Detection of Diseases in Rice Plants using Convolutional Neural Networks

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

B.Tech.

by

Nikhil Singhal (2019IMT-067)

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT GWALIOR-474 015

2022

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the report, entitled **Detection of Diseases in Rice Plants using Convolutional Neural Network**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of my own work carried out during the period *June 2022* to *September 2022* under the supervision of **Dr. Joydip Dhar**. I have also cited the references about the text(s)/figure(s)/table(s) from where they have been taken.

Date: Signatures of the Candidate

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: Signatures of the Research Supervisor

ABSTRACT

India is an agricultural nation, hence a large portion of its population depends on it. We are the world's fourth-largest producer of rice. Rice diseases are the principal factor affecting rice productivity. Therefore, the goal of this research is to create a machine learning model for identifying the rice diseases i.e., Leaf Blast, Brown Spot, and Hispa. The primary emphasis is on the convolutional neural network technology and feature extraction methods used to identify patterns in the images and categorise the diseases. Image collecting, image augmentation, feature extraction, and classification are all part of the methodology.

ACKNOWLEDGEMENTS

I would first like to thank my supervisor, Dr. Joydip Dhar, whose constant guidance and support helped me to sharpen my thinking and brought my work to a satisfactory level. Their patient support and all of the opportunities they provided were of immense help. I would also like to thank my parents for their valuable advice and support

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
MBGD	Mini-Batch Gradient Descent
SGD	Stochastic Gradient Descent
PIL	Python Imaging Library
Tf	Tensorflow
CV	Computer Vision
SVM	Support Vector Machine
HOG	Histogram of Oriented Gradient
RGB	Red Green Blue
SDAE	Stacked Denoising Auto Encoder
BGD	Batch Gradeint Descent

NOTATIONS

θ	Optimization Parameter
C(θ; X_i,Y_i)	Cost of X_i,Y_i
x_i,y_i	Data point in the training set
L(y,$\hat{y}$)	Loss function
y	Actual value
$\hat{y}$	Predicted value
m	Total number of datapoints
n	Total number of classes

Chapter 1

Introduction

This chapter covers an outline of the research that was done. "Detection of Diseases in Rice Plants Using Convolutional Neural Networks" is the title of the study. Context, Problem/Motivation, Objectives, and Research Workflow make up this section. The Thesis is described in the Context. The issues that develop are described in Problem/Motivation. The research's objectives are a list of goals that must be met. The final phase in the research process is summarised in the research workflow.

1.1 Context

Rice is one of the most consumed food in the world and Asia is a huge producer of it. More than 50% of the world's population relies on it and it is a very crucial crop around the world. In various places, rice is eaten on a daily basis. Many factors affect the production of rice, one of them is paddy diseases.

A condition that reduces the yield of the crop or injures it is called crop disease. These are characterized by their symptoms. Examples of rice diseases are Brown Spot, Hispa, Leaf Blast. Computer Vision and Convolutional Networks technologies are proven to help the agricultural industry by increasing their yield. These technologies have a lot of potential to contribute to the agricultural industry.

Detection of diseases through technological methods are proven to be the most beneficial to the agricultural industry. It can speed up the process of disease detection. The aim of this project is to develop a model which can detect and classify rice diseases using convolutional neural networks technique.
In Fig. 1.1, we can find the data illustrating the yield of rice in India from 1991-2001.

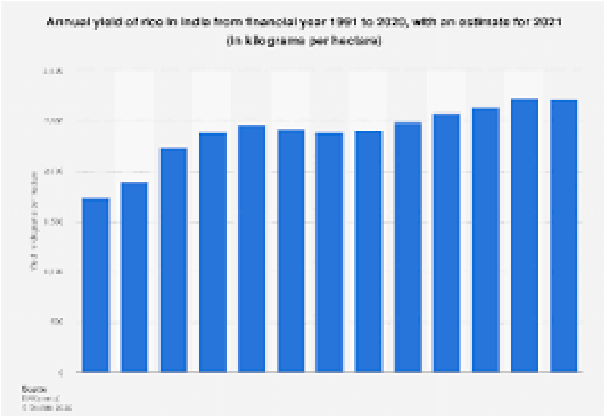


Figure 1.1: Yield of Rice(1991.2021)[9]

1.2 Problem/Motivation

Rice crops are harvested twice in a year. Most of the farmers have to face various problems to harvest their crops as they are attacked by snails, worms and fungi. Furthermore, when one part of the field is infected, it spreads to the other parts as well quickly. Thus, it can decrease farmer's increase and can cause significant losses. Currently, the diseases are detected manually. This manual method can cause error while detection of disease. Also, farmers have to spend a lot of time detecting the type of disease.

1.3 Objectives

The objective of research is to approach and solve the problem of diseases in paddy. The three main goals are:

- to propose a model for paddy disease detection framework

- to classify the diseases by identifying the patterns present on leaves through CNN[8]

- to apply different machine learning architecture to investigate the diseases

1.4 Research work flow

According to the research objectives, the report will describe the work flow as below:

Step 1 Importing the dataset from Kaggle. Dataset is then further uploaded to Google Drive and accessed by mounting the Google Drive to the colaboratory notebook.

Step 2 Resizing the images using Python Imaging Library(Pillow)[5] for downsizing the images. Antialiasing technique is used to reduce the visual defects that occur when high-resolution images are presented in a lower resolution.

Step 3 Developing and analyzing different architectures to find the best method for detecting diseases in rice plants.

Chapter 2

Literature review

2.1 Background

There has been various research in the field of rice disease detection, but all have some or other shortcomings. Below are some of the papers which cite various details of projects which help in this project.

2.2 Key related research

2.2.1 "Detection and classification of rice plant diseases," Intelligent Decision Technologies

In this paper[3], Image processing and machine learning techniques were used for the detection and classification of rice plant diseases. It uses K-means clustering to segment the diseased area and Support Vector Machine(SVM) for classification. It has achieved an accuracy of 93.33% on the training dataset and 73.33% on the testing dataset. The same dataset is also used by us but it resulted in higher accuracy.

2.2.2 Plant disease detection using machine learning

It[7] uses an ensemble learning method to classify between healthy and diseased leaves. Histogram of Oriented Gradient(HOG) was used for extracting the features of an image. Their work has claimed an accuracy of 82.33%.

2.2.3 Content based paddy leaf disease recognition and remedy prediction using support vector machine

In this study[2], K-means clustering was used to separated affected parts from the rice leaf and the model model was trained using texture, shape and color as classifying features using Support Vector Machine(SVM).

2.2.4 A faster technique on rice disease detection using image processing of affected area in agro-field

In this paper[9], using image processing to extract the percentage of the RGB value of the affected area, a model was developed. RGB values were then fed into a Naive Bayes classifier to classify the diseases into three different categories. The accuracy achieved in this approach is 87%.

2.2.5 A deterministic approach for disease prediction in plants using deep learning

This paper[4] was aimed to develop a model to detect plant diseases using Deep Learning techniques. Stacked Denoising Auto Encoder(SDAE), Deep Belief Network(DBN) were used. To train the model using the Caffe deep learning framework, 30880 images were used and 2589 images were used to test the model. For achieving a better accuracy, 10-fold cross validation technique was used on their dataset. The accuracy of this model is 96.77%.

2.3 Analysis

From the above literature, we noticed that many Plant disease detectors perform well in terms of accuracy. However, on the other hand, they have some downsides as well. Some algorithms and network structures can help increase the algorithm's efficiency and achieve real-time prediction speeds.

2.4 Research gaps

There are several options available for each stage of plant disease detection. As stated by the writers in [7, 2, 4], they produced their own image collection by acquiring photos of rice fields. Image preprocessing is needed to get photos ready for more processing. To reduce the noise in pictures, most authors used a median filter.. Although noise is frequently removed using the median filter, there are some instances when disease spots become blurred when a median filter is applied if they are too small. Other filters are also employed, such as mean filter and laplacian filter. Some writers employed the histogram equalisation technique to eliminate noise. After preprocessing, segmentation—the procedure of removing the diseased area of a leaf—is carried out. Thresholding, Otsu's thresholding, 8-connected component labelling, and K-means clustering are some of the segmentation approaches that are accessible. The next phase is the illness part feature extraction.

1. The mean and standard deviation of the R, G, and B components of the sick section.
2. The area of the diseased portion.
3. Texture properties such as contrast, uniformity, and linear correlation.

2.5 Problem formulation

From the above literature review, we are sure that various plant disease detection techniques are suitable in accuracy, but they are way too slow to be used in real-time.

2.6 Conclusion

We need to develop an algorithm that is fast enough to be used in real-time, and is not very resource hungry.

Chapter 3

Methodology

3.1 Proposed hypothesis

1. The images in the dataset[6] are of variable sizes. So first, they are resized to the same dimensions otherwise it can degrade the accuracy of our model.

2. Then, the dataset is augmented to generalize the model. Certain methods like contrast change, transformation of image, blurring images was applied to generalize the dataset.

3. This augmented dataset is then passed to the neural network to extract the features. Three different models, namely EfficientNetV2, InceptionV2 and InceptionV3 are used for transfer learning.

4. These models have 4 different nodes attached at the end to predict to correct image class.

5. At last, heatmaps and confusion matrix are used to compare the predicted outputs to actual outputs.

3.2 Working of Algorithm

In fig. 3.1, I have attached the complete block diagram of the algorithm used. Each step is elaborated in the further sections.

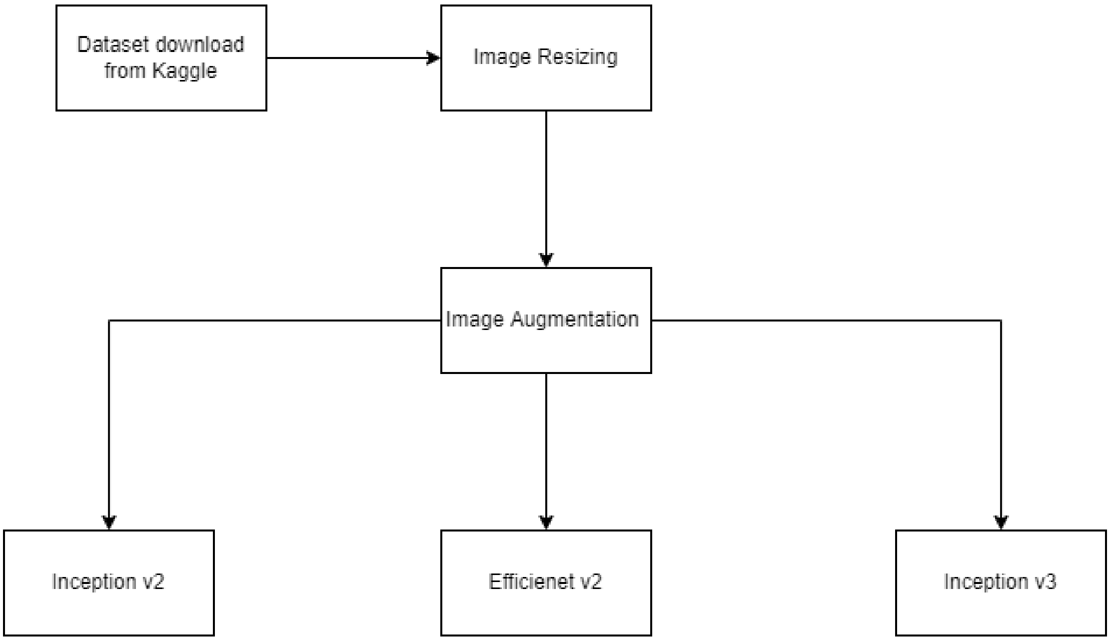
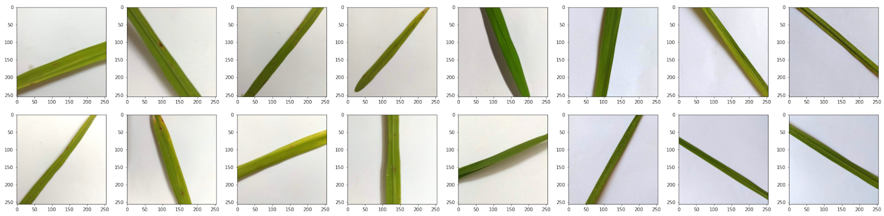


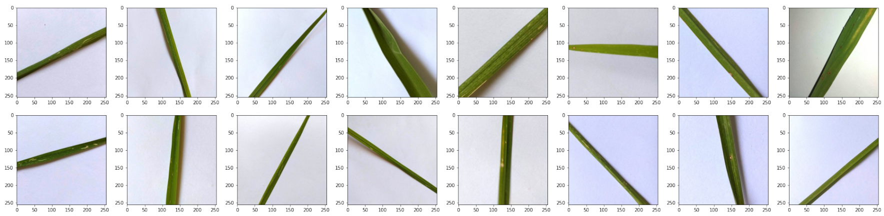
Figure 3.1: Block diagram of the procedure used

3.2.1 Dataset download from Kaggle

I have downloaded the Kaggle dataset for rice leaf diseases. Due to 16GB dataset size, the google drive is first mounted into our notebook using the drive class from the Google.Colab package. There are many different pictures of rice leaves in the collection. Pictures of both healthy and damaged leaves can be found in the dataset. 3355 photos of rice leaves make up our dataset, of which 2684 are used for training and 671 for testing and validation. The photos are 3120x3120 pixels in size and are further downsized to 255x255 pixels using the Python-provided Pillow package. Image Antialiasing is used to smoothen the jagged edges of the image by averaging the colors of the pixels at boundary. The images of different diseases are contained in different folders so labeling is done using that. In Fig. 3.2, we have shown different types of diseased images present in our dataset.



(a) Brown Spot (b) Healthy



(c) Hispa (d) Leaf Blast

Figure 3.2: Images of different kinds of paddy diseases

3.2.2 Working of Network Architecture

The suggested network performs classification using a sequential model. The implementation uses the Python language's Keras module. There are about 21 million parameters in the model. The optimization algorithm that we used in our model is Mini-Batch Gradient Descent(MBGD). In comparison to the traditional gradient descent algorithm, this algorithm is more effective. For model training, the provided photos are input in batches. 2684 photos from 4 distinct classes are used to train the model, with the image size set to 255x255 pixels. The batch size parameter that we utilised is 32. We describe data augmentation and methodologies in the subsections that follow, which describe how we developed the right algorithm to deliver reliable findings.

3.2.2.1 Data Augmentation

To make variations in the dataset, technique of Data Augmentation is used. It alters the input images by applying transformations, image blurring, and contrast adjustments. To help the model learn more effectively, it produces an expanded dataset with a variety of photos. In some cases, this method also helps to prevent overfitting. Instead of training the model on the complete leaf, only the affected portion of the leaf will be used. Therefore, we just consider the diseased portion. The in-place data augmentation technique is applied in our model. Python uses the Keras package to implement it. This method is used in runtime, which guarantees that a fresh converted image is sent into the network at each epoch, increasing accuracy.
In Fig. 3.3, we can find the augmented images of a rice plant leaf.

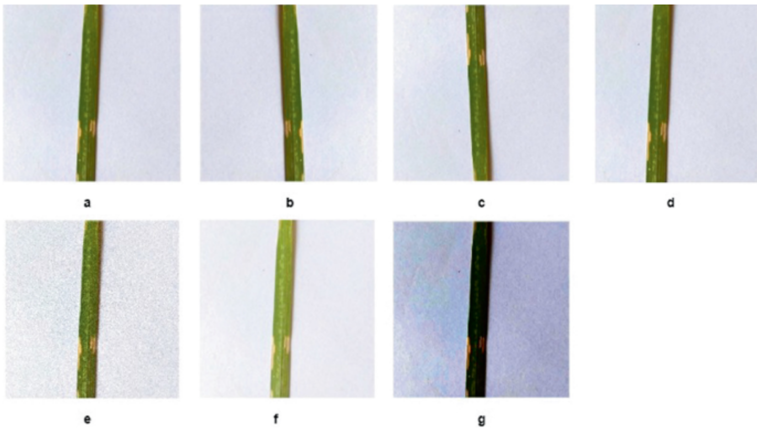


Figure 3.3: Augmented Images of a diseased rice leaf (a)Original Image, (b)Rotation by 180 degrees, (c) Vertical flip, (d)Gaussian noise, (e)Horizontal flip, (f)High brightness, (g)Low brightness

3.2.2.2 Transfer Learning

The transfer learning methodology makes use of a pre-trained neural network. It uses what it learns from one dataset to train on other datasets. Using this data to train on a sizeable dataset, it uses weights and biases, which are parametric variables. In order to classify an image, there are two processes, techniques for feature extraction and classification. Transfer learning enables the model to be trained on a new dataset for classification while reusing a component of the model that was used in the feature extraction process.

3.2.3 Models Used

In our approach, we have used 3 different models for transfer learning. We are training these three models and comparing them in terms of accuracy and efficiency. The three models are as follows:

3.2.3.1 EfficientNet v2

In terms of efficiency and training speed, the convolutional neural network EfficientNet v2 is much better that other models. To maximize the training speed, training aware neural architecture search and scaling is done.
There are the following primary architectural features:

1. Both MBConv and the recently developed fused-MBConv are heavily employed in the initial layers of EfficientNetV2.

2. EfficientNetV2 advises smaller expansion ratios for MBConv as they frequently have less memory access overhead.

3. EfficientNetV2 favours smaller 3x3 kernel size. To compensate the reduced receptive field, more layers are added to it.

The final stride-1 step in the original EfficientNet is completely deleted in EfficientNetV2 due to its huge parameter size and memory access overhead. In our method, we also expanded the EfficientNet v2 model by adding a dropout layer and two dense layers. To prevent the model from being overfitted, a dropout layer is included. To prevent overfitting, dropout layer selects inputs randomly, and set them to 0, scaling non-zero inputs up by 1/ maintains the total of all inputs (1 - rate). The model was trained using a batch size of 24 photos per batch and 10 epochs. The learning rate (step size) for each iteration is set at 0.0001.
In Fig. 3.4, we can find the architectural representation of the EfficientNet v2 model.

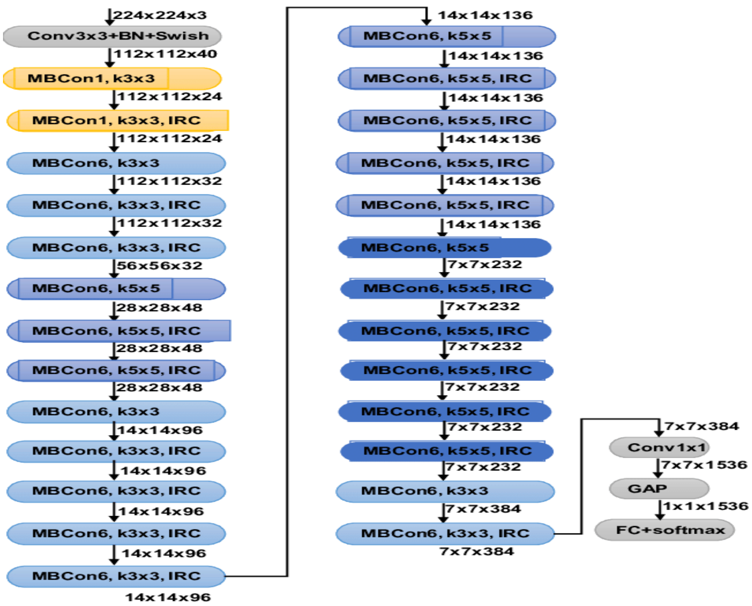


Figure 3.4: Architectural representation of EfficinetNet v2[8]

3.2.3.2 Inception v2

In the Inception V2 architecture, two 3x3 convolutions are used. Due to the fact that a 5x5 convolution is 2.78 more expensive than a 3x3 convolution, this also reduces

calculation time and hence boosts computation speed. Therefore, using two 3x3 layers rather than 5x5 layers improves architecture performance. Also nxn factorizations are transferred in 1xn and nx1 factorizations . As was mentioned above, a 3x3 convolution can be reduced to a 1x3 convolution, which is then followed by a 3x1 convolution, which has a 33% lower computational complexity than a 3x3 convolution. Instead of making the module deeper, the feature banks of the module were increased to address the issue of the representational bottleneck. This would prevent the loss of information that causes when we make it deeper. Fig. 3.5 shows the architectural representation of the Inception v2 model.
In our model, we have added one flatten and two dense layers which have activation functions as relu and softmax. The loss is calculated using the categorical crossentropy function. The optimizer used is Mini-Batch Gradient Descent. We are training the model with the learning rate of 0.001 and the epoch size is 10.

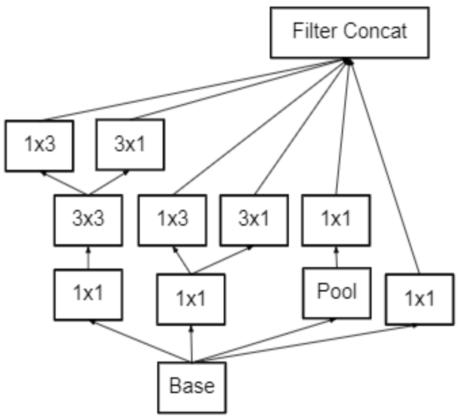


Figure 3.5: Architectural representation of Inception v2[1]

3.2.3.3 Inception v3

Inception v3 is an image recognization model. On the ImageNet dataset, it has been demonstrated that the Inception v3 can achieve higher than 78.1% accuracy. The model is the result of numerous concepts that have been established by various researchers over the years. Convolutions, average pooling, max pooling, concatenations, dropouts, and fully linked layers are some of the symmetric and asymmetric building components that make up the model itself. The model makes considerable use of batch normalisation, which is also applied to the activation inputs. To calculate loss, Softmax is used. The architectural representation of Inception v3 model is demonstrated in Fig. 3.6. In-

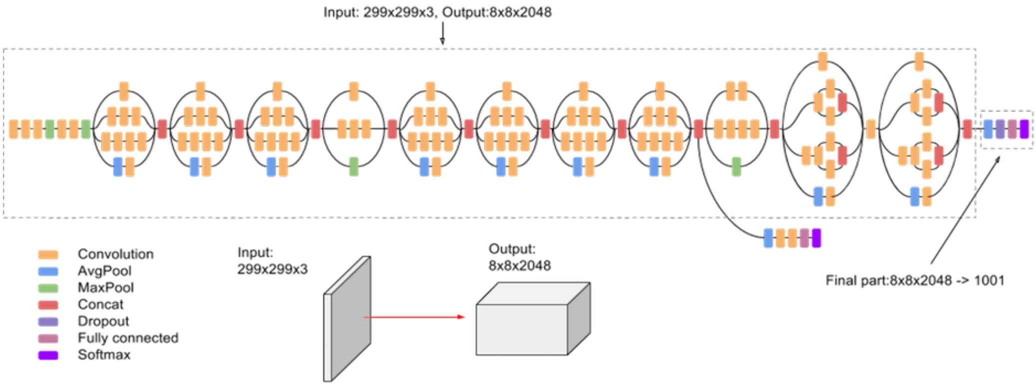


Figure 3.6: Architectural representation of Inception v3[1]

ception v3 model is used for transfer learning in our algorithm. It consists of a 48-layer deep neural network. To train this model, more than a million of images from more than 1000 different classes from the ImageNet Database were used. one flat and one dense layer are also added to this model. It prevents in the overfitting problem.

3.2.4 Optimization algorithm

Optimization algorithms are used to minimize or maximize a mathematical function F(x), when the model is trained iteratively. In the class of first-order optimization algorithms, Gradient Descent is one such method.

$$\theta = \theta - \nabla C\,(\theta; X_i, Y_i) \tag{3.1}$$

where θ is optimization parameter, C(θ; X_i,Y_i) is the cost, (X_i,Y_i) represents the data-point from the training set.
To process the dataset, MBGD technique is used. Dataset is partitioned with 32 images in each batch. MBGD combines both SGD and BGD techniques. BGD's flaw is that it iteratively navigates the full dataset to reach the minimum. SGD doesn't have the advantages of vectorization and also prone to error.

3.2.5 Categorical Cross-Entropy

To evaluate the model's effectiveness, loss functions are used. Loss function will increase if our predictions are different from the actual results. So, we should have small loss value for better predictions. Other loss functions are Mean Squared Error, Binary Cross-entropy, cosine similarity, etc. The one we are using is categorical cross-entropy, which can do multi-class categorization. The value of the correct class is 1, and of all other classes is 0.

$$L(y, \hat{y}) = -\sum_{j=0}^{m}\sum_{i=0}^{n}(y_{ij} * log(\hat{y_{ij}})) \tag{3.2}$$

where y is the actual value and $\hat{y}$ is the predicted value, m represents the total number of datapoints and n represents the total number of classes.

3.3 Conclusion

We can predict the diseases in rice plants by data augmentation, transfer learning and Mini Batch Gradient Descent optimizer using a small dataset of just 3355 images with good accuracy.

Chapter 4

Experiments and results

4.1 Experiment

We used 2684 images for training and 671 images for testing and validation to create this model. Data augmentation is used to generate a diverse dataset. The new output images are 299x299 pixels in size. We used the pre-trained network InceptionV3 structure to shorten training time and improve performance. It contains convolutional layers with biases and weights that have already been trained. A fresh collection of pictures is created after each epoch and sent to the inception model. It makes up the network's feature extraction section.
Our sequential model is then expanded with a dense and flat layer. These layers are used in the model's classification section. A convolution layer's multidimensional output is transformed into a linear output by the flattening layer so that it can be fed directly into the neural network layer or regular dense layer. Following that, the dense layer applies the Softmax activation function and completes the calculations.
As a loss function, categorical cross-entropy is employed. We evaluated the model using various learning rate values. The accuracy of the model initially grew as the learning rate increased, peaking at 0.001 with an accuracy of 90.35% before beginning to decline. As a result, the model's learning rate has a final value of 0.001. As an optimizer, mini-batch gradient descent is employed. Accuracy suffers when batches are small. For maximum accuracy, a batch size of 32 is employed.

4.2 Results

After the implementation of the alogrithm using Inceptionv3 architecture, we have been able to achieve an accuracy of 90.35% on the testing data and 98.75% on the training data of the Rice leaf dataset. This is a publically available dataset which have 2684 images for training and 671 images for testing, all this data is collected from internet.

We have used Mini-Batch Gradient Descent optimizer to train our model. We have used a batch size of 32 and have ran the algorithm of 12 epochs.
Instead of directly using the dataset of photos of rice plant leaves, we applied the Data Augmentation technique to increase the variety of training data, creating a model that is considerably more trustworthy and accurate while also avoiding the overfitting issue. The Transfer Learning approach is used, which accelerates and optimises the training procedure. Additionally, the network is expanded with flat and dense layers that aid in classification. The confusion matrix based on the anticipated outcomes is displayed in Fig. 4.1.

```
Confusion Matrix
[[ 80   7   5  13]
 [  1 272   3  21]
 [  0  17  81  15]
 [ 10   4   9 133]]
```

Figure 4.1: Confusion Matrix

The heatmap for the overall validation set is displayed in Fig. 4.2.

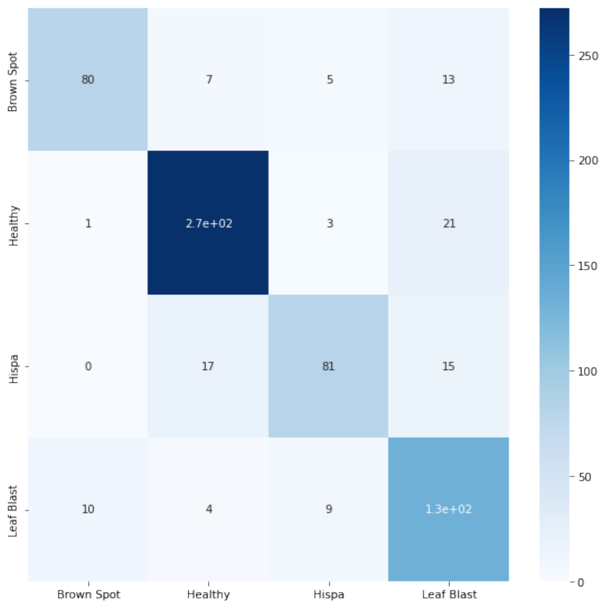
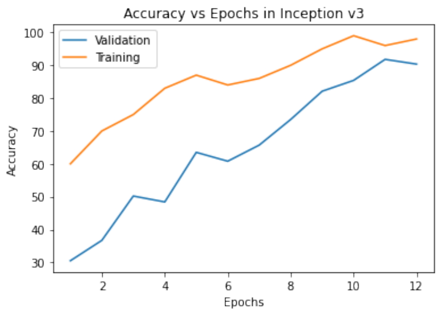
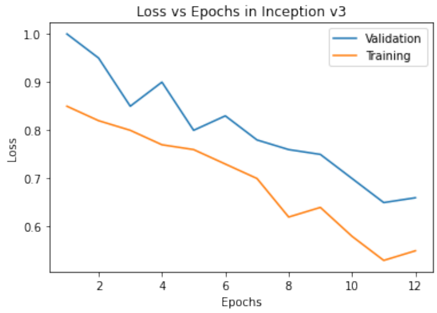


Figure 4.2: Heat Map

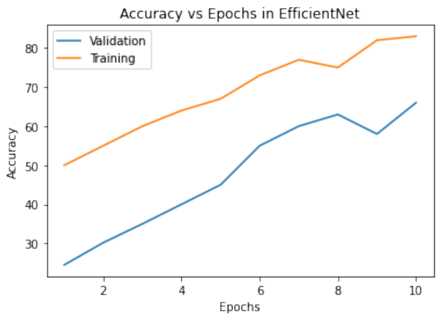
Fig. 4.3 shows the variations of accuracy and loss with respect to number of epochs in our algorithm.



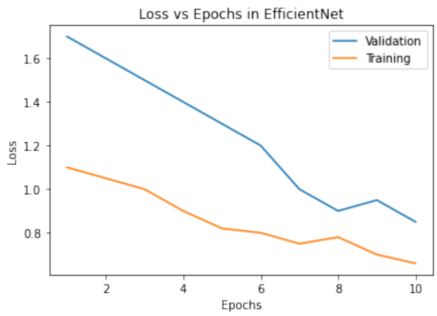
(a) Accuracy Graph for Inception



(b) Loss Graph for Inception



(c) Accuracy Graph for Efficientnet



(d) Loss Graph for Efficientnet

Figure 4.3: Results for Efficinetnet v2 and Inception v3

In our base paper, i.e., Detection and classification of rice plant diseases[7], they achieved an accuracy of 73.33% using the techniques of Image processing and machine learning. Using the same dataset, we have achieved an accuracy of over 90.35% using data augmentation and transfer learning methods.

4.3 Overall conclusion

Results above shows that we have been able to make the algorithm fast, that is we are able to make the algorithm more efficient while also not compromising much with the accuracy. Hence we achieved our target.

REFERENCES

[1] *Diseases in rice plants*: 2020, http://www.knowledgebank.irri.org/step-by-step-production/growth/pests-and-diseases/diseases.

[2] F. T. Pinki, N. K. and Islam, S. M.: 2019, "content based paddy leaf disease recognition and remedy prediction using support vector machine, *20th International Conference of Computer and Information Technology* .

[3] H. B. Prajapati, J. P. S. and Dabhi, V. K.: 2020, Detection and classification of rice plant diseases, *Intelligent Decision Technologies* **12**.

[4] Kaur, R. and Kaur, V.: 2020, A deterministic approach for disease prediction in plants using deep learning, *The International Arab Journal of Information Technology* .

[5] *Pillow library documnetation*: n.d., https://pillow.readthedocs.io/en/stable/.

[6] *Rice leaf disease dataset*: n.d., http://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases.

[7] S. R. Maniyath, P. Vinod, M. N.-R. P. N. S. R. H.: 2019, Plant disease detection using machine learning, *2019 International Conference on Design Innovations for 3Cs Compute Communicate Control* .

[8] Saad Albawi, Tareq Abed Mohammed, S. A.-Z.: 2018, Understanding of a convolutional neural network, *IEEE Explore* .

[9] T. Islam, M. Sah, S. B. and RoyChoudhury, R.: 2020, A faster technique on rice disease detection using image processing of affected area in agro-field, *Second International Conference on Inventive Communication and Computational Technologies* .