Plant Disease Detection using Deep Learning and Convolutionary Neural Network

Anshika Agarwal[1], Akash Sanghi[2], Gaurav Agarwal[3], Y. D. S. Arya[4], Shruti Agarwal[5]

[1]Ph.D. Scholar, Department of CSE, Invertis University, Bareilly, India

[1]Assistant Professor, ABES Engineering College, Ghaziabad

[2]Assistant Professor, Department of CSE, Invertis University, Bareilly, India

[3]Associate Professor, Department of CSE, Invertis University, Bareilly, India

[4]Professor, Department of CSE, Invertis University, Bareilly, India

[5] Assistant Professor, Department of CSE, SRMSCET, Bareilly, India

anshika1993agarwal@gmail.com, anshika.agarwal@abes.ac.in, sanghiakash@gmail.com, gauravagarwal95@gmail.com, yds.a@invertis.org, shrutiagarwal235@gmail.com

Article Info
Page Number: 2499-2516
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 March 2022
Revised: 30 April 2022
Accepted: 15 June 2022
Publication: 19 August 2022

Abstract-
Modern agriculture has evolved into much more than a means of feeding ever-increasing populations. Agriculture affects a country's economy in some way. With the population growing by the day, the primary sector must be given more attention. According to a World Bank report, three out of every four people in developing countries live in rural regions and earn less than Rs.200 per day. Agricultural progress is required to improve the quality of agro-based industry goods, particularly in emerging countries. As a result, early detection of plant diseases could be crucial in preventing agricultural losses. Plant disease detection is based on the principle that all information that aids people in growing food should be freely available to everyone on the earth. The next important disease diagnostics tool that aids in making a vision of productive agriculture into a reality is developing algorithms that can reliably detect a disease based on an image. The goal of this project is to develop an artificial intelligence software that can identify and classify plant illnesses. We'll be using PlantVillage, a public dataset of 54,444 photos, and PyTorch as our deep learning platform. We have worked for Potato and Pepper plant leaves, but the approach can be used for any plant and dataset can be extended accordingly anytime. After Data collection, model have been built using tensorflow, CNN, data augmentation is done by using tf dataset. Tf server will built backend server using FastAPI and then we can also deploy our server Model to Google cloud (GCP). Also the Google cloud function that is running in front end and these functions will be invoked by mobile application return in react native so it is an end to end project. For optimization we have used quantization process Tensorflow lite. Frontend is by using ReactJs and react native deployment is by using GCP (Google Cloud Platform) and GCF (Google Cloud Functions). Plant diseases will be detected using photos of plant leaves. As a result, we feel that early diagnosis of plant diseases will undoubtedly aid in the maintenance of

agricultural stability and the advancement of a country's growth.

Keywords:- ML ops, Data Collection, Keras, Tensorflow, CNN, smart agriculture, Plant disease detection, ResNet50, Deep learning, PlantVillage, PyTorch, Disease diagnostics tool

I. Introduction

Farmers grow potato are facing lot of economical problem because of various diseases that affect potato plant. There are usually two common diseases. One is early blight and other is late blight. **Early Blight** is caused by fungus and **Late Blight** is caused by a specific microorganism. If a farmer is able to detect these diseases early and can apply appropriate treatment then can save wastage of potatoes and can prevent the economical loss. The treatments for early blight and late blight are little different so it’s important to accurately identify whether disease is late blight or early blight. Although this model can work for any plant but for now we have considered two plant leaves. One is Potato and other is Pepper Bell plant.

The basic working of this paper is farmer just need to go to his farmland and will just take a picture of potato plant and upload that picture to application and he will get to know about whether the plant is healthy, or having late blight disease or early blight disease. For this detection system we have used Deep learning and convolutional neural network.

The main objective of this paper is to identify plant disease and also the type of disease. Another main objective of this paper is to help those who are new in farming by providing them diseased plant information.

This paper is designed in five parts. First part gives the detailed introduction about the topic. In the second part i.e. Literature Review, previous work about the topic is discussed. In the third part of the paper i.e. Research Methodology, how our paper depicts the plant disease with ResNet algorithm is in detail described. In part forth, results of our approach are described. Lastly the paper is followed by Conclusion and references.

II. Literature Review

This part of paper basically elaborates existing research work based on plant disease detection with its advantages and disadvantages discussed too.

This has been globally known that in India farmer’s faces majority of their problem because of crop yield loss. And most of the time this loss is because of diseases in plant but if this plant disease detection can be diagnosed during its early phase then crop loss can be somehow decrease. Mainly these plant diseases are of three types i.e. Viral, Bacterial, and Fungal.

1. Viral- Infections like Measles, Scorch, Spider Mite, Target Spot, Leaf Curl, Mosaic leads to degradation in plant’s health. This Viral is actually made from genetics like DNA and RNA with a coating of some kind of protein.

2. Bacterial- Whenever plants have Spot, Blight or Huanglongbing, it means plant have a kind of Bacterial plant disease. Huanglongbing also known as Citrus huanglongbing (HLB), later termed as citrus greening disease. This disease is one of the most harmful disease. These types of diseases can only be seen using microscope. Bacteria sometimes are also useful for digestion process in animal, in case of plant root it is useful for fixing nitrogen level. But they are harmful for plants and cause serious disease and infection in plants.

3. Fungal- Fungi like Scab, Rot, Rust, Mildew, Leaf Mold or Septoria is caused because of deficiency of chlorophyll that leads to degrade in the process of food photosynthesis.

Plant Disease	Examples
Viral	Measles, Scorch, Spider Mite, Target Spot, Leaf Curl, Mosaic
Bacterial	Spot, Blight or Huanglongbing
Fungal	Scab, Rot, Rust, Mildew, Leaf Mold or Septoria

Table. 1: Plant Diseases Category

Abirami Devarai et al. [1] in his paper proposed a technique for plant disease detection with the type of disease plant has by capturing plant leaf images. Since if in advance plant disease can be identified, it will help to treat the disease accordingly. After images of plant leaf taken, clusters of same type of images have been formed using K-mean clustering and random forest classifiers. These two algorithms have been also used for training of machine also. So in this way by taking images and then by creating clusters of images of same type and then by recognizing the disease by using predefined training data, this paper is able to find out the type of disease of plant.

Shima Ramesh et al. [2] in his paper classify the data using Random forest algorithm. This paper has used plant leaves data and used leaves of both types for healthy plant and for sick plant. Directional Gradient Histogram has been used for feature extraction of plant leaves. Authors have used machine learning algorithm for plant disease detection.

Yuan Tian et al. [3] in his paper converted image colors from RGB (Red, Green, Blue) to HSI (Hue, Saturation, Intensity). Generalised Linear Model have been used to perform this process. This paper have also used Support vector Machine (SVM) for wheat plant disease detection and have used 7 parameters for describing image shape.

Few papers have also used K-Mean Clustering to classify infectious leaf from a group of images. And Fuzzy logic is also been used by many papers for ranking purpose. This means it can be used to find the level of infection in leaf. Means it can identify whether the plant disease is in its initial stage, medium stage or higher stage. Means plant disease severity can be checked by Fuzzy logic.

Sanjay B. Dhaygude et al. [4] have divided his work into 4 steps. In first step input leaf image is converted from RGB i.e. used for generation of color to HSI that is used for description of color into Hue, Saturation and Intensity. In second step green part of image is hide and are removed by using threshold value of HSI calculated in step one.

In third step, all the green pixels are removed and finally in step 4 image segmentation is done.

We have also explored various Deep Learning Models that can be used for plant disease detection. In 2012 AlexNet was introduced. AlexNet is 8 layers Convolutionary Neural Network. In this model, extracted features that are obtained by learning of the model are transcended automatically. After AlexNet in 2013 ZFNet has been proposed that is based on Convolutional Neural Network. In this model also extraction of features is done and depending upon this images are been segmented. In 2013 two more models also introduced i.e. NiN (Network in Network) used for image classification and OverFeat used for object localization. It actually identifies an object in an image.

In 2014 VGG, GoogleNet, FCN, RCNN were proposed. Unlike AlexNet, VGG basically focuses upon depth aspect of Convolutionary Neural Network. AlexNet Model basically focuses on smaller window sizes and strides Convolutionary layer. Architecture of VGG has parameters like Input in which image is of 224*224 pixels RGB. Second Component is Convolutional layers of VGG that are very small i.e. 3*3 only. The convolutional filters here acts as linear transformation of input which is followed by ReLu unit. The third component of VGG is fully Connected layers and they are of three types. Two among them have 4096 channels and third one have 1000 channels, one channel for each class. Forth component is hidden layet that uses Relu. There are three Relu units and because of this the decision function is mode discrimative. If more no of layers are there in any model then it represents more improved performance.

After that in 2015, three models i.e. ResNet, Segnet and U-Net were proposed. Residual Network or ResNet is a deep learning model that uses residual blocks to classify image. These blocks solve the problem of very deep networks. The activation function f() and output of Residual block is given as

$H(x)= f(wx+b)$ or $H(x)=f(x)$ and with the addition of new connection the output $H(x)$ will be changed to $f(x)+x$.

In 2016 again three models, FractalNet, YOLO and SSD came in the field of Deep Learning Models and in 2017, DenseNet, CapsuleNet, IRCNN, IRRCNN, RefineNet, PSPNet, Mask-RCNN and Fast-RCNN models were proposed. And in 2018 three models DCRN, R2U-Net and DeepLab were came in the field of Deep Learning. Table 2 shows all Deep learning models with its year of invention.

Year	**Deep Learning Model**
2012	AlexNet
2013	ZFNet
2013	NiN
2013	OverFeat
2014	VGG
2014	GoogleNet
2014	FCN
2014	RCNN
2015	ResNet
2015	SegNet
2015	U-Net
2016	FractalNet
2016	YOLO
2016	SSD
2017	DenseNet
2017	CapsuleNet
2017	IRCNN
2017	IRRCNN
2017	RefineNet
2017	PSPNet
2017	Mask-RCNN
2017	Fast-RCNN
2018	DCRN
2018	R2U-Net
2018	DeepLab

Table 2: Deep learning algorithms for plant disease classification

III. Comparison of various Plant disease detection methods

PAPER No.	DataSet Used	No. of Images	No. of images after preprocessing	Preprocessing and Augmentation Techniques	CNN Architecture	Transfer Learning	Accuracy
5	Plant Village	3700	3700	Re	Modified LeNet	NO	92.88
6	Plant Village	54309	87848	Cr, Re	VGG	NO	99.53
7	Captured and self created	5000	43398	AB,AC,Cr,FL,NR, Re,Ro	R-FCN and ResNet50	NO	85.98
8	Captured and self created	299	NA	Nr,Re,Sg,AB,AC, AS	Modified LeNet	YES	98.6
9	Captured and self created	1053	13689	AT,NR,MS,PC,A JITTERING	Modified AlexNet	NO	97.62
10	Captured and self created	1567	46409	BR,Re,Sg	GoogleNet	YES	94
11	Plant Village	54306	NA	Re,Sg	GoogleNet	YES	99.35
12	Captured and self created	4483	33469	AT,CR,PT,RE,RO	Modified CaffeNet	YES	96.3
13	Plant Village	54323	55038	CR,RE,DCGAN	Incetion V3	YES	99.76
14	Plant Village	2086	NA	RE,NO,RO,FL,ZO	VGG16	YES	90.49

Table 3: Summary of different DL methods for plant disease classification, while Table 5 contains the limitations of different DL methods for plant disease classification.

Augmented Techniques				
Abbreviations				
AT	Affine Transformation(Translations and Rotations)		**AB**	Adjusting Brightness
Cr	Cropping		**AC**	Adjusting Contrast
Fl	Flipping		**AS**	Adjusting Sharpness
MS	Mirror Symmetry		**BR**	Background Removal
PT	Perspective Transformation		**NR**	Noise Removal
Re	Resizing		**Sg**	Segmentation
Ro	Rotation		**Zo**	Zooming

Table 4: List of Various Abbreviations used in Table 4

Paper	Small Number of Examples in Dataset	Small number of plant diseases	Low Accuracy when testing in Real Conditions	Complex Background	Multiple Disease in same Sample	Location	Infection Status	Train and Test Data are from the
5	Unresolved	Unresolved	Unresolved	Partially Resolved	Unresolved	Unresolved	Unresolved	Unresolved
6	Partially Resolved	Resolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved
7	Unresolved	Unresolved	Partially Resolved	Resolved	Resolved	Resolved	Resolved	Unresolved
8	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved
9	Unresolved	Unresolved	Resolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved
10	Unresolved	Resolved	Partially Resolved	Unresolved	Resolved	Unresolved	Unresolved	Unresolved
11	Partially Resolved	Resolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved
12	Unresolved	Partially Resolved	Unresolved	Resolved	Unresolved	Unresolved	Unresolved	Unresolved
13	Partially Resolved	Resolved	Unresolved	Partially Resolved	Unresolved	Unresolved	Unresolved	Unresolved
14	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Unresolved	Resolved	Unresolv

Table 5: Limitations of various Deep Learning methods for plant disease detection

IV. Research Methodology

Any supervised machine learning project starts with data collection. This data can be used as a training dataset. In our case we need to collect images of a healthy potato plant leaf and the potato plant which has early blight and late blight disease. And for this data collection we have use readymade dataset PlantVillage that can be downloaded from Kaggle.

Then comes Data Cleaning and Preprocessing step and for that we have used tf (Tensor Flow dataset and data augmentation. Data augmentation because we might not have enough diverse set of images so we need to rotate, flip and adjust contrast to create more training samples. And once we will have that then can use model building using convolutional neural network. CNN is a standard way of image classification, so we have used CNN and export the train model. Then we will use ML OPs concept using TF (Tensor Flow) server where we will have a TF server serving server running on top of these exported models which can solve different versions of these models and tf serving will be called from fast API. This website is built basically in React JS. It is a fast growing technology now a days for website application development and that will be calling a fast API server where we can drag and drop the image and it will tell you the label whether it us a healthy plant, early blight or late blight plant. And then we convert these exported float models into TF lite model using quantization. Quantization is a way to reduce the size of model so that model occupying less model. We can deploy it on cell phone on edge devices and also the inference speed is much faster. Once we have exported tf lite model then we will deploy those to Google cloud and will write Google Cloud functions which are similar to AWS lambda.

For model building we have used TensorFlow

Data Collection:- We can use readymade data i.e. kaggle. Or we can collect data in form of images from farmers and annotate data on our own. In this from we take images from agricultural land and classify data manually i.e. whether it is a healthy plant, late blight or early blight. It is actually an expensive deal as it requires budget to deal with the raw data.

We have used Kaggle's PlantVillage dataset for our model. Here we have three categories of data. One is of Potato; other is of Tomato and third is of Pepper. We have worked on Potato plant leaves. Here leaves are categorized in three categories. One is healthy, second is early blight and third is late blight.

Healthy:- No spot on leaf, they look fresh

Early Blight:- there are black dots, that shows there is some disease in the plant.

Late Blight:- they look horrible, much damaged.

Download the dataset as tf dataset.

Size of image: 256*256

The dataset images are basically categorized in RGB (Red, Green, and Blue), Black and White or have images which are being cropped from its background.

This dataset is then classified into training data that is used to train the machine, validation data that is used to validate the machine output and test data that is used to test the machine.

First type of leaves used is Healthy potato leaves. Figure 1 shows healthy potato leaves. They are fresh, healthy and without any disease.

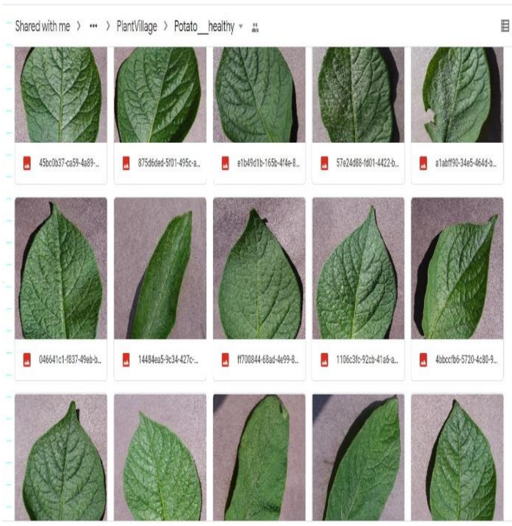

Figure 1: Healthy Potato leaves.

Second type of leaf is Potato with Early Blight Disease. This type of disease has bacterial spots. Figure 2 shows Early Blight Potato leaves.

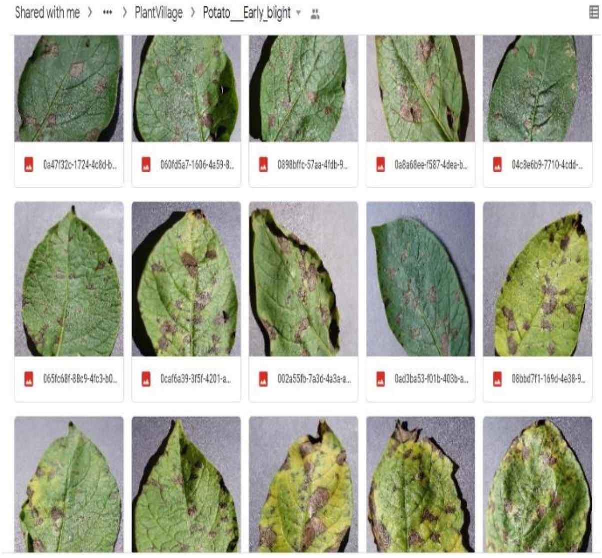

Figure 2: Potato Early Blight

Next type of leaves in dataset is Potato with Late Blight disease. This type of plant is more damaged kind off. Figure 3 shows Potato leaves having Late Blight disease.

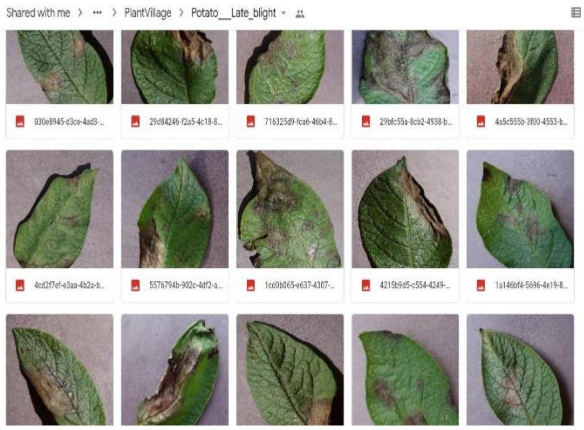

Figure 3: Potato Late Blight

Another type of leaves used in dataset is Healthy Pepper plant leaves. They are also fresh, healthy and without any disease. Figure 4 Shows Healthy Pepper leaves.

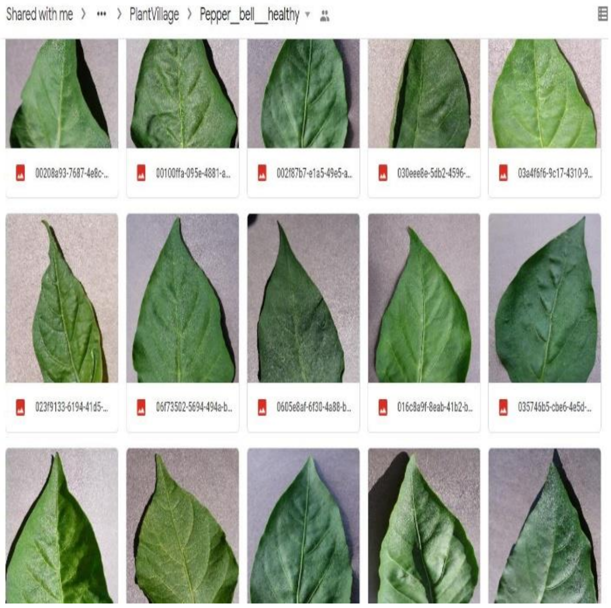

Figure 4: Pepper Healthy

Lastly we have Pepper leaves with Bacterial Spots.

Figure 5 shows Pepper leaves having Bacterial spots.

Figure 5: Pepper Bacterial Spot

V. **V. Technique used: Augmentation:-** Vast datasets are used when we have to use deep learning algorithms and this can lead to no over fitting problem. To collect data is no doubt a very tedious job; again to categorize this collected data is another job which requires a lot of time and energy. We can use augmented strategies in order to increase the size of dataset. This paper has used 2 augmented techniques have been used. In first augmentation technique that is a traditional one was basically used by many research scholars in there paper to detect plant disease [6, 7, 8, 9, 10, 11]. Basically methods like rotation or blurring etc, a kind of simple transformation are applied in augmentation. They add distortion to an image.

We have applied transformations in the dataset that includes the preserving shearing in size, changes and copying the size of image.

In the second method that includes Generative adversarial networks [15] is used for generating semantic data that is actually based on database. Basically architecture of GAN has two neural networks that are used as input to any architecture. It is used to find out whether the data is original or duplicate and for that purpose generator used plays an important role. Through this process, Generative adversarial networks produce an image that will be used in training process in order to find out disease in the plant. These kind of model are used in various platforms instead of Variation Auto Encoders (VAEs), Restricted Boltzmann machines etc. These days many other Generative adversarial networks are also there. This paper used much architecture in order to explore the chance of using semantic data for the training purpose of plant disease classifier. This semantic data have feature like color, shape, size, texture same as original plant leaves. While training at higher resolutions is extremely unstable since one network grows stronger than the other, which halts the learning process, the Deep Convolutional Generative Adversarial Network (DCGAN) architecture is designed. Higher resolution images must be created in order to employ this syntactic data in the training phase because most cutting-edge convolutional neural networks are built for input image sizes around 256 256. This is the only reason, 256*256 sized plant leaves were generated using Progressively Growing GAN (ProGAN) [50]. ProGAN begins by producing a little image of 4 × 4 or 8 x 8 pixels and continues doing so until the discriminator rates the image as realistic. Higher-resolution layers are added by ProGAN and trained after the initial learning phase is finished. Until 1024 x 1024 pixels (or smaller dimensions depending on the number of layers) are taught, this process is repeated. The network was stable during the plant leaf picture training phase, however it was unable to produce images with a sufficient number of characteristics to accurately represent plant leaves. Style GAN [15], a novel architecture that combines ProGAN and neural style transfer, produced plant leaf images in higher dimensions (256*256) employed in the experiment) the most successfully. For the purpose of producing the desired input dimension of 256 * 256, the original architecture was used. This was accomplished by adding a stylized convolutional block of 128*64*3 as the last layer of the generator network and a 64*128*3 convolutional block as the first layer of the discriminator network, both following 3-64 convolutions. The following settings were used to train Style GAN on the whole PlantVillage dataset for producing innovative plant leaves: a minibatch size of 5, a learning rate of 3 10*3, and an AMSGrad optimizer. Style GAN , a novel architecture that blends ProGAN and neural style transfer, produced plant leaf images in higher dimensions (256* 256) employed in the experiment) the most successfully. For the purpose of producing the desired input dimension of 256*256, the original architecture was used. For the generating network, a styled convolutional block of 128*64*3 was added after a 64*3 convolutional block, and for the discriminator network, a 64*128*3 convolutional block was added after a 3*64 convolution as the first layer. The following settings were used to train Style GAN on the whole PlantVillage dataset for producing innovative plant leaves: an AMSGrad optimizer, a minibatch size of 5, and a learning rate of 3*103.

VI. **VI. Methodology Used**: ResNet (Residual Networks)- ResNet is a kind of Neural Network and is a subclass of Convolutionary Neural Network. We can extend this network upto 152 layers. Authors in paper [16], this process is not done by representing signals directly but actually by learning the process of representing residual functions. In residual network we can also minimize or completely finish the connection in order to learn from next layer without manipulate the inputs of its previous layer. And this will obviously help in building strong and deep connections.

There are many advantages of using Resnet.

1. Like one is we need not to use plain networks. We now have additional facility of skiping unwanted intermediately layers. Let's call them simple networks. The deeper the simple network, that is, the more layers, the more gradient disappearance / explosion problems occur**.**

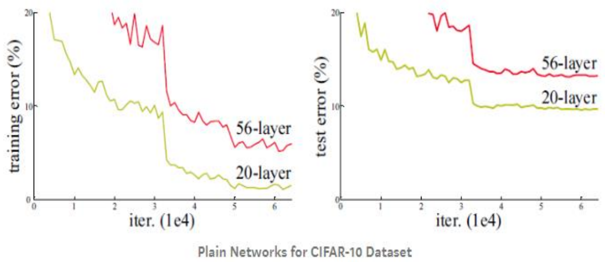

Figure 6: Vanishing Gradient Problem; Image source: [17]

Above figure shows that deep networks suffer from gradient loss / explosion problems more than shallow networks. Residual Network helps by skipping the connections and this way this problem can be solved. Shortcut is added to add the input x to the ouput after few weight layers as shown in Fig.

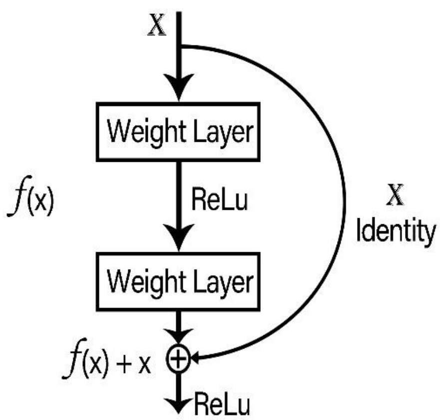

Figure 7: ReLu

The connections that are skipped are mentioned as "Identity" and they are responsible for network to learn the identity function. And they are also responsible to pass the input to required blocked, and need not to pass it to other layers. So the ouput is given as H (x) = F (x) + x.

The weight layer is responsible to learn residual mapping, such as F (x) = H (x) −x. This allows you to smooth out the disappearance gradient by stacking additional layers to build a deeper network and allowing the network to skip layers that may not be relevant to your training. Even if the gradient disappearance appears in the weight layer, there is an identity x to bring back to the previous layer.

ResNet vs Plain Networks: If you use a simple network, a low-rise network is always better. For example. We recommend using a simple network with an 18-layer network rather than a 34-layer network. For high-rise networks, Resnet's performance improves. This is because in deep networks, introducing a skip connection is better than a simple network, eliminating the vanishing gradient problem.

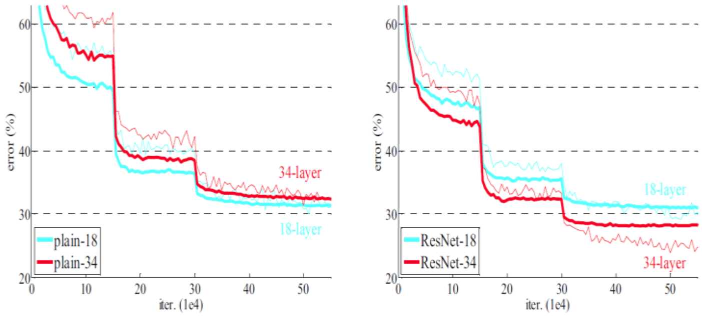

Figure 8: Plain Networks v ResNet; Image source: [15]

Feed Forward Mechanism: Feedforward means that the control system sends a signal over the path from the source to the destination. Feedforward neural networks or multi-layer perceptrons are the essence of deep learning models. The goal of feedforward networks is to approximate mathematical functions. f *. These models are called feedforward because they flow through a function whose information is evaluated from x to output y. There are no feedback connections in which output of the model is fed back into itself.

Data Training: Here Previously clean and converted data will be trained in the training set. The images in the Train Data folder are used to train the neural network, and the Validate Data folder is used to validate the results obtained from the trained model. During training, the model analyzes the input dataset and finds its own meaning. Later, we will test the model using "test data". The trained neural network is tested on a series of images to see if the model works as expected or if there are any errors.

Sanity Check: A sanity test is a basic test to quickly assess whether the results of a calculation may be true, or to make sure that the material produced is reasonable. The point of validation is to exclude certain classes with apparently inaccurate results and not detect all possible errors. Obviously, the advantage of running a sanity test over running a full or rigorous test is speed. If the result of the NN is unreasonable, the result is sent back to the feedforward function. If the result passes the validation test, it is marked as acceptable and added to the result.

VII. Results and Discussion:

The system is trained for 10 epochs for ResNet50 architecture. The accuracy for training and accuracy for testing is 98.90. The accuracy obtained after each epoch is shown in graph for training, testing and training and validation loss.

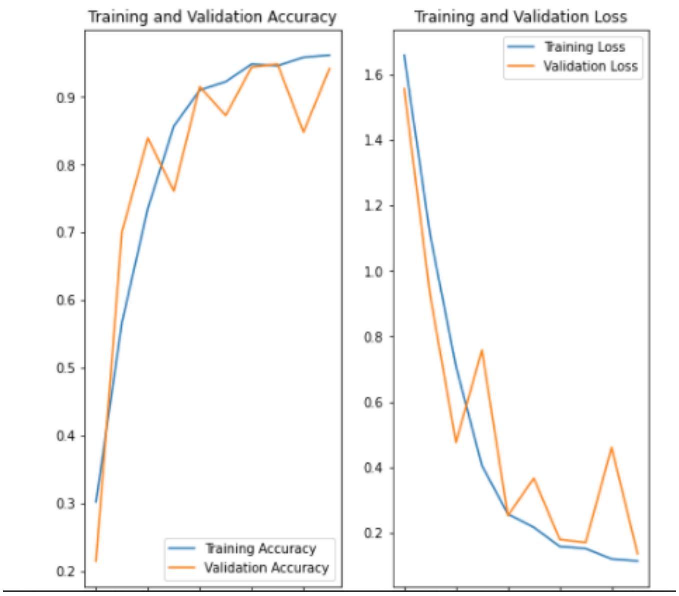

Figure 9 : Training and Validation Accuracy and Loss

As we can easily see in graph there is a huge decrement in training and validation loss as we increase the number of epochs.

Run prediction on a sample image

```
import numpy as np
for images_batch, labels_batch in test_ds.take(1):

    first_image = images_batch[0].numpy().astype('uint8')
    first_label = labels_batch[0].numpy()

    print("Predict First Image")
    plt.imshow(first_image)
    print("Actual label:",class_names[first_label])

    batch_prediction = model.predict(images_batch)
    print("Predicted label:",class_names[np.argmax(batch_prediction[0])])
```

```
Predict First Image
Actual label: Pepper__bell___Bacterial_spot
Predicted label: Pepper__bell___Bacterial_spot
```

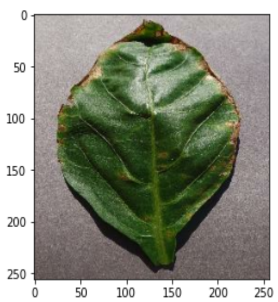

Figure 10 : Prediction on Sample Image

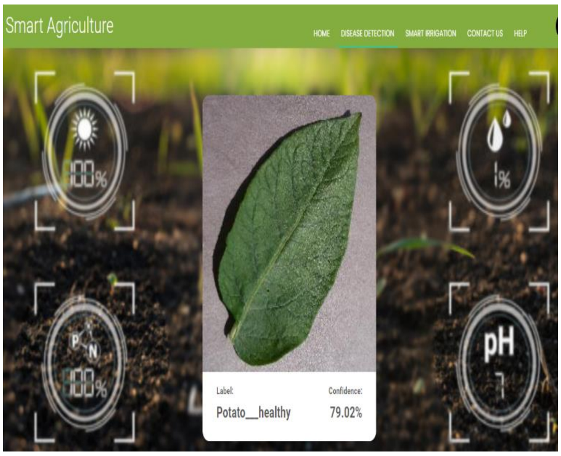

Figure 11 Result shows potato is healthy and is 79.02% confident.

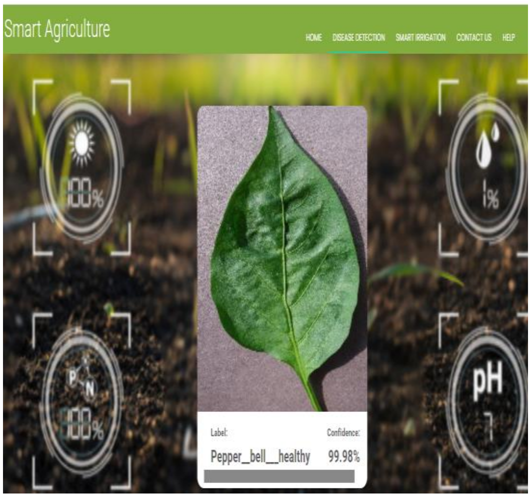

Figure 12: Result shows Pepper leaf is healthy and is 99.98% confident.

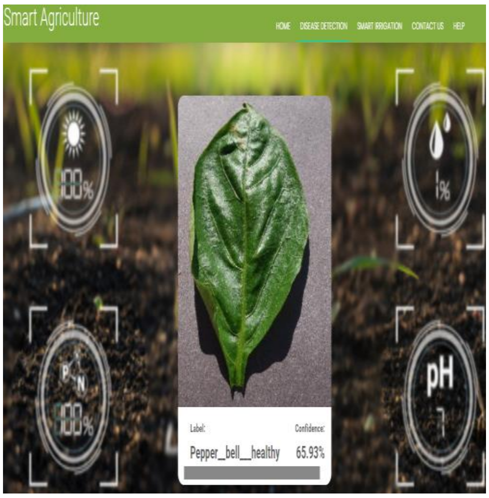

Figure 13 : Result shows Pepper leaf is healthy and is 65.93% confident. Since confidence is low means possibility of being this result wrong is high and for this case it is true as this is potato leaf not pepper.

We have used Kaggle PlantVillage dataset that is having untrained 3000 sample images to check the accuracy of the system.

So the performance of ResNet50 for plant disease detection has been seen throughout the paper.

The training took 4 h to finish without GPU. This makes the network more complex and challenging to train in standard computers.

In opposition, validation and testing are pretty much faster.

ResNet50 gives better performance during training of data but results is not that much satisfactory during the testing of plant disease detection.

Conclusion:-

For centuries, humans have carefully selected and cultivated plants for use as food, fiber, medicine, clothing, and protection. Illness is just one of many dangers to consider when growing crops. Therefore, it is important to improve food quality and pay attention to a stable agricultural sector in order to secure a country of food security. The project "Crop Disease Detection by Deep Learning" aims to build a neural network that can detect 14 crop species and 26 common diseases. Using ResNet34 as the neural network, the accuracy of the model was 96.21%. We would like to create a project that will be useful for early detection of plant diseases, and we sincerely hope that this project will be the basis of further technology for plant diseases.

This paper basically explores various potato and pepper bell plant diseases. For automatically identification of plant disease we have used deep learning ResNet50. PlantVillage a Kaggle dataset has been used for training the machine.

The images are captured under different environmental conditions such as illumination, varying backgrounds, view-invariant, and various noises. The dataset is divided into 80% training and 20% validation.

This paper analyzes the performances of the ResNet50 deep learning algorithms and shows the prediction results for the out of-sample training data and the sample dataset images.

Though we can see the differences in training and validation accuracy, the overall prediction accuracy for 2000 approx test samples of Potato leaves disease classification by ResNet50 architecture is equal to 87.7%.

This paper can be enhanced in terms of prediction, classification of diseases in potato and Pepper Bell plant, and also a solution can be found based on the type of diseases which will be helpful to the farmers to boost the productivity of potato and Pepper bell plant. Thereby helping the food industry and it will help many farmers to live happily**.**

REFERENCES

1. A. Devaraj, K. Rathan, S. Jaahnavi and K. Indira, ”Identification of Plant Disease using Image Processing Technique,” 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0749-0753.
2. S. Ramesh et al., ”Plant Disease Detection Using Machine Learning,” 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, 2018, pp. 41-45.
3. Yuan Tian, Chunjiang Zhao, Shenglian Lu and Xinyu Guo, ”SVM-based Multiple Classifier System for recognition of wheat leaf diseases,” World Automation Congress 2012, Puerto Vallarta, Mexico, 2012, pp. 189-193.
4. Prof. Sanjay B. Dhaygude, Mr.Nitin P.Kumbhar “Agricultural plant Leaf Disease Detection Using Image Processing” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 2, Issue 1, January 2013.
5. Amara, J.; Bouaziz, B.; Algergawy, A. A deep learning-based approach for banana leaf diseases classification, Gesellsch. Inf. Bonn 2017, 79–88.
6. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric.2018, 145, 311–318.
7. Fuentes, A.; Yoon, S.; Kim, S.; Park, D. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 2017, 17, 2022.
8. Cruz, A.C.; Luvisi, A.; De Bellis, L.; Ampatzidis, Y. X-FIDO: An e_ective application for detecting olive quick decline syndrome with deep learning and data fusion. Front. Plant Sci. 2017, 8, 1741.
9. Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry 2018, 10, 11.
10. Arnal Barbedo, J.G. Plant disease identification from individual lesions and spots using deep learning. Biosyst. Eng. 2019, 180, 96–107.
11. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. Front. Plant Sci. 2016, 7, 1419.
12. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. 2016, 2016, 1–11.
13. Brahimi, M.; Arsenovic, M.; Laraba, S.; Sladojevic, S.; Boukhalfa, K.; Moussaoui, A. Deep learning for plant diseases: detection and saliency map visualisation. In Human and Machine Learning; Zhou, J., Chen, F., Eds.;Springer International Publishing: Cham, Switzerland, 2018; pp. 93–117. ISBN 978-3-319-90402-3.
14. Wang, G.; Sun, Y.;Wang, J. Automatic image-based plant disease severity estimation using deep learning. Comput. Intell. Neurosci. 2017, 2017, 1–8.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Advances in Neural Information Processing Systems 27; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2672–2680.

16. Sik-Ho Tsang, 2018, Review: ResNet — Winner of ILSVRC 2015 (Image Classification, Localization, Detection), https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015image-classification-localization-detection-e39402bfa5d8.
17. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, ”Deep Residual Learning for Image Recognition” arXiv:1512.03385v1, 10 December 2015
18. Agarwal, Mohit, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, and Suneet Gupta, “Toled: Tomato Leaf Disease Detection Using Convolution Neural Network,” Procedia Computer Science, Vol.167, Pp.293-301, 2020.
19. Kanika Gupta, “Study of Deep Learning Techniques on Image Denoising” IOP Conference Series: Materials Science and Engineering, Volume 1022, 1st International Conference on Computational Research and Data Analytics (ICCRDA 2020) 24th October 2020, Rajpura, India, IOP Conf. Ser.: Mater. Sci. Eng. 1022 012007
20. P. K. Aggarwal, P. Jain, P. Chaudhary, R. Garg, K. Makar and J. Mehta, "AII0T for Development of Test Standards for Agricultural Technology", Intelligence of Things: AI-I0T Based Critical Applications and Innovations, pp. 77-99, 2021.
21. Mandeep Singh, Namrata Sukhija, Anupam Sharma, Megha Gupta, Puneet Kumar Aggarwal, “Security and Privacy Requirements for IoMT-Based Smart Healthcare System”. Big Data Analysis for Green Computing Edition1st Edition First Published 2021 ImprintCRC Press Pages 21 eBook ISBN9781003032328.
22. NanditaGoyal, MuneshChandra Trivedi, “Breast cancer classification and identification using machine learning approaches”. https://doi.org/10.1016/j.matpr.2020.10.666