DETECTION OF DISEASE IN LEAVES USING DEEP LEARNING

A PROJECT REPORT

Submitted by

Shivang Dogra (18BCE0200)
Tushar Takshak (18BCE0203)
Lakshit Dhanuka (18BCE0217)
Naveen Bellani (18BCE0876)
Himanshu Singh Chauhan (18BCE0885)

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE4020 MACHINE LEARNING
SLOT: F2+TF2

Computer Science and Engineering
JUNE 2021

CONTENTS

TEAM MEMBERS COLLAGE

Naveen Lakshit

Shivang Himanshu Tushar

ABSTRACT

Plants play an essential role in climate change, agriculture industry and a country's economy. Thereby taking care of plants is very crucial. In this paper, a Convolutional Neural Network (CNN) architecture for plant leaf disease detection using techniques of Deep Learning is proposed. Using a public dataset of 54,306 images of diseased and healthy plant leaves collected from “Plant Village Dataset”, we train a deep convolutional neural network to identify 14 crop species and 26 diseases. First of all, augmentation is applied on dataset to increase the sample size. Later Convolution Neural Network (CNN) is used with multiple convolution and pooling layers. The trained model achieves an accuracy of 96.05% on a held-out test set, demonstrating the feasibility of this approach. When testing the model on a set of images collected from trusted online sources - i.e. taken under conditions different from the images used for training - the model achieves an accuracy of 96.1%.

CHAPTERS

CHAPTER 1 – INTRODUCTION

1.1. Introduction

Agriculture is the backbone of economy. Yield loss of more than 50% due to pests and diseases. “At a global scale, pathogens and pests are causing wheat losses of 10 percent to 28 percent, rice losses of 25 percent to 41 percent, maize losses of 20 percent to 41 percent, potato losses of 8 percent to 21 percent, and soybean losses of 11 percent to 32 percent”
-Study, published in the journal Nature, Ecology & Evolution.
Computer vision, and object recognition in particular, has made tremendous advances in the past few years. In 2012, a large, deep convolutional neural network achieved an error of 16.4% for the classification of images into 1000 possible categories. In the following 3 years, various advances in deep convolutional neural networks lowered the error rate to 3.57%. Thus, to combine the power of Deep learning convolutional neural network and real-life expertise to Combat the Problem of loss of yield due to diseases in plants by Automated Detection from Image and Suggestion of Preventive Measures.

Computer vision, and object recognition in particular, has made tremendous advances in the past few years

1.2 Background For the Project Idea

The goal is to detect different plant diseases by processing the leaf picture.
We will be developing a web application that will provide login, forum for discussions and portal to upload the leaf image for detection. The output will tell the disease in the plant by processing the image of leaf.
Product will take input of an image and will process the image and will give disease if present as a output. It will be identifying disease in the leaves of tomato, apple blueberry, cherry, grapes, corn, orange, peach, raspberry, soya bean, squash and strawberry Input leaf image will be classified into 38 categories.

1.3 Problem Statement

1. Plants play an essential role in climate change, agriculture industry and a country's economy. Thereby taking care of plants is very crucial.
2. Identification of the plant diseases is the key to preventing the losses in the yield and quantity of the agricultural product but it is very difficult task in agriculture field. Wrong decision can lead to devastating loss on the production of crop and economical value of market.
3. Farmers try to identify and diagnose the diseases and monitor the health of plants with their own knowledge and experience. Their naked eyes can't observe big plantation area and it can be expensive too.
4. Leaf disease detection demands huge amount of work, knowledge in the plant diseases, and also require the more processing time. Identification includes finding visually observable patterns seen on the plant Existing models are used for limited crops.
5. If the scope of project is big then accuracy is less.
6. These kinds of software don't have user friendly interface.

1.4 Objective

In light of the literature survey conducted the following solutions are implemented:
• To reduce the error percentage and enhance the time complexity, using a better deep

learning model, with enhanced layers of neurons and a better algorithm.
• To develop an application with a model to recognize the leaves even with background
disturbance, using noise removal techniques.
• To use hyperspectral image generation (increase color band) in order to achieve better
clarity images thus reducing the number of images to be fed into the model.
• To increase the usability of the mobile application by making it more user friendly, for it to
be accessible by people with all backgrounds.

CHAPTER 2 - LITERATURE REVIEW

2.1 Using Deep Learning for Image-Based Plant Disease Detection

By
Sharada P. Mohanty David P. Hughes Marcel Salathé

2.1.1 Abstract

One of the major threats to the food security is crop diseases Their identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. In this paper a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions and trained a deep convolutional neural network to identify 14 crop species and 26 diseases. The trained model has an accuracy of 99.35%

2.1.2. Dataset

Dataset contains 54,306 images of plant leaves, which 38 class labels. Images were resized to 256 × 256 pixels using python. Three kind of images were used that is colored, gray-scaled version and segmented where all the background as been removed.

2.1.3 Algorithm

1. Two learning architecture were used that are
    - Alex Net,
    - Google Net.
2. Two training mechanism were used
    - Transfer Learning,
    - Training from Scratch.
3. Three type of dataset was used:
    - Color,
    - Gray scale,

Leaf Segmented.

4. Various combination of training-testing set were taken:

Train: 80%, Test: 20%,
Train: 60%, Test: 40%,
Train: 50%, Test: 50%,
Train: 40%, Test: 60%,
Train: 20%, Test: 80%.

Google Net architecture was trained using transfer learning on the gray-scaled images dataset was split into a train—test 60–40 ratio and for Google Net Transfer Learning was used with testing training ratio of 60–40 on grey scale images.

30 epochs were run for each experiment where one epoch is defined as the number of training iterations in which the particular neural network has completed a full pass of the whole training set.

Hyper-parameters were standardized across all the experiments using following methods

- Solver type: Stochastic Gradient Descent,
- Base learning rate: 0.005,
- Learning rate policy: Step (decreases by a factor of 10 every 30/3 epochs),
- Momentum: 0.9,
- Weight decay: 0.0005,
- Gamma: 0.1,
- Batch size: 24 (in case of Google Net), 100 (in case of Alex Net)

2.1.4 Result

Random guessing would have only achieve an overall accuracy of 2.63% on average whereas this model has an overall accuracy varying from 85.53% in case of alex net (training from scratch gray scale (80–20)) to 99.34% in case of google net (transfer learning color (80–20))

2.2.Deep neural networks based recognition of plant diseases by leaf image classification

By srdjan sladojevic
Marko arsenovic
Andras anderla
Dubravko culibrk

2.2.1 Abstract

Classification of healthy and infected leaves are done using of deep convolutional networks. Model identifies 13 types of plant diseases out of healthy leaves, with the ability to differetiate plant leaves from their surroundings. Deep learning framework used was caffe framework accuracy obtained were between 91% and 98%, for separate class tests and average accuracy obtained was 96.3%.

2.2.2 Dataset

Images were downloaded from internet from various sources and were stored together into fifteen different classes.

All duplicated images taken from different sources were deleted by using python script applying opencv the comparing procedure.
Image augmentation was performed. 30880 images for training and 2589 images for validation was used.

2.2.3 Image preprocessing and labelling

Preprocessing on images was done which include cropping of all the images, making the square around the leaves, in order to highlight the region of interest (plant leaves).

2.2.4 Augmentation process

Image augmentation was done using several transformation techniques including affine transformation, perspective transformation, and simple image rotations. Affine transformations were applied to express translations and rotations where all parallel lines in the original image are still parallel in the output image.

2.2.5 Neural network training

Caffe, an open-source deep learning framework used, along with the set of weights learned on a very large dataset, imagenet.

Caffe framework is basically used for research experiments and industry deployment. It is written in c++ and provides command line, python, and matlab interfaces.

Caffe net architecture is considered a starting point, but modified and adjusted to support our 15 categories (classes).

Since the pre trained model contain different number of output classes as they were orignally trained on image net data set but here author needs only 15 types of classes so last layer of the café model was altered.

Another important layer of cnns is the pooling layer, which is a form of nonlinear down sampling. Pooling operation gives the form of translation invariance it operates independently on every depth slice of the input and resizes it spatially. To reduce the overfitting overlapping pooling is applied. Furthermore, to reduce overfitting a dropout layer is used in the first two fully connected layers. The disadvantage of dropout is that it slow down training time 2-3 times comparing to a traditional neural network of the exact architecture

2.2.6 Testing

10-fold cross validation technique is used for the accuracy test. The cross-validation procedure was repeated after every thousand training iteration.

2.2.7 Fine tuning

Fine-tuning seeks to increase the accuracy by making small changes in the hyper parameters.

2.2.8 Result

After the 100th training iteration an accuracy of 95.8% was obtained while after fine-tuning an accuracy of 96.3% was obtained.

2.3 Plant disease detection using machine learning

by

Vinod p v
Shima ramesh maniyath
Niveditha m

Abstract

Random forest is used to identify healthy and diseased leaf. Various process include in this paper are dataset creation, feature extraction, training the classifier and classification. Histogram of an oriented gradient is used for extracting features of an image, we use (hog).

Proposed methodology

Steps followed to implement the model are
Preprocessing,
Feature extraction,
Training of classifier and classification

To a reduced the image to uniform size preprocessing of image is performed.
Features extraction on a preprocessed image are done with the help of hog which is a feature descriptor used for object detection. Feature descriptor use intensity gradients to tell the appearance and the outline of the image.

Three feature descriptors are used that are
Hu moments
Hu moments tells the outline of a leaf and are calculated over single channel only. Images are converted from rgb to gray scale and then the hu moments are calculated. This step gives an array of shape descriptors and image moments

Haralick texture
Healthy leaves and diseased leaves texture varies from each other. Haralick texture feature differentiate between the healthy and diseased leaf. It is based on the adjacency matrix which stores the position of each pixel. Frequency of the pixel (i,j) occupying the position next to pixel is used to calculate the texture. Before that image need to be converted into gray scale

Color histogram
Color histogram tell about the colors in the image. Rgb is first converted to hsv color space and the histogram is calculated for it. Leaf is converted from rgb to hsv because hsv model optimum similarity with how human eyes perceive the colors. Histogram plot tells the description about the number of pixels available in the given color ranges

Algorithm description
Model is implemented by random forests classifier.
The labeled datasets are splited into training and testing data. The feature vector is produced for the training dataset using hog feature extraction. The developed feature vector is trained under a random forest classifier. The feature vector for the testing data created through hog feature extraction is passed to the trained classifier for prediction. Labeled training datasets are transformed into their corresponding feature vectors by hog feature extraction. These extracted feature vectors are stored under the training datasets. Random forest classifier is used to trained feature vectors. The feature vectors are extracted for the test image by using hog feature extraction. These generated feature vectors are given to be stored in dataset and trained classifier for predicting the results.

Result
First image is converted from rgb image into gray scale image because hu moments shape descriptor and haralick features can only be calculated over a single channel only. To calculate histogram the image first be converted to hsv (hue, saturation and value). And then histogram is obtained. Accuracy obtained in the final histogram is 70.15%.

2.4 Detection of plant leaf diseases using image segmentation and soft computing techniques

By

Vijai singh
A.k. mishra

Methodology

1. Images of leaves for dataset are taken by capturing the images from camera.

2. Then following steps and algorithm are applied to implement the model.

3. Image acquisition is the initial step that require capturing an image with a digital camera.

4. Preprocessing of input image to enhance the quality of image and to remove the unwanted distortion from the image is done. Clipping of the leaf image is done to get the interested image region and then image smoothing is performed on the leaf image using the smoothing filter. To enhance the contrast image enhancement is also done.
5. Then thresholding is applied to green color of the image that is the leaf part. A threshold value is set and if the intensity of the pixel is less than threshold than zero intensity is applied

6. Mask is removed in the infected part of the leaf

7. Then important segments are obtained to classify the leaf diseases. Segment the components using genetic algorithm

The search capability of gas is used, to set of unlabeled points in n-dimension into k clusters. Similarly, it has been applied to these images. A color image of size $m \times n$ has been with 3 color channels (rgb) . Every chromosome represents a solution, which is a sequence of k cluster centers. Population is initialized in various rounds randomly which from existing chromosome best chromosome survives in each round for the next round processing.

In the first step of fitness computation the points of pixel is clustered according to nearest corresponding cluster centers such that each pixel x_i of color image is put into the respective cluster with cluster center z_j for $j = 1, 2, \ldots, k$ by the following given formula

$$\text{If} \quad ||x_i - z_j|| < ||x_i - z_l||,$$

$$i = 1,2,\ldots,m \times n, \quad l = 1,2,\ldots,K, \text{ and } p \neq j.$$

Then new cluster centers are formed by calculating the mean of each pixel of the assigned clusters. The new center of cluster has a formula of

$$Z_i(r,g,b) = \frac{1}{n_i} \sum_{xj \in c_i} (x_j(r,g,b)) \quad i = 1,2,\ldots.k$$

Then fitness function is calculated by calculating euclidean distance between the pixels and their corresponding cluster by

$$M = \sum M_i$$

$$M_i = \sum_{x_j \in c_i} |(x_j(r,g,b) - z_i(r,g,b))|$$

8. Computing the features using color co-occurrence methodology

   Co-occurrence method for feature extraction is used. In this method in both the texture and color of an image are assumed to come to the unique features, which represents that image.

   The three major mathematical processes in the color co-occurrence method. First is conversion of the rgb images of leaves into his color space representation. After this pixel map is used, which results into three color co-occurrence matrices, one for each of h, s, i.

   Features known as texture features, which include local homogeneity, contrast, cluster shade, energy, and cluster prominence are computed for the h image

$$\mathrm{CONTRAST} = \sum_{i,j=0}^{N-1} (i,j)^2 C(i,j) \tag{4}$$

$$\mathrm{Energy} = \sum_{i,j=0}^{N-1} C(i.j)^2 \tag{5}$$

$$\mathrm{Local\ Homogeneity} = \sum_{I,j=0}^{N-1} C(i,j)/(1+(i-j)^2) \tag{6}$$

$$\mathrm{Entropy} = -\sum_{i,j=0}^{N-1} C(i.j) \log \mathrm{C}(i.j) \tag{7}$$

9. Classification of disease

The measure of success of classification is performed by using the classification gain and following

Gain (%) = number of correct classification/total no of test images∗100

Results

Model is implemented in matlab. Model is implemented for leaves like rose with bacterial disease, beans leaf with bacterial disease, lemon leaf with sun burn disease, banana leaf with early scorch disease and fungal disease in beans leaf are considered.

2.5 Plant disease identification based on deep learning algorithm in smart farming

By

Yan guo
Jin zhang
Chengxin yin

Abstract

To detect and recognize disease based on deep learning a mathematical model has been developed using region proposal network (rpn) identify and find the leaves in complex surroundings. Then images segmentation is performed on the basis of the output of rpn algorithm which contain the feature of symptoms through chan–vese (cv) algorithm. At last, the segmented leaves are input to the transfer learning model and trained by the dataset of diseased leaves under simple background. The model is tested with black rot, bacterial plaque, and rust diseases and has an accuracy of 83.57%,

Proposed model

It has mainly three steps.

1. Locating the diseased leaves.
   The algorithm used for training the leaf dataset in the complex environment is rpn, and algorithms used for locating and retrieving the diseased leaves in the complex environment are the frame regression neural network and classification neural network
2. Segmentation of diseased leaves.
   The chan–vese algorithm is used for segmenting the image of diseased leaves. On the basis of set zero level set and the minimum energy function as the goal, leaf's contours are developed by calculating, to find the image segmentation of diseased leaves in that is the part which is infected by disease in the complex environment.
3. Identification of leaf disease species.
   The technique used for training is pretrain transfer learning. It is trained to find plant diseases recognition in the plain back -ground (white or black).

Result

The average accuracy rate of the proposed method obtained is 83.75%,

---

2.6 Plant disease prediction using machine learning algorithms

Authors:

G.prem rishi kranth ,m. Hema lalitha, laharika basava, anjali mathur

Abstract :

The paper suggests the detection of diseases and pest infestation in plants suing machine learning, based on parameters and factors like : dryness , moisture, powdery mildew , galls .they used k-means, decision tree, naïve bayes, neural network prediction algorithms . The results obtained conveyed that decision trees are the best to predict the infestation in plants with an accuracy of 89.93 %. The naive bayes had an accuracy of 83 % and decision trees gave an accuracy of 89.97%. The main pitfall in the paper is that it is very difficult to use decision trees for image classification , rather neural network are preferred in order to handle images , apart from this the parameters or features chosen are also not accurate due to unpredictable/non-static weather conditions.

2.7 Neural networks in plant disease detection using hyperspectral data

Authors:

Kamleshgolhani, siva k. Balasundram , ganesan vadamalai, biswajeet pradhan

Abstract:

This paper suggests the usage of hyperspectral images in cnn to increase the detection accuracy. Hyperspectral images have a broad amount of spectral than the normal images thus allowing the neural networks to solve the problems of image preprocessing , and increasing accuracy . The results were very accurate as they used the concept of sdi's (spectral disease indices) in order to analyze the data. The main pitfall over here is the hughes problem ,or the curse of dimensionality problem, which states that beyond a certain amount of dimensionality or training sample the predictive performance starts deteriorating rather than increasing , therefore this needs to be avoided using the optimal dimensionality.

2.8 Agricultural plant leaf disease detection and diagnosis using image processing based on morphological feature extraction

Authors:

Mr. Sachin b. Jagtap, mr. Shailesh m hambarde

Abstract:

This paper suggests the spots or patches formed on the leaves to be considered as diseases/insect trails, with the help of which the leaves can further be classified into infested, diseased or healthy leaves .they suggested the use of image processing to identify the leaf patches and the disease type. To accomplish this task the images were processed in four stages: image enhancement , image segmentation, feature extraction and classification. The results obtained were accurate though there was only a classification between infested and non infested leaves , the results are of very less use as the model was not successful in detecting the type of insect or the type of disease the plant is infected with . Unless there is any information about the type of insect/disease the specialized treatment cannot be given to the plants , apart from this this, there is also a need for spatial care in the model, and for accurate results many images need to be taken thereby making the application not very user friendly.

2.9 A smartphone image processing application for plant disease diagnosis.

Author :

Nikos petrellis

Abstract

In this paper the author used color,area and the number of spots in order to find out the type of disease or infestation .the algorithm used in the image processing ,proceses the image and extracts a few features like: amount of spots and gray area in the image, finally a histogram is drawn using the images , which indicates the specific rgb color percentages , and then the places/pixels with high color concentration levels are considered as spots and then they are examined for the specific disease detection . The author has also proposed an application for detection and examination of the diseased areas . Though the accuracy of this proposal is around 90% ,the threshold accuracy comes around 70 %,a apart from this, the mobile application is not very farmer friendly, and cannot be used by local people living in rural areas . Therefore it requires a bit jargon simplification for a customer friendly experience.

Title :

2.10 Deep learning based mobile application for plant disease diagnosis.

Authors :

Shraddha verma , anuradha chug , amit prakash singh , shubham sharma

Abstract:

This research paper suggest the pathogen outbreak reduction, in early stages of infection , which is quite impressive as it is important to eliminate the infestation at early stages so that it does not spread to different parts of the plants .they used the general regression neural network , back propagation neural network , regularization and support vector machine algorithms to train the neural networks which helped in increasing the accuracy level. The final accuracy as mentioned in the paper was 99% which is the required amount of accuracy for an application to be deployed . Flask api was used to create the application ,which is very developer friendly .the only pitfall observed in this was that they used only three environment variables : temperature, humidity and rainfall , and too many images were used without proper preprocessing ,which could be avoided using the techniques mentioned in the above papers .

CHAPTER 3 – PROBLEM FORMULATION

3.1 Methodology

Preprocessing

Dataset

Dataset contains 54,306 images of plant leaves, which 38 class labels. The dataset is split into two parts, training and validation.

Data augmentation

For the training, following transformations were applied to the dataset

Random Rotation
Image is rotated horizontally
Random Resized Crop
This will extract a patch of size (224, 224) from your input image randomly. So, it might pick this path from top left, bottom right or anywhere in between.

Random Horizontal Flip - Image is flipped horizontally

Resize - Input image is resized to be of size (256, 256)

Data normalization

Input images has been normalize using the formula for the all the three channels RGB

image = (image - mean) / std

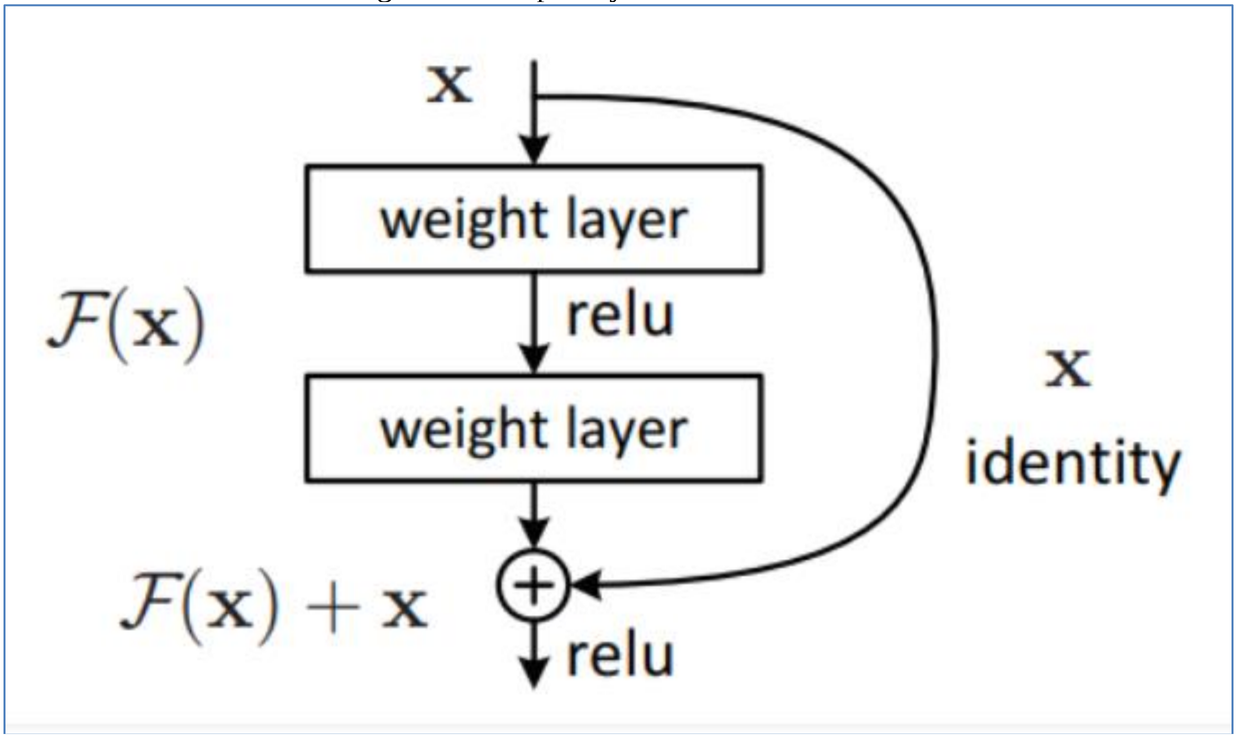
For the means, it's ' [0.485, 0.456, 0.406]' and for the standard deviations "[0.229, 0.224, 0.225]' These values will shift each color channel to be centered at 0 and range from -1 to 1.

MODEL – RESNET 152

Model used is resnet-152 for training.
In this network we use a technique called skip connections . The skip connection skips training from a few layers and connects directly to the output.

The approach behind this network is instead of layers learn the underlying mapping, we allow network fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit,

F(x) := H(x) – x which gives *H(x) := F(x) + x*.
ResNet-152 Pre-trained Model for PyTorch with a depth of up to 152 layers---8x deeper than VGG nets but still having lower complexity.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

ResNet Architectures

Each ResNet block is either 2 layer deep (Used in small networks like ResNet 18, 34) or 3 layer deep (ResNet 50, 101, 152).

ResNet-152, both have about 60M parameters so we Freeze parameters so we don't backprop through them.
We Define a new, untrained feed-forward network as a classifier, using ReLU activations

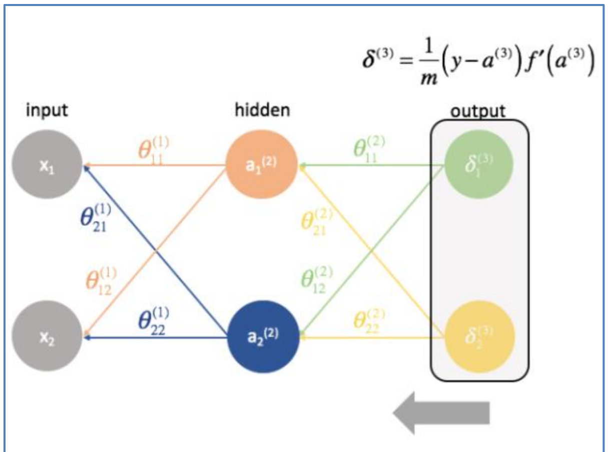
We create classifier ordered dictionary first and since the pretranined model has 1000 classes in the final layer and we need only 38 classes, we replace the final layer of the pretrained model with our untrained feed-forward network

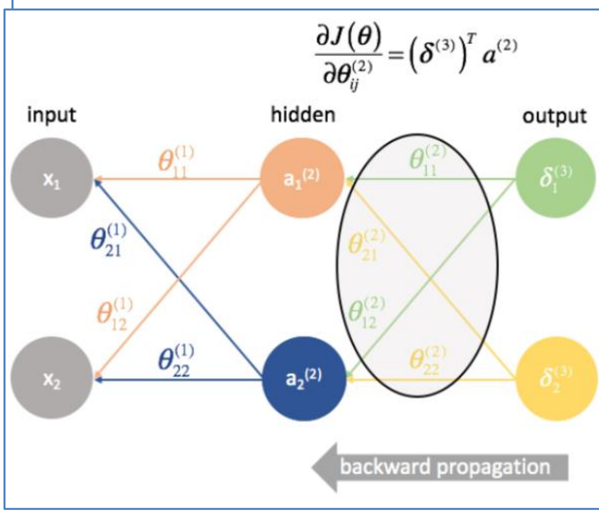
Training - backpropagation

In order to calculate the partial derivative of cost function with respect to parameters, this method is used.

Basically, we reuse the set of calculated partial derivatives as we go layer by layer in the backwards direction, in the network of layers.

In order to understand this better let us observe the following diagrams as an example:





Using the below formula, we propagate the layers backwards.

l = 1,2,3,4…L

1) $\boxed{\delta(L)=1m(y-a(L))f'(a(L))}$ - while finding the partial derivative for the layers

2) $\boxed{\partial J(\theta)/\partial\theta(L-1)ij=(\delta(L))Ta(L-1)}$ for finding the cost function with respect to parameters that are fed into the output layer
3) For the hidden layer we use: $\boxed{\delta(l)=\delta(l+1)\Theta(l)f'(a(l)),\ \text{here } l=l-1}$ because we move one layer back
4) Finally, we calculate the cost function of current layer and repeat the steps above for each layer backwards.

Testing

Output is compared with the actual answer and if it is correct 1 is added to the score and if it is wrong 0 is added to the answer.

Mean of all the scores divided by total number of images is taken as final score

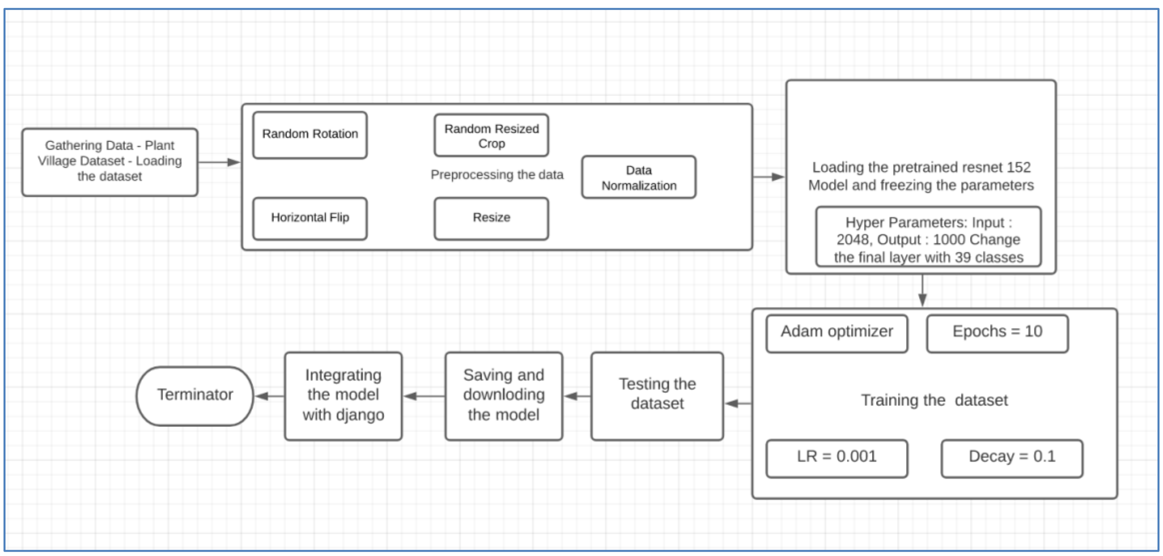
Accuracy score = s/n

Where,

s = sum of correct answer during testing

N = number of images taken for testing

3.2 ARCHITECTURE



CHAPTER 4 – RESULT AND DISCUSSION

4.1 DATASET DESCRIPTION

This dataset contains an open access repository of images on plant health to enable the development of mobile disease diagnostics. The dataset contains 54, 309 images. The images span 14 crop species: Apple, Blueberry, Cherry, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato. It contains images of 17 fundal diseases, 4 bacterial diseases, 2 mold (oomycete) diseases, 2 viral diseases, and 1 disease caused by a mite. 12 crop species also have images of healthy leaves that are not visibly affected by a disease.

The PlantVillage dataset is the only public dataset for plant disease detection. The data set curators created an automated system using GoogleNet and AlexNet for disease detection. The images in PlantVillage dataset are taken in laboratory setups and not in the real conditions of cultivation fields.

The PlantVillage dataset contains images taken under controlled settings. This dataset limits the effectiveness of detecting diseases because, in reality, plant images may contain multiple leaves with different types of background conditions with varying lighting conditions.

We analyze 54,306 images of plant leaves, which have a spread of 38 class labels assigned to them. Each class label is a crop-disease pair, and we make an attempt to predict the crop-disease pair given just the image of the plant leaf. We resize the images to 256 x 256 pixels, and we perform both the model optimization and predictions on these downscaled images. Segmentation was automated by the means of a script tuned to perform well on our particular dataset.

It will be identifying disease in the leaves of tomato, apple blueberry, cherry, grapes, corn, orange, peach, raspberry, soya bean, squash and strawberry
Input leaf image will be classified into 38 categories.

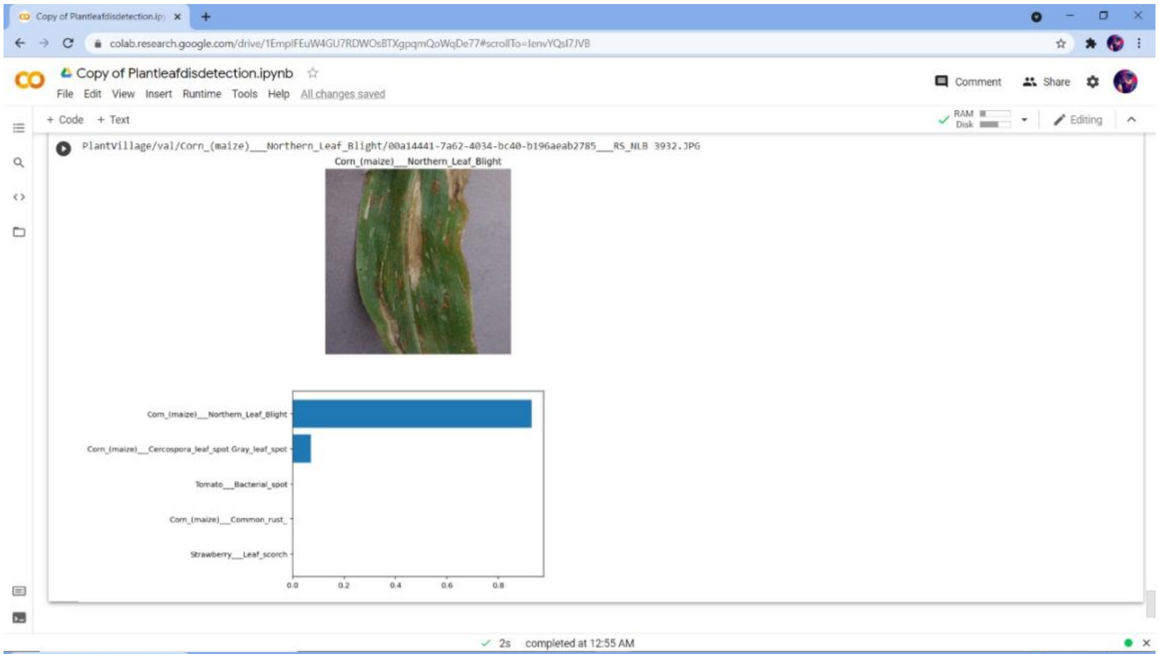
The Plant Village dataset contains 38 different plant disease classes and one background class from Stanford's open dataset of background images.

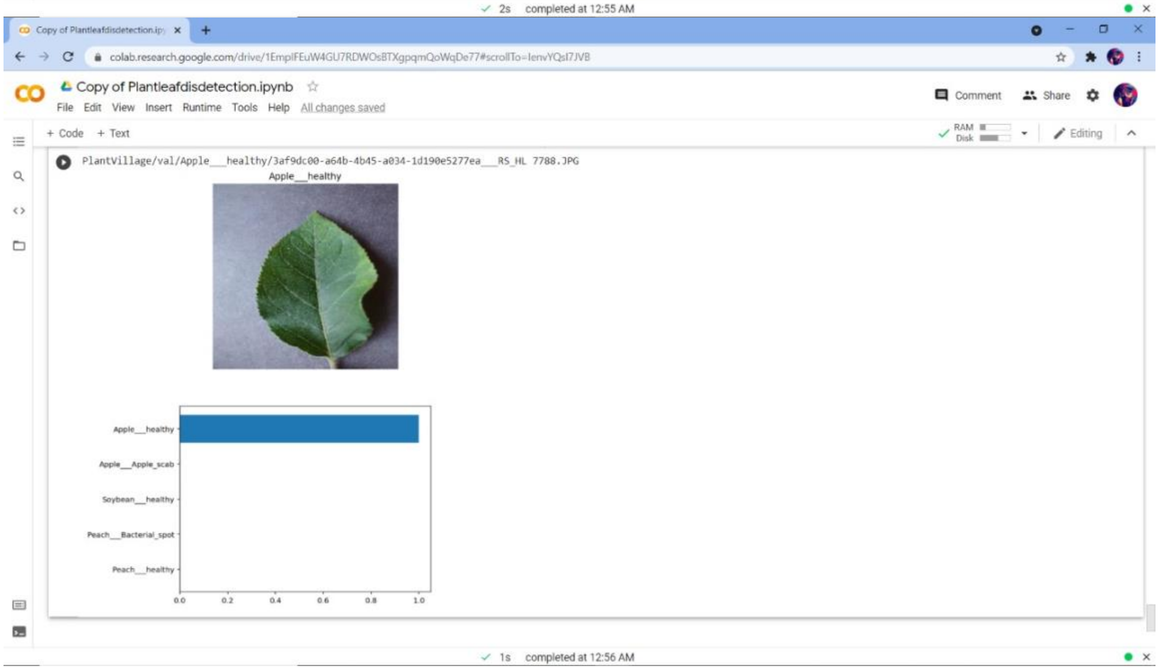
The classes are:

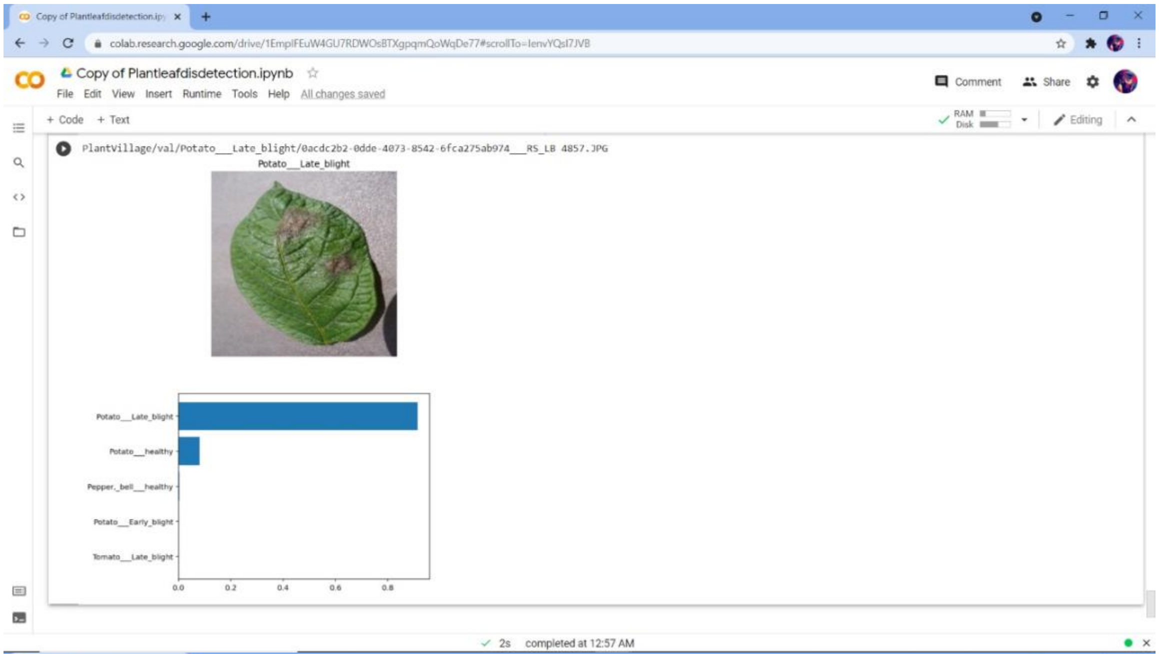
- Apple___Apple_scab
- Apple___Black_rot
- Apple___Cedar_apple_rust
- Apple___healthy

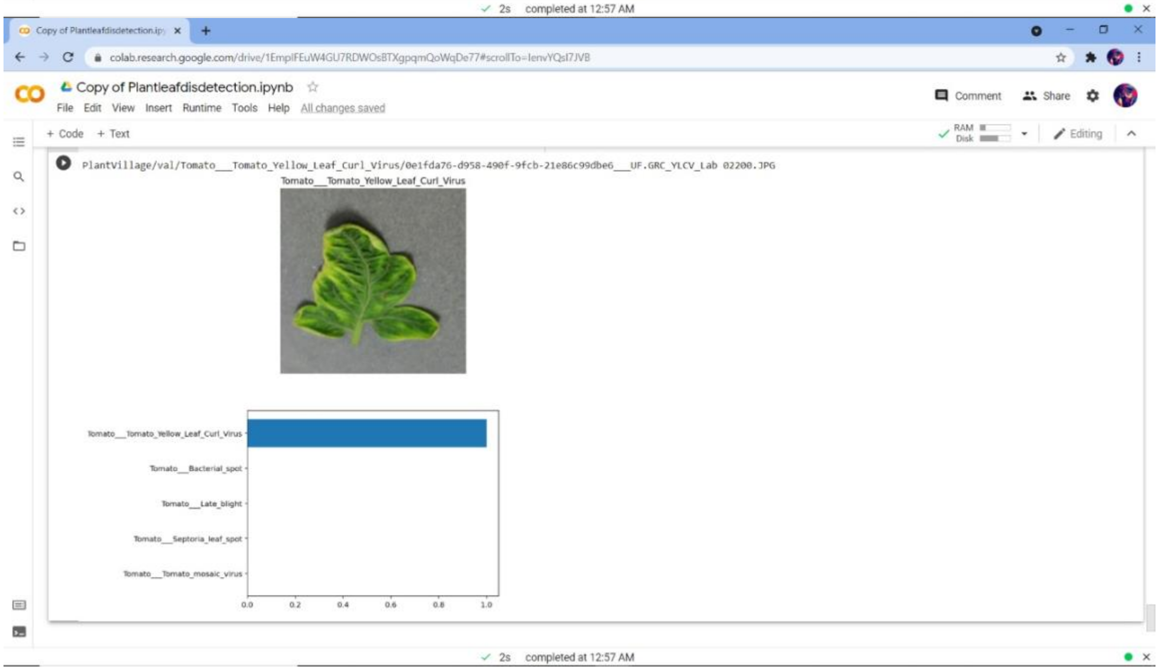
- Blueberry___healthy
- Cherry_(including_sour)___Powdery_mildew
- Cherry_(including_sour)___healthy
- Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
- Corn_(maize)__*Common_rust*
- Corn_(maize)___Northern_Leaf_Blight
- Corn_(maize)___healthy
- Grape___Black_rot
- Grape___Esca_(Black_Measles)
- Grape___Leaf_blight_(Isariopsis_Leaf_Spot)
- Grape___healthy
- Orange___Haunglongbing_(Citrus_greening)
- Peach___Bacterial_spot
- Peach___healthy
- Pepper,_bell___Bacterial_spot
- Pepper,_bell___healthy
- Potato___Early_blight
- Potato___Late_blight
- Potato___healthy
- Raspberry___healthy
- Soybean___healthy
- Squash___Powdery_mildew
- Strawberry___Leaf_scorch
- Strawberry___healthy
- Tomato___Bacterial_spot
- Tomato___Early_blight
- Tomato___Late_blight

4.2 RESULT SCREENSHOTS






Copy of Plantleafdisdetection.ipynb
colab.research.google.com/drive/1EmpIFEuW4GU7RDWOsBTXgpqmQoWqDe77#scrollTo=lenvYQsI7JVB
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Editing
PlantVillage/val/Potato___Late_blight/0acdc2b2-0dde-4073-8542-6fca275ab974___RS_LB 4857.JPG
Potato___Late_blight
Potato___Late_blight
Potato___healthy
Pepper,_bell___healthy
Potato___Early_blight
Tomato___Late_blight
0.0
0.2
0.4
0.6
0.8
2s completed at 12:57 AM


Copy of Plantleafdisdetection.ipynb
colab.research.google.com/drive/1EmpIFEuW4GU7RDWOsBTXgpqmQoWqDe77#scrollTo=lenvYQsI7JVB
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
Editing
PlantVillage/val/Tomato___Tomato_Yellow_Leaf_Curl_Virus/0e1fda76-d958-490f-9fcb-21e86c99dbe6___UF.GRC_YLCV_Lab 02200.JPG
Tomato___Tomato_Yellow_Leaf_Curl_Virus
Tomato___Tomato_Yellow_Leaf_Curl_Virus
Tomato___Bacterial_spot
Tomato___Late_blight
Tomato___Septoria_leaf_spot
Tomato___Tomato_mosaic_virus
0.0
0.2
0.4
0.6
0.8
1.0
2s completed at 12:57 AM

4.3 RESULT ANALYSIS

Why we got 96.1% accuracy?

Since our dataset was similar to ImageNet dataset, it was more efficient for us to use a pretrained model. The model we choose is Residential network. Basically, Deeper neural networks are more difficult to train. This network is deeper than other models available. evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth.

We obtained the hyper-parameters of Resnet- *50 layers*. After that we select the same but more layer's net that is *ResNet-152 layers* to increase the accuracy.

LOGICAL ANALYSIS

We compare our deep learning model with another project based on Citrus Plant Disease Identification using Deep Learning with Multiple Transfer Learning Approaches

DESCRIPTION

- The dataset used for logical analysis comprised of a small number of citrus plant leaves divided into five categories; 4 of them are disease affected, and the fifth category includes healthy leaves.
- Inception is used as a deep learning model to train the dataset.
- First, the model is prepared without a transfer-learning approach and then the model is pre-trained on external data of plant leaves, ImageNet dataset, and a combination of external data and ImageNet data.
- Model without transfer learning failed to identify the diseases.
- Model pre-trained on external data resulted in an accuracy of 92%, AUC score of 98.8%, and F1-score of 95%.
- The model with combined pretraining resulted in an accuracy of 88% and F1-score of 88%.
- Pre-training on ImageNet data resulted in an accuracy of 82% and F1-score of 87%.

We trained our model on this dataset and got an accuracy of 95.8%.

REASON

- The main base element of ResNet is the residual block. As we go deeper into the network with a large number of layers, computation becomes more complex. These layers put on top of each other and every layer try to learn some underlying mapping of the desired function and instead of having these blocks, this make this model better than inception and we get high accuracy

CHAPTER 5 – CONCLUSION AND FUTURE WORK

People are not aware of how to take care of their fields. There are old methodologies in agriculture.
Identifying plant disease wrongly leads to huge loss of yield, time, money and quality of product. Identifying the condition of plant plays an important role for successful cultivation.
In olden days identification is done manually by the experienced people but due to the so many environmental changes the prediction is becoming tough. So, we can use image processing techniques for identification of plant disease. Generally, we can observe the symptoms of disease on leaf's, stems, flowers etc. so here we use leafs for identification of disease affected plants.

There are many methods in automated or computer vision plant disease detection and classification process, but still, this research field is lacking. In addition, there are still no commercial solutions on the market, except those dealing with plant species recognition based on the leaves images.
Using the deep convolutional neural network architecture, we trained a model on images of plant leaves with the goal of classifying both crop species and the presence and identity of disease on images that the model had not seen before. Within the PlantVillage data set of 54,306 images containing 38 classes of 14 crop species and 26 diseases, this goal has been achieved as demonstrated by the top accuracy of 96.1%. Thus, without any feature engineering, the model correctly classifies crop and disease from 38 possible classes in 961 out of 1000 images.

In this paper, a new approach of using deep learning method was explored in order to automatically classify and detect plant diseases from leaf images. The developed model was able to detect leaf presence and distinguish between healthy leaves and 23 different diseases, which can be visually diagnosed. The complete procedure was described, respectively, from collecting the images used for training and validation to image preprocessing and augmentation and finally the procedure of training the deep CNN and fine-tuning. Different tests were performed in order to check the performance of newly created model.

The final overall accuracy of the trained model was 96.3%. Fine-tuning has not shown significant changes in the overall accuracy, but augmentation process had greater influence to achieve respectable results.

As the presented method has not been exploited, as far as we know, in the field of plant disease recognition, there was no comparison with related results, using the exact technique. In comparison with other techniques used and presented in Section 2, comparable or even better results were achieved, especially when taking into account the wider number of classes in the presented study.

APPENDIX

- **Link To Dataset**

  https://drive.google.com/uc?id=18DbC6Xj4NP-hLzI14WuMaAEyq482vNfn

- **Link To GitHub**

  https://github.com/trixtun/DETECTION-OF-DISEASE-IN-LEAVES-USING-DEEP-LEARNING

INDIVIDUAL CONTRIBUTION

1. **Shivang Dogra**
   - **Understand all the libraries and implementing them**
   - **Understanding deep learning model that we used called resnet 152 containing 152 layers as we have used already a trained model**
2. **Naveen Bellani**
   - **Training the dataset with num_epochs =20**
3. **Lakshit Dhanuka**
   - **Test set validation using torch library**
   - **Saving the trained model on the drive**
4. **Tushar Takshak**
   - **Task to update the earlier model with existing trained model**
5. **Himanshu Singh Chauhan**
   - **Predicting the Outcomes from the test data**

REFERENCES

- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.* 110, 346–359. doi: 10.1016/j.cviu.2007.09.014

- Chéné, Y., Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., et al. (2012). On the use of depth camera for 3d phenotyping of entire plants. *Comput. Electron. Agric.* 82, 122–127. doi: 10.1016/j.compag.2011.12.007

- Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. (IEEE) (Washington, DC).

- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei L. (2009). "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (IEEE).

- Ehler, L. E. (2006). Integrated pest management (ipm): definition, historical development and implementation, and the other ipm. *Pest Manag. Sci.* 62, 787–789. doi: 10.1002/ps.1247

- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* 88, 303–338. doi: 10.1007/s11263-009-0275-4

- K. A. Garrett, S. P. Dendy, E. E. Frank, M. N. Rouse, and S. E. Travers, "Climate change effects on plant disease: genomes to ecosystems," *Annual Review of Phytopathology*, vol. 44, pp. 489–509, 2006.

- A. J. Tatem, D. J. Rogers, and S. I. Hay, "Global transport networks and infectious disease spread," *Advances in Parasitology*, vol. 62, pp. 293–343, 2006.