	0.58	50
accuracy			0.67	150
macro avg	0.67	0.67	0.67	150
weighted avg	0.67	0.67	0.67	150

Figure 6: ResNet50 Classification Report

6.2 Experiment 2 - CNN-RF

The hybrid CNN-RF model works very well with the limited dataset. The first phase, i.e. CNN, focuses on extracting features from the images, and the output is supplied to Random Forest (RF) classifier. The CNN as a classifier, achieves an accuracy of only 71.88%, but when Random Forest is used as a classifier, it achieves an accuracy of 84%. It is also performing well when provided with unseen images. The precision, recall and f1 score achieved by RF is excellent as the data on which it is trained is very less. The Figure 7 shows the classification report of the model.

Classification Report

	precision	recall	f1-score	support
Healthy	0.80	0.82	0.81	50
Powdery	0.82	0.90	0.86	50
Rust	0.91	0.80	0.85	50
accuracy			0.84	150
macro avg	0.84	0.84	0.84	150
weighted avg	0.84	0.84	0.84	150

Figure 7: CNN-RF Classification Report

6.3 Experiment 3 - MobileNetV2

A MobileNetV2 model pre-trained on the ImageNet dataset is used here to extract and learn all the relevant features, and this is used to train the plant village dataset for disease detection. The model achieves an accuracy of 95.33% and also performs well with unseen images. The precision, recall and f1 score are also close to 1. There is no over-fitting in training and test data as the hyperparameters like early stopping, learning rates, epoch and batch size are selected appropriately. The model also tries to predict 20 unseen images and predicts them correctly. The figure shows the confusion matrix for all the evaluation metrics.

Classification Report

	precision	recall	f1-score	support
Healthy	0.96	0.96	0.96	50
Powdery	0.96	0.90	0.93	50
Rust	0.94	1.00	0.97	50
accuracy			0.95	150
macro avg	0.95	0.95	0.95	150
weighted avg	0.95	0.95	0.95	150

Figure 8: MobileNetV2 Classification Report

6.4 Discussion and Model Comparison

This research entirely focuses on assessing the performance of models with a limited set of data. Three models, ResNet50, MobileNetV2 and CNN-RF, compared each other to evaluate their performance against a limited set of data for plant disease detection. The dataset contains 1532 images in total, out of which the training dataset has 1322 images, the test dataset has 150, and the validation set has 60 images. Each dataset is divided into three categories health, powdery and rust. Out of all three models, MobileNetV2 outcasts the other two models by achieving an accuracy of 95.33%. CNN-RF model performs better than ResNet50 getting an accuracy of 84%. Regularization techniques like data augmentation, dropout and early stopping have been applied wherever necessary to avoid over-fitting. Figure 9 and 10 shows the loss and accuracy graph plot for MobileNetV2 to demonstrate its superiority over other models.

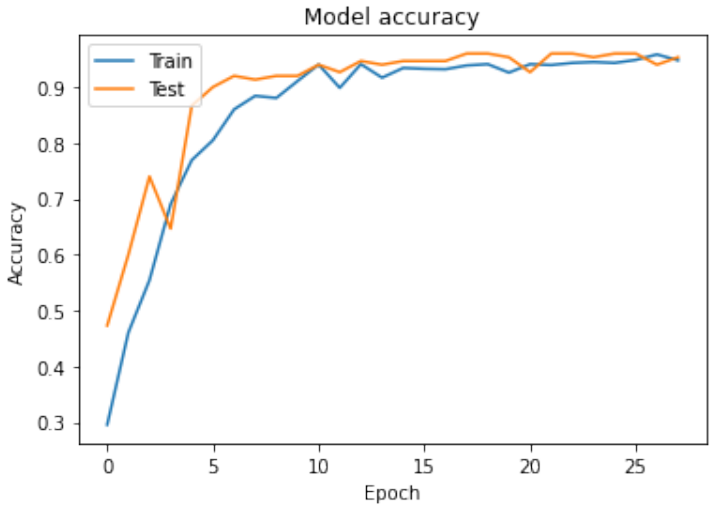

Figure 9: MobileNetV2 Accuracy

Also, the MobileNetV2 model is converted into tf.lite, a new tensor flow model which can be directly used in android and ios mobile for real-life disease prediction. It is evident from this research that deep learning models perform well only when there is a large dataset. Hence, to overcome this limitation of deep learning, the transfer learning method can be applied to achieve higher accuracy and accurate prediction. Transfer

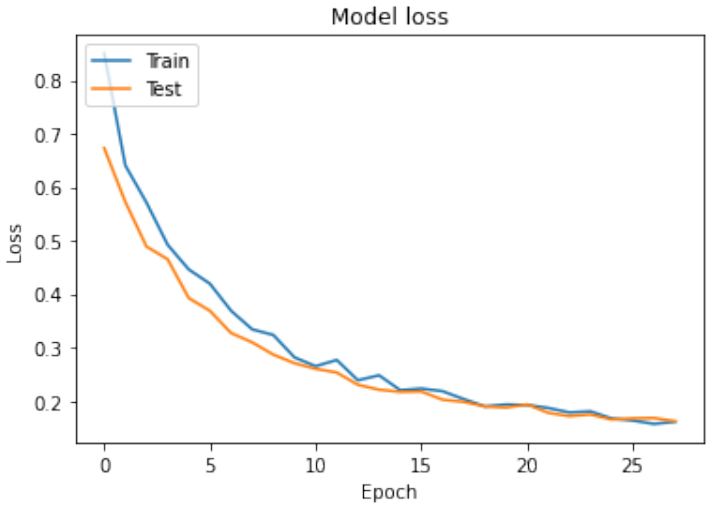

Figure 10: MobileNetV2 Loss

learning is still a novel approach in deep learning fields, and this study shows how well it can be used when there is limited set of data.

7 Conclusion and Future Work

Food is essential for human existence, and hence, it becomes extremely necessary to detect diseases on plants as early as possible. Data collection is a very tedious task, and therefore, a suitable model needs to be built that can even work with a smaller dataset. Deep learning is a technique that has evolved in recent years for all computer vision-related tasks, but it comes with the liability that it cannot be used with a limited dataset. This research has clearly shown that transfer learning can be used as a great rescuer for such scenarios, instead of deep learning, and should be extensively used for building a good model. The pre-trained MobileNetV2 model has surpassed both ResNet50 and CNN-RF models in predicting plant diseases effectively, achieving an accuracy of 95.33%. The model is then exported in tf.lite extension to be easily deployed in mobiles for making real-life predictions.

Many publicly available datasets are even smaller than the one used in this study. As a result, transfer learning models such as MobileNetV2 should be evaluated on those datasets to determine their efficacy. A hybrid model should be created by combining data augmentation techniques such as Deep Convolutional Generative Adversarial Network (DCGAN), to generate enough synthetic images on which a deep learning model can be built for prediction.

8 Research Acknowledgement

My supervisor Dr. Bharathi Chakravarthi provided consistent support and comments throughout the research endeavour. He was educated and helpful, and guided me in the correct route for finishing my master's project by providing the essential help for the project implementation.

References

Abbas, A., Jain, S., Gour, M. and Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with c-gan synthetic images, *Computers and Electronics in Agriculture* **187**: 106279.

Ashwinkumar, S., Rajagopal, S., Manimaran, V. and Jegajothi, B. (2021). Automated plant leaf disease detection and classification using optimal mobilenet based convolutional neural networks, *Materials Today: Proceedings* .

Barbedo, J. G. A. (2018). 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.

Bedi, P. and Gole, P. (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network, *Artificial Intelligence in Agriculture* **5**: 90–101.

Bourke, P. A. (1964). Emergence of potato blight, 1843–46, *Nature* **203**(4947): 805–808.

Chen, J., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y. (2020). Using deep transfer learning for image-based plant disease identification, *Computers and Electronics in Agriculture* **173**: 105393.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis, *Computers and Electronics in Agriculture* **145**: 311–318.

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Deep residual learning for image recognition.

Hernández, S. and López, J. L. (2020). Uncertainty quantification for plant disease detection using bayesian deep learning, *Applied Soft Computing* **96**: 106597.

Jethwani, B., Dave, D., Ali, T., Phansalkar, S. and Ahhirao, S. (2021). Indian agriculture gdp and non performing assets: A regression model, *IOP Conference Series: Materials Science and Engineering*, Vol. 1042, IOP Publishing, p. 012007.

Kaur, N. and Devendran, V. (2020). Novel plant leaf disease detection based on optimize segmentation and law mask feature extraction with svm classifier, *Materials Today: Proceedings* .

Max Roser, H. R. and Ortiz-Ospina, E. (2013). World population growth, *Our World in Data* .

Mohanty, S. P., Hughes, D. P. and Salathé, M. (2016). Using deep learning for image-based plant disease detection, *Frontiers in Plant Science* **7**: 1419.

Mukti, I. Z. and Biswas, D. (2019). Transfer learning based plant diseases detection using resnet50, *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, pp. 1–6.

Mutka, A. M. and Bart, R. S. (2015). Image-based phenotyping of plant disease symptoms, *Frontiers in plant science* **5**: 734.

Padmanabhan, S. (1973). The great bengal famine, *Annual Review of Phytopathology* **11**(1): 11–24.

Ristaino, J. B., Anderson, P. K., Bebber, D. P., Brauman, K. A., Cunniffe, N. J., Fedoroff, N. V., Finegold, C., Garrett, K. A., Gilligan, C. A., Jones, C. M., Martin, M. D., MacDonald, G. K., Neenan, P., Records, A., Schmale, D. G., Tateosian, L. and Wei, Q. (2021). The persistent threat of emerging plant disease pandemics to global food security, **118**(23).

Sagar, A. and Jacob, D. (2021). On using transfer learning for plant disease detection.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C. (2019). Mobilenetv2: Inverted residuals and linear bottlenecks.

Shah, D., Trivedi, V., Sheth, V., Shah, A. and Chauhan, U. (2021). Rests: Residual deep interpretable architecture for plant disease detection, *Information Processing in Agriculture* .

Sujatha, R., Chatterjee, J. M., Jhanjhi, N. and Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection, *Microprocessors and Microsystems* **80**: 103615.

Wang, D., Wang, J., Li, W. and Guan, P. (2021). T-cnn: Trilinear convolutional neural networks model for visual detection of plant diseases, *Computers and Electronics in Agriculture* **190**: 106468.

Yogeshwari, M. and Thailambal, G. (2021). Automatic feature extraction and detection of plant leaf disease using glcm features and convolutional neural networks, *Materials Today: Proceedings* .