Department of Electrical and Computer Engineering
North South University

CSE 299 Junior Design

Corn Disease Detection Using Machine Learning

Section: 12

Group: 5

Md. Mahbub Hasan Rakib – 1813223042

Ajmain Ahmed Prottay - 1821756642

Mohammed Newaz Sharif – 1911191642

Faculty Advisor:

Dr. Tanzilur Rahman
Assistant Professor
Department of ECE

Spring 2022

Table of Contents

Corn Disease Detection Using Machine Learning

Abstract: The field of computer science is progressing at such a quick rate that we are seeing more and more advanced technology every year. Artificial Intelligence is one of the most promising of these developments. With the help of AI we can tackle problems like detecting flaws or predicting things with great precision, and in some circumstances, they have even outperformed people. With the growth of technology and artificial intelligence, we can create systems that can assist farmers remotely without creating too much trouble.

During the growth and production stages of corn, farmers face a complex issue in accurately diagnosing corn crop diseases. Traditional disease detection methods are ineffective because majority of the time farmers are unaware of the disease that has infected their plants and getting help from the government in remote areas is not always possible. Furthermore the farmers only know that their fields are affected only when the large portion of their fields are infected.

Farmers will be able to detect diseases and take the necessary steps to diagnosis them if they can test their fields every 7 days during the growing stage of corns. As a consequence, they will be able to diagnose crops early, which will substantially reduce production loss. To solve this issue, we are building a deep learning model and developing an android-based application that will employ our deep learning model to detect and properly diagnose three most common corn leaf diseases: leaf blight, leaf gray spot and rust. Our program can be utilized in the agricultural sector to protect crops by detecting them early.

1. Introduction:

Bangladesh is primarily an agricultural country, and the agriculture industry is critical to the country's economic prosperity. Among all the grain corps, maize/corn is the third most significant. It has huge export potential but the main obstacle in corn production are various kinds of diseases. There are more than 20 major diseases in corn and farmers are not qualified enough to identify them moreover lack of experts are also there. As a result farmers have to face less production each year.

As a result, we want to use machine learning and Android to create a system that can assist farmers in detecting plant diseases. Without using the internet, our app will be able to snap images with the phone and detect plant disease. The appearance of the leaves can reveal the problems that we are willing to work on. Gray spot, Leaf Blight, and Rust all create different markings and patterns on corn leaves, allowing us to recognize them from healthy corn leaves. However, prior knowledge of diseases is required to identify the exact disease. Our goal is to create a system that can diagnose a specific disease from leaf images.

1.1 Machine Learning:

Machine learning allows computers to perform jobs that were previously solely performed by humans. It is a branch of Artificial Intelligence that allows software applications to improve their prediction accuracy without being expressly designed to do so. In order to predict new output values, machine learning algorithms use historical data as input. For example, in our project, we will supply data on healthy and diseased corn leaves, and our computer software will learn how to determine which corn leaf is healthy and which has a disease using machine learning methods. The computer first assigns particular values to data features in order to link them, and then it measures the accuracy using loss functions. The system then sets new values for addressing the problem based on the outcome.

1.2 Android:

Android is by far the most popular operating system on the planet. As of January 2022, Android has 69.74 percent of the mobile OS market share.**[1]** Apple's iOS, which is the second-most-popular mobile operating system in the world with 29.49 percent, pales in comparison. The official

Google Play Store has over 2.6 million apps accessible, but you can also sideload apps from the web. Because of this variety, android phones are extremely powerful and adaptable.

1.3 Motivation:

Healthy and diseased plants can be identified by their leaves. Healthy leaves are green fresh and without any spots whereas disease leaves carry different patterns of spots of different colors rather than green. The diseases can be cured easily if one can identify them. So, here we are stepping in to make a product that will ease the process of identifying diseases. Our goal is to create a system that can identify a specific disease from leaf images.

So, we want to develop an application that can assist the farmers with recognizing the illnesses in plants with the assistance of Artificial Intelligence. Through our application farmers will be able to take pictures from the phone and detect disease without having access of the internet connection.

2. Background Studies:

To properly understand the concept, we chose the following 3 research papers for our study.

2.1 Identification of Diseases in Corn Leaves using Convolutional Neural Networks and Boosting.[2]

Published By: TCS Research and Innovation, Mumbai, India

For classification they used different methods such as for architecture they used VGG-16, Inception-V2, ResNet-50, MobileNet-v1 and for Activation used Softmax, Random Forest and SVM to figure out which combination gives the best performance. They achieved a test accuracy of 98% with classification score of {precision, recall, f1-score} = {0.97, 0.98, 0.97}.

2.2 Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition.[3]

Published By: Amity University, Lucknow Campus, India

Their network is fed with 150X150 input pictures. For all weights, they used a batch size of 32, with Gloret uniform initialization. Except for the last layer of the network, their model uses max-pooling operation with 2x2 pool size and relu activation function in all pooling layers. The last layer's output is a prediction of corn leaf disease using the softmax activation function. To

attain accuracies above 96 percent, they modified network hyper-parameters such as learning rate and maximum epoch throughout the training phase. Their learning rate's optimal value is.0004. Then, using feature visualization, they created a lightweight model for real-time data inference. Their validation process uses a total of 679 photos from the database.

2.3 High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module.[4]

Published By: College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

They used different architectures with multi activation function to show the comparison with tradition machine learning algorithm that demonstrates outstanding performance of CNNs in corn leaf disease detection and for dataset, they used PlantVillage dataset. The comparison of their different architecture is given below:

Model	Tanh	ReLU	LeakyReLU	Sigmoid	Mish	Accuracy
SVM						83.18%
RF						87.13%
baseline						92.82%
MAF-AlexNet	✓	✓	✓	✓	✓	93.11%
	✓		✓	✓		93.49%
		✓	✓	✓		92.80%
baseline						93.92%
MAF-VGG19	✓	✓	✓	✓	✓	94.93%
		✓		✓	✓	95.30%
	✓			✓	✓	95.18%
baseline						95.08%
MAF-ResNet50	✓	✓	✓	✓	✓	95.93%
	✓	✓		✓		97.41%
			✓	✓	✓	96.18%
baseline						96.18%
MAF-DenseNet161	✓		✓	✓		95.90%
	✓			✓	✓	96.75%
	✓	✓		✓		97.01%
baseline						94.27%
MAF-GoogLeNet	✓	✓	✓	✓	✓	95.01%
	✓		✓	✓		95.09%
	✓		✓	✓		94.27%

3. Methodology:

For our project we are building a system that will predict plant diseases using a smartphone. We must go through a number of steps in order to achieve our aim.

We divided our project into 3 sections:

1. **Dataset:** Exploration, Preprocessing and Augmentation
2. **Model:** Exploration, Training and Evaluation
3. **Android Application:** Development, Deployment and testing

From week 1 to week 5 we have worked on Dataset and Model building. We are working on model to make it stable enough so that it can accurate detect diseases that we are aiming for. Till now our model have achieved training accuracy of 93% and validation accuracy of 94%. When we are done with model building. Only then we will develop android application for deployment other than that we will keep working on the model.

3.1 System Diagram:

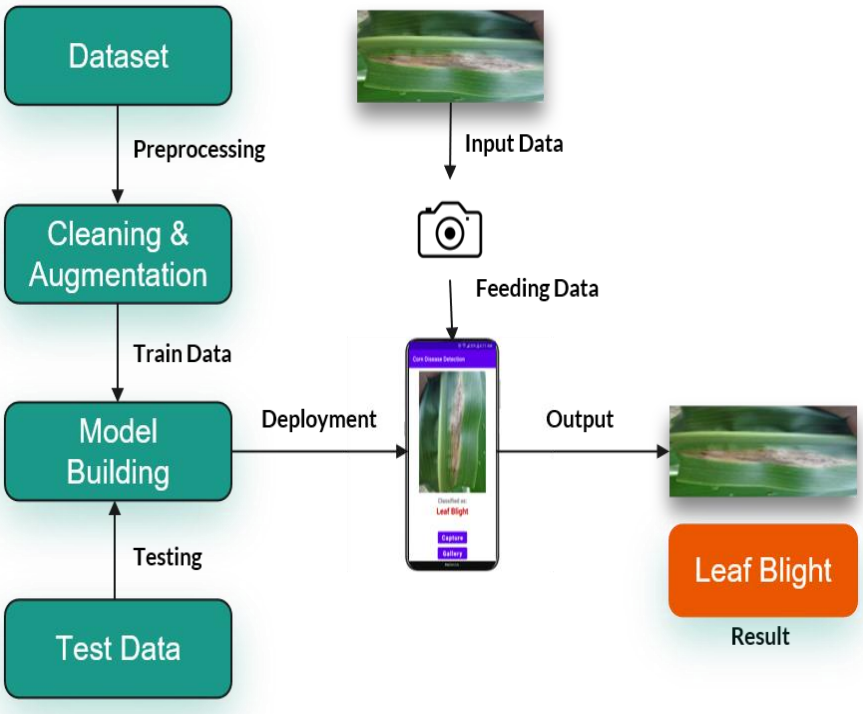


Figure 1. System Diagram

3.2 Software & Technologies Required:

a) Model Building:

a. **TensorFlow & Keras:** TensorFlow and Keras are Python frameworks for developing deep learning and machine learning tasks. These include all of the algorithms and features required to build, train, and process machine learning and deep learning models.

b. **Google Colab:** Google Research's Colaboratory, or "Colab" for short, is a cloud-based platform that allows anyone to write and run arbitrary Python code. It's notably useful for machine learning, data analysis, and education. Colab provides us with all of the essential training modules, software, and hardware.

b) Android Application:

a. **Android Native:** The Native Development(NDK) is app development framework. It employs Java/Kotlin as its programming language. Flutter is readily incorporated into Android Studio for developing Android apps. It includes platform frameworks for managing native activities and interacting with physical device components like sensors and touch input.

c) Dataset:

a. **Kaggle:** Kaggle is a dataset repository. Users can use Kaggle to search and publish data sets, explore and construct models in a web-based data science environment, collaborate with other data scientists and machine learning experts, and compete in data science challenges.

3.3 Dataset Exploration:

A proper dataset is necessary for all phases of classification and detection tasks. In this project, we used an open-source dataset obtained from the Mendeley Data repository (ARUN PANDIAN J, 2019).**[5]**

In this data-set, 39 different classes of plant leaf and background images are available. The data-set containing 61,486 images. We used only the corn leaves dataset categorized into 4 different classes (Healthy, Cercospora leaf spot, Common rust, Northern leaf blight) containing 3,852 images.

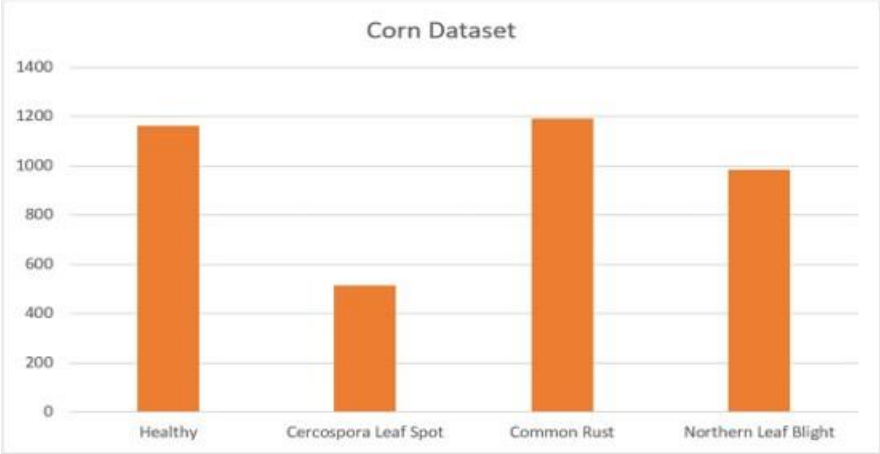
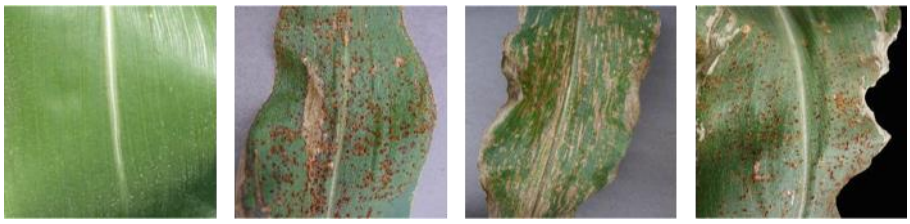


Figure 2. Dataset



a) b) c) d)

Figure 3. Corn leaf Images from the PlantVillage database
(a) Healthy (b) Leaf Blight (c) Leaf Spot (d) Common Rust

3.4 Dataset Process and Augmentation:

When there aren't enough training samples, the data augmentation method is used. The training of the network model will be insufficient or the model will be overfitting if the sample size of the training set is too small. To avoid overfitting the model and improve accuracy on images collected under various situations, data augmentation is done with rotation, flipping, cropping, decolorized etc.

First we will visualize data to see if any corrupt data is there then we will resize if it is not in appropriate data size for any particular task. In our case data is already resized and the resolution is 256X256. We will further resize it to 256X256 from 256X256 to make sure that all the data we feed in to our model remain same in dimension.

Data Distribution: We will split the data into 3 parts Training, Validation and test and the ratio will be 80:10:10

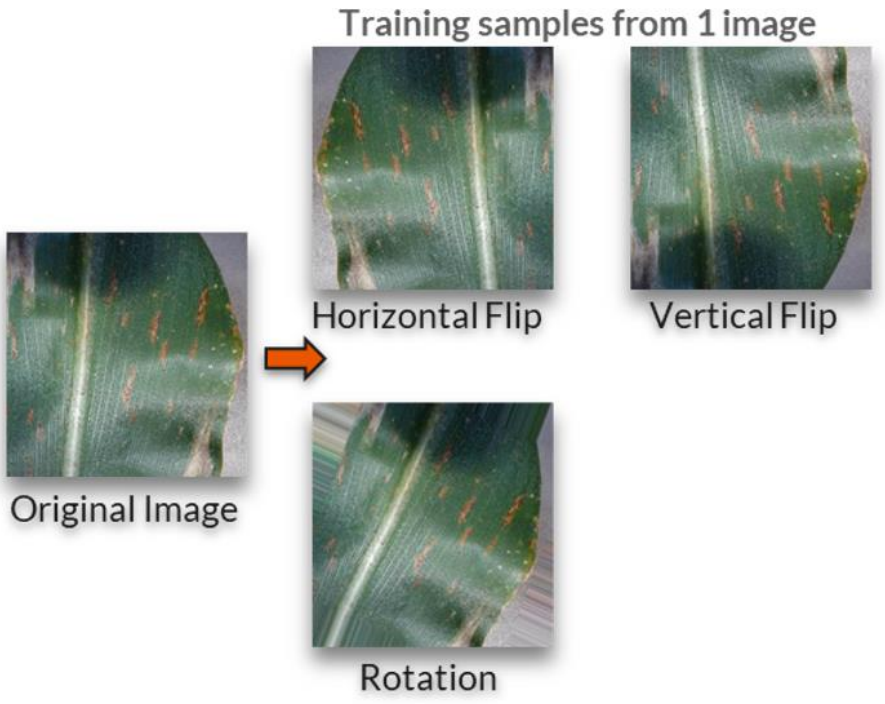


Figure 4. Augmented Data

3.5 Model Exploration:

For our project we choose to work with CNN architecture with ReLU and Softmax Activation Function.

CNN: A CNN architecture is made up of a series of discrete layers that use a differentiable function to transform the input volume into an output volume. There are a few different sorts of layers that are regularly utilized.

ReLU: Relu makes zero if given input is negative and if greater than or equal to 0 the output remains same. Fast and More computationally efficient to compute.

Softmax: Sigmoid delivers values in the range of 0 to 1, which is appropriate for probabilistic distributions. As a result, sigmoid is used to classify binary data. So, we're utilizing Softmax because we're working with four classes.

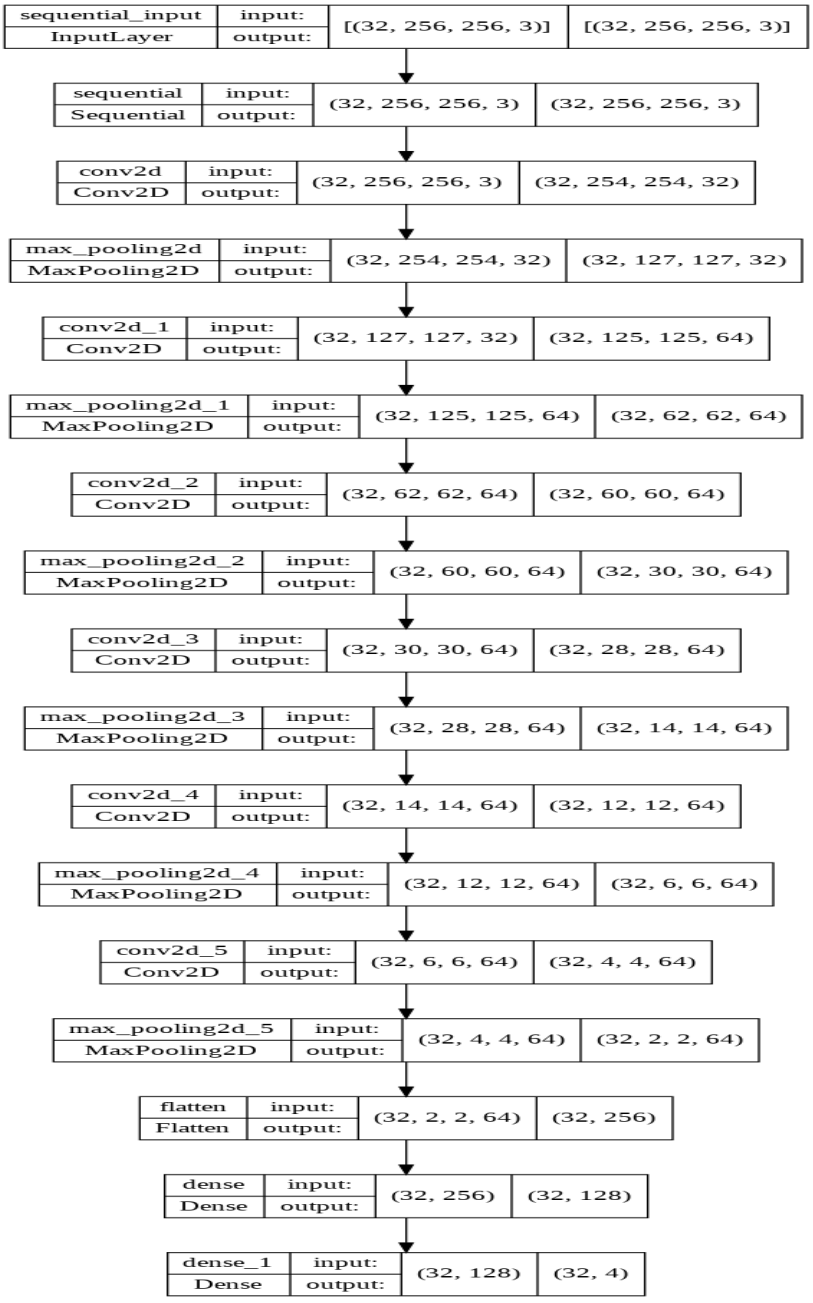


Figure 5. CNN Architecture

We have used 6 convolutional layers with kernel size of 3X3, each followed by 6 max-pooling layers with kernel size of 2X2. We can't take more than 6 CNN layers as our picture size is 256X256 if we take more than 6 CNN layers the size of the picture becomes less than zero or negative. Thus we took the maximum 6 layers. Then we flatten it to convert it into 1 dimensional array. In the last layer we used softmax activation function for classification as we are working with 4 classes.

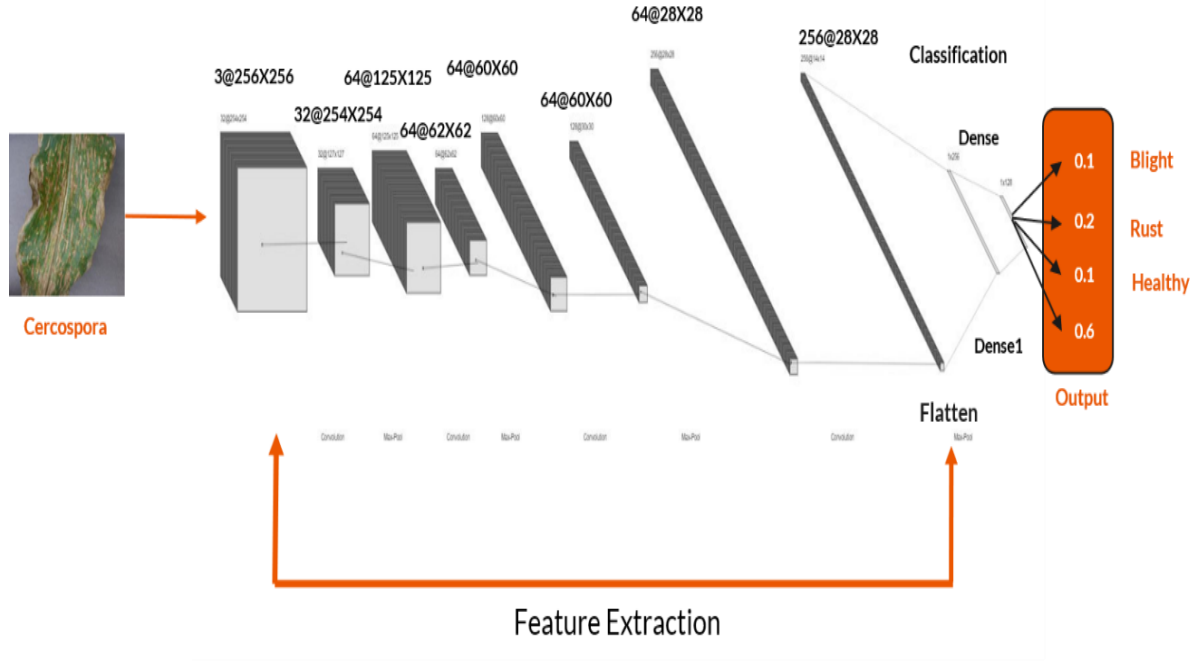


Figure 6. CNN Diagram

3.6 Model Training and Evaluation:

The step where we ultimately train our model with the dataset we've picked is called model training. Machine learning algorithms assist the model in learning from the data provided throughout this process. Simultaneously training and testing the model is performed. The loss was calculated using Categorical Cross Entropy. It's a metric for calculating loss in multiclass detection tasks. We used the Adam optimizer as the model's optimizer, which can be used in place of the SGD approach. The optimizer is the central algorithm that aids the model's learning. SoftMax was implemented in the very last layer of our model. This function returns probabilities for each class in a multiclass classification issue.

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 256, 256, 3)         0

 conv2d (Conv2D)             (32, 254, 254, 32)        896

 max_pooling2d (MaxPooling2D  (32, 127, 127, 32)       0
 )

 conv2d_1 (Conv2D)           (32, 125, 125, 64)        18496

 max_pooling2d_1 (MaxPooling  (32, 62, 62, 64)         0
 2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_2 (MaxPooling  (32, 30, 30, 64)         0
 2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_3 (MaxPooling  (32, 14, 14, 64)         0
 2D)

 conv2d_4 (Conv2D)           (32, 12, 12, 64)          36928

 max_pooling2d_4 (MaxPooling  (32, 6, 6, 64)           0
 2D)

 conv2d_5 (Conv2D)           (32, 4, 4, 64)            36928

 max_pooling2d_5 (MaxPooling  (32, 2, 2, 64)           0
 2D)

 flatten (Flatten)           (32, 256)                 0

 dense (Dense)               (32, 128)                 32896

 dense_1 (Dense)             (32, 4)                   516

=================================================================
Total params: 200,516
Trainable params: 200,516
Non-trainable params: 0
_________________________________________________________________
```

Figure 7. No of total features

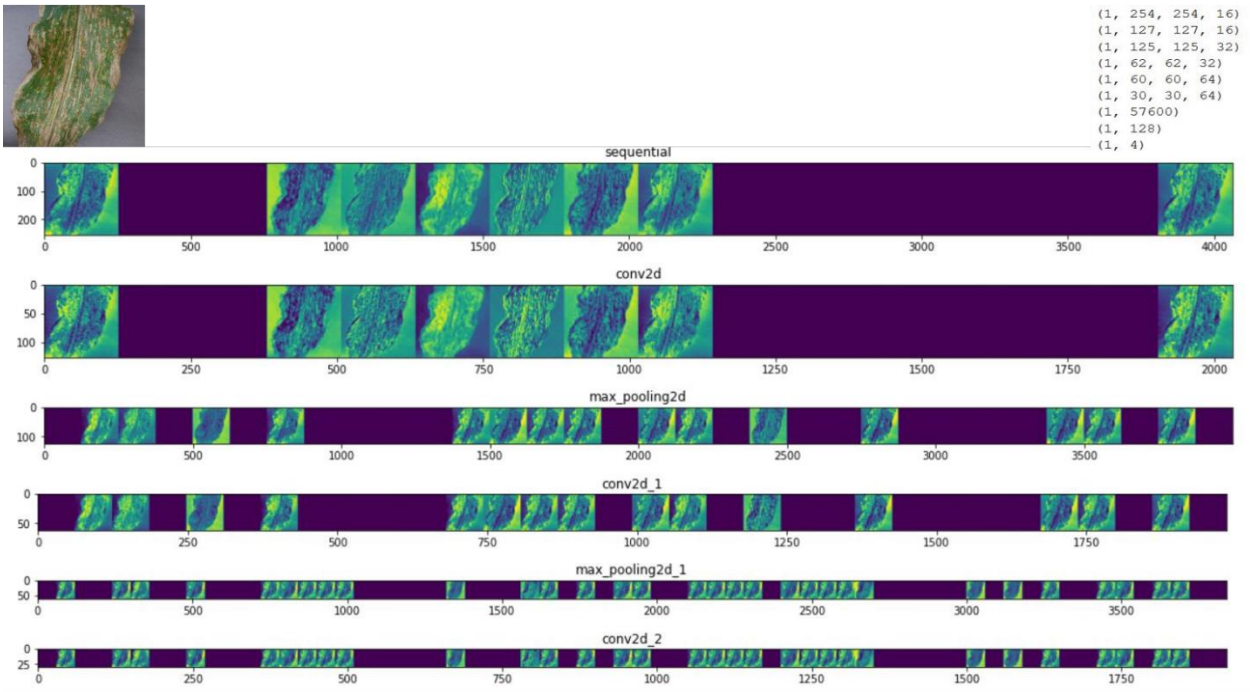


Figure 8. Feature map

We can see in the figure 8 that after each convolutional layer the size of the image is decreasing while keeping the feature. And there are some Black section in the figure that suggests that in the model haven't find anything new to train and learn.

3.7 Classifying Real World Images:

When testing with unbiased data we found that one of the class that is common rust can not be identified. Instead it predicts Northern Leaf Blight. We found that 2 things that was causing this problem.

a) Black Background
b) Multiple disease within single image

3.8 Black Background Playing a Role:

One of the class contains images with black background and that is common rust. When we test common rust image with black ground our model predicts with 100% accuracy. But when we cropped the same picture and omit the background our model predicts different class.

Figure 11. Prediction with Black Background

Figure 12. Prediction without Black Background

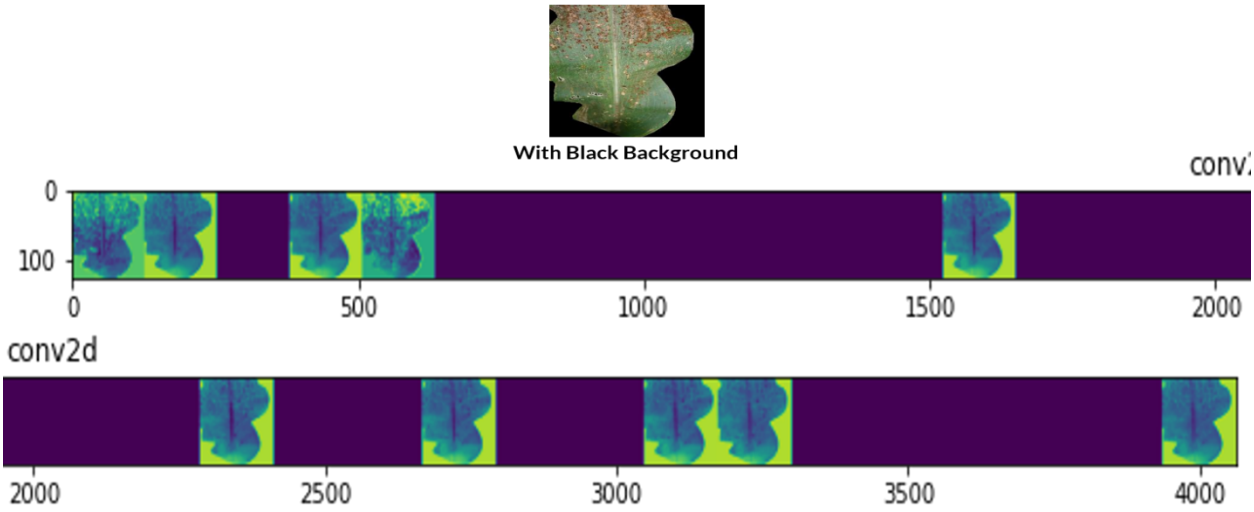


Figure 13. Feature Map of image with Black Background

With feature map visualization we can see that black background is counted as feature.

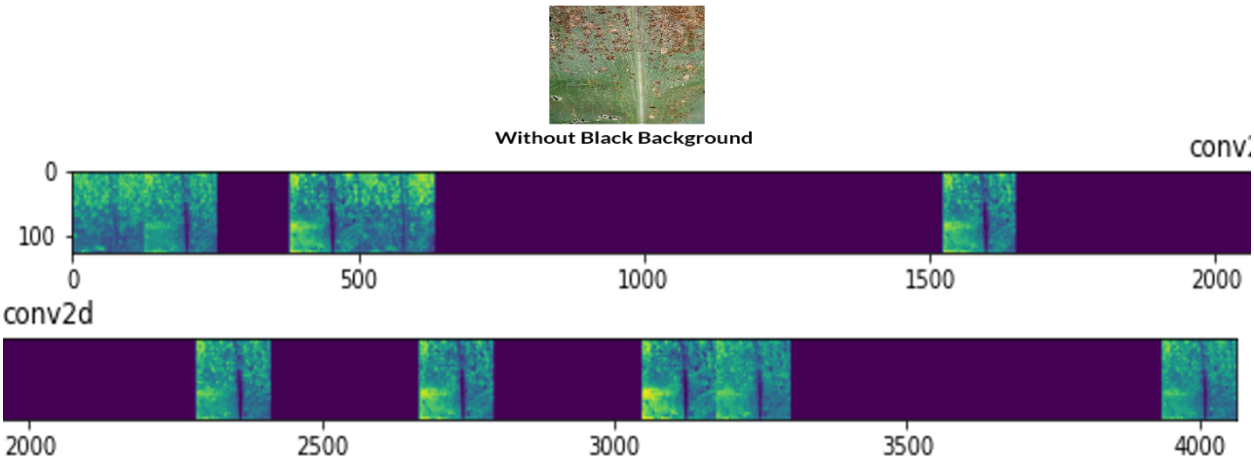


Figure 14. Feature Map of image without Black Background

So, when we cropped the image, we found that now the features are being extracted correctly.

3.9 Multiple Disease Within Single Image:

We figured out the reason behind mispredictions and the reason is, Apart from Healthy, other 3 classes contain images that have multiple diseases in single image.

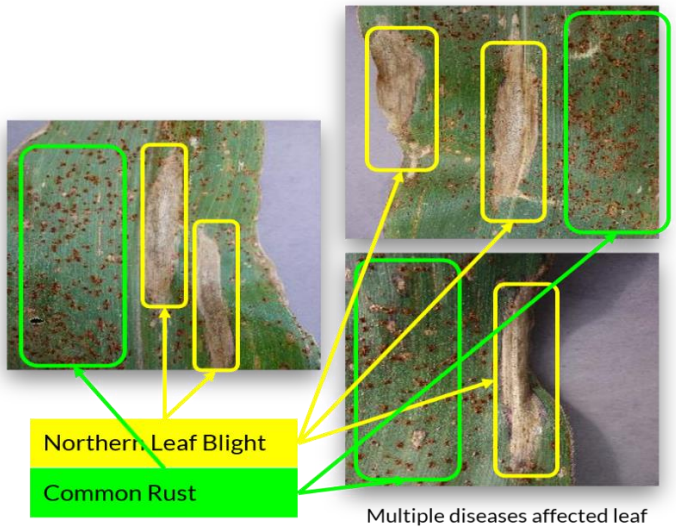


Figure 15. Multiple disease carrying leaf

Multiple disease carrying leaves are one of the reason for wrong prediction. To solve this issue we have removed those images that contains multiple diseases. Our dataset size was 3852. We have found 764 images that have multiple diseases. After cleaning dataset now we have dataset size of: (3852 - 764) = 3088

3.10 Android Application:

We converted our model into tflite format using tflite converter to work with mobile devices. TF-Model size is 11.6 MB. After converting Tf-lite Model size is 4.2 MB. Then we designed and implemented the model in our application with basic UI and functionality. Our application have 2 features it can capture a image using the camera and it can access gallery for image. Once the picture is provided the application returns the prediction.

Application Details:

Apk Size: 15.56MB

Minimum OS: Android 7 (Nougat)

Permission Required: Camera & Storage

4. Results:

For the corn disease detection in PlantVillage dataset after training with 50 epoch we got the accuracy of 97% in validation and 95% in training.

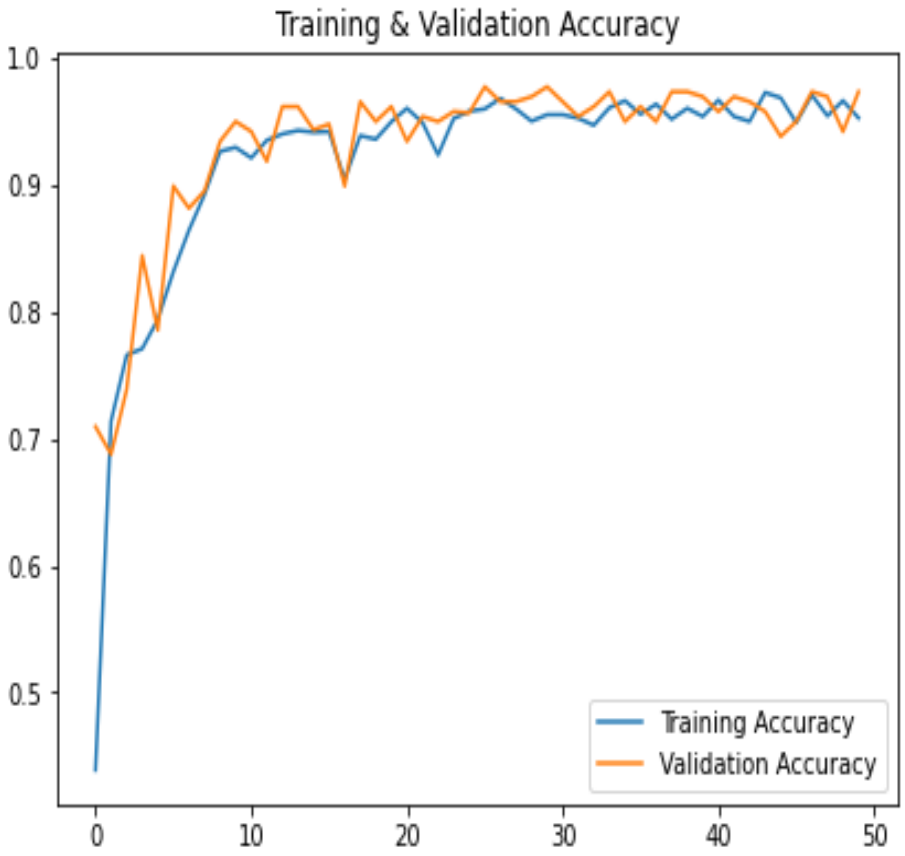


Figure 10. Training and Validation Accuracy Graph

Study	Model (Best Performing)	Dataset Distribution (Train:Validation:Test)	Train Accuracy	Validation Accuracy
1	Dense Net	Not Mentioned	Not Mentioned	96.18%
2	VGG-16	8:1:1	Not Mentioned	97.41%
3	Inception-V2	7:2:1	98%	90%
Our Model	CNN	8:1:1	95%	97%

Table 1. Result compared to Background studies

Class	Correct Prediction	Wrong Prediction	Accuracy
Healthy	15/15	0/15	100%
Common Rust	14/15	1/15	93.3%
Grey Leaf Spot	13/15	2/15	86.6%
Northern Leaf Blight	14/15	1/15	93.3%

Table 2. Application Accuracy Test

Device	CPU	Ram	Android Version	Initialization Time(ms)	Avg Latency (ms)
Pixel 2	Snapdragon 835	4GB	8	29.3	26.7
Galaxy S10	Exynos 9820	8GB	12	38.9	19.1
OnePlus 6	Snapdragon 845	6GB	11	22.4	26.6
Redmi 5i	Snapdragon 665	3GB	9	102	55.4
Poco X2	Snapdragon 730	8GB	11	39.9	31.1
Samsung J2	Exynos 3475	1GB	5	NA	NA
Redmi 4	Snapdragon 430	2GB	6	NA	NA

Table 2. Application Compatibility Test

(Red marked devices does not work due to lower OS version)

Across all devices, the application always identified correctly and never crashed. However, depending on the storage kind of the device we were testing, we found some lag. The application, for example, operated much faster on UFS 3.0 storage devices than on ordinary storage systems.

5. Discussion:

Though we tried to build a corn disease detection system, our main goal was to implement the idea that we can detect plant diseases with our phones without accessing the internet. Throughout the development of this project, we faced various challenges and learned various workarounds.

We mainly struggled with our dataset. The development of the model and application was not that challenging but working with the data was. We faced various performance issues due to our first dataset and we figured out that black background was playing a role. So we cropped the images and remove the background and It is working now. We concluded that the previous dataset was not made properly. So in conclusion we can say that the dataset can be the most important thing in Ai development. If the dataset has issues, no matter how many sophisticated algorithms we use we won't get proper results.

Last but not least we have a drawback in our app. When we try any picture other than that belongs to 4 of the classes we are working with our app gives prediction from the 4 classes.

6. References:

1. https://gs.statcounter.com/os-market-share/mobile/worldwide
2. https://www.sciencedirect.com/science/article/pii/S187705092030702X
3. https://www.scitepress.org/Papers/2019/76876/76876.pdf
4. https://www.mdpi.com/2072-4292/13/21/4218/pdf
5. https://data.mendeley.com/datasets/tywbtsjrjv/1

Group Contribution:

1. Md. Mahbub Hasan Rakib-1813223042:
    - Model Implementation
    - Android Application Development

- Slide Preparation
- Android Application Testing(Virtual IDE)

2. Ajmain Ahmed Prottay-1821756642
    - Background Study(Exploration, Analysis)
    - Real-World Android Application Testing
    - Android Application Testing(Virtual IDE)
    - Image Enhancement (Background)
3. Mohammed Newaz Sharif-1911191642
    - Data Exploration(Corn Disease)
    - Data Processing
    - Android Application Testing(Virtual IDE)
    - Raw Data Collection.