

VEGETABLE DISEASES DETECTION USING MACHINE LEARNING

BY

SUMON BHUIYA
ID: 181-15-10601

MD. MAHMUDUL HASAN
ID: 181-15-10930

FAIUS MOJUMDAR NAHIN
ID: 181-15-10965

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Mr. Riazur Rahman
Sr. Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Mr. Md. Mahade Hasan
Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

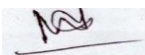
DHAKA, BANGLADESH

JANUARY 2022

APPROVAL

This Project titled “**Vegetable Diseases Detection Using Machine Learning**” submitted by ***Sumon Bhuiya ID: 181-15-10601***, ***Md. Mahmudul Hasan ID: 181-15-10930***, ***Faius Mojumdar Nahin ID: 181-15-10965*** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 4th January 2022.

BOARD OF EXAMINERS

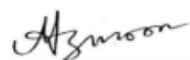


Chairman

Dr. Md. Ismail Jabiullah

Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Internal Examiner

Nazmun Nessa Moon (NNM)

Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Internal Examiner

Aniruddha Rakshit (AR)

Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



External Examiner

Dr. Md Arshad Ali

Associate Professor


Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology
University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Mr. Riazur Rahman, Sr. Lecturer, Department of CSE** Daffodil International University.

We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diplo

Supervised by:



Mr. Riazur Rahman
Sr. Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:



Mr. Md. Mahade Hasan
Lecturer
Department of CSE
Daffodil International University

Submitted by:




Sumon Bhuiya

ID: -181-15-10601

Department of CSE

Daffodil International University



Md. Mahmudul Hasan

ID: -181-15-10930

Department of CSE

Daffodil International University



Faius Mojumdar Nahin

ID: -181-15-10965

Department of CSE

Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Mr. Riazur Rahman, Sr. Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Md. Ismail Jabiullah, Professor, and Head**, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Our country cultivates to a great extent of leafy vegetable. Meanwhile a lot of amount goes to wastage responsible being the lack of care. This happens due to incapability in peaking up the right illness of plants, making farmers huge loss. As a result, farmers are turning their back from planting vegetables. To solve this calamity, from our by heart efforts we are presenting vegetable diseases detection using machine learning algorithm. This machine can ease difficulties of detecting incurable vegetables with its features. Modern medical terms cannot upgrade its facilities in no alternate means of using machine learning algorithm. Through this algorithmic way we can easily detect the diseases. Medical procedure is a long process to know health report. In the meantime, losing a lot of time does make customer dissatisfied. To save time and for customer's satisfaction this machine learning algorithm demand is at its rise. Our main focus in this research was to detect diseases of vegetations easily. Our country is in lot of demand of vegetables. For any reason there comes any failure to detect diseases, all harvest will go rotten. Late detection of this diseases will not lead to any solution. The initiative of this research was taken to detect diseases fast. Our little trial is for those farmers, to not let their harvest go crumble and so to detect diseases fast.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii-iii
Declaration	iv-v
Acknowledgements	vi
Abstract	vii
CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1-2
1.2 Motivation	2
1.3 Objective	2
1.4 Our Uniqueness	2
1.5 Research Questions	3
1.6 Expected Outcome	3
1.7 Report Layout	3
CHAPTER 2: BACKGROUND	4-7
2.1 Introduction	4
2.2 Related Works	4-6
2.3 Research summary	6
2.4 Scope of the problem	7
2.5 Challenges	7

CHAPTER 3: RESEARCH METHODOLOGY	13-20
3.1 Research Subject and Instrumentation	13
3.2 Data Collection & Pre-Processing	14-16
3.3 Proposed Methodology	16-19
3.4 Implementation Requirements	20
CHAPTER 4: EXPERIMENT RESULTS & DISCUSSION	21-28
4.1 Introduction	21
4.2 Data Segmentation	21-24
4.3 Data Preprocessing	24
4.4 Input File Creation	24
4.5 Model Building & Dataset Fitting	24-25
4.6 Experiment Accuracy Loss and Output	26-8
4.7 Summary	28
CHAPTER 5: SUMMARY, RECOMMENDATION & CONCLUSION	29
5.1 Research Summary	29
5.3 Conclusion	29
Reference	30-31

LIST OF FIGURES

FIGURES	PAGE NO
Fig 2.1: Model Train	5
Fig 2.2: Test Model	5
Fig 2.3: Data Loss	6
Fig 3.1 Methodology flow diagram	8
Fig 3.2: sample use of opencv & numpy	10
Fig 3.3: sample of tqdm	10
Fig 3.4: sample output of csv	11
Fig 3.5: tensorflow architecture	12
Fig 3.6: keras model	13
Fig 3.7: mlxtend output	13
Fig 3.8: data pre-processing model	14
Fig 3.9. Convolutional Neural Network	17
Fig 3.10: Convolution 2D model structure	19
Fig-4.1: Beans Brown spots	21
Fig-4.2: Cabbage blight	22
Fig-4.3: Early blight of potato	23
Fig-4.4: Late blight tomato	24
Fig-4.5: Scab of potato	24
Fig-4.6: Model Training	25
Fig-4.8: Model Accuracy	27
Fig-4.9: Model Loss	27
Fig-4.10: Model Output	28

Chapter-1

INTRODUCTION

1.1 INTRODUCTION

In our daily life there is no substitute rather than vegetable in food list. Almost 80% of the farmland in Bangladesh cultivate different type of vegetables. Day by day people are moving towards farming and showing interest to avoid Joblessness. So, to make them interested more, there is need for good vegetations. the main reason for the harm of vegetables are its diseases. These diseases aren't easy enough to detect. For this we have to go to different places and much time wastes. Consequently, the attack of diseases goes high and right solution in the right time cannot be done. For plantation face lot of harm and for this farmer turn their back. As a consequence, we are not getting our daily lives, our needs are not being full filled. That being so it is very important to turn farmers away from this uninterested. and necessarily to detect the disease in right time. and to easily detect the use of technology immense.

At present the important of technology has become boundless. Our daily life rigid without technology. Technology is making our life easier from easier by daily basis. In our daily life we do face a lot of diseases. And there is no alternative means rather than machine learning algorithm to detect diseases. we can easily detect by using machine learning algorithm. through this way we can detect diseases can detect fast in less time.at present the use of machine learning has expanded. Hence it made our life very flexible. Machine learning is to shorten huge sized dataset but the technique of searching worthy information. Thus, to decide the phenomena of next stage in the beginning instance vegetable diseases detection. this system can easily detect diseases in a short time and tell if pictures of diseases sent to this.

Vegetable disease detection is a part of Machine learning (ML). Through data train these diseases can be easily detected. In our country lives of majority of people depend on farming to meet their daily earnings. so, to make their life easy to machine learning

algorithm is limitless. We design a dataset through our research and we did care at the matter that this dataset can detect diseases subtle.

1.2 MOTIVATION

Day by day use of machine learning algorithm is increasing and developing the technology in every fields. Machine learning algorithm do ease our daily life. As most of the people trying to do things with less effort, that's why we using machine learning algorithm as Vegetable diseases detection.

Most of the farmers in our country don't know about scientific farming method. Most of the vegetables are damage/spoiled in field. They don't know what type of disease and why it's spoiled. If we find out the solution of these problems it'll be very helpful for our farmers.

1.3 OBJECTIVE

Our Primary goal is

- The various machine learning techniques are applied on user to collect data.
- The data is selected to what type of disease is occurred. And how can solve this disease.
- The results obtain by this algorithm are useful for farmers to take decision about this problem

1.4 OUR UNIQUENESS

- We use efficient algorithm
- Free service and easily accessibility
- More than 90% accuracy

1.5 RESEARCH QUESTIONS:

- Would we able to gather real-data?
- Can we apply this in real life?
- Which is the helpful algorithm those can be utilized for the detect disease?

1.6 EXPECTED OUTCOME

- Most of the vegetables are damage/spoiled in field. Many farmers don't know what type of disease and why it's spoiled.
- We can find the disease and to give them solutions
- If we find out the solution of these problems it'll be very helpful for our farmers

1.7 REPORT LAYOUT

Our report consists of 5 chapter

1. First chapter contains the Introduction, Motivation, Objective, Our Uniqueness, Research Questions, Expected Outcome.
2. Second chapter contains Introduction, Related Works, Research Summary, Scope of the Problem, challenges.
3. Third chapter contains Research subject and Instrumentation, Data collection and preprocessing, Proposed Methodology, Implementation Requirements.
4. Fourth chapter contains Introduction, Data segmentation, Data preprocessing, Input File Creation, Model building and Dataset Fitting, Experiment Accuracy, Loss and Output, Summary.
5. Fifth chapter contains Summary & Recommendation

Chapter 2

BACKGROUND

2.1 INTRODUCTION

In this section, we will briefly portray related works with our project. Where we will describe how we have built this project, what are the problems we faced while building this project and a short description of the expected outcome of it. At the same time, there will be some short descriptions related to this project, which has been done before. This project has been developed by using only Machine Learning Algorithm. This application will be used only detect some disease.

2.2 RELATED WORKS

Since vegetables are our daily necessity food, so there is a lot of research and a lot of work going on about vegetables. But most of them are on vegetable cultivation. Most research is happened on what kinds of diseases occur in vegetable trees or saplings and how to fix them. But when a vegetable is grown up, there is little research on the sets of diseases that occur. Sometimes many farmers can't find out what has happened to their vegetables. This is our initiative thinking about them. We think that our initiative will play a valuable role in agriculture.

A study was conducted on vegetable leaves in 2016 [1]. In these studies, they use leaves of different vegetables to see what diseases they have had. Based on this they create a model that had the result like this-

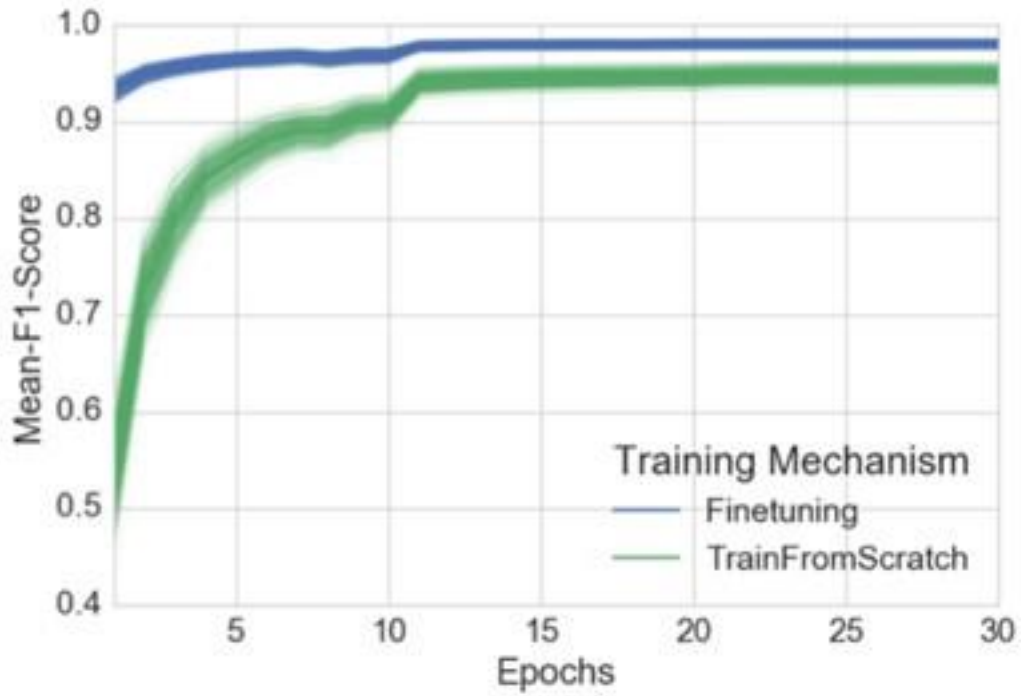


Fig-2.1: Model Train

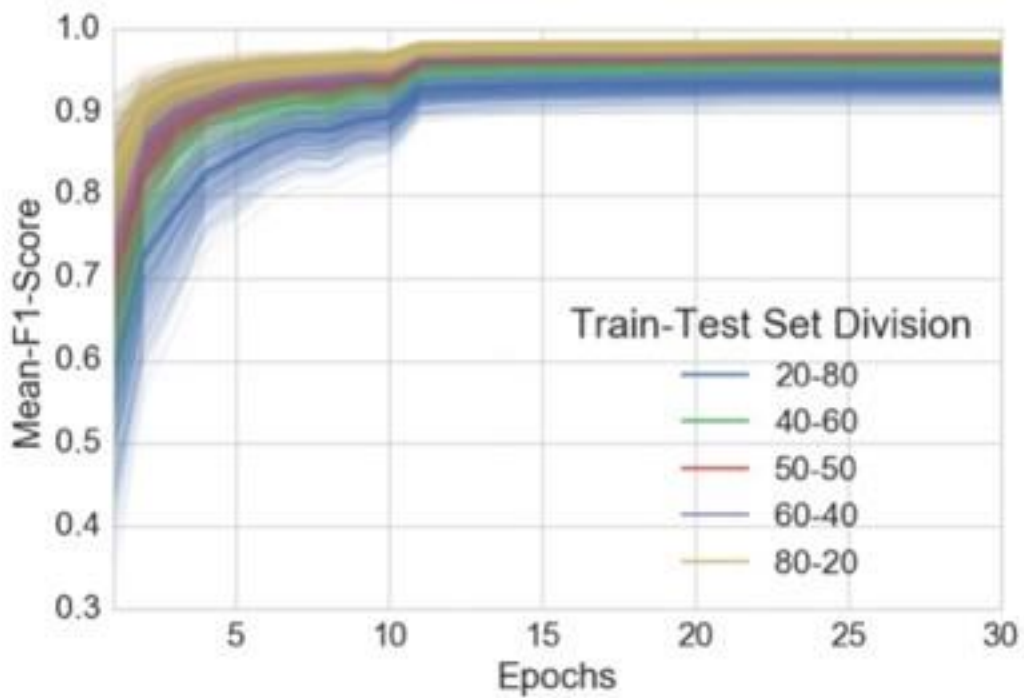


Fig-2.2: Test Model

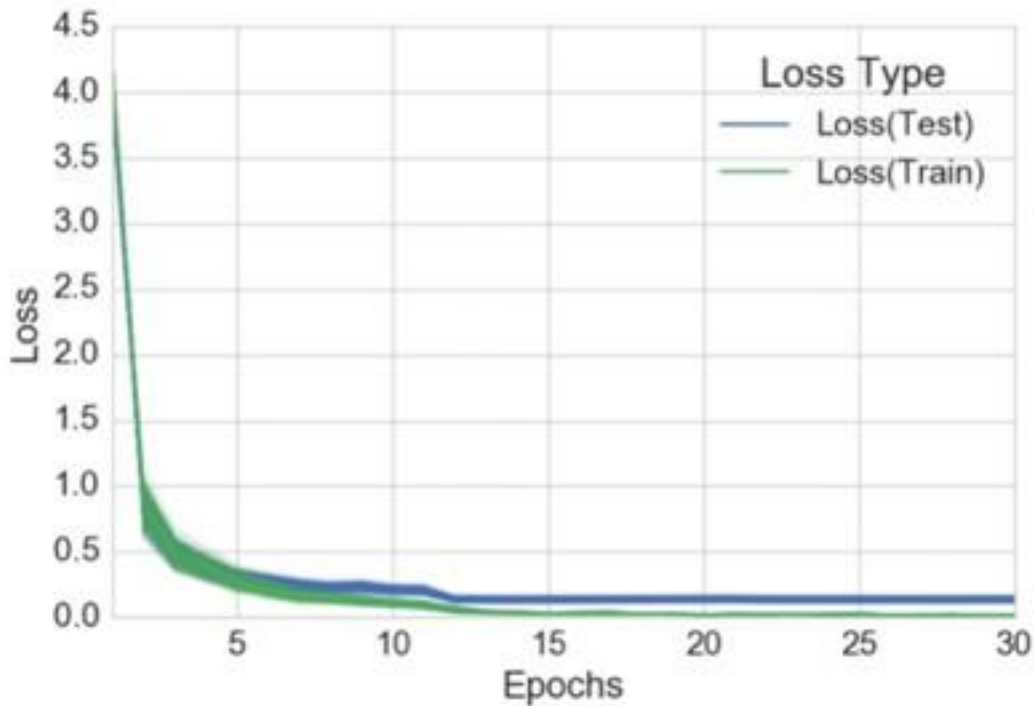


Fig-2.3: Data Loss

2.3 RESEARCH SUMMARY:

So, it has been seen that most of the farmers in our country don't know about the scientific farming method to protect the vegetables from several diseases. To overcome this farming problem a proper disease detection is necessary for a better outcome. For vegetable disease detection different types of data or images are used with a different machine learning algorithm. The majority of them utilized the best measure of foundation data as training data and the remainder of the data as test data. Moreover, two sorts of data give better feedback with the help of the applied classifier model.

2.4 SCOPE OF THE PROBLEM:

Production and marketing of vegetables must be vale into the production side, marketing side, processing, numerous works on disease detection for plants and individual vegetables and there is not much work about vegetable disease detection for several items of vegetables. By the collaboration of this model, we can detect many vegetable diseases. In our country growth of the economy, rate depends on the significant role of the agriculture sector. So, if we want to improve our farming, we need to innovate new technologies and systems. The advantage of this research project is farmers can aware of their crops from harmful diseases. And manufacturing side. There is remarkable production of vegetables in a shorter period. Hence to avoid the post-harvest loss and increase solid profit to develop vegetable disease detection. There have been

2.5 CHALLENGES:

Collecting Data

Data collection was pretty hard. Cause our data is image-based. From various online portals, google images, blog posts, we got our data. Most of the data are from google images.

Selecting model

Fixing a model is a kind of difficult job. If we choose the wrong one whole project would be a dumb and waste of time. We used some tires to create a better model. We used CNN (Convolutional Neural Networks) of which might operate large amounts of data. One of the most popular deep neural networks to analyze image processing and pattern recognition.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 RESEARCH SUBJECT AND INSTRUMENTATION

Machine Language is the most momentous piece of modern science and technology. Our research subject was vegetable disease detection. And for doing this work here we used image processing and authentic technologies like CNN. We are using this method so that the illiterate farmer can get immediate result with high accuracy. For doing this process we need to think this work some small pieces. Like collect dataset or image, process data as fundamental, need action about this dataset, data or image expansion, and finally give decision or analytical analysis.

For this reason, a user does needn't any additional knowledge. A farmer without sufficient sense disease detection knowledge, he or she can use modern techniques for their farming. The dataset which is used as input is mixed of healthy and distorted images and after completing the action of input dataset the system output provides the disease name. A chart is introduced below as the proposed methodology.

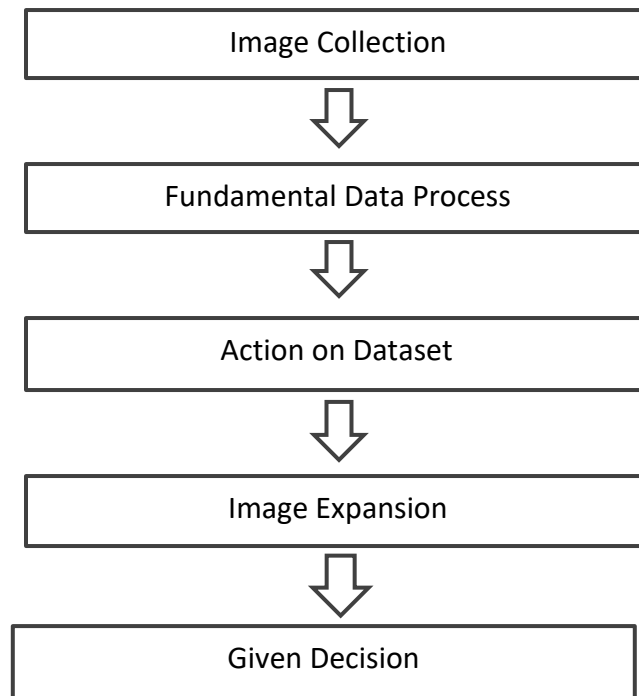


Fig 3.1: Methodology flow diagram

As it is a machine learning process, so we need to learn our device or machine. For this process, we are using python language, which version was 3.8.0. And need some requirements also, which are-

OpenCV-Python:

We are using image for our project. So it's important to understanding image process into our machine. And we know that computer receives our image as visualization with range of value between 0 and 255.

Opencv is a python library which can solve our image vision problem. Actually, opencv supports some algorithm like C++, python, java etc. Here we use opencv python algorithm. Also we need to know header or need to detect any image. Opencv makes it easier. Initial command

```
pip install opencv-python
```

NumPy:

Numpy is an open source numerical python library that contains a multi-dimensional array and matrix data structure. It actually used for working with array or sequentially. This makes our project faster by sorting arrays one by one and taking short memory to store this data. Installation command

```
pip install numpy
```

Sample like-

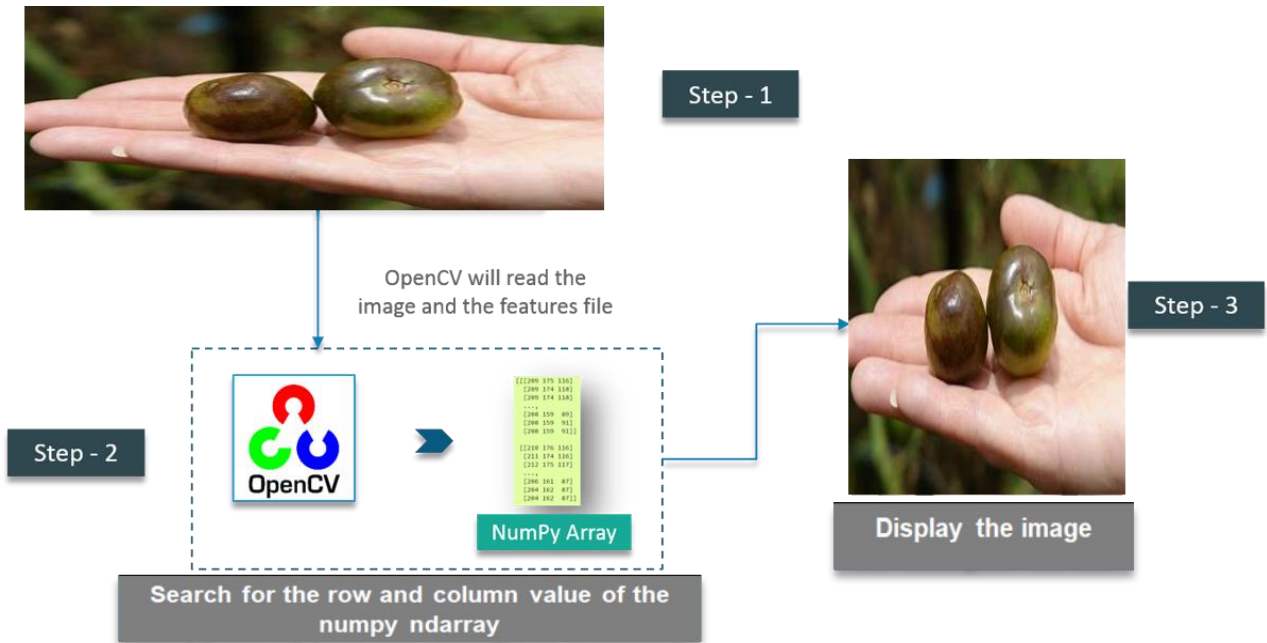


Fig 3.2: sample use of opencv & numpy

Tqdm:

We use tqdm for smart progress bar for wrapping around any iteration. When we train our model or need to find something then we need to wait sometime. We also want to see how much it works or how much time will take to complete the task. Progress bar show us this percentage. It shows us not only how much time has elapsed, but also shows the estimated time remaining for the iterable. Also **utilizing these progress bars, we can see if we are getting stuck somewhere** and work on that immediately. Installation command

```
pip install tqdm
```

like-

```
processed: 1
67%|██████████| 2/3 [00:01<00:00, 1.99it/s]
processed: 2
100%|██████████| 3/3 [00:02<00:00, 1.53it/s]
```

Fig 3.3: sample of tqdm

Python-csv:

Generally, csv files are used to store large number of data. Because they are incredibly simplified spreadsheets –like excel. So we can say that, csv is just a simple file format which is used to store tabular data in a plain text. And it's separated by commas.

We are also using csv file for this kind of file formatting. Where we can store our pictures data into a csv file. Initial command-

```
pip install python-csv
```

Dataset be like-

```
['2010', '102.634', '103.003', '103.52', '103.943', '103.278']  
['2011', '103.078', '103.42', '103.233', '103.948', '103.422']  
['2012', '104.095', '104.7', '104.521', '104.056', '104.341']  
['2013', '104.28', '104.11', '104.541', '105.687', '104.656']  
['2014', '104.698', '105.135', '106.206', '105.827', '105.469']  
['2015', '106.135', '106.414', '106.895', '106.349', '106.448']  
['2016', '106.174', '106.147', '107.013', '107.479', '106.704']
```

Fig 3.4: sample output of csv

Scikit-Learn:

Scikit-learn is a most useful library in python for machine learning. It's used to solve three different kinds of problems in machine learning. They are namely supervised learning, unsupervised learning and reinforcement learning.

We use scikit-learn to help our machine, learn properly and pick the correct variables. It's used for manipulating and summarizing data. It also helps our machine to learn various type of dataset. Installation command-

```
pip install -U scikit-learn
```

TensorFlow:

TensorFlow is a symbolic math library that uses dataflow and differentiable program. It uses in various tasks focused on training and inference of deep neural networks.

In our project when we give any input for information then it need to search them, because we need to find the disease. We can find it in different way, but tensorflow can make it easier. Tensorflow can try to search, one by one letter spacing. We are using here tensorflow version 2.3.0. Command-

```
pip install tensorflow==2.3.0
```

Here given tensorflow working architecture-

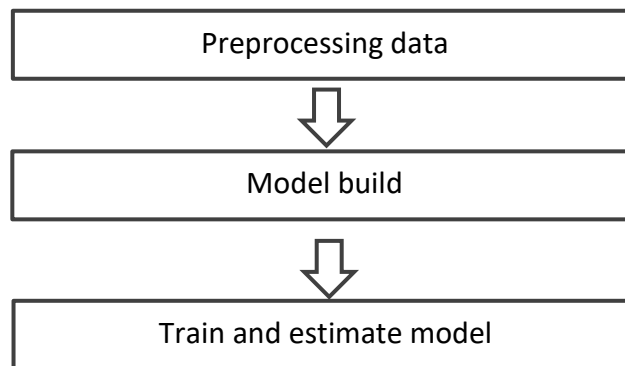


Fig 3.5: tensorflow architecture

Pyppeteer:

When we need to passing any objects then we use pyppeteer as a passing option. There have another version of pyppeteer is puppeteer. But we use pyppeteer, because pyppeteer functions or methods accept both dictionary and keyword arguments.

Pyppeteer is like a chrome browser, which is hidden. Pyppeteer installation-

```
pip install pyppeteer
```

keras:

keras is also like tensorflow, but keras is a high level API that runs on top of tensorflow. Keras simplifies the implementation of complex neural networks with its easy to use framework.

We are using keras for making an API design, which can visible by user not machine. We also run our project on NVIDIA, so keras can help us for doing this. Because keras has native support for mixed-precision training. Install command-

```
pip install keras
```

Sample model-

flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 128)	65664
batch_normalization_6 (Batch Normalization)	(None, 128)	512
dropout_6 (Dropout)	(None, 128)	0

Fig 3.6: keras model

MLxtend:

Here we need to match the input image and train dataset image. For this mathematical reason and get good result we use this python mlxtend library. Installation-

```
pip install mlxtend
```

Output-

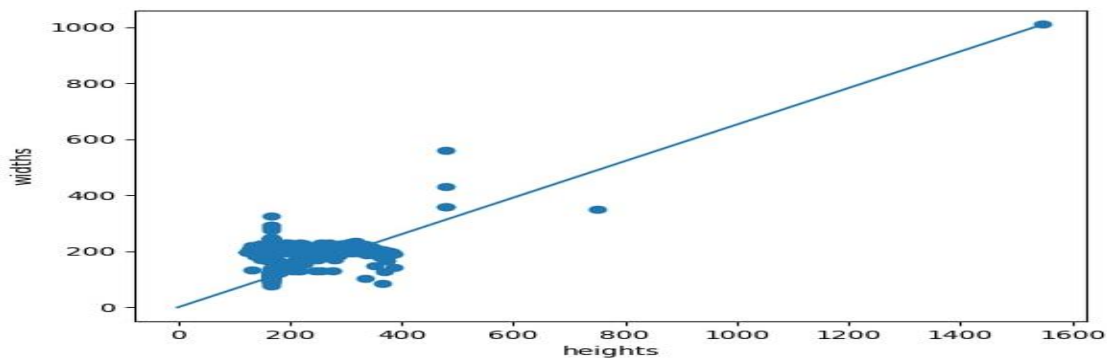


Fig 3.7: mlxtend output

3.2 DATA COLLECTION & PRE-PROCESSING

For doing any kind of reaches of making any kind of project or any profound learning algorithm it needs a huge collection of data. As enormous the dataset is the outcome is that vastly improved. We also need an attractive number of data to implement in this project. There is not enough sufficient dataset on the web. On the other hand, covid-19 criteria con not permit us to go outside for collecting data. So, it is difficult for us to collect data.

For this reason, we visit or browse many sites and collect data one by one from the different web. Our 100% data are collected from different web.

If we want to train our machine then we need to resize or pre-processing our dataset. So, it is the first step for any grey scale to learn our machine. It is important to remember that, whenever we want to train our machine then the bias-variance needs to trade-off. We also need to look at image color, because if the colors in the images didn't contain any distinguishing information, then it can reduce the number of inputs by a third by greyscaling.

Also, we know that computer doesn't see an image. For using image and train model we are using here CNN. CNN is mainly used in image analysis tasks like image recognition, object detection and segmentation.

Pre-processing model-

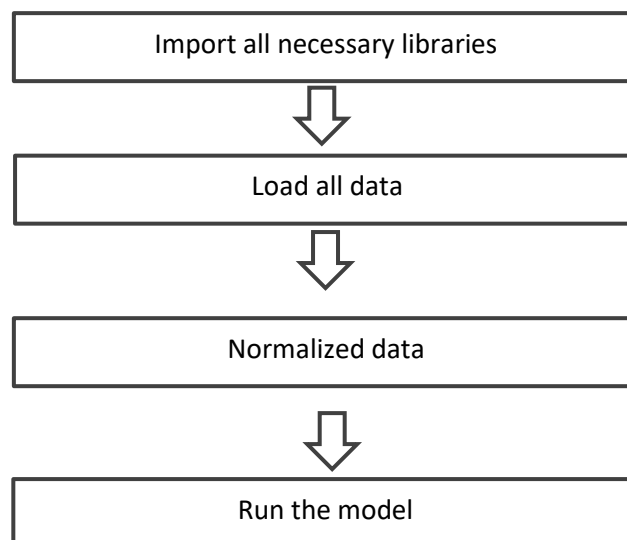


Fig 3.8: data pre-processing model

1. Import all necessary libraries:

```
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense, Dropout, BatchNormalization, Conv2D, MaxPool2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing import image
from tensorflow.keras import layers
```

2. Load all data:

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=0, test_size=0.20)
print("creating dataset file...")
np.savez_compressed('dataset/train_test_set/compressed_processed_train_test_set_01.npz',
x_train=x_train, x_test=x_test, y_train=y_train, y_test=y_test)
```

3. Normalized data:

```
for i in range(1, (t_cat+1)):
    j = 1
    dirc = i_direct + str(i) + '/'
    print(dirc)
    for ip in tqdm(os.listdir(dirc)):
        img = cv2.imread((dirc+ip))
        if img is not None:
            (w, h) = img.shape[:2]
            filename = str(i) + "-" + str(j)
            f_path = t_direct + filename + ".png"
            cv2.imwrite(f_path, img)
            _images.append(filename)
            _index.append(str(i))
            np_sets.append([filename, str(i), str(w), str(h)])
            j = j+1
    print("All Train Image Transferred for index " + str(i))
print("Transferred and Rename Successfull!")
```

4. Run the model:

```
Epoch 95/100
66/66 [=====] - 35s 525ms/step - loss: 0.1570 - accuracy: 0.8331 - val_loss: 0.2501 - val_accuracy: 0.7317
Epoch 96/100
66/66 [=====] - 35s 525ms/step - loss: 0.1284 - accuracy: 0.8576 - val_loss: 0.2043 - val_accuracy: 0.7439
Epoch 97/100
66/66 [=====] - 35s 525ms/step - loss: 0.1285 - accuracy: 0.8790 - val_loss: 0.2327 - val_accuracy: 0.6890
Epoch 98/100
66/66 [=====] - 35s 524ms/step - loss: 0.1478 - accuracy: 0.8270 - val_loss: 0.1956 - val_accuracy: 0.7378
Epoch 99/100
66/66 [=====] - 34s 523ms/step - loss: 0.1191 - accuracy: 0.8744 - val_loss: 0.2115 - val_accuracy: 0.7195
Epoch 100/100
```

3.3 PROPOSED METHODOLOGY:

In our research, we used CNN (Convolutional Neural Network) multi-level model. For the first layer we use 2D convolution and dimensional vector is (32), Kernel_size (3×3), activation function as rectifier and input shape is (0), BatchNormalization used for transform the output value as close to 0 or 1, Max_Pool_size (2×2) and Dropout (0.3). For the second layer, the dimensional vector is (64), Dropout is (0.4), Kernel_size, the activation function, and input shape, and Max_Pool is the same as the first layer. For the third layer, the dimensional vector is (128), Dropout (0.5), Kernel_size, activation function, and input shape, and Max_Pool as same as the previous layer.

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

For the fourth layer, the dimensional vector is (256), Dropout (0.6), Kernel_size, activation function, and input shape, and Max_Pool as same as the previous layer. For the fifth layer, the dimensional vector is (512), Dropout (0.5), Kernel_size, activation function, and input shape, and Max_Pool as same as the previous layer. For the sixth layer we use dense layer and dimensional vector is (128), activation function as rectifier and input shape is (0), BatchNormalization used for transform the output value as close to 0 or 1 and Dropout is (0.5). The output layer used sigmoid function for activation and batch_size is (10). With the sigmoid function, the output value should be between 0 and

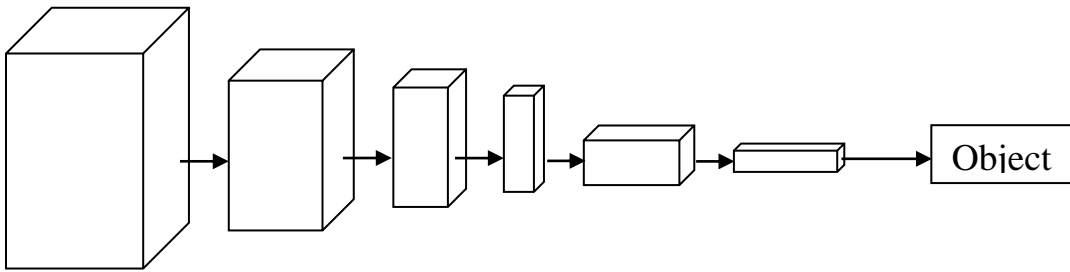


Fig 3.9. Convolutional Neural Network.

Since we used a Dense layer with activation function ‘sigmoid’. For Dense layer used a dimensional vector and then Batch Normalization to normalize the data. The sigmoid function returns the output line between 0 to 1 with a characteristic curve. The Dense layer also reduces the loss rate, time of every epoch we used 10, and batch size 100. At last, we compile with adam optimization and fit the model and set the input and output sequence meaningfully.

The optimization algorithm ‘Adam’ is selected for enough alternative results of deep learning. Several types of application like NLP and computer vision shows perfect adaption with the optimization algorithm. The optimization algorithm shows a great performance for the speed of training. The optimization technique is efficient for isolated learning rate estimation dealing with numerous parameters from the first and second instance of a gradient.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

The model served to use 0.001 as the learning rate in the Adam '3' optimization. The neural network and cross-entropy function take measures acceptable output in the classification and prediction and this result is more grantable than mean square error. Commonly, the training is not halted out in disable of getting enough fraction using Cross-entropy error and weighting variation.

In the following way algorithm used in our model is: -

1. **Load Data()**
2. declare **Sequential ()**
3. add(**Conv2D(kernal_size(units), activation, input shape)**)
4. add(**BatchNormalization()**)
5. add (**MaxPool2D()**)
6. add(**Dropout()**)
7. add(**Dense(output shape, activation)**)
8. add(**BatchNormalization()**)
9. set (**epochs value**)
10. set (**batch_size value**)
11. compile(**optimizer , loss function, accuracy**)
12. model fit (**epochs, batch_size, validation_data**)

The coding part used Keras sequential model. From this model, we used convolution 2D as the input layer and 'relu' as an activation function which is called a rectifier. Then normalized the data With Batch Normalization layer for transform the value as close to 0 or 1. MaxPool2D used for an input signal fixed of several input planes also used Dropout layer for outfitting. For output, we used the Dense layer with activation function 'sigmoid' and Batch Normalization.

Also increased learning rate to reduce loss and used batch size like 100 to reduce training time for each epoch. We fit more than 800 images in the model and found the accuracy near 90%.

Here is the structure of the convolution 2D model-

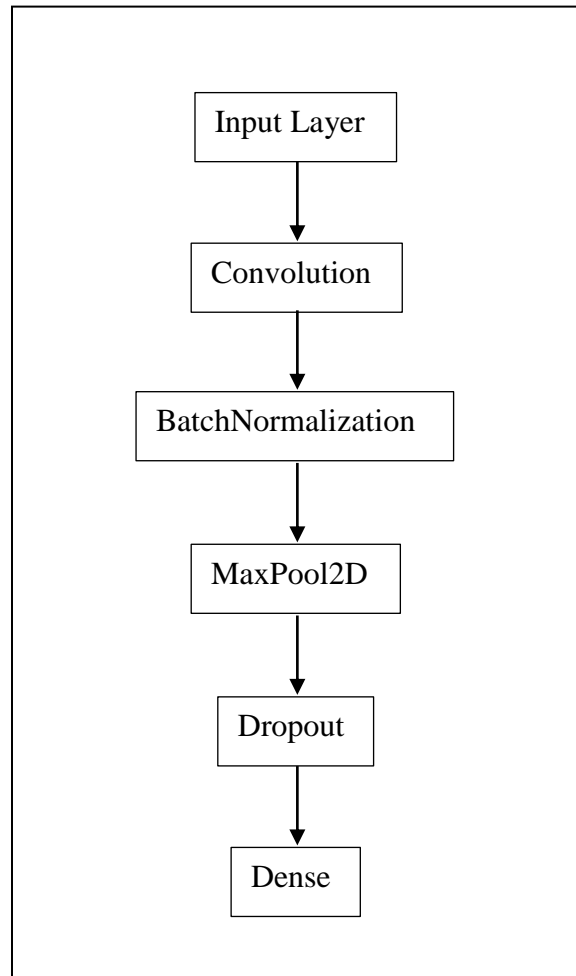


Fig 3.10: Convolution 2D model structure.

3.4 IMPLEMENTATION REQUIREMENTS:

We needed some hardware and software to train our data in the model. For software purposes we selected Python. It is a high-level and most popular interpreted dynamic programming language. It turns into a more popular language because it emphasizes code authenticity. It has a large collection of libraries, packages, methods to do several works. Writing code with python it's like commanding the machine.

And it has automatic memory management. Multiple programming instances are supported in it and have a large number of libraries like data science, machine learning, data analysis, image processing, data mining, etc. It is best for research-based work because it naturalizes the code quality. For our research, we used python 3.8.0, PyCharm as I.D.E, Opencv-python for visualizing an image, Numpy for multi-dimensional array and matrix data structure. Numpy makes our project faster by sorting arrays one by one and taking short memory to store this data. Used Tqdm for smart progress bar for wrapping around any iteration. Python-CSV as stores a large number of data. Because they are incredibly simplified spreadsheets –like excel sheets. Scikit-learn is used to solve three different kinds of problems in machine learning. They are namely supervised learning, unsupervised learning, and reinforcement learning. TensorFlow uses dataflow and differentiable programs. It uses in various tasks focused on training and inference of deep neural networks. Pyppeteer for passing any objects. Also used Keras for making an API design, which can visible to the user. Mlxtend to match the input image and train dataset image.

Hardware requirements:

1. Operating system (Windows 10)
2. Hard disk (1 TB)
3. Ram (8 GB)

Chapter 4

EXPERIMENTAL RESULTS & DISCUSSION

4.1 INTRODUCTION

We have attempted to build up an algorithm Regarding Vegetable Diseases Detection, which is Convolutional Neural Network (CNN) by which one can detect Vegetable Diseases. In section 3 we have depicted the research project and the technique of its working. Presently in section 4 we have attempted to show the yield results outcome and procedure of the Vegetable Diseases Detection.

4.2 DATA SEGMENTATION

For this research project, we gathered preprocessed data from prestigious site for example: google, government sites and other informative sites. An image file is put on a jpg, jpeg, gif file. At this point we change file jpg/jpeg/gif to PNG. A perspective on portioned data is given below.



Fig-4.1: Beans Brown spots



Fig-4.2: Cabbage blight



Fig-4.3: Early blight of potato



Fig-4.4: Late blight tomato

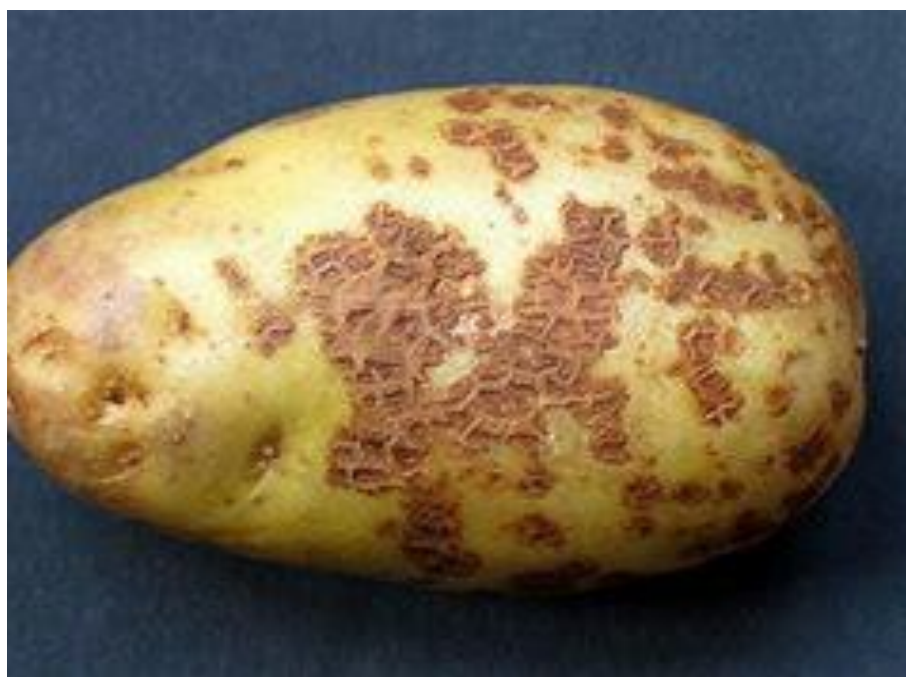


Fig-4.5: Scab of potato

In all figure we can see an all file are not covered with PNG file. So, information preprocessing is expected to lessen information redundancy to locate the specific information or data. So, we cannot input the data into our model. We must convert our data into the face of machine understanding format which we can feed then to the model.

4.3 DATA PRE-PROCESSING

We have Structure a Python content script to start an information preparing task. The content is fundamentally working for.

- Removing unwanted data
- File convert jpg/gif to PNG

4.4 INPUT FILE CREATION

After a productive preprocessing process, we have some absolute file close by. At that point, we need to perform Machine Learning algorithm which is Convolutional Neural Network (CNN).

4.5 MODEL BUILDING & DATASET FITTING

For building our model and training our data we are using more than 800 pictures. Using CNN, we fit this data into our model.

Thinking on saving time and better performance first we teach transfer (TensorFlow). Then we take images and do data augmentation (TensorFlow), which helps us with different kinds of data. At last, we fit the dataset. To ignore overfitting, we avoid pulling or pushing any data from the dataset and for underfitting, we give enough dataset for training

```
C:\Windows\System32\cmd.exe
66/66 [=====] - 35s 524ms/step - loss: 0.1591 - accuracy: 0.8224 - val_loss: 0.2723 - val_accuracy: 0.6585
Epoch 83/100
66/66 [=====] - 34s 522ms/step - loss: 0.1370 - accuracy: 0.8499 - val_loss: 0.2149 - val_accuracy: 0.7317
Epoch 84/100
66/66 [=====] - 35s 523ms/step - loss: 0.1364 - accuracy: 0.8453 - val_loss: 0.3408 - val_accuracy: 0.6220
Epoch 85/100
66/66 [=====] - 34s 522ms/step - loss: 0.1532 - accuracy: 0.8300 - val_loss: 0.2072 - val_accuracy: 0.7378
Epoch 86/100
66/66 [=====] - 35s 524ms/step - loss: 0.1245 - accuracy: 0.8821 - val_loss: 0.2069 - val_accuracy: 0.7561
Epoch 87/100
66/66 [=====] - 42s 635ms/step - loss: 0.1357 - accuracy: 0.8714 - val_loss: 0.2251 - val_accuracy: 0.7256
Epoch 88/100
66/66 [=====] - 35s 532ms/step - loss: 0.1114 - accuracy: 0.8821 - val_loss: 0.2983 - val_accuracy: 0.6768
Epoch 89/100
66/66 [=====] - 35s 524ms/step - loss: 0.1216 - accuracy: 0.8729 - val_loss: 0.2206 - val_accuracy: 0.7317
Epoch 90/100
66/66 [=====] - 35s 524ms/step - loss: 0.1284 - accuracy: 0.8622 - val_loss: 0.3278 - val_accuracy: 0.6159
Epoch 91/100
66/66 [=====] - 34s 522ms/step - loss: 0.1246 - accuracy: 0.8606 - val_loss: 0.2930 - val_accuracy: 0.6341
Epoch 92/100
66/66 [=====] - 34s 523ms/step - loss: 0.1541 - accuracy: 0.8315 - val_loss: 0.1963 - val_accuracy: 0.7317
Epoch 93/100
66/66 [=====] - 34s 522ms/step - loss: 0.1372 - accuracy: 0.8392 - val_loss: 0.2262 - val_accuracy: 0.6890
Epoch 94/100
66/66 [=====] - 35s 526ms/step - loss: 0.1770 - accuracy: 0.7979 - val_loss: 0.2592 - val_accuracy: 0.7073
Epoch 95/100
66/66 [=====] - 35s 525ms/step - loss: 0.1570 - accuracy: 0.8331 - val_loss: 0.2501 - val_accuracy: 0.7317
Epoch 96/100
66/66 [=====] - 35s 525ms/step - loss: 0.1284 - accuracy: 0.8576 - val_loss: 0.2043 - val_accuracy: 0.7439
Epoch 97/100
66/66 [=====] - 35s 525ms/step - loss: 0.1285 - accuracy: 0.8790 - val_loss: 0.2327 - val_accuracy: 0.6890
Epoch 98/100
66/66 [=====] - 35s 524ms/step - loss: 0.1478 - accuracy: 0.8270 - val_loss: 0.1956 - val_accuracy: 0.7378
Epoch 99/100
66/66 [=====] - 34s 523ms/step - loss: 0.1191 - accuracy: 0.8744 - val_loss: 0.2115 - val_accuracy: 0.7195
Epoch 100/100
```

Fig-4.6: Model Training

4.6 EXPERIMENT ACCURACY, LOSS & OUTPUT

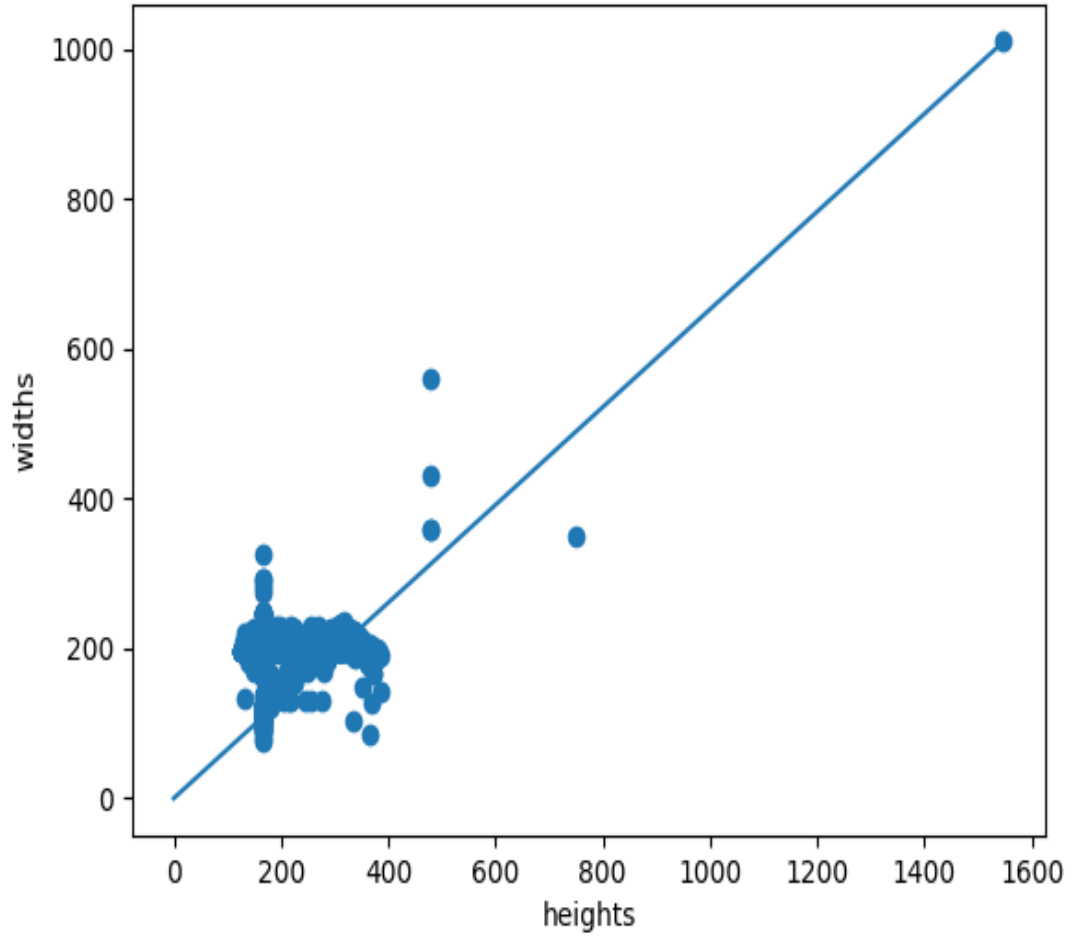


Fig-4.7: mlxtend output

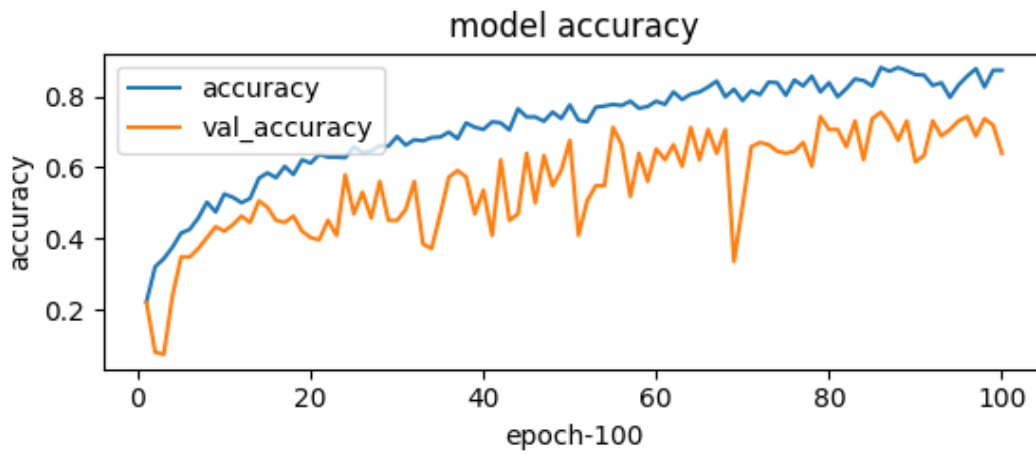


Fig-4.8: Model Accuracy

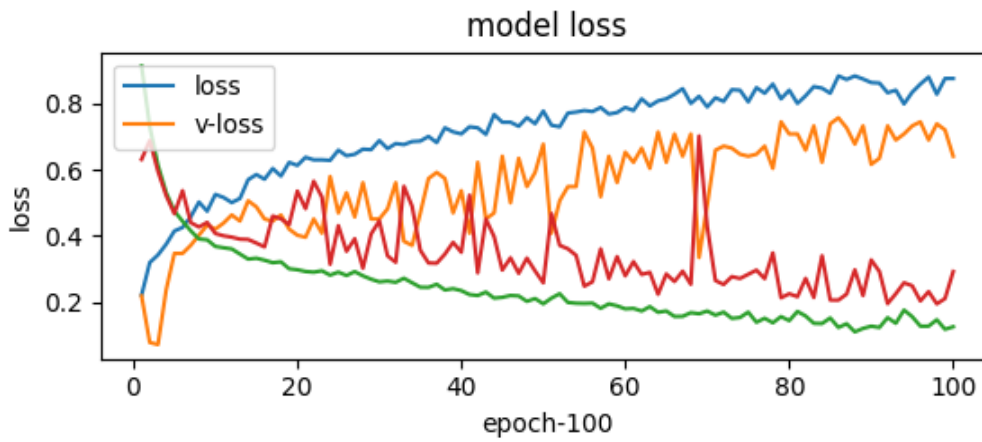


Fig-4.9: Model Loss

```
(VegetableDiseasesDetectionenv) F:\Thesis\Vegetale Disease Detection>python test_model.py
2021-12-02 20:54:39.446479: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2021-12-02 20:54:39.446995: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2021-12-02 20:54:46.672641: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2021-12-02 20:54:46.672841: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2021-12-02 20:54:46.688997: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: DESKTOP-7BAD28J
2021-12-02 20:54:46.689290: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-7BAD28J
2021-12-02 20:54:46.691346: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-12-02 20:54:46.710198: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x1fab522ea10 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-12-02 20:54:46.710344: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
Input Image File Name: leaf_curl_tomato.png
Leaf curl tomato = 96.59%
```

Fig-4.10: Model Output

4.7 SUMMARY

After completion of all the processes successfully we get our anticipated result “Leaf curl tomato” 96.59% accuracy.

CHAPTER 5

SUMMARY OF RESEARCH & CONCLUSION

5.1 SUMMARY OF RESEARCH

The research project is for vegetable diseases detection. We need to collect dataset. we collect dataset from google, government sites and other online sources. After that dataset is processed to feed into the system, we clean the data, we feed it into the system to train and the output is diseases detection.

5.2 CONCLUSION

In this paper, we used many vegetables sample, Example: Beans Brown spots, Cabbage blight, Scab of potato, Leaf Curl Tomato and proposed machine learning algorithm that can detect vegetable diseases. We used Convolutional Neural Network (CNN) for detect vegetable diseases. This model provides best possible results instantly. As a result, the farmer who can want to detects vegetable diseases manually can save their time. Our model gives satisfaction output. Our model gives 88% accuracy (Fig: Model Training). We'll work on it in the future.

.

REFERENCE

- [1] Plant Disease Detection using CNN Model and Image Processing, Md. Tariqul Islam, (2020)
- [2] Disease Detection in Vegetables Using Image Processing Techniques, Gouri C.Khadabadi , Vijay S. Rajpurohit , Arun Kumar, V.B.Nargund, (2015)
- [3] Disease Prediction Using Machine Learning Akash C. Jamgade, Prof. S. D. Zade, (2019)
- [4] Plant Disease Detection Using Machine Learning, Shima Ramesh Maniyath, Hebbar Ram, (2018)
- [5] Scope, Status and Importance of Fruits and Vegetables Industries in INDIA & Role of Fruits and Vegetables in Indian Economy, Sugam Birthare, (2018)
- [6] Comparison Of Models For Forecasting Of Stewart's Disease Of Corn In Iowa, Esker, P. D., Harri, J., Dixon, P. M., And Nutter, F. W., Jr., (2006)
- [7] Predicting Crop Diseases Using Data Mining, Umair Ayub, Syed Atif Moqurrab, (2018).
- [8] Morphological Identification And Management Of Fungal Post-Harvest Pathogens Of Peach (*Prunus Persica* L), Naeem, M., Irshad, G., Naz, F., Noorin, S., Aslam, F., & Rafay, A., (2018).
- [9] A Survey Paper On Crop Disease Identification And Classification Using Pattern Recognition And Digital Image Processing Techniques, Goutum Kambale¹, Dr.Nitin Bilgi, (2017).
- [10] Using deep learning for image-based plant disease detection. Mohanty, S. P., Hughes, D. P. & Salathe, M, (2016).
- [11] A Review on the main challenges in automatic plant disease identification based on visible range images. Barbedo, J. G. A, (2016).
- [12] Visual tea leaf disease recognition using a convolutional neural network model, *Symmetry*, 11, (2019).
- [13] Identification of rice diseases using deep convolutional neural networks. Lu, Y., Yi, S., Zeng, N., Liu, Y. & Zhang, Y, (2017).
- [14] A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Ma, J. et al, (2018).
- [15] Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Ferentinos*, K. P, (2018).

- [16] Rice blast disease recognition using a deep convolutional neural network. Liang, W., Zhang, H., Zhang, G. F. & Cao, H. X, (2019).
- [17] K. & Moussaoui Deep learning for tomato disease: classification and symptoms visualization. Brahim, M., Boukhalifa, (2017).
- [18] Report of the Plenary of the Intergovernmental Science-Policy Platform on Biodiversity Ecosystem and Services on the work of its fourth session, (2016)
- [19] A simple Way To Prevent Neural Networks From Overfitting, Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, (2014)
- [20] Cotton Plant Leaf Diseases Identification Using Support Vector Machine. Jyoti And Tanuja, December, (2017).

VEGETABLE DISEASES DETECTION USING MACHINE LEARNING

ORIGINALITY REPORT

29%
SIMILARITY INDEX

28%
INTERNET SOURCES

4%
PUBLICATIONS

%
STUDENT PAPERS

PRIMARY SOURCES

1 dspace.daffodilvarsity.edu.bd:8080 **21%**
Internet Source

2 www.ijert.org **4%**
Internet Source

3 www.coursehero.com **1%**
Internet Source

4 hub.packtpub.com **1%**
Internet Source

5 Chinmay R C, Mrinal Raj, Sarthak Mishra, Shobha K. "Record.Ai - An Ai Based Solution To Classify Calls Based On Conversation", 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021 **1%**
Publication

6 www.activestate.com **<1%**
Internet Source

7 datascience.stackexchange.com **<1%**
Internet Source

8

medium.com

Internet Source

<1 %

9

www.analyticsvidhya.com

Internet Source

<1 %

10

www.lincoln.ac.nz

Internet Source

<1 %

11

link.springer.com

Internet Source

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On