Plant Disease Detection using CNN and Transfer Learning

A. Introduction and Problem Statement

Plant diseases can cause significant losses in crop yields, posing a threat to farmers and plantation businesses. Traditionally, agriculture experts have relied on manual techniques to detect plant diseases, which can be time-consuming and ineffective due to the lack of expertise. However, with the emergence of deep learning-based techniques, particularly Convolutional Neural Networks (CNNs), there is a possibility of automating the process and achieving better accuracy in disease detection [15] [5]. Various CNN architectures such as VGG-19, ResNet-18, ResNet-34, MobileNetV2, EfficientNet, GoogLeNet, and InceptionV3 have been proposed, allowing farmers to quickly and accurately identify plant diseases, preventing their spread, and increasing crop yields [10] [12] [13].

In this study, we aim to use CNNs to detect plant diseases using three different datasets, each labelled with the type of disease or whether the plant is healthy. The primary objective is to assess the performance of three CNN architectures - Resnet18, MobileNetv2, and InceptionV3 - under the same training environment and hyperparameters to determine which model works best for this problem. The pre-processed data will be used to train the models, and performance evaluation will be carried out using various metrics such as accuracy, precision, recall, and F1-score [11]. Transfer learning will also be performed on two models with lower accuracy to improve their performance by reusing the learned features from a pre-trained model and fine-tuning it on a new dataset [4].

B. Proposed Methodologies

Dataset 1 [3]: This dataset contains 2662 images of different plant leaves which are further divided into 2 classes, healthy, and unhealthy. All the images in this dataset are raw images and hence pre-processing the data helps in achieving better training accuracy.

Dataset 2 [2]: This dataset consists of 54303 healthy and unhealthy leaf images divided into 38 categories by species and disease. For the purpose of this study we have to reduce the number of categories to 10 and hence we considered only 10 classes containing 15250 healthy and unhealthy images of 5 different plants. We are using plant village dataset for this with is a very popular dataset in this domain.

Dataset 3 [1]: This dataset consists of 31000 healthy and unhealthy leaf images divided into 50 categories by species and diseases. For the purpose of this project we have considered 20 labels with 18600 healthy and unhealthy images of 10 different plants. Most of the images in this dataset are surrounded by noisy and the number of images per label are also comparatively low so we are expecting some underfitting issues and low accuracy.

ResNet18:
We have used ResNet18 architecture which is a 72 layer architecture with 18 deep layers and has 11.4 million trainable parameters and 5 residual blocks with 1.8 billion FLOPS. Residual blocks addresses performance degradation problems by introducing skip connections and identity mapping, which helps the model to learn features more effectively and increases accuracy.

MobileNetV2:
MobileNetV2 architecture has 53 convolution layers and 1 AvgPool. It has 3.5 million trainable parameters and is designed to perform well with low computational cost. It has 5.79 million FLOPS which is the lowest among all other models we are using. We are expecting a low computation time on all datasets compared with other models.

InceptionV3:
The inception V3 is just the advanced and optimized version of the inception V1(GoogleNet) model. It uses Batch normalization which aims to reduce internal covariate shift, and in doing so aims to accelerate the training of deep neural nets. It also uses auxiliary classifiers as regularizes. It is 22 layers deep with 9 inception modules stacked linearly in total.

C. Attempts at solving the problem

We divided the dataset into two parts: training and testing. We used the 'Stratified Shuffle Split' cross-validation method to split the dataset in a ratio of 70% for training and 30% for testing [9]. The training dataset was further divided into five folds, where four folds were used for training and one fold for validation for 10 class and 20 class dataset. For remaining dataset, test, validation and test splits are already available from the source.

As part of data pre-processing, we applied normalisation using mean and standard deviation [6]. We performed resizing to meet the model requirements and transformations such as random horizontal flip, and random rotation to stabilise the gradient descent step and help the models converge faster for a given learning rate [8] .

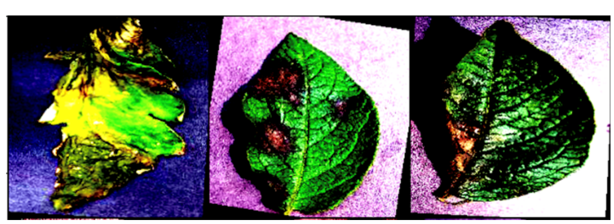


Figure 1. Sample pre-processed Image

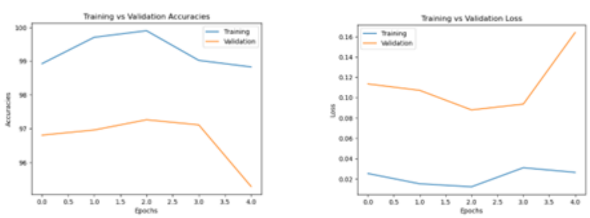
We successfully trained and evaluated ResNet-18, MobileNetV2, and inception v3 on the Binary class dataset. We also performed pre-processing on the remaining datasets. To optimise our models, we considered both Adam and SGD optimizers. Although Adam optimizer [7] converges well, we ultimately chose to use SGD with momentum as it has higher generalisation than Adam optimizer and eliminates the need for bias rates in our model [16]. Additionally, we applied the shuffle operation for the same purpose.

Model Parameters	
Epochs	100
Input Image Size	224*224
Batch Size	32
Loss Function	Cross Entropy
Learning Rate	0.0002

After training with above parameters we observed below accuracies.

Model	ResNet-18	MobileNet V2	Inception V3
Binary Dataset	99.7	99.5	72.2

ResNet-18

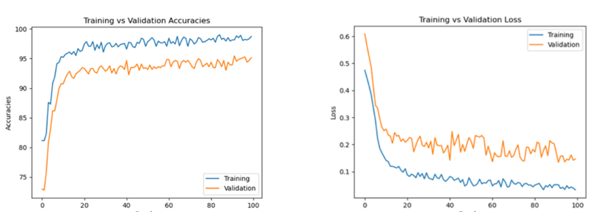


(a) Accuracy Plot (b) Loss Plot

Figure 2. Training vs Validation ResNet18

Accuracy for ResNet18 started coming down when it is approaching 100 epochs.

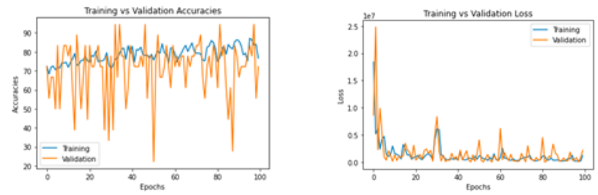
MobileNetV2



(a) Accuracy Plot (b) Loss Plot

Figure 3. Training vs Validation MobileNetv2

InceptionV3



(a) Accuracy Plot (b) Loss Plot

Figure 4. Training vs Validation InceptionV3

With Inception v3 we have observed a lot of fluctuation spikes accuracy. We are trying to figure out the reason for this behaviour.

Below are results of the remaining metrics.

Metrics	ResNet-18	MobileNet V2	Inception V3
Precision Macro	1.00	0.99	0.09
Precision Micro	1.00	0.99	0.19
Recall Macro	1.00	0.99	0.50
Recall Micro	1.00	0.99	0.19
F1 Score Macro	1.00	0.99	0.16
F1 Score Micro	1.00	0.99	0.19

D. Future Improvements

As per our schedule, we need to finish the remaining data pre-processing and implement models on two additional datasets within the next two weeks. Following that, we have set aside two weeks for hyperparameter tuning and transfer learning. During hyperparameter tuning, we plan to experiment with different values for the number of epochs, including 50, 70, and 120, as well as varying the learning rate with values of 0.0001, 0.0003, and 0.0005, and adjusting the batch size to 10, 50, and 64. Currently, we are not using dropout, but we plan to incorporate it into our models [14]. If time permits, we will also explore changes to the number of layers of one of our models.

References

[1] Plant disease [50 classes]. www.kaggle.com/datasets/fabinahian/plant-disease-50-classes. 1

[2] Plantvillage-dataset. www.github.com/spMohanty/PlantVillage-Dataset/tree/master/raw. 1

[3] potato-tomato-dataset. www.kaggle.com/datasets/alyeko/potato-tomato-dataset. 1

[4] Transfer learning for computer vision tutorial — pytorch tutorials 1.7.0 documentation. 1

[5] Sk Mahmudul Hassan, Arnab Kumar Maji, Michał Jasiński, Zbigniew Leonowicz, and Elżbieta Jasińska. Identification of plant-leaf diseases using cnn and transfer-learning approach. *Electronics*, 10:1388, 06 2021. 1

[6] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 1

[7] Diederik Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization, 01 2017. 2

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:1097–1105, 05 2012. 1

[9] Sebastian Mirjalili. *PYTHON MACHINE LEARNING - THIRD EDITION : machine learning and deep learning with python, scikit... -learn, and tensorflow 2*. Packt Publishing Limited, 2019. 1

[10] Mehmet Alican Noyan. Uncovering bias in the plantvillage dataset, 06 2022. 1

[11] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv:2010.16061 [cs, stat]*, pages 37–63, 10 2020. 1

[12] Nishant Shelar, Suraj Shinde, Shubham Sawant, Shreyash Dhumal, and Kausar Fakir. Plant disease detection using cnn. *ITM Web of Conferences*, 44:03049, 2022. 1

[13] Nishant Shelar, Suraj Shinde, Shubham Sawant, Shreyash Dhumal, and Kausar Fakir. Plant disease detection using cnn. *ITM Web of Conferences*, 44:03049, 2022. 1

[14] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv:1803.09820 [cs, stat]*, 04 2018. 2

[15] Bulent Tugrul, Elhoucine Elfatimi, and Recep Eryigit. Convolutional neural networks in detection of plant leaf diseases: A review. *Agriculture*, 12, 08 2022. 1

[16] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Hoi, and Salesforce Research. Towards theoretically understanding why sgd generalizes better than adam in deep learning. 2