iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VII **Month of publication:** July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36582

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Plant Leaf Disease Prediction

Vaishnavi Monigari[1], G. Khyathi Sri[2], T. Prathima[3]
[1, 2, 3]*Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India*

Abstract: ***The Indian economy relies heavily on agriculture productivity. A lot is at stake when a plant is struck with a disease that causes a significant loss in production, economic losses, and a reduction in the quality and quantity of agricultural products. It is crucial to identify plant diseases in order to prevent the loss of agricultural yield and quantity.***
Currently, more and more attention has been paid to plant diseases detection in monitoring the large acres of crops. Monitoring the health of the plants and detecting diseases is crucial for sustainable agriculture. Plant diseases are challenging to monitor manually as it requires a great deal of work, expertise on plant diseases, and excessive processing time. Hence, this can be achieved by utilizing image processing techniques for plant disease detection. These techniques include image acquisition, image filtering, segmentation, feature extraction, and classification.
Convolutional Neural Network's(CNN) are the state of the art in image recognition and have the ability to give prompt and definitive diagnoses. We trained a deep convolutional neural network using 20639 images on 15 folders of diseased and healthy plant leaves. This project aims to develop an optimal and more accurate method for detecting diseases of plants by analysing leaf images.

I. INTRODUCTION

Agriculture production in the Indian economy is more than just food. Today's agricultural land mass has grown so large that it has become an important part of its economy. In India, 60-70% of population relies on agriculture sector. Plant diseases often cause severe loss of vegetables and crops. Plant diseases can also affect human health by secreting toxic metabolites. The study of plant disease involves detection of visual patterns in the plants. Diagnosis of plant diseases is an important part of cultivation as failure will affect the quantity and quality of product and human health . There are various types of plant diseases caused by organisms like virus, bacteria and fungus. An automated disease identification process can be helpful in identifying plant pathology at an early stage. The early detection of disease has a positive effect on plant health. In most of the cases, disease symptoms are seen on the leaves, stem and fruit. The indications on the plant leaves are used to diagnose the disease faster, more reliably and at lower costs.
In general, the technique for diagnosing plant diseases is naked eye inspection by farmers, which allows for disease recognition and detection. A large number of specialists and constant plant monitoring is required for this, which incur a cost when dealing with large farms. However, in certain nations, farmers lack adequate facilities or even the knowledge of how to contact experts. This means that consulting professionals is both expensive and time consuming.
Image processing and machine learning methods were also used to identify various plant diseases before the emergence of deep learning. To prepare images for the next steps, image processing methods such as image enhancement, segmentation, colour space conversion, and altering are used. The image's key features are then extracted and used as the input for the classifier. The overall classification precision is determined by the image processing and feature extraction techniques.
However, recent research has shown that networks trained on generic data will achieve state-of-the-art efficiency. CNNs are supervised multi-layer networks that can dynamically learn features from datasets. In almost all significant classification tasks, CNNs have recently achieved state-of-the-art results. In the same architecture, it will isolate features and classify them.

II. LITERATURE SURVEY

The paper presents a deep learning model called the plant disease detector, which is able to detect different diseases of plants based on images of their leaves. This model is developed by applying advance neural network techniques. Initially dataset is augmented to increase sample size, and subsequently Convolution Neural Network (CNN) with multiple convolution and pooling layers is applied. A model is trained and then tested properly to validate its results. Proposed model has achieved 98.3% testing accuracy.85% of the data collected is used for training and 15% of the data is used for testing from PlantVillage dataset. These images show healthy and diseased plants. This research focuses on deep learning models to detect disease in plant leaves. But in future, these models can be integrated with drones or other systems to locate diseases living in plants in order to treat them accordingly[1].

©IJRASET: All Rights are Reserved

A public dataset of 54,306 images of healthy and diseased plant leaves has been used to train a deep convolutional neural network to identify 14 crops and 26 diseases. An accuracy of 99.35% was achieved for this model on a held-out test set, showing the success of this approach. The general approach of training deep learning models on increasingly large and publicly accessible image datasets presents a path toward the mass deployment of smartphone-aided crop disease detection[2]

Image processing and machine learning can be used to improve plant diseases detection techniques, thereby reducing the time, effort, and knowledge necessary for the detection of infected plants. It involves image acquisition, filtering, segmentation, feature extraction, and classification. This paper proposes a way to best detect disease by detecting its appearance from plant images and, if present, evaluating its type among Alternaria Alternata, Anthracnose, Bacterial Blight and Cercospora Leaf Spot. As the minimum accuracy is 95.774 percent and the maximum accuracy is 99.874 percent, this process gives almost accurate results. The process detects the diseases by the area of disease, although it has a low affected region[3]

A neural network was trained on simple leaf images of healthy and diseased plants in this study using deep learning to detect and diagnose plant diseases. The models were trained on an open database of 87,848 images from 25 different plants in 58 distinct plant-disease combinations. The best performing model architecture had a success rate of 99.53% in indicating the corresponding plant-disease combinations (or healthy plants). Since the model has a high success rate, it is an excellent early warning tool that could be further developed to support the implementation of an integrated disease identification system in real-time[4].

A mathematical model is proposed that detects and recognizes plant diseases through deep learning, improving its accuracy, generality, and training efficiency. After recognizing leaves placed in complex surroundings, the region proposal network (RPN) is applied to extract symptom features from the pictures following Chan-Vese algorithm. The segmented images are then input into the transfer learning model with the training dataset of diseased leaves provided. Using three types of diseases (black rot, bacterial plaque, and rust), the model shows higher accuracy than the traditional method, thus reducing the influence of disease on production and making it more beneficial to sustainable agriculture. This paper presents a deep learning algorithm that is of great significance to intelligent agriculture, environmental protection, and agricultural production[5]

The current shortcomings of current plant disease detection models are discussed. The new dataset contains 79,265 leaf images with the aim of being the largest dataset to contain leaf images. The images were taken in various weather conditions, under various lighting conditions and during daylight hours with an unreliable background resembling realistic scenarios. Traditional augmentation methods and state-of-the-art style generative adversarial networks were used to augment the number of images in the dataset. Tests were conducted to verify the effectiveness of training in a controlled setting and usage in the real world to accurately identify diseases of plants on natural and detection of multiple diseases in a single leaf. The trained model achieved an accuracy rate of 93.67%. Finally, a new two-stage architecture of a neural network was proposed for plant disease classification in a real environment[6]

In this paper, a system was proposed for classifying three diseases affecting grapes– Anthracnose, Powdery Mildew and Downy Mildew and identifying the severity of these diseases using image processing and machine learning algorithms. U 900 images of disease infected grapes leaves were acquired by the farmers and field workers from the fields. Images of single leaf or bunch of leaves were captured with background from different distances and at different angles using mobile phone cameras with varying resolution starting from less than 1 megapixel to 13 megapixel. This proposed disease detection algorithm consists of 4 main stages: (a)Pre-processing of the input images, (b) Leaf extraction from the background, (c) Disease patch identification and (d)Background removal. Performance of four machine learning algorithms namely, PNN, BPNN, SVM and Random Forest are compared, for separating the background from disease patches and classifying between the different diseases. The performance of different texture features like local texture filters, Local Binary Patterns, GLCM features, and some statistical features in RGB plane for classification are also observed. It is observed that the proposed system achieves best classification accuracy of 86% using Random Forest and GLCM features. [7]

In this paper, a real-time decision support system integrated with a camera sensor module is designed and developed for the identification of plant disease. The performance of three machine learning algorithms, Extreme Learning Machine (ELM) and Support Vector Machine (SVM) with linear and polynomial kernels was analyzed. A real-time decision support system using extreme learning machine was designed and developed using Raspberry PI hardware. Results demonstrated that the performance parameters, namely accuracy and sensitivity of the extreme learning machine, is 95% and is higher when compared to the other adopted classifiers. It is also observed that the developed real-time hardware with Extreme Learning Machine classifier is highly capable of detecting three different plant diseases and can be extended to detect many more plant diseases by training it with wide range of train datasets. [8]

©IJRASET: All Rights are Reserved

Table 1: Comparison of performance of existing works

S No	Title	Author	Approach	Dataset	Accuracy Reported
1	Plant Disease Detection using Deep Learning	Rozina Chohan & Murk Chohan	Convolutional Neural network	-	98.3%
2	Using deep learning for Image-based plant disease detection	Sharada P.Mohanty & David P.Hughes	Deep Convolutional Neural network	Github	98.21%
3	Plant Leaf disease detection and classification using Machine Learning	Vijeta Shrivastava & Indrajit Das	Gray co-occurrence Matrix method and Support Vector machine	Kaggle	Minimum:95.774% and Maximum:98.874%
4	Deep learning models for plant Leaf disease detection	K P. Fereninos	Convolutional Neural network	-	99.53%
5	Plant disease Identification based on Deep learning algorithm in Smart Farming	Yan Guo & Jin Zhang	Region Proposal Network (RPN) algorithm and CV algorithm	-	83.57%
6	Solving Current limitations of deep learning based approaches for plant leaf disease detection	Marko Arsenovic & Mirjana Karanovi	Traditional Augmentation methods , GAN and neural networks.	Kaggle	93.47%
7	Application of Random Forest for Classification of Grapes Diseases from Images Captured in Uncontrolled Environments	Sandika Biswas, Avil Saunshi & Sanat Sarangi	Probabilistic Neural Network (PNN) ,Back Propagation Neural Network (BPNN) ,Support Vector Machine (SVM) and Random Forest	-	86%
8	Intelligent Plant Disease Identification System Using Machine Learning	Paramasivam Alagumariappan, Najumnissa Jamal Dewan	Extreme Learning Machine (ELM) and Support Vector Machine (SVM) with linear and polynomial kernels	-	95%

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

III. PROPOSED METHODOLOGY

CNN algorithms analyze an image and extract its features. Convolutional neural networks are deep learning algorithms that can process large datasets containing millions of parameters, modeled on 2D images, and connect the resulting representations to the corresponding outputs. A CNN is a supervised multilayer network that can dynamically learn new features from datasets. In nearly all significant classification challenges, CNNs have achieved state-of-the-art results recently. In the same architecture, they are also able to systematically isolate features and categorize them.

A. Flow Chart

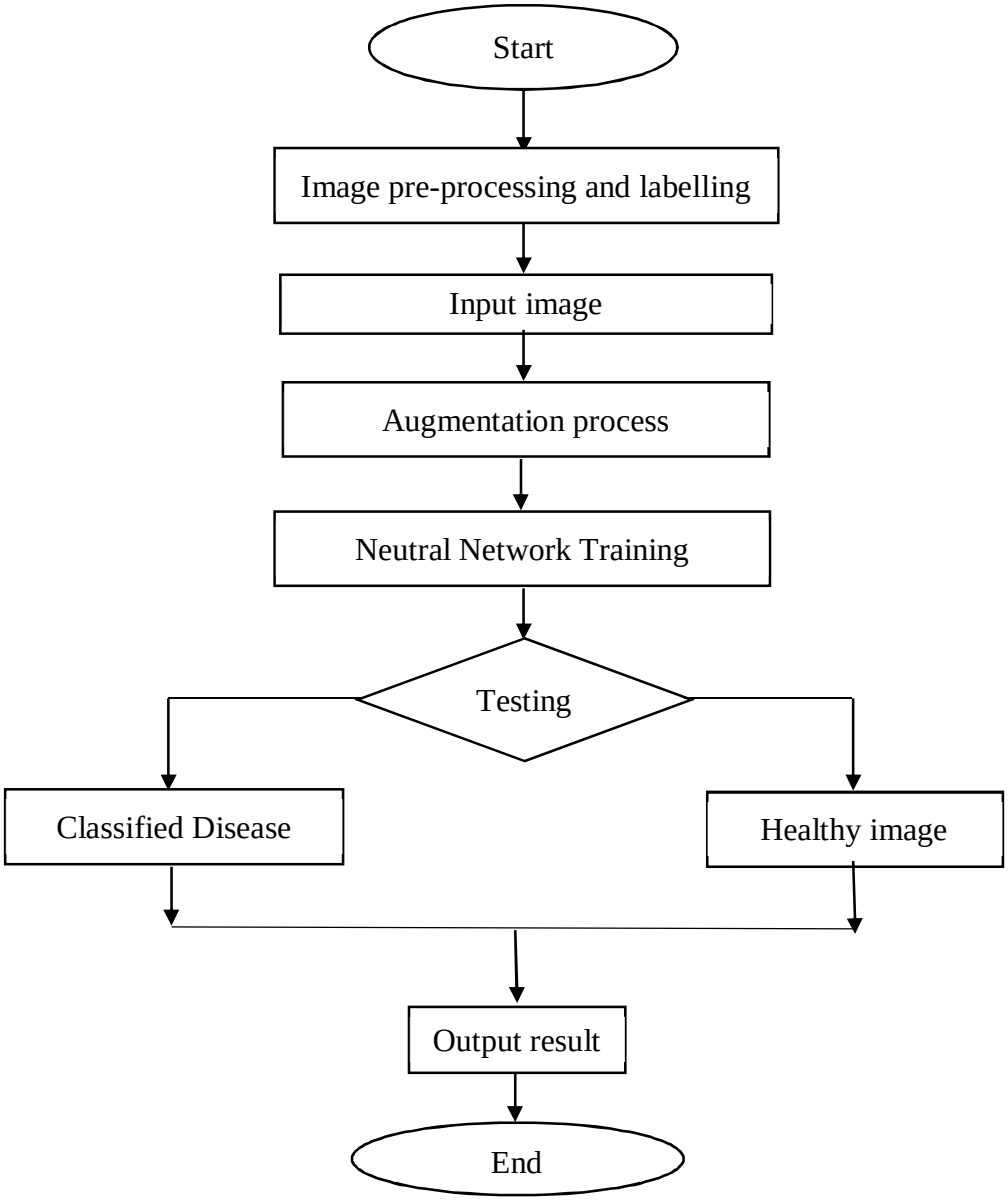


Fig. 1: System Design

Collect images of plants with and without disease. A Python script calculated the training time by automatically resizing the images, which was calculated using the OpenCV framework. By augmenting the dataset and adding distortion to the images, overfitting can be reduced during the training period. The Deep Neural Network is trained on datasets of healthy and diseased crop leaves. It serves its purpose by classifying images of leaves into diseased or healthy categories based on their pattern of defect. As the leaves have texture and visual similarities, they are attributes for identifying disease types. Hence, computational vision applied to deep learning provides an efficient way to solve the problem.

©IJRASET: All Rights are Reserved

B. Dataset Description

This dataset consists of 20,639 images of diseased and healthy plant leaves, which were classified into 15 classes to train a deep convolutional neural network which can identify the diseases.

Table 2: Description of the dataset

Class	Plant Name	Healthy or Diseased	Disease Name	Images (Number)
C_1	Pepper	Diseased	Bacterial Spot	997
C_2	Pepper	Healthy	-	1478
C_3	Potato	Diseased	Early Blight	1000
C_4	Potato	Healthy	-	152
C_5	Potato	Diseased	Late Blight	1000
C_6	Tomato	Diseased	Target Spot	1401
C_7	Tomato	Diseased	Mosaic Virus	373
C_8	Tomato	Diseased	Yellow leaf Curl Virus	3209
C_9	Tomato	Diseased	Bacterial Spot	2127
C_10	Tomato	Diseased	Early Blight	1000
C_11	Tomato	Healthy	-	1591
C_12	Tomato	Diseased	Late Blight	1909
C_13	Tomato	Diseased	Leaf Mold	952
C_14	Tomato	Diseased	Septoria Leaf Spot	1771
C_15	Tomato	Diseased	Spider Mites two spotted Spider Mite	1676

C. Data Preprocessing

The dataset included images that were resized to minimize training time, which was calculated automatically by a Python script that uses the OpenCV framework. The input data is pre-processed by scaling the data points from [0, 255] (the minimum and maximum RGB values of the image) to [0, 1]. The dataset is divided into two parts, one for training and one for testing. 80% of the dataset is for training, and 20% for testing. A training dataset consists of 16,511 images and testing is made of 4,128 images. The training dataset is used to train the model while the testing dataset is kept unseen so that accuracy of the model can be tested.

D. Data Augmentation

Data augmentation is a technique for increasing the number of images in a database. Various operations such as shifting, rotating, zooming, and flipping are applied to image datasets to diversify our dataset. By augmenting the dataset and adding distortion to the images, overfitting can be reduced during the training period. The Keras ImageDataGenerator class implements in-place data augmentation or on-the-fly data augmentation. Through this type of augmentation of data, we can make sure that our network, when trained, sees new variations every time epoch. It allows us to come up with high results utilizing a smaller dataset[9].

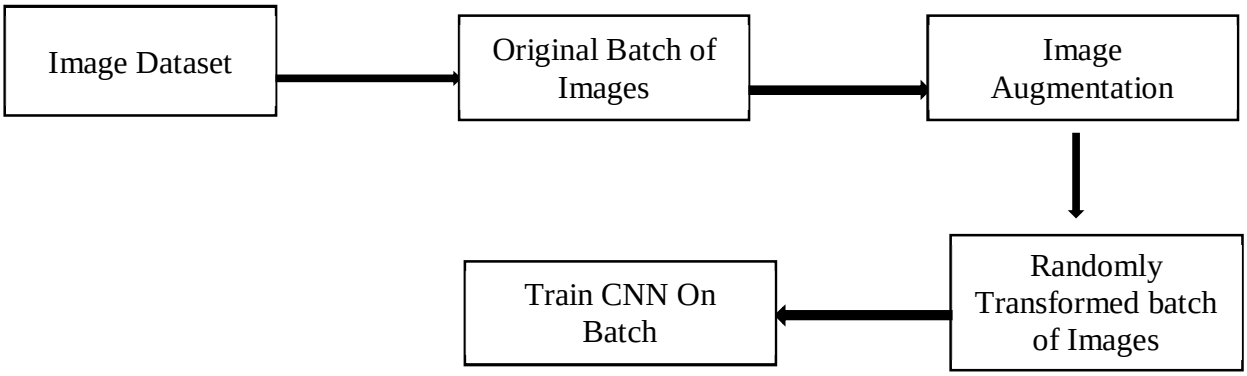


Fig. 2. In-place Data Augmentation

©IJRASET: All Rights are Reserved

Figure 2 demonstrates the process of applying in-place data augmentation

1) *Step 1:* The ImageDataGenerator is presented with an input batch of images.
2) *Step 2:* Next, the ImageDataGenerator transforms each image into a random series of rotations, flipping , cropping etc.
3) *Step 3:* The randomly transformed batch is trained by using CNN.

E. Architecture of Convolutional Neural Network

CNN architecture is divided into two main parts :

1) A convolution tool that separates and categorizes the various features of images for analysis in what is called Feature Extraction.
2) Convolution is applied to the output of the fully connected layer and predicts the class of image based on the features extracted before.

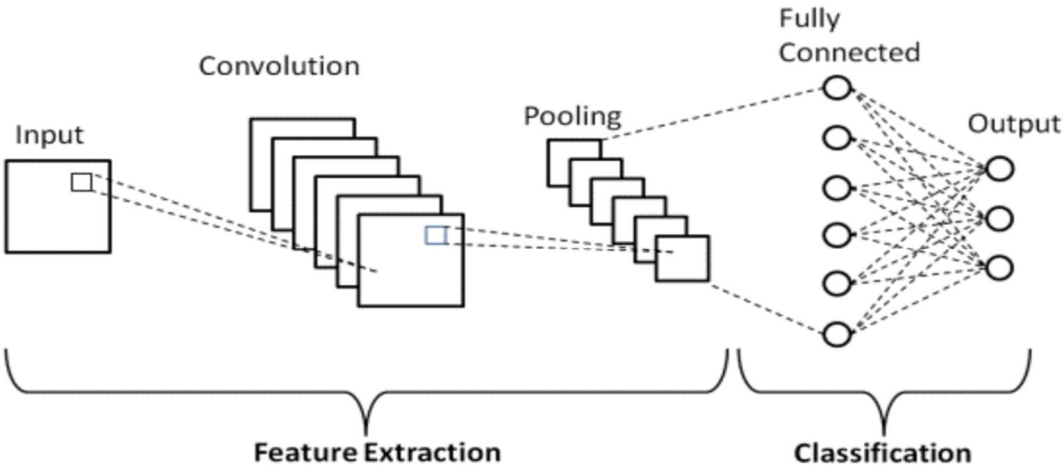


Fig. 3: CNN Architecture

F. Convolution Layers

The three layers that make up the CNN is the convolutional layer, pooling layer, and fully-connected layer (FC layer). When these layers are stacked, a CNN is formed. There are two additional layers to these three which are the dropout layer and the activation function. Convolutional Layer is the first layer that focuses on extracting features from the input images. In this layer, the convolution mathematical process is performed between each input image and a set of convolution filters of a particular size. By sliding the filter over the input image, a dot product is computed between the filter and the parts of the input image corresponding to the size of the filter. Feature maps represent the output. Later, the Feature map can be used to as input to other layers[10]

The Conv2D function takes the following arguments:

1) *Filters* - The number of different filter methods (feature detectors) that will be applied to the original image for creating the feature map. There are different types of filters, such as the Edge Detection Filter and Blur Filter.
2) *Kernel Size* - It gives the dimension of the (n x n) matrix of a convolution filter.
3) *Activation* - The activation function for the neurons. We use a Rectifier Linear Unit (Relu) function as an activation function for every layer besides the output layer. We have also added non-linearity to our network using ReLU. This is essential in identifying any linear relationships within the feature map.
4) *Input Layer* - It takes the shape of the Input Images and the number of channels (3 for color)

G. Pooling Layer

In our convolutional neural network, the next layer is called the pooling layer. One of the main objectives of the pooling layer is to minimize the spatial dimension of the data propagating through the network. Pooling can be achieved in two different ways in convolutional neural networks. Max- pooling and average pooling. In Max Pooling which is the most common in two, for each section of image we scan the highest value. Average Pooling calculates the average of an image's elements within a predefined sized region. Pooling Layer serves as the bridge between Convolutional Layer and the Fully Connected Layer.

©IJRASET: All Rights are Reserved

H. Fully Connected Layer

Fully Connected Layers (FC) consist of weights and biases as well as neurons, and they are used to connect neurons between different layers. In this layer, we flatten the output of the last convolutional layer and connect every node of the current layer with every other node of the next layer. This layer basically takes its input from the preceding layer, whether it is a convolutional layer, ReLU, or pooling layer. At this stage, the classification process begins.

I. Dropout

When all the features are connected to the FC layer, it can lead to overfitting of the training dataset. A model is said to be overfitted if it can perform proficiently on training datasets but then shows negative performance when applied to new datasets. To solve this problem, a dropout layer is used wherein a few neurons are removed during the training process, thus reducing the size of the neural network model. On passing a dropout of 0.2, 20%of the nodes are removed randomly from the neural network.

J. Activation

Activation functions plays a major role in the process of neural network. It determines what information from the model should be fired in the forward direction, and which information should not at the end of the network. Hence, it adds nonlinearity to the network. It has been observed that there are quite a few widely used activation functions. The most frequently utilised activation functions are Sigmoid, tanH, Softmax, and ReLU. Each activation function has its own specific application. For a multi class classification we generally use ReLU and Softmax functions.

1) *ReLU:* The rectified linear unit (ReLU) function is the most widely used activation function in today's networks. There is an advantage of using the ReLU function compared to the other activation functions in that it does not activate all the neurons at once. If the input is negative, then it is converted to 0, and the neuron is not activated. If the input is positive, it returns the positive value of x and the neurons get activated. Consequently, only a few neurons are activated at a time, making the network sparse and very efficient. The ReLU function also served as a significant advancement in the field of deep learning by overcoming the vanishing gradient problem.

$$\text{ReLU} = \max(0,x)$$

2) *Softmax:* The softmax function is ideally used in the output layer of the classifier where we are actually trying to get the probabilities to define the class of each input. As a result, it is easier for us to categorize data points and determine to which category they belong[11]

A convolutional neural network will be used to classify images without relying on pre-trained models. There are a number of popular pre-trained models available that can tell the difference between hundreds of classes without training each of them. These models have relatively complex architectures that help them handle hundreds of thousands of classes. The architecture can be difficult for a beginner to visualize. Keras make building of custom CNN's easier. We developed this project using Custom CNN.

K. Model

We now make use of Sequential model. Sequential model API is a way to build deep learning models in which sequential classes and model layers are created and added.

The input to a convolutional neural network, is an (n x m x 3) for colored images, where the number 3 represents the red, green, and blue components of each pixel in the image.

For this model, we first create a 2D convolutional layer with 32 filters of 3 x 3 kernels and a Rectified Linear Unit (ReLU) activation. In the following layers, we perform batch normalization which is used to scale data by a certain factor and pooling we use maximum pooling with a pooling size of two. Next, two blocks of 2D Convolutional layer are created with 64 filters and ReLU activation followed by a pooling layer. Finaly we add a layer of Convolutional layer with 32 filters followed by a layer of ReLU activation and pooling.

Then we flatten the output from these layers so the data can proceed to fully connected layers. Flatten is used to convert data into a 1-Dimensional form. We add another 512 dense layers with a dropout of 0.2. Finally, we use the softmax activation function to convert the outputs into probability values.

©IJRASET: All Rights are Reserved

L. Training the Model

The next step is to compile the model and then train it on a training dataset.

The following parameters need to be specified for compiling the model:

Optimizer - An optimizer is an algorithm or methodology used to reduce losses by modifying the weights and learning rate of a neural network. Optimizers train models faster and work more efficiently. As we have a multi class classification problem ,we use the Adam optimization technique because it always leads to a smoother way than other optimization techniques. Adam is an optimization algorithm that uses adaptive moment estimation to generate more efficient neural network weights[12].

loss → The cost function that calculates the difference between predicted and actual values. In our case we will be using " sparse categorical_cross entropy". Sparse categorical cross entropy can be used for integer targets instead of categorical vectors[13]

In order to fit the model, we have to specify the following parameters:

batch_size → Number of images to be used for training our CNN model before back-propagating the weights.

epochs → An epoch is a measure of how many times the whole training set of images is used once.

We train our model over 20 epochs and 30 epochs. A higher number of training epochs increases its accuracy along with lowering the loss.

IV. RESULTS

A. Training v/s Validation

We plot a graph to illustrate the maximum accuracy the model achieved during training and validation while minimizing the loss.

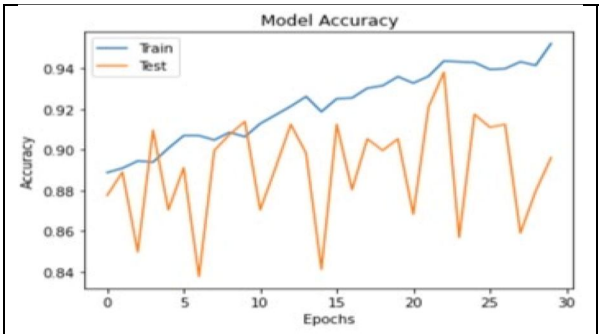


Fig. 4: Training v/s Validation Accuracy

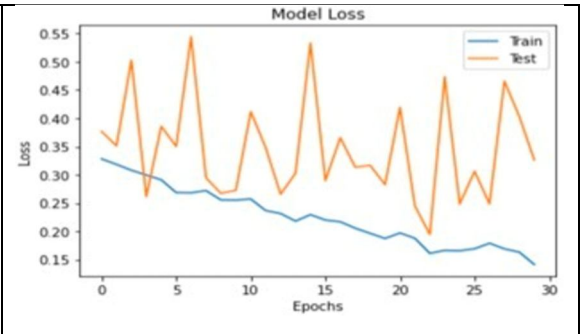


Fig. 5: Training v/s Validation Loss

From the above graphs, we observe that as the training accuracy increases, validation accuracy increases. Similarly as the training loss decreases, the validation loss decreases too.

B. Confusion Matrix

```
[[ 78   2   0   0   0   0   0   0   0   0   0   0   0   2   0]
 [  0  97   0   1   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0 108   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0  31   0   1   0   1   0   0   0   0   0   0   1]
 [  0   0   3   2  81   0   0   0   0   2   0   0   0   0   2]
 [  0   0   0   0   1  64   2   0   0   2  18   0   0   4   4]
 [  0   0   0   0   0   0  81   0   0   0   0   0   0   3   0]
 [  1   0   0   0   0   0   0 101   3   1   0   0   0   0   1]
 [  0   0   0   0   3   1   0   2  85  10   0   0   0   0   0]
 [  0   0   0   0   1   3   0   2   1  86   0   1   1   4   2]
 [  0   0   0   0   0   0   0   0   0   0 111   0   0   0   0]
 [  2   0   1   0   4   0   0   0   0  10   0  70   0   1   3]
 [  0   0   0   0   0   0   0   0   0   0   0   1  93   5   1]
 [  1   0   0   0   0   0   2   0   0   2   0   0   0  78   0]
 [  1   0   0   0   0   2   4   0   0   1  15   0   1   1  95]]
              precision    recall  f1-score   support

           0       0.94      0.95      0.95        82
           1       0.98      0.99      0.98        98
           2       0.96      1.00      0.98       108
           3       0.91      0.91      0.91        34
           4       0.90      0.90      0.90        90
           5       0.90      0.67      0.77        95
           6       0.91      0.96      0.94        84
           7       0.95      0.94      0.95       107
           8       0.96      0.84      0.89       101
           9       0.75      0.85      0.80       101
          10       0.77      1.00      0.87       111
          11       0.97      0.77      0.86        91
          12       0.98      0.93      0.95       100
          13       0.80      0.94      0.86        83
          14       0.87      0.79      0.83       120

    accuracy                           0.90      1405
   macro avg       0.90      0.90      0.90      1405
weighted avg       0.90      0.90      0.90      1405
```

Fig. 6: Confusion Matrix & Evaluation metric results

©IJRASET: All Rights are Reserved

All the pairs both having disease and not having disease were plotted on a confusion matrix. A confusion matrix measures the degree of accuracy of a classification model with respect to each classification category. A trained model's evaluation and output is determined by True positives(TP), True negatives(TN), False positives(FP), False negatives(FN). For evaluation we also used F1, which combines both precision and recall in one term. The higher the F1-Score, the better the model. For all three metrics, models with 0 perform the worst while models with 1 perform the best[14]. Figure 5.3 displays the precision, recall, F1 and support for each class. The overall accuracy reported is 90%.

1) *Precision:* Precision describes all the positive classes correctly predicted by the model; how many of those are actually positive. The precision is calculated by taking the number of correctly classified positive examples divided by the number of predicted positive examples. The equation can be written as:

$$Recall = \frac{TP}{TP + FP}$$

2) *Recall:* It defines how much the model predicted correctly among all positive classes. Recall is the ratio between the number of correctly classified positive examples and the total number of positive examples. The equation can be written as:

$$Recall = \frac{TP}{TP + FN}$$

3) *F1-score:* F1-score gives an overall estimation of the precision and recall of a test subject. It is the harmonic mean of the precision and recall of a test subject[14]. Formally, F1-call can be defined as,

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall}$$

4) *Accuracy:* Accuracy is a metric for assessing classification models. Informally, accuracy is the fraction of predictions that are correct. Formally, accuracy can be defined as follows:

$$Accuracy = \frac{No.of\ correct\ predictions}{Total\ no.of\ predictions}$$

C. Outputs Screenshots

A random sample of images is taken from the dataset and predicts the plant image's disease and class.

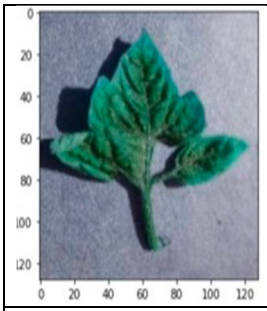


Fig. 7: Tomato Yellow leaf curl virus

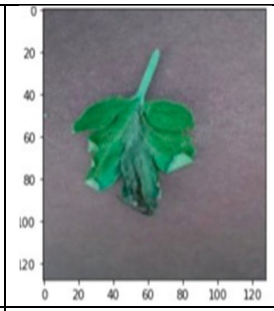


Fig. 8: Tomato Late Blight

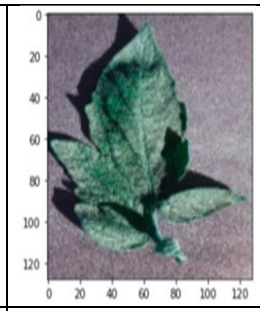


Fig. 9: Tomato Health leaf

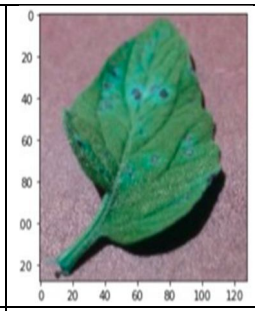


Fig. 9: Tomato Septoria Leaf Spot

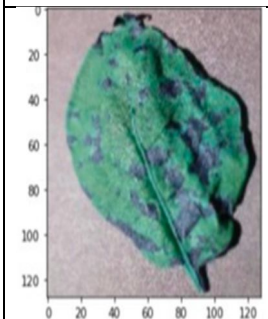


Fig. 10: Potato Early Blight

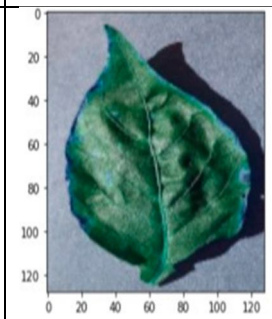


Fig. 11: Pepper bell Bacterial Spot

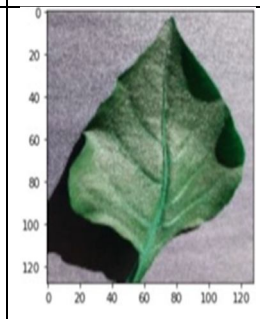


Fig. 12: Pepper bell Healthy

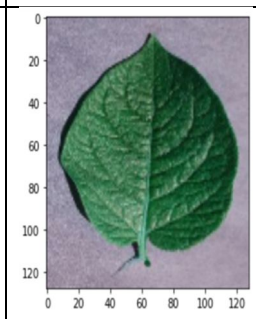


Fig. 13 Potato Healthy leaf

©IJRASET: All Rights are Reserved

D. Comparison of Results

Table 3: Results of existing versus custom CNN

S. No	Architecture	Dataset	Accuracy
1	AlexNet	PlantVillage	96.30% [15]
		PlantVillage	83.63% [16]
		PlantVillage	97.49 [17]
2	GoogLeNet	PlantVillage	85.74% [16]
3	MobileNet	PlantVillage	97.1% [14]
4	ResNet50	PlantVillage	98.2% [14]
5	InceptionV3	PlantVillage	97.1% [14]
6	VGG-16	PlantVillage	97.23% [17]
7	Custom Model	PlantVillage	90%

V. CONCLUSION

Even though there are various methods for detecting and classifying plant diseases using automatic or computer vision, research into this field has been lacking. In addition, there are few commercial options, with the exception of those focusing on the identification of plant species via photographs.

Over the last few years, there has been tremendous progress in the performance of convolutional neural networks. The new generation of convolutional neural networks (CNNs) has shown promising results in the field of image recognition. A novel approach to automatically classifying and detecting plant diseases from leaf images was examined through this project utilizing deep learning techniques. With an accuracy of 90%, the developed model could distinguish healthy leaves from eight diseases that could be observed visually. On the basis of this high level of performance, it becomes apparent that convolutional neural networks are highly suitable for automatic diagnosis and detection of plants.

VI. FUTURE SCOPE

The main goal for the future project is to develop a complete system comprising a trained model on the server, as well as an application for mobile phones that display recognized diseases in fruits, vegetables, and other plants based on photographs taken from the phone camera. This application will aid farmers by facilitating the recognition and treatment of plant diseases in a timely manner and help them make informed decisions when utilizing chemical pesticides[6].

Also, future work will involve spreading the use of the model across a wider land area by training it to detect plant diseases on aerial photos from orchards and vineyards captured with drones, in addition to convolution neural networks for object detection.

Drones and other autonomous vehicles, such as smartphones, to be used for real-time monitoring and dynamic disease detection in large-scale open-field cultivations. A future possibility for agronomists working at remote locations could be the development of an automated pesticide prescription system that would require the approval of an automated disease diagnosis system to allow the farmers to purchase appropriate pesticides. Thus, the uncontrolled acquisition of pesticides could be severely restricted, resulting in their excessive use and misuse, with their potentially catastrophic effects on the environment.

REFERENCES

[1] Chohan, Murk & Khan, Adil & Chohan, Rozina & Hassan, Muhammad. (2020). Plant Disease Detection using Deep Learning. International Journal of Recent Technology and Engineering. 9. 909-914. 10.35940/ijrte.A2139.059120.

[2] Mohanty SP, Hughes DP. Using Deep Learning for Image-Based Plant Disease Detection. Front Plant Sci. 2016 Sep 22;7:1419. doi: 10.3389/fpls.2016.01419. PMID: 27713752; PMCID: PMC5032846.

[3] Vijeta Shrivastava, Pushpanjali, Samreen Fatima, Indrajit Das, " Plant leaf diseases detection and classification using machine learning", International Journal of Latest Trends in Engineering and Technology , Vol.10, Issue.2, April 2018.

[4] Ferentinos, Konstantinos. (2018). "Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture." 145. 311-318. 10.1016/j.compag.2018.01.009.

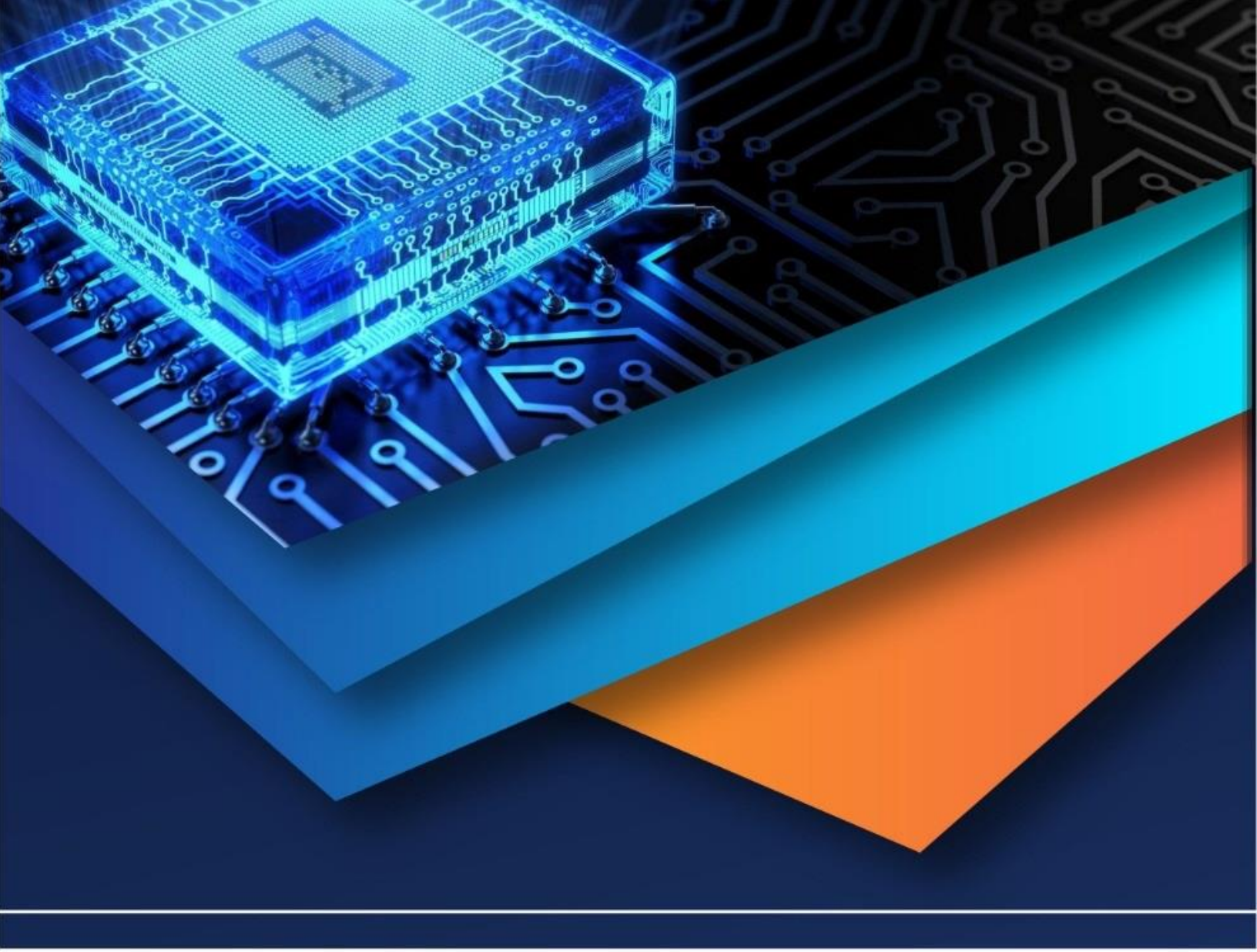
[5] Guo, Yan, et al. "Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming." Discrete Dynamics in Nature and Society, Hindawi, 18 Aug. 2020.

[6] Arsenovic Marko, Karanovic Mirjana, Sladojevic S. "Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection." Symmetry. 2019; 11(7):939.

©IJRASET: All Rights are Reserved

[7] B. Sandika, S. Avil, S. Sanat and P. Srinivasu. "Random forest based classification of diseases in grapes from images captured in uncontrolled environments," 2016 IEEE 13th International Conference on Signal Processing (ICSP), 2016, pp. 1775-1780, doi: 10.1109/ICSP.2016.7878133.
[8] Alagumariappan P, Dewan NJ, Muthukrishnan GN, Raju BKB, Bilal RAA, Sankaran V. Intelligent Plant Disease Identification System Using Machine Learning. Engineering Proceedings. 2020; 2(1):49
[9] Adrian Rosebrock. “Keras ImageDataGenerator and Data Augmentation PyImageSearch.” PyImageSearch, 8 July 2019.
[10] MK Gurucharan. “Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | UpGrad Blog.” UpGrad Blog, 7 Dec. 2020.
[11] Michael A. Nielsen. “Neural Networks and Deep Learning.” Neural Networks and Deep Learning.
[12] DeepAI. “Adam Definition | DeepAI.” DeepAI, DeepAI, 5AD.
[13] Christian Versloot. “Sparse Categorical Crossentropy Loss with TF 2 and Keras – Machine Curve.” Machine Curve, 6 Oct. 2019.
[14] Abhinavsagar. “GitHub – Abhinavsagar/Plant-Disease : Code for the Paper On Using Transfer Learning for Plant Disease Detection.” GitHub
[15] vipool. “Plant Diseases Classification Using AlexNet | Kaggle.” Kaggle: Your Machine Learning and Data Science Community, Kaggle, 29 Nov. 2018.
[16] Lincy, Babitha & Rubia, Jency. (2021). Detection of Plant Leaf Diseases using Recent Progress in Deep Learning-Based Identification Techniques.
[17] Aravind Krishnaswamy Rangarajan, Raja Purushothaman, Aniirudh Ramesh. “Tomato Crop Disease Classification Using Pre-Trained Deep Learning Algorithm - ScienceDirect.” ScienceDirect.Com | Science, Health and Medical Journals, Full Text Articles and Books.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)