Intelligent Disease Management in Maize using Deep Learning Techniques

Karnika Bhardwaj, Sakshi Jha , Sanjay Keshwar, Savitri Khyadad, Ms. Priyadarshini Patil,Dr. Meena S.M

KLE Technological University, Hubli, Karnataka 580031, India.
https://www.kletech.ac.in

Abstract. Maize is one of the most significant crops in India. Every year the Maize gets affected by various fungal diseases. Diseases such as Northern Blight, Southern Blight, and Rust are causing a reduction in the yield to an extent of 28–91%. Usually, at the flowering stage, the symptoms start to appear on the leaves in the affected plants. It has been recognized by the agricultural sector that, acquiring relevant information about plants is needed for crop management to thrive. However, studies have shown that agricultural problems remain difficult in many parts of the world due to the lack of the necessary infrastructures and knowledge. In this paper, we have carried out an automation system for detection of maize diseases using image processing and deep learning techniques. Automating the process of disease detection in plants would lead to a rapid increase in the yield. To address this challenge we use convolutional neural networks(CNNs). Different CNNs were trained for classification and their respective results were then compared. The results obtained are encouraging for five classes with an accuracy of 79.40% using VGG19.

Keywords: Machine Learning,Deep Learning, Disease Management, Maize Disease Classification.

1 Introduction

According to resources,58% of India's population is supported by agriculture. There has been an increase in the contributions of agriculture to the economy of India.The contribution of agricultural sector to the GDP is about 20%.There has been a lot of improvements and advancements in the field of agriculture. Introducing artificial intelligence,IoT devices and other technologies has led to a greater yeild.Maize is cereal grain grown in several parts of the world.It is one of the most important crops in India.It is sensitive to excessive moisture as well as the moisture stress.Maize is often referred to as the queen of cereals.Consumption of maize has been increasing over the years.In order to meet the rapidly increasing demand of maize,there is a need to increase the production.However,various diseases that affect maize crop are reducing the yield.There is a need to pay more attention to the disease management in maize.In today's time, artificial Intelligence is contributing to the agriculture industry in many aspects. In addition to yielding healthy crops AI plays a role in controlling pests, monitoring soil, irrigation, harvesting etc. Artificial Intelligence is constantly making an effort to increase the yield of crop. Techniques like image processing and Big Data Analysis are being used to make judgements on soil fertility, crop quality and pest infestation. Smart agriculture based on geospatial technology, IOT, sensor technology, machine learning and high performance computing can mark the existing production challenges related to cropping system optimization for improving productivity and reducing environmental impacts.We propose a system which classifies the given image using convolutional neural network pretrained on ImageNet data. Classification is done using different CNN architectures like VGG-19, ResNet50 and Inception-V3 used with softmax classifier. Features extracted from VGG-19 gave the highest accuracy.

2 Literature Survey

Maize is the third most important food crop in India after rice and wheat and contributes about 9% in the national food basket. Maize is one of the most versatile crop. In addition to staple food for human being and quality feed for animals, maize serves as a basic raw material as an ingredient to thousands of industrial products that includes starch, oil, protein, alcoholic beverages, food sweeteners, pharmaceutical, cosmetic, film, textile, gum, package and paper industries etc. Maize is cultivated throughout the year in all states of the country for various purposes. The predominant maize growing states that contributes more than 80% of the total maize production are Andhra Pradesh (20.9%), Karnataka (16.5%), Rajasthan (9.9%), Maharashtra (9.1%), Bihar (8.9%), Uttar Pradesh (6.1%), Madhya Pradesh (5.7%), Himachal Pradesh (4.4%). Like all the crops, pest and diseases affect the yield of maize too. Prior identification of these diseases and pests improve the overall yield of the crop and reduce wastage.

Common diseases which affect maize are - Sorghum downy mildew, Non-Insect Pest-Maize cystnematode, Rajasthan downy mildew, Polysora rust,Common rust, Bacteria stalk rot, Fusarium stalk rot, Charcoal rot, Banded leaf and sheath blight, Curvularia leaf spot, Turcicum leaf blight (TLB) / Northern Corn Leaf Blight, Maydis leaf Blight (MLB) /Southern Corn Leaf Blight, Cercospora leaf spot (Gray leaf spot). Pest that attack maize are - Spotted Stem borer, chilo partellus, Pink stem borer, Fall Armyworm, Shoot fly.

[1] classifies the problem into tasks-biotic stress classification and severity estimation making it a multitask system. The dataset in [1] consists of the following stresses-leaf minor,rust,brown leaf spot and cercospora leaf spot. It also included healthy leaf giving a total of 1747 images of coffee leaf. The photos were manually collected. Among the 5 trained network used, ResNet50 was the one that obtained the best result.[2] presents a pipeline for the visulatization and classification of agricultural pest insects by computing a saliency map and applying deep convolutional neural network (DCNN) learning.

3 Challenges

In today's technical world Artificial Intelligence and concepts of Machine Learning are used in number of agricultural applications. But we have to keep this in note that Machine Learning cannot be the entire solution. Gaining trust of users is always a key challenge for any provider. Ensuring farmers to adopt new technologies is not proven easy. Due to lack of understanding of practical applications of AI tools the farmers remain cautious to accept any unrecognized technical models. These new technologies also appear to be highly confusing and often expensive to use, hence challenge is not only about convincing them but also training them to implement the facility in their work.

Obtaining relevant data at farmer's level is another huge challenge in India due to privacy concerns. Appropriate dataset plays crucial role in predictive analysis. A good amount of information about the object is indispensable for a machine to identify or classify it. Authenticity of the data provides efficiency to the work and best results can be obtained by processing the authentic data in the methods of machine learning and it's real-time applications.The application that is developed by us is user-friendly.It gives the results with good accuracy.The farmer can just hold the device camera towards the affected maize leaf to identify the disease.

4 Proposed Solution

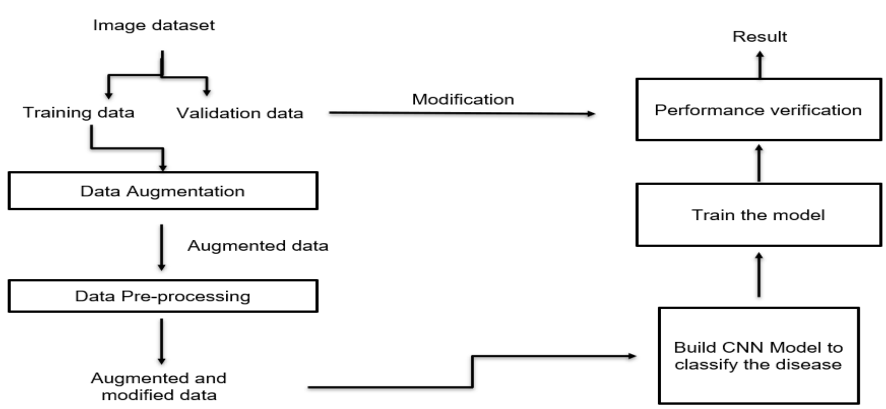


Fig. 1: Proposed System

The Fig 1. shows how the proposed system is designed. An android application is designed for live detection of maize diseases and pest infestation using

VGG19 architecture. The application starts accepting the live image as soon as the application is opened. The identified image must belong to one of the classes - Leaf Blight, Common Corn Rust, Gray Leaf Spot, Fall Armyworm infestation or Healthy. The percentage value for each class is displayed and the class holding the highest percentage value is the class to which the image belongs to. The application is supported by the android versions 4.0 and above.
The dataset is split into training set,validation set and, testing set.Data augmentation and preprocessing is performed on the dataset to obtain better results.A pre-trained VGG19 model is used to build the application.The model is trained on the training dataset and validated using the validation dataset.The model is further tested using the testing dataset.

4.1 Data Augmentation

Models trained on a small dataset tend to overfit. Overfitting is a scenario where a model fits the training dataset but performs poorly on the unseen data. Data augmentation is a technique used to increase the data by creating modified copies from the existing data. It acts as a regularizer and helps to prevent overfitting. We have used data augmentation techniques to create new images and hence increase the training set. It is carried out by using ImageDataGenerator class from Keras library. Augmentation techniques that we have used include zoom, rotate, and fill mode. Following is a brief explanation of these techniques.

- Zoom: The zoom augmentation technique generates new images by randomly zooming in or out within the given range
- Rotate: The images are rotated at some angle, and new images are created.
- Fill mode: This technique generates images by filling the points outside the boundaries of the input images.

4.2 Data Preprocessing

The raw data has to be preprocessed before training and evaluating the model. The preprocessing techniques used for our dataset include resizing and rescaling. All the images have to be in the same shape before they are used for training and evaluation, and it is done by resizing the images. The raw images have pixel values between 0 and 255, and these values are too high for the model to process, so the pixel values are rescaled between 0 and 1 to reduce the computational complexity. We multiply the data by the rescale value, that is, by a factor of 1/255. Once the preprocessing is done, the dataset is split into training data, validation data, and test data in the ratio of 60:20:20.

4.3 Models

Four CNN models were trained on the dataset. The models are- ResNet50, Inception V3, a CNN architecture built from scratch, and VGG19. Out of the four models, the model with the highest accuracy was selected to build an android application that can detect and classify the type of disease or pest infestation in the given input.

ResNet 50 is a CNN that is 50 layers deep. The pre-trained version of the network trained on the ImageNet dataset was used for training. The model takes an input of size 224 x 224. The number of neurons in the last layer was changed layer was changed to 5. L2 regularization was applied to reduce overfitting in the model. The model was then trained and tested. It achieved a training accuracy of 75.57% and a testing accuracy of 68.40%.

Inception V3 was another pre-trained model that was used to train the model. It is a 48 layer deep CNN model. The model pre-trained on the ImageNet dataset was used for training. It takes an input of size 224 x 224. . The number of neurons in the last layer was changed layer was changed to 5. Batch normalization is used in the model and applied to the activation inputs. The model was trained and tested on the dataset and achieved a training accuracy of 99% and a testing accuracy of 70.04%.

The third model that the dataset was trained on, was a CNN architecture built from scratch. The kernel size in all the convolution layers is 3 x 3. The first has 32 kernels and it is followed by a max pool layer with stride size 2. Max pool layer is followed by two convolution layers with 32 kernels and 64 kernels respectively. Another max pool layer with stride 2, followed by convolution layer with 64 kernels. The is followed by a max pool layer with stride 2 and a convolution layer with 64 kernels. There are 3 dense layers with 256, 128, and 64 neurons respectively. A dropout layer is added between the second and the third dense layer. The output layer has 5 neurons with a softmax activation function. The model takes an input of size 150 x 150. It has 9,47,045 trainable parameters. The model was trained and tested on the dataset and it achieved a training accuracy of 89.53% and a testing accuracy of 70.40%.

5 Implementation

5.1 Model Architecture

VGG19 architecture is selected to perform the classification task. It contains 19 layers. There are 16 convolution layers, 3 fully connected layers, 5 maxpool layers, and a softmax layer. Figure 1 represents the layers in VGG19. A group named Visual Geometry Group at Oxford created the architecture and hence the name VGG. A pre-trained VGG19 architecture is used for the work. The pre-trained VGG19 model is a model that is previously trained on a dataset and contains the weights of that dataset it was trained on. VGG19 is trained on the ImageNet dataset. ImageNet is an image dataset that consists of 14,197,122 images divided into 1000 classes.

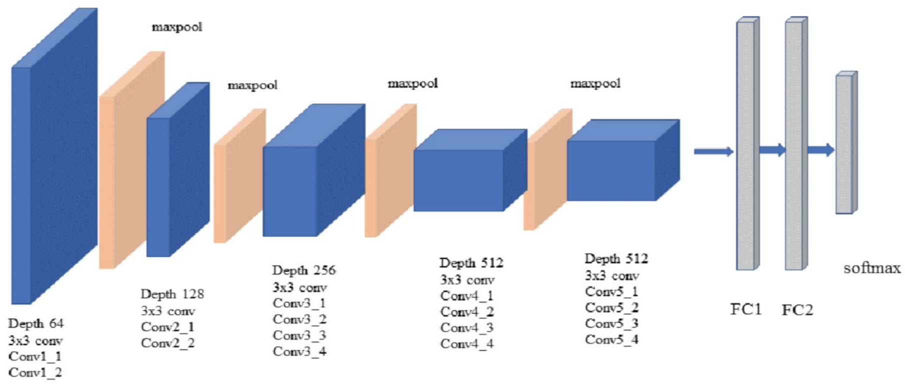


Fig. 2: VGG19 Architecture. source:[3]

VGG19 has a total of 138 million parameters. It takes a fixed size of 224 x 224 RGB image as input and uses kernels of 3 x 3 with a stride size of 1. Max pooling is performed over a 2 x 2 pixel window with stride 1. Each maxpool layer is followed by a ReLu unit. ReLu is a non-linear activation function that does not activate all the neurons at a time. It deactivates the neuron only if the input is less than 0. There are three fully connected layers, the first two with 4096 neurons, and the third one has 1000 neurons with a softmax activation function.

5.2 Model Compilation and Training

The TensorFlow and Keras libraries are used to perform the classification task. All the required python libraries are imported before starting the compilation and training of the model. In this work, a customized VGG19 model is used for the classification task. The pre-trained VGG19 has 1000 neurons in the last layer. Since there are 5 classes, 5 neurons are added to the output layer. The weights of the VGG19 model are imported without the output layer by setting (IncludeTop = false). Two dense layers are added, one with 128 neurons and the output layer with 5 neurons. The first dense layer has a ReLu activation function and, the last layer has a softmax activation function. As the pre-trained weights are used, all the weights are set to be non-trainable. Only the dense layers that we add manually are trained.

Now that the model is ready, the training dataset has to be loaded using ImageDataGenerator class. ImageDataGenerator is an inbuilt class in Keras that generates batches of data with real-time data augmentation. The model receives modified images after every epoch. It only returns the transformed images instead of adding them to the dataset. The augmentation techniques used were zoom range, rotate range, and fill mode with an image shape of 224 x 224 and a batch size of 32. The dataset must be stored in a particular order before it is fed to the model. The main data folder should have two folders, namely train and test. Both train and test folders should have a folder for each class in the dataset containing corresponding images. The names of the folders in the train and test folders should be the names of their respective classes.

Once the training dataset and validation dataset are generated using ImageDataGenerator, the model is compiled and trained. The Keras library provides a method, compile() for compilation the model. Loss function, optimizer and, metrics are important arguments for the compile method. The loss function used is CategoricalCrossEntropy. The purpose of the loss function is to calculate the loss that has to be minimized during training. The optimizer used is the Adam optimizer with a learning rate of 0.001. Optimizers are used to update the weights and training rate to reduce the loss. Accuracy is used as the metric for the mode. Metric is a function used to assess the performance of the model. After compiling the model, the model is trained using the method fit(). The fit function is used to train and validate the model.

The model achieved a training accuracy of 98% and a validation accuracy of 87%. Figure 3 represents the graph of training loss vs validation loss. It can be observed that there is no much difference between the training and validation loss. Figure 4 represents the graph of training accuracy vs validation accuracy.

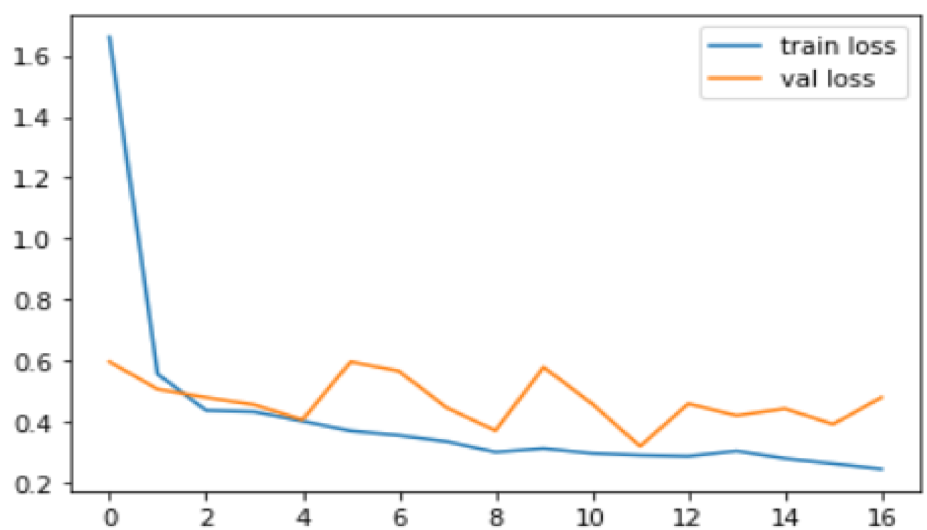


Fig. 3: Training and Validation Loss

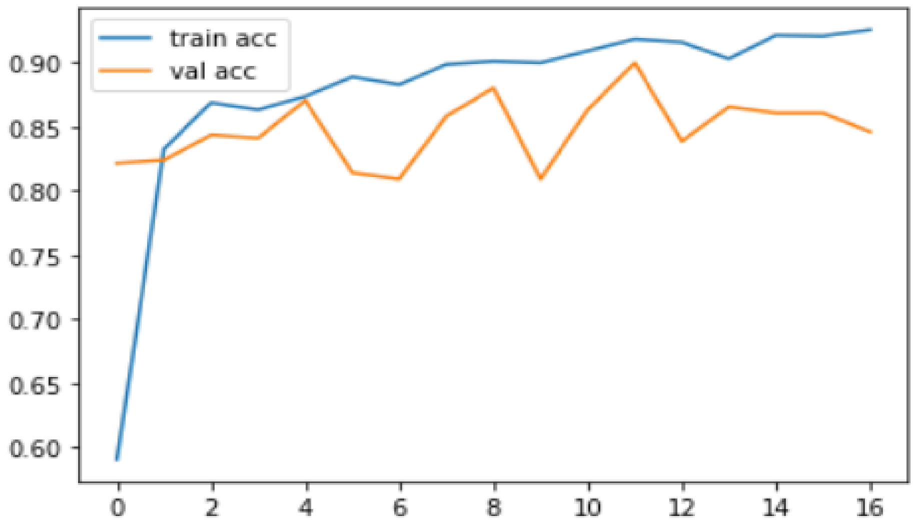


Fig. 4: Training and Validation Accuracy

5.3 Model Testing and Evaluation

The main goal of model evaluation is to measure the generalization accuracy of the unseen data. The model is evaluated on the test dataset using evaluate() method.

The model achieved a testing accuracy of 79.40%. A confusion matrix is a way to examine the performance of the model on the test data. Figure 1 shows the confusion matrix calculated for the testing data and Table 2 shows the precision and recall of the proposed model.

Table 1: Classification scores of the model

	Precision	Recall
Healthy	0.9	0.968
Leaf Blight	0.66	0.635
Common Corn Rust	0.612	1.131
Gray Leaf Spot	0.91	0.636
Fall Armyworm infestation	0.853	0.879

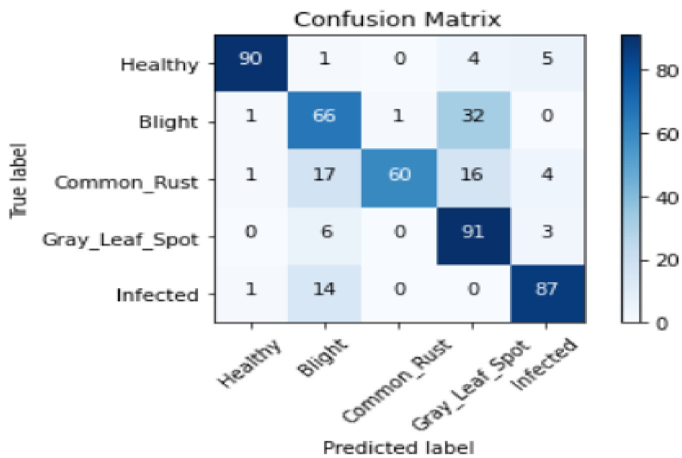


Fig. 5: Confusion Matrix

6 Dataset

The dataset collected for this work consists of maize leaf images. The images include healthy leaves, leaves infected with Fall Armyworm, and leaves affected by Leaf Blight, Common Corn Rust, and Gray Leaf Spot. Images of maize leaves infected by Fall Armyworm were captured from the University of Agricultural Sciences, Dharwad, using different cameras. The photos were captured by placing the leaves on a black background. The healthy and diseased images were collected from the Kaggle website. A total of 2,546 images of maize leaves were collected. Figure 1 shows examples of diseased maize leaves, healthy maize leaves, and those infected with Fall Armyworm.

Table 2: This table represents the number of images for each class in the dataset.

Image Class	Number of Images
Infected with Fall Armyworm	552
Leaf Blight	500
Common Corn Rust	493
Gray Leaf Spot	500
Healthy	500

(a)

(b)

(c)

(d)

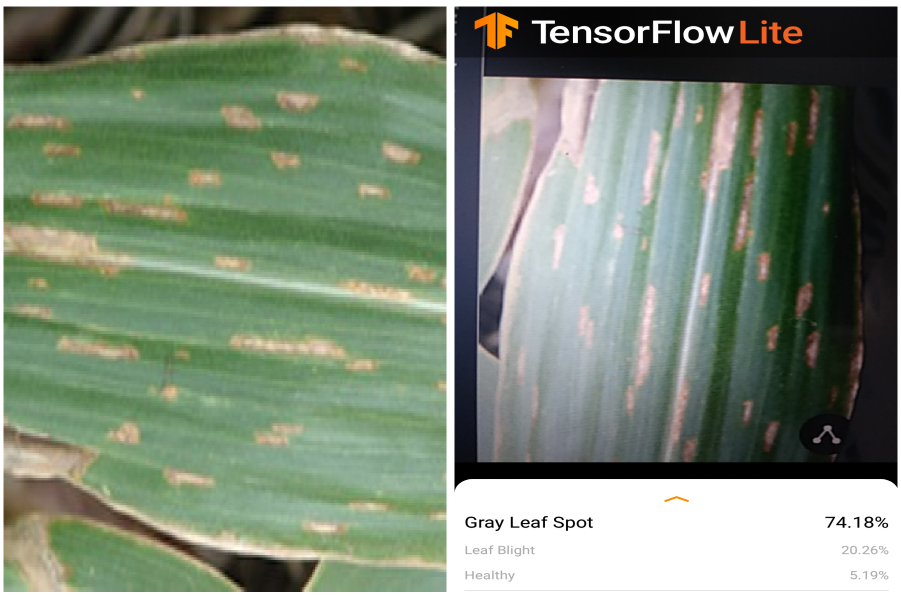
(e)

Fig. 6: Examples of diseased maize leaves : Leaf Blight (a), Common Corn Rust(b), Gray Leaf Spot (c), healthy maize leaves (d) and those infected with Fall Armyworm (e)

7 Results and Discussions

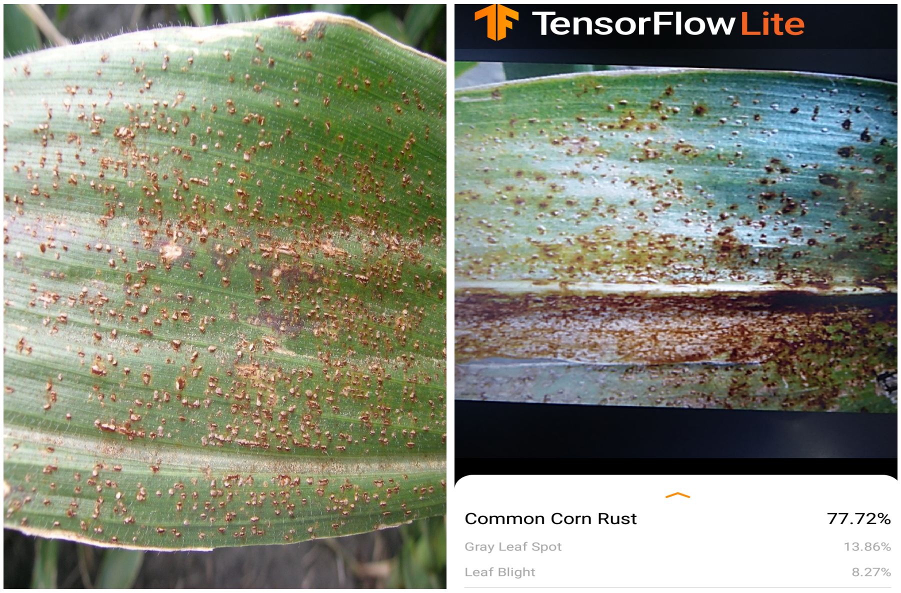
Initially, the model used for training on the available dataset was RESNET50, which gave an accuracy of 68.40%. To get better accuracy, other models were used. Inception V3 and VGG19 were used which gave an accuracy of 78.23% and 79.40% respectively. We also implemented our model which gave an accuracy of 70.04%. VGG19 was used as the final model as it gave the highest accuracy out of all the models.

Following are the results obtained from the android application that was built using VGG19 architecture:



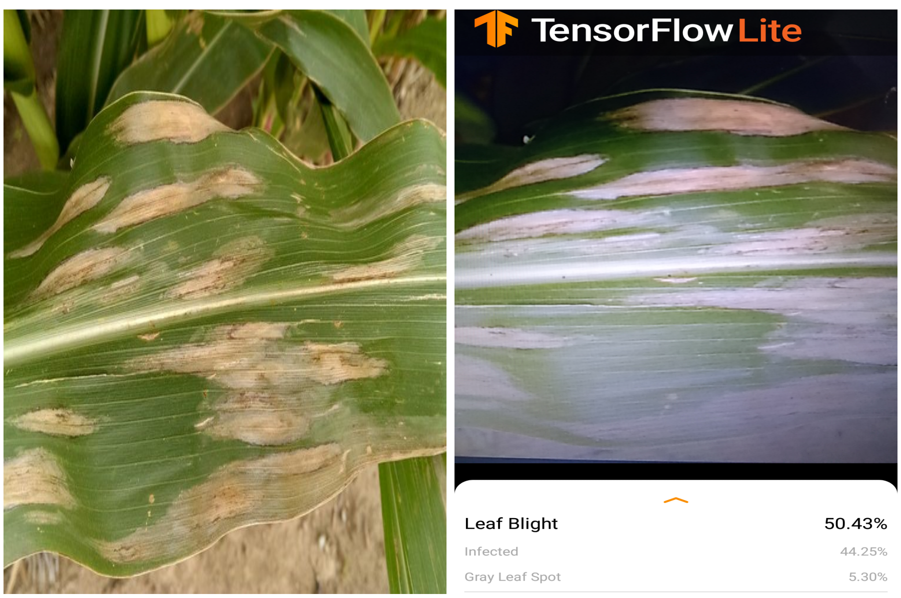
(a) Input (b) Output

Fig. 7: Input(a) and Output(b) of Maize Leaf infected with Gray Leaf Spot.



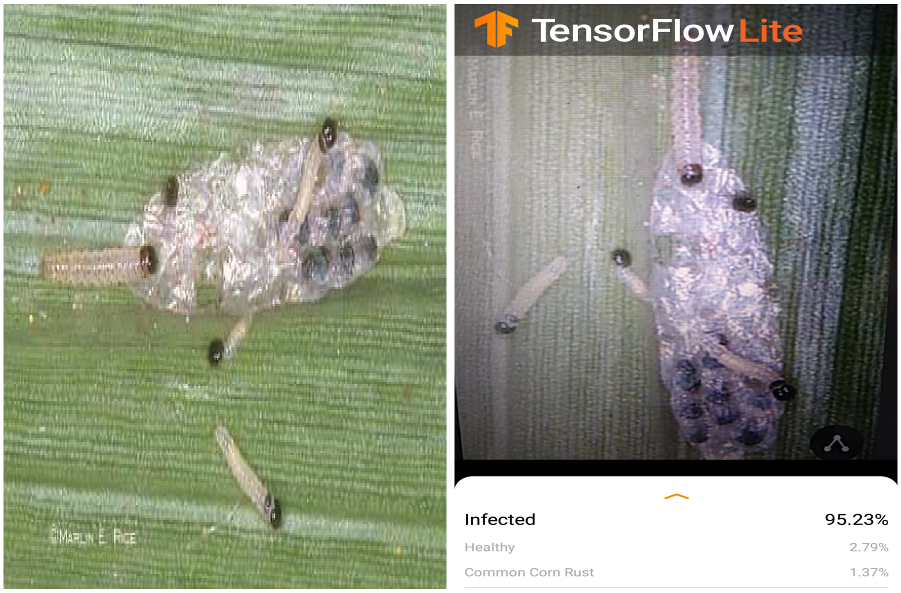
(a) Input (b) Output

Fig. 8: Input(a) and Output(b) of Maize Leaf Infected with Common Corn Rust.



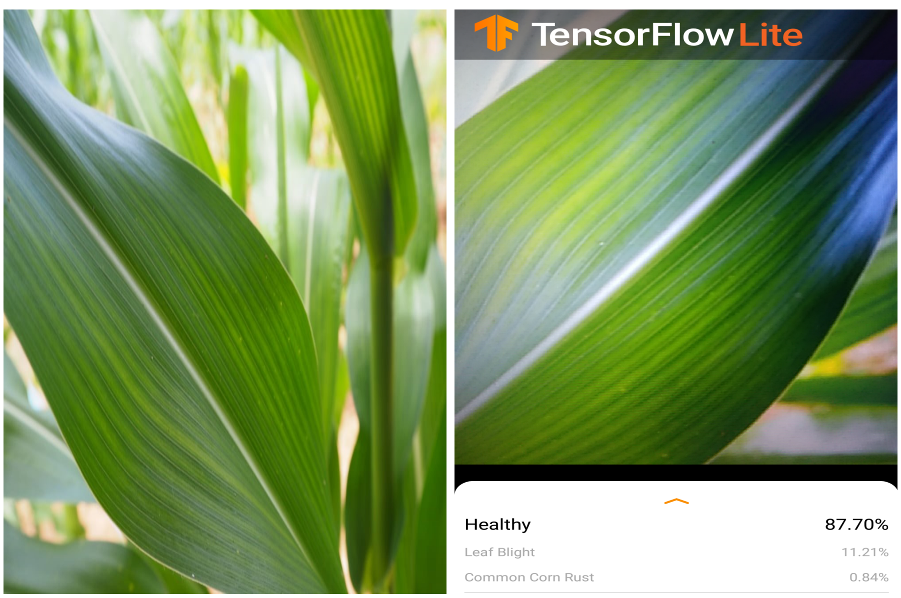
(a) Input (b) Output

Fig. 9: Input(a) and Output(b) of Maize Leaf Infected with Leaf Blight.



(a) Input (b) Output

Fig. 10: Input(a) and Output(b) of Maize Leaf Infected with Fall Armyworm.



(a) Input (b) Output

Fig. 11: Input(a) and Output(b) of Healthy Maize Leaf.

8 Conclusion and Future Scope

A classification system based on improved VGG19 for quickly and accurately detecting diseases or infestation in the live image was proposed. A pre-trained VGG19 architecture has been implemented using the Keras framework and achieved successful feature extraction. The experimental results based on test accuracy and validation loss on currently available data show that the model is faster and accurate. The application designed helps farmers and agricultural experts to detect diseases and take required actions. However, still some issues can be identified in this proposed method. This performance of the model can be further improved by expanding the database by collecting images of various diseases in the maize manually in the future.

References

1. Zhang, Xihai, Yue Qiao, Fanfeng Meng, Chengguo Fan, and Mingming Zhang. "Identification of maize leaf diseases using improved deep convolutional neural networks." IEEE Access 6 (2018): 30370-30377.
2. Esgario, José GM, Renato A. Krohling, and José A. Ventura. "Deep learning for classification and severity estimation of coffee leaf biotic stress." Computers and Electronics in Agriculture 169 (2020): 105162.
3. https://towardsdatascience.com/extract-features-visualize-filters-and-feature-maps-in-vgg16-and-vgg19-cnn-models-d2da6333edd0.
4. Machine Learning Mastery — Transfer Learning in Keras with Computer Vision Models— Published: May 15,2019 — Accessed: November 2020.
5. Detection & prediction of pests/diseases using deep learning
6. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.
7. PaperspaceBlog — A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet — Published: 5 June, 2020 — Accessed October 2020.