International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

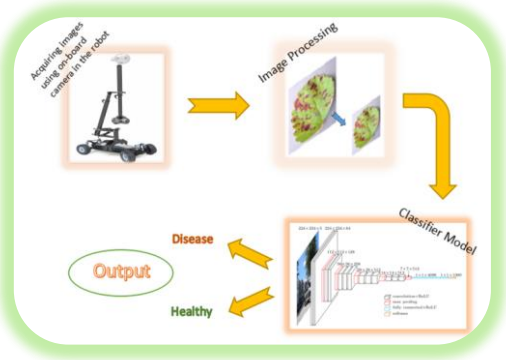
Automated Plant Disease Detection using Deep Learning Architectures with Autonomous rover

Jothilakshmi R, Sharanesh R

Abstract: ***Agriculture is the backbone and plays a vital role in many Asian countries. Farmers mainly depend on their agricultural produce for their living. A report says one-third of the farmers income account's for the agricultural loss which is primarily due to plant diseases. To combat this farmers are in need of a early plant disease identification mechanism. Observation of individual plants in the farm for detecting the disease is labor-intensive and time consuming work, if the farm is vast and multiple plants are cultivated then it's even worse. To solve such issues, current technologies like the Internet of Things (IoT) and artificial intelligence (AI) and Machine Learning (ML) are used to predict the diseases more effectively. Farmers usually detect plant diseases with the help of images captured manually and analyzed separately by experts. The proposed system renders an efficient solution for detecting multiple diseases in several plant varieties. The system is designed to detect and recognize several plant varieties, specifically pepper, grapes, and strawberry. The proposed system discovers various plant's various diseases based on the inputs obtained by capturing images from a built-in camera present in the Autonomous rover. The rover also record's it's GPS location and makes a map of the entire farm traced and checked by the robot. The images are processed and are classified into their respective categories using deep learning algorithms. Convolutional neural networks the powerful methodology for image classification is the underlying principle applied. The deep learning model's architecture namely, VGG16 and InceptionResNetV2, are used to train the model. These models are primarily made of convolutional layers. On testing, we recorded am accuracy of 93.21% was obtained from VGG16, and 95.24% from InceptionResNetV2.***

Keywords: Deep learning, Disease detection, Precision farming, Convolutional Neural Networks(CNN), Location mapping, VGG16, InceptionResNetV2

Graphical Abstract:



Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
***** Correspondence Author
Dr. Jothilakshmi R*, Department of Information Technology, R.M.D Engineering College, Chennai, India. E-mail: rjothilakshmi@gmail.com
Sharanesh R, Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India. E-mail: ee19b116@smail.iitm.ac.in

I. INTRODUCTION

A large number of people in India depend on agriculture for their income. In rural areas, more than 75 percent of the people are engaged in agricultural work either directly or indirectly in some way. According to a recent survey conducted, the agricultural loss that occurs accounts for around one-third of the actual outcome produced. If we take a more in-depth insight as to why such huge losses are incurred to the farmers, we see that it is due to the following reasons: lack of water supply, plant disease, and lack of sufficient storage for the produce. Out of them, water supply and storage issues can not majorly be solved by an individual alone. It is a community issue, and government actions will address such issues. What we being as an individual can solve is the problem of plant disease. We can solve this issue by using exponentially growing technology and newly emerging domains such as Artificial intelligence and automation.To solution proposed detects diseases with the help of deep learning. Looking deeper into the solution's use case, we find that this is more applicable to the hillside farmers. However, the drawback is that they mainly concentrate on the crops that are generally planted in plain areas. We are specifically addressing the problem by predicting plant diseases and alerting the user regarding the same. In the typical stable lands, the farmer can quickly move around all the plants and find out for any signs of disease. The same isn't that easy in hill stations where there might be frequent rains and some irregular terrains and inference of wild animals at some times. To help them out, we are designing an all-terrain robot that shall move around the farm and capture images and store the GPS location precisely. The robot comprises a camera module and a micro-controller that will capture images of the plant and maps it along with the GPS coordinates detected using a GPS module. Now, the image captured by the camera module is processed by the machine learning model that detects the plant disease using the features extracted from the leaf's image captured. Few existing solutions address the same kind of solutions for conventional crops such as Rice (Oryza sativa), Corn (Zea mays), Wheat (Triticum). Our solution deals with the same ideology, but with the crops that are generally planted in the hilly areas such as Pepper (Capsicum annuum), Strawberry (Fragaria ananassa), Grapes (Vitis vinifera). According to the recent world survey conducted on the global production of these crops is 5 lakh tonnes, 9 million tonnes, 750 million tonnes respectively. Nevertheless, these figures are largely dependant on the plant diseases caused by pests and other insects. The major diseases that affect these plants are Pepper-bacterial spot, Strawberry-Leaf scorch, Grape-(Black rot, Esca(Black measles), Leaf blight)[20].

If these diseases aren't detected in advance, they will have severe consequences on the produce. So in order to automate that work, we use the trained machine learning model. Precision farming is a method wherein technologies are embedded in agriculture with few devices such as sensors for remote sensing, robotics in agricultural fields. Examples of precise agricultural farming include spraying medicine only to the affected area, etc. Similarly, we use deep learning here to detect plant diseases using the images of the leaves of the plant. We train the model with the help of the data available in the Plant village dataset, which includes the leaf images of the diseased plants. It has all the datasets for the diseases as mentioned above. It is an open-sourced dataset available in Github. The image captured by the robot's camera is processed into a single leaf image and then passed to our trained model. Instead of deploying the classical methodologies, we use the more robust and more powerful deep learning algorithms to classify the plants.The models which we use here to find out and compare their performances are the InceptionResNetV2 and the VGG16. These are quite famous models that are used for image classification and their underlying architecture is Convolutional Neural Networks(CNN)[9]. So, we use both these models and compare their performances based on the training and testing accuracy.

II. RELATED WORK

L.Sherly Puspha Annabel[10] applied Image processing with Artificial Intelligence techniques. They used all the standard image processing techniques such as pre-processing, segmentation, feature extraction and image classification, RGB to grayscale conversion, thresholding, etc. A random forest classifier with a randomized decision tree is used to train the system to predict the diseases. Halil Durmus [4] applied AlexNet and SqueezeNet deep learning architecture to train and classify the tomato plant diseases. They used the Plant village dataset to train the system and achieved a good result using the SqueezeNet architecture. Devie Rosa Anamisa [2] presented the systematic survey on technologies and the methods used to predict, monitor, detect, and control the plant diseases using a computer or mobile technology. Sammy V. Militante [17] applied Convolutional Neural Networks with Adam optimizer and categorical cross-entropy techniques to predict the diseases. They used Data augmentation techniques to enhance the image dataset by rotating, flipping, and shifting of images horizontally and vertically and achieved an accuracy rate of 96.5%. Poojan Panchal [15]proposed the method which comprises image segmentation using K-means clustering and HSV dependent classification to identify the infected part of the plant leaf and feature extraction using GLCM. They achieved an accuracy of 98% by applying the Random Forest classifier. Parul Sharma[14] proposed the automatic plant disease detection system KrishiMitr, with a convolutional neural network (CNN) model, in which the user interface asks the user to select the plant type to detect disease type. Nuttakarn Kitpo [13] proposed the system with a drone and IoT system to predict the early detection of rice plant diseases. The system displays the analytic results, including the position of infected rice plants, mapping them on rice fields by using the GPS sensor to define the position in real-time.

III. MATERIALS AND METHODS

A. Image Dataset

PlantVillage dataset is an open-sourced plant disease database prepared by Hughes[19]; it contains around 60,000 images of more than 35 diseases of 16 plant species. We in this work used the images only of Pepper, Grapes, and Strawberry. Four diseases occur in these three plant species, Black rot and Black measles in Grape, Bacterial spot in pepper, and Leaf scorch in strawberry. The dataset contains both an augmented and an un-augmented set of images[19]. Our work uses un-augmented data because, in the real-time application of the model, augmentation needs additional processing. The below table shows the distribution of the images used for implementing the model.

Table 1: Dataset images count

Species	Disease	Number of Images
Pepper	Bacterial Spot	907
	Healthy	1477
Grapes	Black rot	1180
	Black Measles	1383
	Healthy	423
Strawberry	Leaf scorch	1109
	Healthy	456

Figure 1: Sample Images from the dataset

B. Algorithms implemented

Deep learning is a division of machine learning and Artificial Intelligence. These are generally implemented using Artificial Neural Networks(ANN). In the process of training a deep learning model, we extract the features for classification from the model.

Applications of deep learning include computer vision, image classification, speech, and video analysis, etc. In this work, we used two deep learning models, namely the VGG16[6] and the InceptionResNetV2[1]. They are the winners of the prestigious ILSVRC competition during the year 2014 and 2012 in their respective categories. They are pre-trained on the ImageNet's dataset, we in our work just use their architecture and not their hyper-parameters. If we take a look at their algorithmic composition, we see that VGG16 is primarily based on Convolution layers that are connected with some max-pooling layers and a final fully connected layer. On the other hand, InceptionResNetV2[12] is made of Inception block and residual blocks, which are, in-turn, made of many convolution layers within them. So, we see that both are of the type Convolutional neural networks(CNN's)[18]. A convolutional neural network was more efficient in evaluating graphical images and extracts the essential features through its multi-layered structure. Now, let us take an in-depth look at the main algorithms implemented: convolution layer, Pooling layer, Inception blocks, and the residual block.

i. Convolution layer

Convolution layer is the building block of any convolutional neural network[16], CNN's are more similar to the regular neural network except that they are made of neurons that are capable of learning the weights and biases. Each neuron takes an input and dot it with the weights and add biases and optionally includes some non-linearity in the output. The specialty of the convolution layer is that they allow us to add few changes in the architecture; by doing so, we can make the forward function to be implemented efficiently and reduce the numbers of parameters significantly. The other advantage of it is that it allows us to use 3-dimensional neural networks. The parameters of a Convolution layer are much similar to that of the learnable filters. Filters are spatially small(along width and height, typically the first layer will have a size of 5x5x3 in general) but extend along with the complete depth of input volume. A value is calculated based on the filter using a convolution operation for each point on the image as we move the filter across the entire image. Once the filters have passed over the complete input, a feature map is generated, stacked upon each other. Following it, they are passed through an activation function, which decides whether a particular feature is present or not at that specific location in the image. The below figure[27] depicts a simple 3x3 convolution operation.

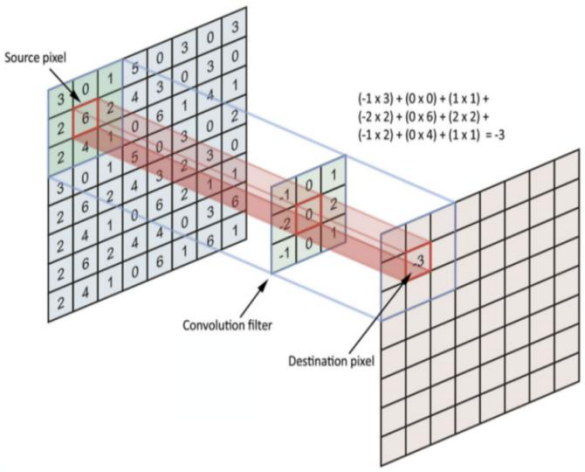


Figure 2: Convolution operation

ii. Pooling layer

It is pretty much common to add pooling layers in between consecutive layers of any Convolutional neural network. They progressively reduce the spatial size of the neural network representation and decrease the number of parameters needing to be tuned and the computation in the neural network. It also means that we can reduce the problem of overfitting, which arises when we use a large number of trained parameters. On each slice of the input data, the pooling layer operates independently and resizes it spatially. Many types of pooling layers are available, the most common being the Maxpooling[14][16], which implements the MAX operation on the independent slices of the input. Some other pooling layers include the average and the L2 norm pooling, which implements the AVERAGE and the L2 Normalization functions to produce the output. The figure[26] below depicts a sample Maxpooling operation.

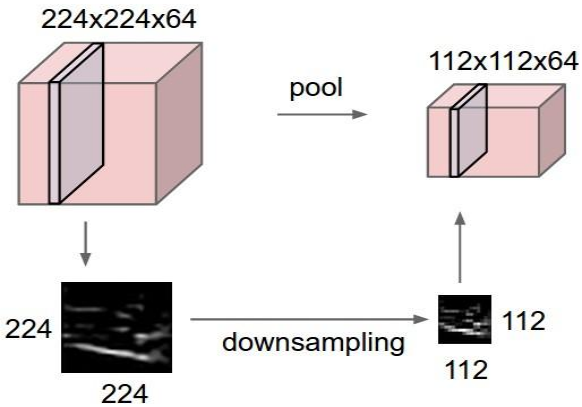


Figure 3: 2x2 Maxpooling

iii. Inception blocks

In a broad sense, Inception blocks are just a combination of convolutional layers of various sizes[1][12]. There are quite a few drawbacks in the simple convolutional layers when the images have the same features of various sizes at different locations. If we use the simple convolutional layer to address the above issue, we must increase the number of such convolutional layers,

which increases the depth and hence the number of hyper-parameters to be tuned. The same issue can be solved with the help of inception blocks since it comprises convolutional layers of multiple sizes and a final filter concatenation, which concatenates all the individual outputs and produces a single final output. The other advantage is that it helps in much efficient feature extraction without stacking more convolutional layers sequentially. The simplest form of an inception block uses three different sized filters(1x1, 3x3, 5x5x, namely) along with a max pooling. Some other forms of inception block help in dimensional reductions too. Another characteristic feature is that it makes the network broader and not deeper because the convolutions are performed within the same level of the network. The below figure[28] shows the layers of the contents.

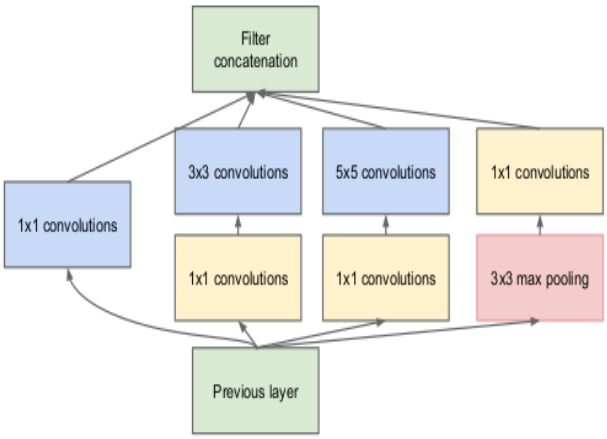


Figure 4: Inception block with Dimensional reductions

iv. Residual blocks

In the regular neural networks, each layer's output is propagated to the next layer. Neural networks, in a general sense, are universal approximators. The general notion is that the accuracy increases as we keep on increasing the accuracy increases. Nevertheless, there is a limit on the number of layers that can be added. Though the accuracy increases with the increasing layers, after reaching a certain value where it saturates, further it starts decreasing even if we went on adding more layers to the network. There have been some weird errors with deep learning where some complex neural networks with a large number of layers may not be able to learn simple functions like an identity function. This can be understood in other words as shallower networks learn better than their deeper counter-parts, which is termed as the degradation problem. To find out a possible solution for it, the hint lies in the problem statement itself. Using the fact that shallower networks outperform deeper and broader networks, we skip the training of the extra deeper layers using the skip-connections or residual connections[12]. Looking deeper into it, we find that the traditional neural network block will try to learn the actual output (H(x)). On the other side, we see that the residual network block will learn the residual output (R(x)), which is the difference between the output(H(x)) and the input(x) as shown in Eqn (1).

$$R(x) = \text{Output} — \text{Input} = H(x) — x \quad (1)$$

Rearranging it we get as shown in Eqn (2),

$$H(x) = R(x) + x \quad (2)$$

. In fact, the practical applications suggest that training the residual output is far easier than training the actual output in many cases. The below figure[29] depicts the functions of a residual block

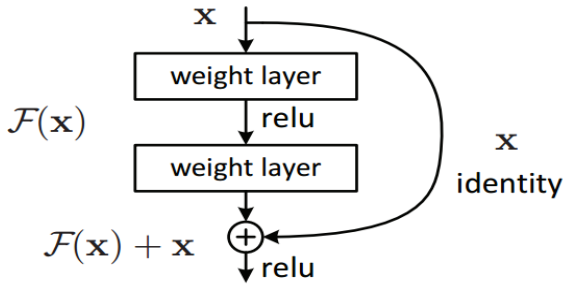


Figure 5: Single Residual Block

C. Architectures - VGG16 and InceptionResNetV2

In our work, we have used the architectures of the pre-trained VGG16[6] and InceptionResnetV2[1][12] models. These ILSVRC models were trained on Imagenet's data. We use their underlying architecture and make our models, train our data on the PlantVillage dataset. The underlying basic implementation in both the architectures is the Convolutional layer. The algorithms used in them were discussed in the previous section. This section shows the architecture of models and the other characteristics of the model. We here provide the comparative study of both the architecture in terms of their basic set of characteristics. Inspecting the architecture of VGG16[6], the input to the first Convolutional layer of the model is of a fixed size of 224x224 RGB image. The inputted image is then sent through a pile of convolutional layers, where filters are used with a small receptive field. The convolutional stride is fixed to 1-pixel; the spatial padding is selected so that the resolution of the image does not image in the process of filtering, i.e., the padding is 1-pixel for 3x3 convolutional layer. The dimensional spatial reduction is carried out by five max-pooling layers that follow some of the convolutional layers. Maxpooling operation is performed over a 2x2 pixel window, with stride 2. Three Fully-Connected(FC) layers are added following the stack of convolutional layers with different dimensions. The first two have 4096 channels each. The third contains the number of channels the same as the number of classes(7 here) to be classified into. Finally, a soft-max layer is added. ReLU non-linearity has been added to all the hidden layers. The optimizer here used is Adam, and the loss function that is being minimized is the categorical cross-entropy. The below image[24] shows the architecture of VGG16 graphically.

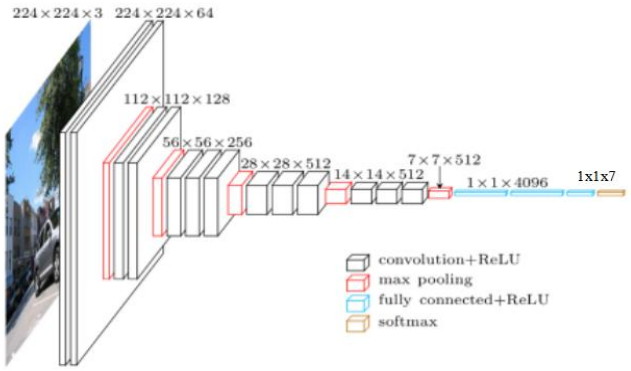


Figure 6: VGG16 Architecture

On the other hand, looking at the architecture of InceptionResNetV2[1][12], the input to this model is a fixed size of a 299x299 RGB image. The model is primarily composed of three Inception ResNet blocks (inception block with a residual module at the end of the block) along with an initial stem block made of few Convolutional layers and some max-pooling layers for dimensional reduction. The inputted image is passed through the stem block, which consists of five convolutional layers and two max-pooling layers. Same as in the case of VGG16, the convolutional stride is 1-pixel so as retain the resolution of the input image passed on. The max-pooling operation here is performed over a 3x3 pixel window, with stride 2. Following this comes the Inception ResNet block[12], which consists of an inception block, followed by a filter concatenation layer and final residual block. Finally, it consists of an average pool layer for dimensional reduction, a dropout layer for reducing the number of parameters, a fully connected layer with dimension as the number of classes(7 here), and a soft-max layer. The below figure[22] shows the architecture pictorially.

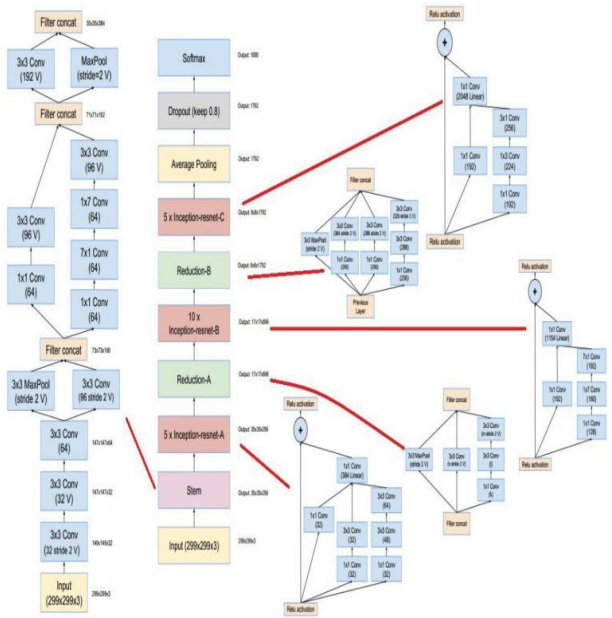


Figure 7: InceptionResNetV2 Architecture

D. Autonomous Robot

We have trained the model to detect diseases using the images of the leaf of the plant. To have a real-time application of it, we use a robot that will transit automatically and capture the images of the plant leaves on a farm or any greenhouse. We have proposed an autonomous robot that uses an obstacle avoidance algorithm to move independently. The robot has a camera pole of adjustable length, to capture a clear image of the leaf. The robot has a GPS location sensor on-board to save the location at which the picture has been taken. First, the robot moves using its obstacle avoiding algorithm. In the meantime, the camera will continuously detect for leaves using the same model we used. Once the model predicts the probability of leaf higher than a threshold, the robot stops capture a clear image of the leaf and records the GPS location[11] and save it with the location coordinates as the file name. The microcontroller used for this purpose is the RaspberryPi 4. This work is still in progress, and to test our model, we took the pictures of the leaves ourselves and tested it with our model. The below shown figure [25] depicts how the autonomous rover will look like.

Figure 8: Sample Rover

IV. EXPERIMENTS AND RESULTS

The entire dataset that we used had around 7000 images[19]. 80-20 rule was applied in the splitting for testing. So, around 5500 images were in the training set and 1500 in the testing set. The images in the dataset were of size 256x256, but for our models, we need 224x224 for VGG16 and 299x299 for the InceptionResNetV2 model. This resizing of the images was achieved by using the image processing techniques in OpenCV. VGG16 architecture-based model had 134,289,223 parameters to be trained whereas the InceptionResNetV2 based model had 58,091,591 parameters. Both models are large. VGG16 is large in terms of the parameters, whereas the InceptionResNetV2 is large in terms of the depth. To train such huge models, it took around 45 hours. We trained the model for 25 epochs with a batch size of 32 images per batch. The metrics that were observed during the training part were the training and validation loss along with the training and validation accuracy. The following graph shows the results of both models.

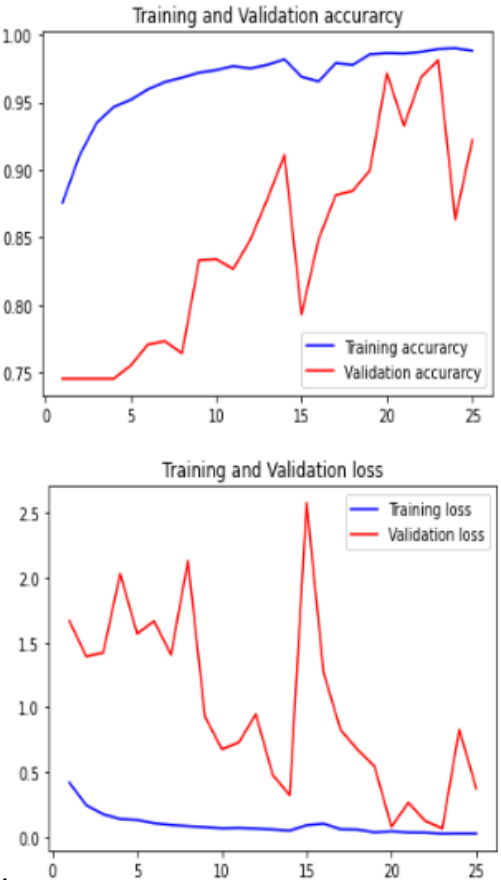


Figure 8: VGG16 Model Results Graph

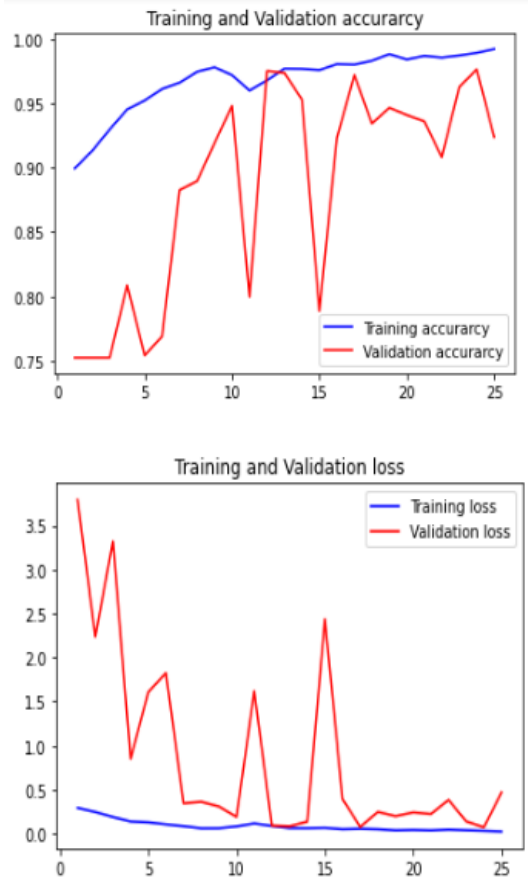


Figure 9: InceptionResNetV2 Results Graph

The below table has a summary of the results of the two models.

Table 2: Result Summary Table

Models	Number of parameters	Training Accuracy	Validation Accuracy
VGG16	134,289,223	97.56 %	93.21 %
InceptionResNetV2	58,091,591	98.32 %	95.24 %

The complete training was done only once, and the best parameters were saved and used in the later testing process. We randomly took a few images from the dataset and tested the model again. The following were the output obtained from them.

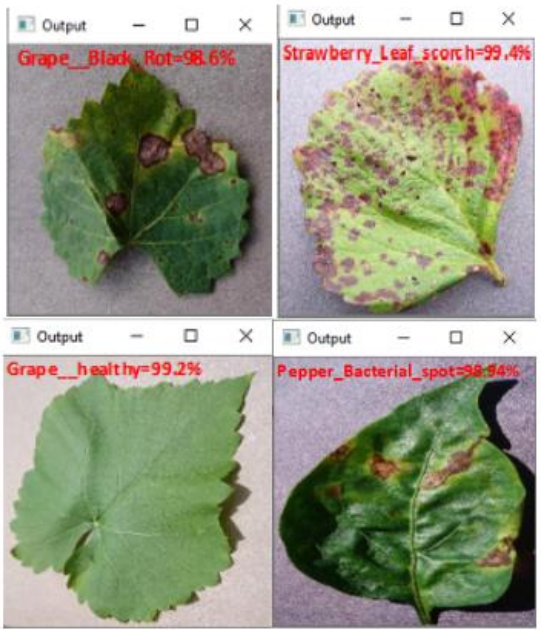


Figure 10: Random Test output

V. CONCLUSION

Plant diseases are a major threat to the farmer's income. Monitoring plant's diseases in a farm with human labor is a tiresome work that consumes a lot of capital and time. Therefore, the need for an automated system to identify plant disease with the help of modern technologies is huge. Deep learning techniques, especially Convolutional neural networks, have proven to produce high impact on the classification of the plants into the categories of their respective diseases. In this work, we employed two deep learning models, namely VGG16 and InceptionResNetV2, whose underlying architecture are based on convolutional neural networks. The PlantVillage which is used here, has images of around 20 species, in which we use three species (Grape, Strawberry, Pepper) in our model. Four major diseases were found in the above mentioned species which are: Black rot and Black measles in grape,

Retrieval Number: B3469079220/2020©BEIESP
DOI:10.35940/ijrte.B3469.079220
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Leaf scorch in strawberry, Bacterial spot in pepper. Around 7000 images were available for these plants, we trained the models on 80% of the data and tested it on the rest 20% of the data. The accuracy we have achieved here are 93.21%(test) and 97.56%(train) using the VGG16 architecture based model, and 95.24%(test) and 98.32%(train) from InceptionResNetV2 architecture based model. Clearly, among the two, InceptionResNetV2 based model outperforms the VGG16 based model in terms of accuracy. However, the accuracy is a bit low in both the models due to a lesser number of data trained to the model. If the dataset used for training had been larger, then the accuracy might increase as well[7]. For training such a large model with huge dataset, we need more robust computation facility. From this result, we conclude that the wider architecture InceptionResNetV2 performs well than the VGG16 architecture, so stacking up the layers does not increase the accuracy as compared with adding wider layers at the same level in the neural network. Soon, we have planned to extend this approach further to all the species of plants, along with a real-time classifier within the microcontroller itself, which enables the output to be more efficient and effective. We have also planned to come up with an app to monitor these stuff and to create alerts for any diseased plant being detected. We would also like to add more data to our training set from different sources and through field visits.

REFERENCES

1. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi,"Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning",Computer Vision and Pattern Recognition, Cornell University,2016.
2. Devie Rosa Anamisa , Muhammad Yusuf, Wahyudi Agustiono ,"Technologies, Methods, and Approaches on Detection System of Plant Pests and Diseases", EECSI 2019
3. Edna Chebet Too,Li Yujian, SamNjuki, Liu Yingchun,"A comparative study of fine-tuning deep learning models for plant disease identification"Computers and Electronics in Agriculture,
4. Halil Durmus, Ece Olcay Gune, Murvet Koro, "Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning", 6th International Conference on Agro-Geoinformatics, 2017.
5. Hilman F. Pardede, Endang Suryawati, Rika Sustika, and Vicky Zilvan, "Unsupervised Convolutional Autoencoder-Based Feature Learning for Automatic Detection of Plant Diseases ", International Conference on Computer, Control, Informatics and its Applications, 2018.
6. Hussam Qassim , Abhishek Verma ,David ,"Compressed residual-VGG16 CNN model for big data places image recognition", IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC),2018.
7. Jayme Garcia Arnal Barbedo,"Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification"Computers and Electronics in Agriculture Volume 153, 2018, Pages 46-53
8. Kamlesh Golhania, Siva.K.Balasundram, Ganesan Vadamalai, Biswajeet Pradhan,"A review of neural networks in plant disease detection using hyperspectral data"Information Processing in Agriculture,Volume 5, Issue 3, 2018, Pages 354-371
9. Konstantinos P.Ferentinos,"Deep learning models for plant disease detection and diagnosis",Computers and Electronics in Agriculture,Volume 145, 2018, Pp: 311-318
10. L. Sherly Puspha Annabel, V. Muthulakshmi, "AI-Powered Image-Based Tomato Leaf Disease Detection", Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), 2019
11. Liqiong Tang , Phillip Abplanalp,"GPS guided farm mapping and waypoint tracking mobile robotic system",9th IEEE Conference on Industrial Electronics and Applications,2014.
12. Long D. Nguyen, Dongyun Lin, Zhiping Lin,Jiuwen Cao,"Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation" IEEE International Symposium on Circuits and Systems (ISCAS),2018.
13. Nuttakarn Kitpo,Masahiro Inoue ,"Early Rice Disease Detection and Position Mapping System using Drone and IoT Architecture", 12th South East Asian Technical University Consortium Sysmposium (SEATUC), Yogyakarta, Indonesia,2018.
14. Parul Sharma, Yash Paul Singh Berwal, Wiqas Ghai, " KrishiMitr (Farmer's Friend): Using Machine Learning to Identify Diseases in plants" , IEEE International Conference on Internet of Things and Intelligence System (IoTaIS),2018.
15. Poojan Panchal, Vignesh Charan Raman, Shamla Mantri,"Plant Diseases Detection and Classification using Machine Learning Models ", 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), 2019.
16. Sammy V. Militante, Bobby D. Gerardo, Nanette V. Dionisio, "Plant Leaf Detection and Disease Recognition using Deep Learning ",IEEE Eurasia Conference on IOT, Communication and Engineering,2019
17. Sammy V. Militante, Bobby D. Gerardo, Ruji P. Medina,"Sugarcane Disease Recognition using Deep Learning", IEEE Eurasia Conference on IOT, Communication and Engineering,2019.
18. Shamse Tasnim Cynthia, Kazi Md. Shahrukh Hossain, Md. Nazmul Hasan, Md. Asaduzzaman, Amit Kumar Das," Automated Detection of Plant Diseases Using Image Processing and Faster R-CNN Algorithm ", International Conference on Sustainable Technologies for Industry 4.0 (STI), 2019.
19. Sharada P. Mohanty,David P. Hughes,Marcel Salathe,"Using Deep Learning for Image-Based Plant Disease Detection"Frontiers in plant science, 2016
20. Suyash S. Patil, Sandeep A. Thorat,"Early Detection of Grapes Diseases Using Machine Learning and IoT",Second International Conference on Cognitive Computing and Information Processing (CCIP),2016
21. https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d ; Raimi Karim; July 29,2019.
22. https://medium.com/@mannasiladittya/building-inception-resnet-v2-in-keras-from-scratch-a3546c4d93f0; Siladittya Manna; April 12, 2019.
23. https://neurohive.io/en/popular-networks/vgg16/ ; Muneeb ul Hassan; November 20, 2018.
24. https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c ;Rohit Thakur; August 16,2019
25. https://cinescopophilia.com/the-worlds-first-remote-dolly-for-360o-vr-cameras-coming-to-nab-2016/ ; Cinescopophilia;
26. https://cs231n.github.io/convolutional-networks/#conv ; CS231 Stanford University
27. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 ; Arden Dertat; November8, 2017
28. https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202 ; Bharath Raj; March 29,2018
29. https://towardsdatascience.com/residual-blocks-building-blocks-of-resnet-fd90ca15d6ec ;Sabyasachi Sahoo; November 27, 2018

AUTHORS PROFILE

Dr. R.Jothilakshmi, working as Associate Professor in the Department of Information Technology, at R.M.D Engineering College .She received her Bachelor of Engineering (CSE) from Bharathidasan University, M.E.(CSE) and Ph.D (ICE) from Anna University. She has more than 20 years of teaching experience. She is member of ISTE, CSI. She has published several research papers in many conferences and Journals in the areas of Semantic Query, Information Retrieval and Image Processing.

Sharanesh R, is a student currently pursuing his B.Tech in Electrical Engineering at Indian Institute of Technology Madras(Expected graduation in 2023). He is an Deep learning, IoT and Embedded systems Enthusiast. He has made few free lancing projects using Embedded systems. His areas of interests include Deep learning, Embedded systems, Internet of Things (IoT), Block chain and Robotics.