

**Savitribai Phule Pune University**

**A**

**Internship Report**

**On**

**“****DETECTION AND IDENTIFICATION OF COTTON PESTS”**

Submitted in partial fulfillment of the requirement for the award of the degree of

#### BATCHELOR OF ENGINEERING IN COMPUTER ENGINEERING

#### By,

**Suyog Magar Exam No: 72037016H**

**Suraj Katariya Exam No: 72037009E**

**Saurabh Madake Exam No: 72037015K**

**Aakif Khan Exam No: 72036978K**



**Department of Computer Engineering**

**ANANTRAO PAWAR COLLEGE OF ENGINEERING & RESEARCH**

**SHAHU COLLEGE ROAD, PARVATI, PUNE - 411009.**

**Academic year 2021-2022**

#### 

#### This is to certify that, the Internship report

#### “DETECTION AND IDENTIFICATION OF COTTON PESTS”

#### submitted by,

**Suyog Magar Exam No: 72037016H**

**Suraj Katariya Exam No: 72037009E**

**Saurabh Madake Exam No: 72037015K**

**Aakif Khan Exam No: 72036978K**

#### for partial fulfillment of the requirement for the award of the Bachelor Of Engineering in COMPUTER ENGINEERING at ANANTRAO PAWAR COLLEGE OF ENGINEERING & RESEARCH, Pune as laid down by the Savitribai Phule Pune University. This is a record of the work carried out under my supervision and guidance during academic year 2021 - 2022.

**Prof.** **(Dr).**  **Sandeep Kadam** (HOD)

#### Prof. Anil Lohar Prof. Pranjali More

#### (INTERNAL GUIDE) (INTERNAL GUIDE)

#### Prof. (Dr). Sunil B. Thakare

#### (Principal)

**Place**: - Pune

**Date**: - / / 2021-22

### ACKNOWLEDGEMENT

The report would not have been completed without the encouragement and support of many people who gave their precious time and encouragement throughout the period. we want to thank our advisers and everyone at the company for their patience and assistance during our on-site training also to my internal guide **Prof. Anil Lohar & Prof. Pranjali More** Thanks to their guidance, we were able to develop **Image Recognition Software** and learn about **CNN , PYTORCH and Django**.

We are also grateful to **Prof.** **(Dr).** **Sandeep Kadam** (**Head Computer Engineering Department)**, ANANTRAO PAWAR COLLEGE OF ENGINEERING & RESEARCH for continuous motivation, support in all aspects.

We are most grateful to our honorable Principal **Prof. (Dr). Sunil B. Thakare** for giving us the permission for internship. We sincerely thank to the entire team of staff members, our college, company, our family and those who knowingly and unknowingly have contributed in their own way in completion of this Internship report.

**Mr. Suyog Magar**

**Mr Suraj Katariya**

**Mr. Saurabh Madake**

**Mr. Aakif Khan**

**Contents:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No. | CHAPTER | Page No. | |
|  | Acknowledgement. . . . . . . . . . . . . . . . . . . . . . | i | |
| 1 | INTRODUCTION. . . . . . . . . . . . . . . . . . . . .1.1 Introduction. . . . . . . . . . . . . . . . . . . . . . . | 11 | |
| 2 | PROBLEM STATEMENT / OBJECTIVES . . . . . . . | 3 | |
| 3 | MOTIVATION. . . . . . . . . . . . . . . . . . . . . . . | 4 | |
| 4 | METHODOLOGICAL DETAILS4.1 Back-end. . . . . . . . . . . . . . . . . . . . . . . . .4.2 Front-end. . . . . . . . . . . . . . . . . . . . . . . . .4.3 Deployment . . . . . . . . . . . . . . . . . . . . . . . | 551216 |
| 5 | RESULTS / ANALYSIS. . . . . . . . . . . . . . . . . . | 17 | |
| 6 | SOURCE CODE. . . . . . . . . . . . . . . . . . . . . . | 18 | |
| 7 | SCREENSHOTS. . . . . . . . . . . . . . . . . . . . . . | 31 | |
| 8 | CONCLUSION. . . . . . . . . . . . . . . . . . . . . . .8.1 References. . . . . . . . . . . . . . . . . . . . . . . . | 3334 | |
| 9 | WEEKLY OVERVIEW. . . . . . . . . . . . . . . . . . . | 35 | |
| 10 | PRESENTATION. . . . . . . . . . . . . . . . . . . . . . | 38 | |

**List of Figures:**

|  |  |  |
| --- | --- | --- |
| FIGURE | ILLUSTRATION | **Page NO.** |
| 1.1 | Cotton Pests . . . . . . . . . . . . . . . . . . . . . . . | 1 |
| 4.1 | CNN Layer Structure. . . . . . . . . . . . . . . . . . . | 6 |
| 4.2 | Pytorch Workflow. . . . . . . . . . . . . . . . . . . . | 8 |
| 4.3 | Project Methodology. . . . . . . . . . . . . . . . . . . . | 8 |
| 4.4 | Image Recognition. . . . . . . . . . . . . . . . . . . . | 10 |
| 5.1 | Accuracy Screenshot. . . . . . . . . . . . . . . . . . . . | 17 |
| 5.2 | Accuracy Graph. . . . . . . . . . . . . . . . . . . . . . | 17 |
| 7.1 | Home Page. . . . . . . . . . . . . . . . . . . . . . . . | 31 |
| 7.2 | Browse Image. . . . . . . . . . . . . . . . . . . . . . . | 31 |
| 7.3 | Submit the Image. . . . . . . . . . . . . . . . . . . . . . | 32 |
| 7.4 | Prediction Image. . . . . . . . . . . . . . . . . . . . . . | 32 |

**Chapter 1**

**Introduction**

**1.1 Introduction**

Cotton is the collective name given to four species of plants in the genus Gossypium, Gossypium hirsutism, Gossypium barbadense, Gossypium arboretum and Gossypium hieracium which are perennial shrubs in the family Malvaceae grown for the fluffy fibre which protects the seeds of the plant.

From emergence until harvest, various pests attack the roots, leaves, stems or fruit (squares, blooms and bolls) of cotton. Growers and their field scouts must be vigilant in locating these pest outbreaks so that timely control measures can be undertaken (Figure 3). Economic threshold levels have been established for many cotton pests. A threshold infestation is the point at which control measures are needed to prevent the target pest from reaching its economic injury level (when control costs equal damage caused by the pest). The goal should be to suppress pest populations not annihilate them. Missing the opportunity to control threshold infestations and targeting the pests' earlier growth stages can lead to a greater number of pesticide applications. This increases production costs and yield losses, secondary pest outbreaks, and pest resistance to future chemical control measures.

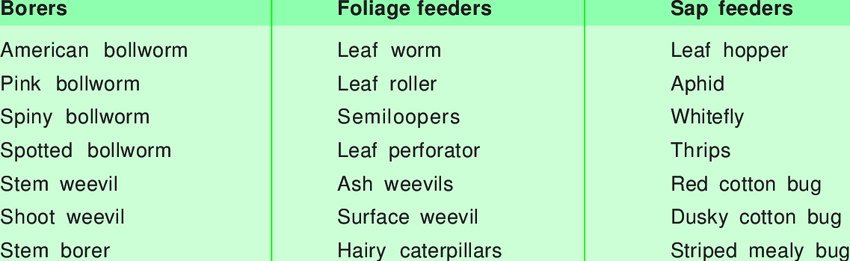


Figure 1.1: Cotton Pests

To prevent the Cotton crop from infestation, identification and Using Preventive Measures to avoid it is extremely important. Hence, We decided to Create a Hands on tool for Farmer enabling them to accurately identify the cotton pest and provide them with enough Knowledge so as to be able to manage it.

For the Correct identification of the Pest, We employed Image Recognition Algorithms specifically CNN of Machine and Deep Learning to train our Module in a Pytorch Environment.

**Chapter 2**

**PROBLEM STATEMENT / OBJECTIVES**

**2.1 Problem statement:**

Detection and identification of Cotton Pests using Image Recognition Algorithms

**2.2 Objectives:**

* To develop a system that is capable of detecting and identifying the type of pest on cotton crop.
* To provide an efficient and accurate methodology for prevention of Cotton Pests.
* To provide organic and inorganic information for eradication of cotton pests.

**Chapter 3**

**MOTIVATION**

The scope of this project is to design a system which can assist farmers in controlling pests on cotton crops and make cotton crops pest free. We use image recognition technique to identify pests and their types on cotton crops and which pesticides can be used to control growth of those pests. This system will help farmers to keep track of crops and will minimize the damage done by pests.

**Chapter 4**

**METHODOLOGICAL DETAILS**

**4.1 BACK END**

**CNN (Convolutional neural network technology)**

**Advantages:**

* Powerful and accurate way of solving classification problems.
* Automatically detects the important features without any human supervision.

**Limitations:**

* Lots of training data is required
* It does not encode the position and orientation of objects

**Applications:**

* Analysing documents
* Predicting Climate change, Earthquakes and Natural Disasters

**What is deep learning?**

Deep learning definition: A subfield of machine learning that structures algorithms in layers to create an “artificial neural network” that can learn and make intelligent decisions on its own.

Deep learning describes algorithms that analyse data with a logic structure similar to how a human would draw conclusions. Note that this can happen both through supervised and unsupervised learning. To achieve this, deep learning applications use a layered structure of algorithms called an artificial neural network (ANN). The design of such an ANN is inspired by the biological neural network of the human brain, leading to a process of learning that’s far more capable than that of standard machine learning models.

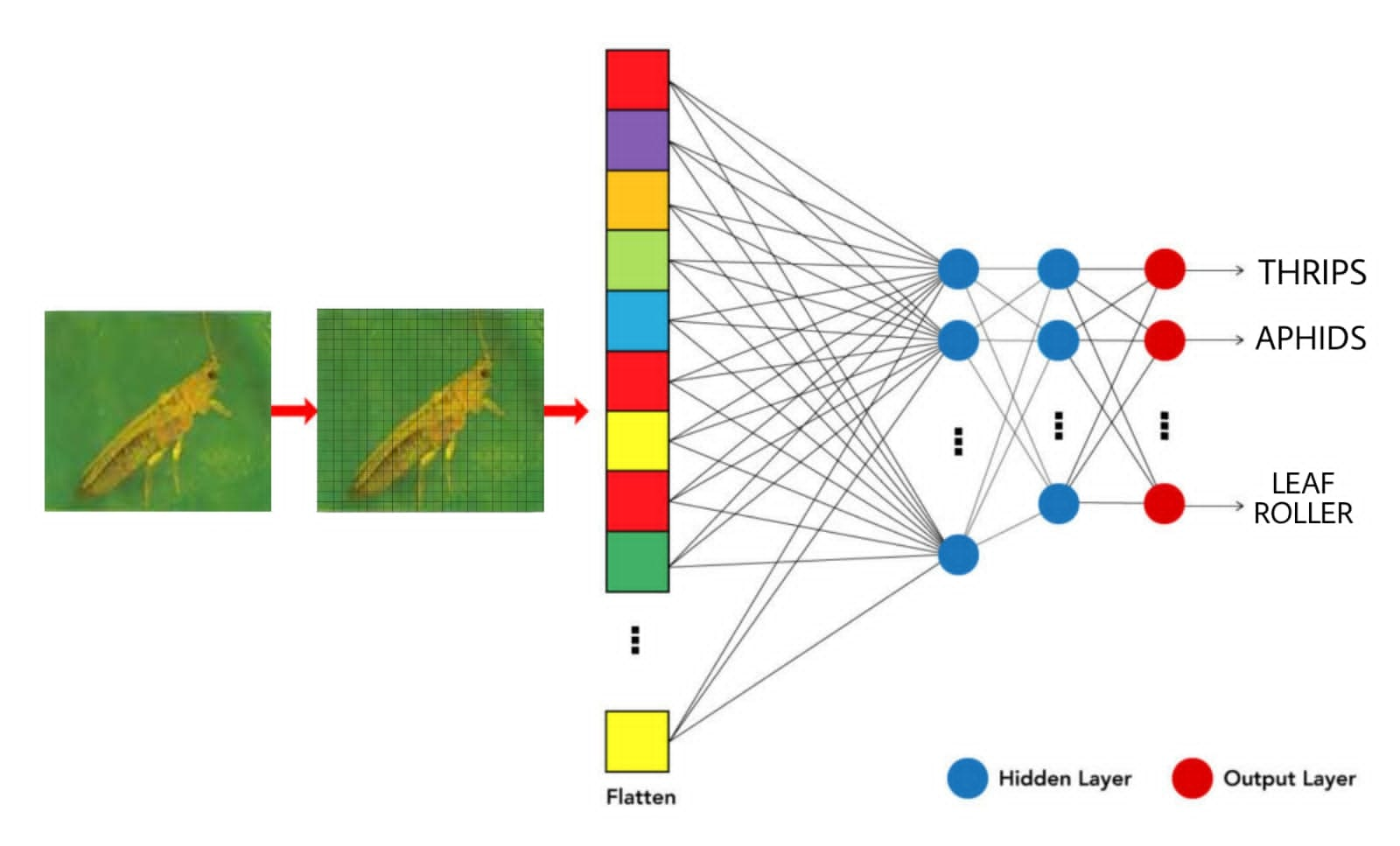


Figure 4.1: CNN Layer Structure

**PYTORCH**

PyTorch is a scientific computing framework that offers broad support for machine learning algorithms. It is a Lua based deep learning framework and is used widely amongst industry giants such as Facebook, Twitter, and Google.

**Its two primary purpose are :**

* Replacing Numpy to use the power of GPUs for faster computation.
* Calculating gradients to perform backpropagation on neural networks.

**Advantages:**

* Rich set of powerful APIs to extend the Pytorch Libraries.
* It is flexible, faster, and provides optimizations.
* It has support for GPU and CPU.
* It supports cloud platforms.

**Limitations:**

* It has been released in 2016, so it’s new compared to others and has fewer users, and is not widely known.
* Absence of monitoring and visualization tools like a tensor board.
* The developer community is small compared to other frameworks

**Why PYTORCH?**

* Torch is an open-source machine learning library, a scientific computing framework, and a script language based on the Lua programming language. It provides a wide range of algorithms for deep learning.
* Pytorch is excellent at rapid prototyping.
* Strong support for GPUs as parallel programs can be implemented on multiple GPUs.
* Provides cleaner interface and is easier to use and learn.
* Facilitates the exchange of data with external libraries.

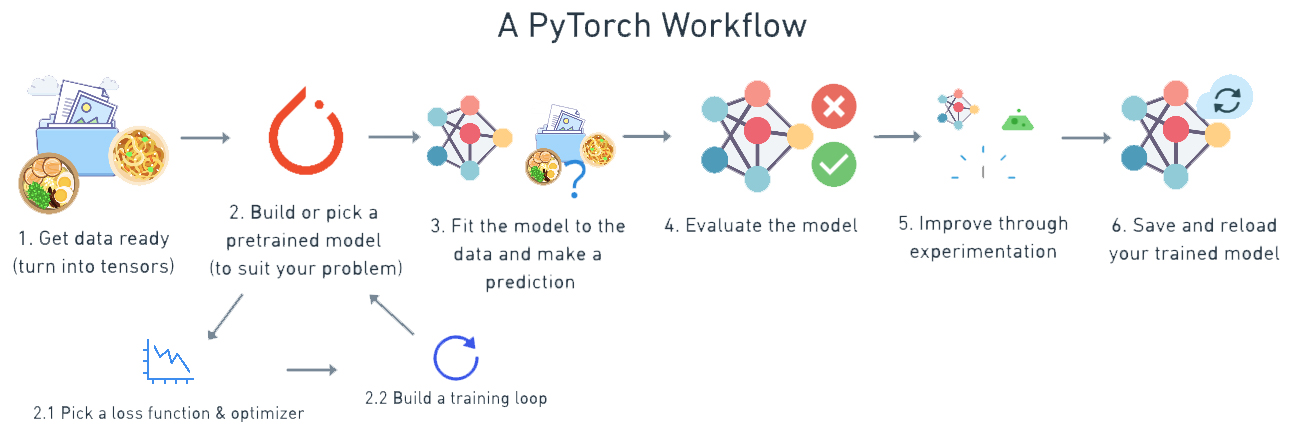


Figure 4.2: Pytorch Workflow

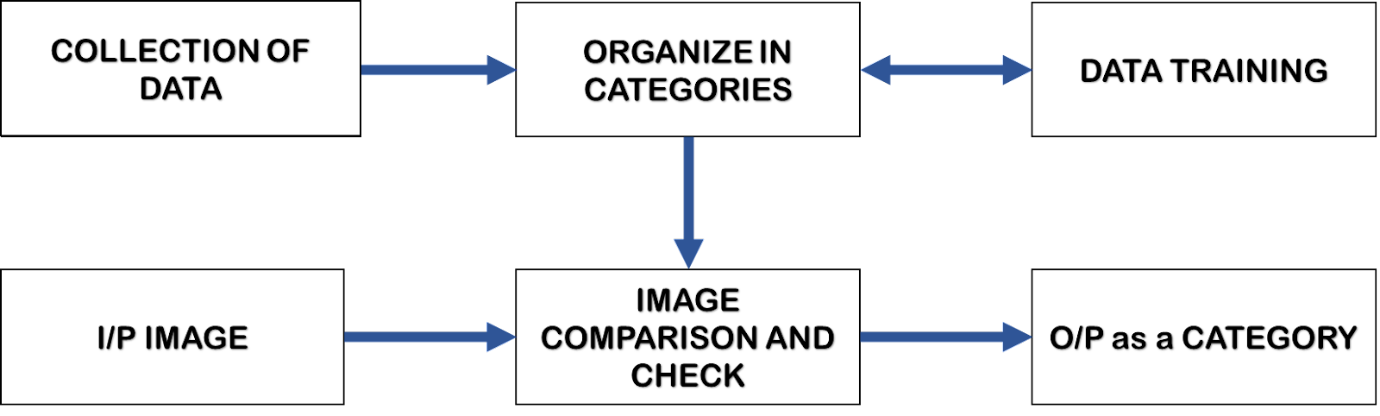


Figure 4.3: Project Methodology

* **COLLECTION OF DATA :**

Data and images of different species of cotton pests will be collected and stored in a database. Preventive Measures for each cotton pest will be assembled.

* **ORGANIZE IN CATEGORIES :**

All the gathered raw data will be processed and organized into different categories based upon type of species and management method.

* **DATA TRAINING :**

The test data will be fed to the model. The model then iterates over the data multiple times and automatically learns the most important features relevant to the images.

* **I/P IMAGE :**

The image of the pest from the user will be taken as input by the model.

* **IMAGE COMPARISON AND CHECK :**

The image of the pest obtained from the user will be compared with the image database using image recognition algorithms to identify the type of cotton pest.

* **O/P as CATEGORY :**

Based upon result from the software model, it’ll suggest the accurate and efficient method for prevention of specific cotton pest.

**What is Image recognition?**

Image recognition refers to technologies that identify places, logos, people, objects, buildings, and several other variables in digital images. It may be very easy for humans like you and me to recognise different images, such as images of animals. We can easily recognise the image of a cat and differentiate it from an image of a horse. But it may not be so simple for a computer.

A digital image is an image composed of picture elements, also known as pixels, each with finite, discrete quantities of numeric representation for its intensity or grey level. So the computer sees an image as numerical values of these pixels and in order to recognise a certain image, it has to recognise the patterns and regularities in this numerical data.

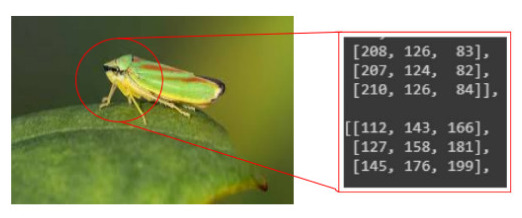


Figure 4.4: Image Recognition

**How does Image recognition work?**

Typically, the task of image recognition involves the [creation of a neural network](https://www.mygreatlearning.com/blog/types-of-neural-networks/) that processes the individual pixels of an image. These networks are fed with as many pre-labelled images as we can, in order to “teach” them how to recognize similar images.

So let me break the process for you in some simple steps:

We need a dataset containing images with their respective labels. For example, an image of a dog must be labelled as a dog or something that we can understand.

Next, these images are to be fed into a Neural Network and then trained on them. Usually, for the tasks concerned with images, we use [convolutional neural networks](https://www.mygreatlearning.com/blog/cnn-model-architectures-and-applications/). These networks consist of convolutional layers and pooling layers in addition to Multiperceptron layers (MLP). The working of convolutional and pooling layers are explained in the below.

We feed in the image that is not in the training set and get predictions.

**Working of Convolutional and Pooling layers**

Convolutional layers and Pooling layers are the major building blocks used in convolutional neural networks. Let us see them in detail

**How does Convolutional Layer work?**

The convolutional layer’s parameters consist of a set of learnable filters (or kernels), which have a small receptive field. These filters scan through image pixels and gather information in the batch of pictures/photos. Convolutional layers convolve the input and pass its result to the next layer. This is like the response of a neuron in the visual cortex to a specific stimulus.

**4.2 FRONT END**

**HTML (Hyper Text Markup Language)**

It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

**What is HTML?**

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content

**Features of HTML:**

* It is easy to learn and easy to use.
* It is platform-independent.
* Images, videos, and audio can be added to a web page.
* Hypertext can be added to the text.
* It is a markup language

**Advantages:**

* HTML is used to build websites.
* It is supported by all browsers.
* It can be integrated with other languages like CSS, JavaScript, etc

**CSS (Cascading Style Sheets)**

CSSis a stylesheet language used to design a webpage to make it attractive. The reason for using this is to simplify the process of making web pages presentable. It allows you to apply styles on web pages. More importantly, it enables you to do this independent of the HTML that makes up each web page.

**Why is CSS used in HTML?**

* **Solves a big problem:** Font, color, background style, element alignments, border, and size tags had to be duplicated on each web page before CSS. This was a lengthy procedure.
* **Saves a lot of time:** Because CSS style definitions are stored in external CSS files, updating only one file can modify the entire website.
* **Provide more attributes:** CSS gives more specific features for defining the look and feel of a website than simple HTML.
* **Pages load faster:** CSS doesn’t require you to write HTML tag attributes all of the time. A tag’s rule can be written once and applied to all instances of the tag. As a result, CSS uses less code, resulting in speedier download times.
* **Easier Website maintenance:** CSS makes website maintenance much easier. If we need to make a global change to the file, we can simply alter the style, which will update all of the elements on the web page.
* **Multiple device compatibility:** We can use CSS with older language versions because it is compatible with them. CSS makes it possible to optimize material for several devices.

**Advantages:**

* Maintenance: It is easy to maintain, changing in a single place will affect globally in your web site. No need to change every specific place.
* Time-saving: You can easily use any single CSS script at multiple places.
* Support: CSS is supported by all the browsers and search engines.
* Cache storing: CSS can store web applications locally with the help of offline cache so you can see the web site when you are offline.
* Native front-end: CSS contains a huge list of attributes and function that is helpful to design the HTML page.
* Selectors: In CSS, there are lots of selectors (ID selectors, Class Selectors, etc.) that will be helpful to perform specific tasks.

**JAVASCRIPT:**

**JavaScript** is a lightweight, cross-platform, and interpreted compiled programming language which is also known as the scripting language for webpages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for [**Client-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [**Server-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments. JavaScript contains a standard library of objects, like [**Array**](https://www.geeksforgeeks.org/arrays-in-javascript/), [**Date**](https://www.geeksforgeeks.org/javascript-date-objects/), and [**Math**](https://www.geeksforgeeks.org/javascript-math-object/), and a core set of language elements like **operators**, **control structures**, and **statements**.

**What is JavaScript used for?**

* Web pages with interactive elements: User interaction with web pages is enabled through JavaScript. On a web page, JavaScript has essentially no bounds.
* Developing online and mobile applications: For web and mobile app development, developers can employ a variety of JavaScript frameworks.
* Creating web servers and server applications: Aside from websites and apps, developers may use JavaScript to create simple web servers and Node.js to construct backend infrastructure.
* Game Development: Browser games can also be made using JavaScript. Beginning developers can use these to hone their JavaScript skills.

**Applications of JavaScript:**

* **Web Development:** Adding interactivity and behaviour to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.
* **Web Applications:** With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces (APIs) that provide extra power to the code. The Electron and React is helpful in this department.
* **Server Applications:** With the help of Node.js, JavaScript made its way from client to server and node.js is the most powerful on the server-side.
* **Art:** Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, make the sound more effective also can be used [p5.js](https://www.geeksforgeeks.org/p5-js-introduction/) library.
* **Machine Learning:** This JavaScript ml5.js library can be used in web development by using machine learning.

**4.3 DEPLOYMENT**

**Django**

It is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because Django provides built-in features for everything including Django Admin Interface, default database – SQLlite3, etc. When you’re building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use and that too for rapid development.

**Why Django?**

* Django is a rapid web development framework that can be used to develop fully fleshed web applications in a short period of time.
* Django is fully functional framework that requires nothing else.
* Excellent documentation and high scalability.
* Used by Top MNCs and Companies, such as Instagram, Disqus, Spotify, YouTube, Bitbucket, Dropbox, etc. and the list is never-ending.
* Easiest Framework to learn, rapid development and Batteries fully included.
* The last but not least reason to learn Django is [Python](https://www.geeksforgeeks.org/python-programming-language/), Python has huge library and features such as Web Scrapping, Machine Learning, Image Processing, Scientific Computing, etc. One can integrate it all this with web application and do lots and lots of advance stuff.

**Chapter 5**

**RESULTS / ANALYSIS**

A CNN-based image classifier is ready, and it gives 80% accuracy. As per the graph below, training loss decrease exponentially as the epochs increase. The losses are in line with each other as seen in the figure, which proves that the model is reliable and there is no underfitting or overfitting of the model.

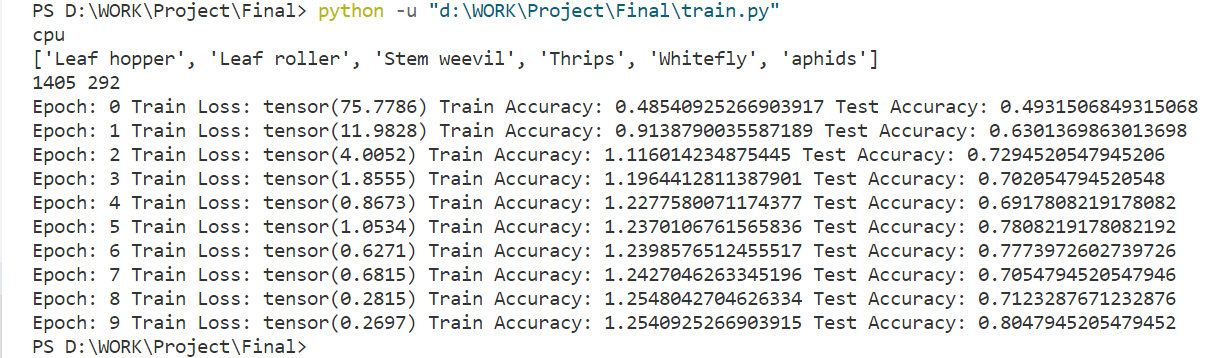


Figure 5.1: Accuracy Screenshot

Figure 5.2: Accuracy Graph

**Chapter 6**

**SOURCE CODE**

**Image Recognition code:**

from distutils.command.config import config

import enum

from fileinput import filename

from typing import final

from django.shortcuts import render,HttpResponse

from django.core.files.storage import FileSystemStorage

import torch

#PYTORCH

import torch.nn as nn

from torchvision.transforms import transforms

from torch.utils.data import DataLoader

import numpy as np

from torch.autograd import Variable

from torchvision.models import squeezenet1\_1

import torch.functional as F

from io import open

import os

from PIL import Image

import pathlib

import glob

import cv2

# Create your views here.

def home(request):

context={'a':1}

return render(request,"home.html")

def about(request):

return render(request,"about.html")

def predictImage(request):

print(request)

print(request.POST.dict())

fileObj=request.FILES['filePath']

fs=FileSystemStorage()

filePathName=fs.save(fileObj.name,fileObj)

filePathName=fs.url(filePathName)

context={'filePathName':filePathName}

train\_path = 'D:/WORK/Project/Final/train'

pred\_path = 'D:/WORK/Project/Final/media'

initial="."+filePathName

final='D:/WORK/Project/Final/media/111.jpg'

try:

os.rename(initial,final)

except FileExistsError:

os.remove(final)

os.rename(initial,final)

# categories

root = pathlib.Path(train\_path)

classes = sorted([j.name.split('/')[-1] for j in root.iterdir()])

# CNN Network

class ConvNet(nn.Module):

def \_\_init\_\_(self, num\_classes=6):

super(ConvNet, self).\_\_init\_\_()

# Output size after convolution filter

# ((w-f+2P)/s) +1

self.conv1 = nn.Conv2d(

in\_channels=3, out\_channels=12, kernel\_size=3, stride=1, padding=1)

self.bn1 = nn.BatchNorm2d(num\_features=12)

self.relu1 = nn.ReLU()

self.pool = nn.MaxPool2d(kernel\_size=2)

# Reduce the image size be factor 2

self.conv2 = nn.Conv2d(

in\_channels=12, out\_channels=20, kernel\_size=3, stride=1, padding=1)

self.relu2 = nn.ReLU()

self.conv3 = nn.Conv2d(

in\_channels=20, out\_channels=32, kernel\_size=3, stride=1, padding=1)

self.bn3 = nn.BatchNorm2d(num\_features=32)

self.relu3 = nn.ReLU()

self.fc = nn.Linear(in\_features=32\*125\*125, out\_features=num\_classes)

# Feed forwad function

def forward(self, input):

output = self.conv1(input)

output = self.bn1(output)

output = self.relu1(output)

output = self.pool(output)

output = self.conv2(output)

output = self.relu2(output)

output = self.conv3(output)

output = self.bn3(output)

output = self.relu3(output)

# Above output will be in matrix form, with shape (256,32,125\*125)

output = output.view(-1, 32\*125\*125)

output = self.fc(output)

return output

checkpoint = torch.load('D:/WORK/Project/Final/best\_checkpoint.model')

model = ConvNet(num\_classes=6)

model.load\_state\_dict(checkpoint)

model.eval()

# Transforms

transformer = transforms.Compose([

transforms.Resize((250,250)),

transforms.ToTensor(), # 0-255 to 0-1, numpy to tensors

transforms.Normalize([0.5, 0.5, 0.5], # 0-1 to [-1,1] , formula (x-mean)/std

[0.5, 0.5, 0.5])

])

# prediction function

def prediction(img\_path, transformer):

image = Image.open(img\_path)

image\_tensor = transformer(image).float()

image\_tensor = image\_tensor.unsqueeze\_(0)

if torch.cuda.is\_available():

image\_tensor.cuda()

input = Variable(image\_tensor)

output = model(input)

index = output.data.numpy().argmax()

pred = classes[index]

return pred

images\_path = glob.glob(pred\_path+'/\*.jpg')

pred\_dict=''

for i in images\_path:

pred\_dict = prediction(i, transformer)

context={ 'mynum': pred\_dict}

if pred\_dict=="aphids":

return render(request,"Aphids.html",context)

elif pred\_dict=="Thrips":

return render(request,"Thrips.html",context)

elif pred\_dict=="Leaf hopper":

return render(request,"LeafHopper.html",context)

elif pred\_dict=="Leaf roller":

return render(request,"LeafRoller.html",context)

elif pred\_dict=="Stem weevil":

return render(request,"StemWeevil.html",context)

elif pred\_dict=="Whitefly":

return render(request,"whitefly.html",context)

else:

return render(request,"default.html",context)

**Saving the Trained Model:**

# Load libraries

import os

import numpy as np

import torch

import glob

import torch.nn as nn

from torchvision.transforms import transforms

from torch.utils.data import DataLoader

from torch.optim import Adam

from torch.autograd import Variable

import torchvision

import pathlib

# checking for device

device = torch.device('cuda' if torch.cuda.is\_available() else 'cpu')

print(device)

# Transforms

transformer = transforms.Compose([

    transforms.Resize((250, 250)),

    transforms.RandomHorizontalFlip(),

    transforms.ToTensor(),  # 0-255 to 0-1, numpy to tensors

    transforms.Normalize([0.5, 0.5, 0.5],  # 0-1 to [-1,1] , formula (x-mean)/std

                         [0.5, 0.5, 0.5])

])

# Dataloader

# Path for training and testing directory

train\_path = 'D:/WORK/Project/Dataset2.0/train'

test\_path = 'D:/WORK/Project/Dataset2.0/test'

train\_loader = DataLoader(

    torchvision.datasets.ImageFolder(train\_path, transform=transformer),

    batch\_size=8, shuffle=True

)

test\_loader = DataLoader(

    torchvision.datasets.ImageFolder(test\_path, transform=transformer),

    batch\_size=4, shuffle=True

)

# categories

root = pathlib.Path(train\_path)

classes = sorted([j.name.split('/')[-1] for j in root.iterdir()])

print(classes)

# CNN Network

class ConvNet(nn.Module):

    def \_\_init\_\_(self, num\_classes=6):

        super(ConvNet, self).\_\_init\_\_()

        # Output size after convolution filter

        # ((w-f+2P)/s) +1

        self.conv1 = nn.Conv2d(

            in\_channels=3, out\_channels=12, kernel\_size=3, stride=1, padding=1)

        self.bn1 = nn.BatchNorm2d(num\_features=12)

        self.relu1 = nn.ReLU()

        self.pool = nn.MaxPool2d(kernel\_size=2)

        # Reduce the image size be factor 2

#....................................................................................................

        self.conv2 = nn.Conv2d(

            in\_channels=12, out\_channels=20, kernel\_size=3, stride=1, padding=1)

        self.relu2 = nn.ReLU()

#....................................................................................................

        self.conv3 = nn.Conv2d(

            in\_channels=20, out\_channels=32, kernel\_size=3, stride=1, padding=1)

        self.bn3 = nn.BatchNorm2d(num\_features=32)

        self.relu3 = nn.ReLU()

        self.fc = nn.Linear(in\_features=32\*125\*125, out\_features=num\_classes)

        # Feed forwad function

    def forward(self, input):

        output = self.conv1(input)

        output = self.bn1(output)

        output = self.relu1(output)

        output = self.pool(output)

        output = self.conv2(output)

        output = self.relu2(output)

        output = self.conv3(output)

        output = self.bn3(output)

        output = self.relu3(output)

        output = output.view(-1, 32\*125\*125)

        output = self.fc(output)

        return output

model = ConvNet(num\_classes=6).to(device)

# Optmizer and loss function

optimizer = Adam(model.parameters(), lr=0.001, weight\_decay=0.0001)

loss\_function = nn.CrossEntropyLoss()

num\_epochs = 10

# calculating the size of training and testing images

train\_count = len(glob.glob(train\_path+'/\*\*/\*.jpg'))

test\_count = len(glob.glob(test\_path+'/\*\*/\*.jpg'))

print(train\_count, test\_count)

# Model training and saving best model

best\_accuracy = 0.0

for epoch in range(num\_epochs):

    # Evaluation and training on training dataset

    model.train()

    train\_accuracy = 0.0

    train\_loss = 0.0

    for i, (images, labels) in enumerate(train\_loader):

        if torch.cuda.is\_available():

            images = Variable(images.cuda())

            labels = Variable(labels.cuda())

        optimizer.zero\_grad()

        outputs = model(images)

        loss = loss\_function(outputs, labels)

        loss.backward()

        optimizer.step()

        train\_loss += loss.cpu().data\*images.size(0)

        \_, prediction = torch.max(outputs.data, 1)

        train\_accuracy += int(torch.sum(prediction == labels.data))

    train\_accuracy = train\_accuracy/train\_count

    train\_loss = train\_loss/train\_count

    # Evaluation on testing dataset

    model.eval()

    test\_accuracy = 0.0

    for i, (images, labels) in enumerate(test\_loader):

        if torch.cuda.is\_available():

images = Variable(images.cuda())

            labels = Variable(labels.cuda())

        outputs = model(images)

        \_, prediction = torch.max(outputs.data, 1)

        test\_accuracy += int(torch.sum(prediction == labels.data))

    test\_accuracy = test\_accuracy/test\_count

    print('Epoch: '+str(epoch)+' Train Loss: '+str(train\_loss) +

          ' Train Accuracy: '+str(train\_accuracy)+' Test Accuracy: '+str(test\_accuracy))

    if test\_accuracy > best\_accuracy:

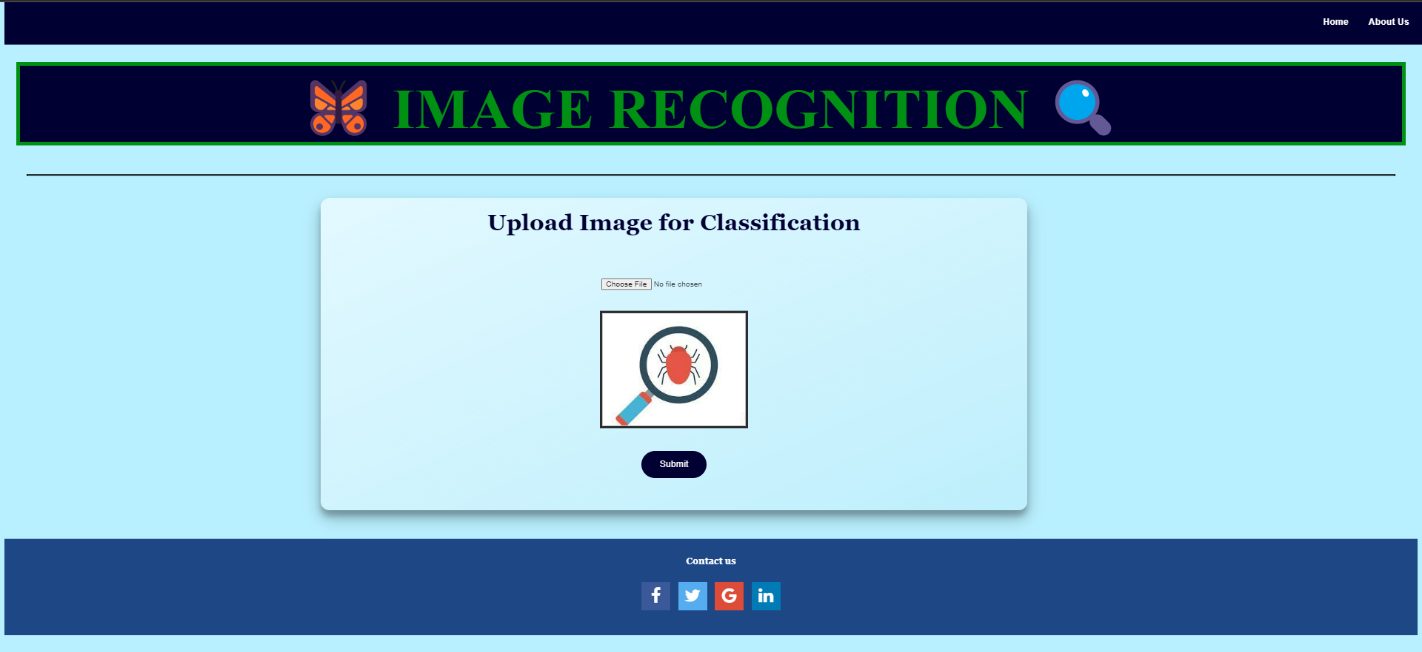
        torch.save(model.state\_dict(), 'best\_checkpoint.model')

        best\_accuracy = test\_accuracy

**Chapter 7**

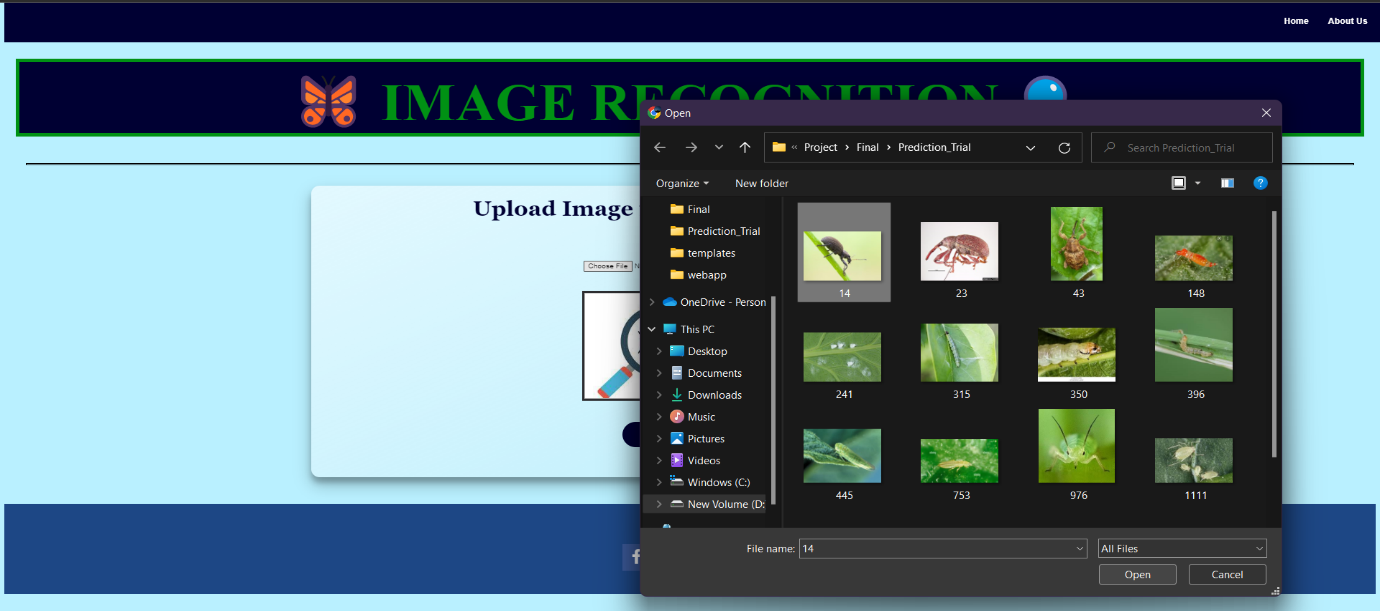
**SCREENSHOTS**

Home Page:



Click here to choose an image

Figure 7.1: Home Page



Browse the image from your device for prediction

Figure 7.2: Browse Image

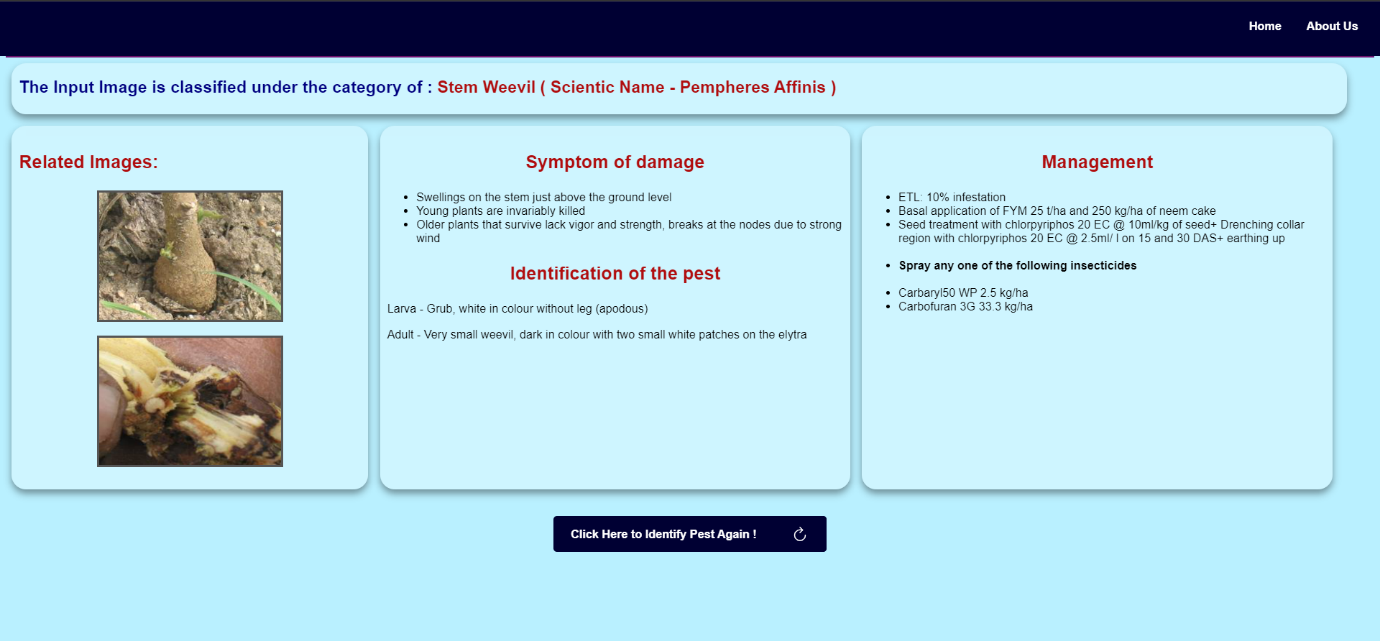


Submit the image to get prediction

After selecting image, it comes into view

Figure 7.3: Submit the Image

Prediction:



Click here to identify another pest

Figure 7.4: Prediction Page

**Chapter 8**

**CONCLUSION**

We have learnt about image processing, image recognition techniques and web designing.

This project has paved a way for problem identification, its management and providing a reliable solution for the same.

* 1. **REFERENCES**

1. [M Manoj krishna](https://www.researchgate.net/scientific-contributions/M-Manoj-krishna-2142507350), [M Neelima](https://www.researchgate.net/scientific-contributions/M-Neelima-2142496936), [Harshali Mane](https://www.researchgate.net/profile/Harshali-Mane)( [K L University](https://www.researchgate.net/institution/K_L_University)[Venu), Gopala Rao Matcha](https://www.researchgate.net/profile/Venu-Matcha)([K L University](https://www.researchgate.net/institution/K_L_University)) – “Image classification using Deep learning”(March 2018)
2. Z. Yan, V. Jagadeesh, D. Decoste, et al., HD-CNN: hierarchical deep convolutional neural network for image classification. Eprint Arxiv 4321-4329 (2014).
3. C. Zhang, X. Pan, H. Li, et al., A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. Isprs Journal of Photogrammetry & Remote Sensing 140, 133–144 (2018).
4. S. Roychowdhury, J. Ren, Non-deep CNN for multi-modal image classification and feature learning: an azure-based model (IEEE international conference on big data. IEEE, Washington, D.C., 2017), pp. 2893–2812.
5. The Research of Intelligent Image Recognition Technology Based on Neural Network LI Huanliang1,a1 Yishui Campus of Linyi University, Yishui 276400, China
6. Adam Paszke ,[Sam Gross](https://www.researchgate.net/scientific-contributions/Sam-Gross-2108176720) ,[Francisco Massa](https://www.researchgate.net/scientific-contributions/Francisco-Massa-2166247441) ,[Adam Lerer](https://www.researchgate.net/profile/Adam-Lerer)([Meta](https://www.researchgate.net/institution/Meta)) - PyTorch: An Imperative Style, High-Performance Deep Learning Library (December 2019)
7. <https://www.geeksforgeeks.org/image-classifier-using-cnn/>
8. <https://www.analyticsvidhya.com/blog/2021/06/image-classification-using-convolutional-neural-network-with-python/>
9. <https://docs.microsoft.com/en-us/windows/ai/windows-ml/tutorials/pytorch-train-model>
10. <https://www.pluralsight.com/guides/image-classification-with-pytorch>
11. <https://github.com/bentrevett/pytorch-image-classification>
12. <https://medium.com/bitgrit-data-science-publication/building-an-image-classification-model-with-pytorch-from-scratch-f10452073212>
13. <https://www.pluralsight.com/guides/image-classification-with-pytorch>
14. <https://marutitech.com/top-8-deep-learning-frameworks/>

**Chapter 9**

**WEEKLY OVERVIEW**

|  |  |  |  |
| --- | --- | --- | --- |
| **1st WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 03/03/22 | Thursday | Project Problem Discussion |
| 04/03/22 | Friday | Research on Problem |
| 05/03/22 | Saturday | Collecting the Data |
| 06/03/22 | Sunday | Research in depth |
| 07/03/22 | Monday | Collecting data in soft copy |
| 08/03/22 | Tuesday | Meeting on data |
|  | 09/03/22 | Wednesday | Creating a folder in Google drive for storing the data |

|  |  |  |  |
| --- | --- | --- | --- |
| **2nd WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 10/03/22 | Thursday | Project Discussion with guide |
| 11/03/22 | Friday | Listing out the cotton pest |
| 12/03/22 | Saturday | Uploading the remaining images |
| 13/03/22 | Sunday | Checking on the duplicate images |
| 14/03/22 | Monday | Remainder for the team members to upload the images |
| 15/03/22 | Tuesday | Checking all folders |

|  |  |  |  |
| --- | --- | --- | --- |
| **3rd WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 17/03/22 | Thursday | Update Data |
| 18/03/22 | Friday | Tasks distributed within the team members |
| 19/03/22 | Saturday | Searching more images |
| 20/03/22 | Sunday | Removing duplicate images |
| 21/03/22 | Monday | Check all the acquired data |
| 22/03/22 | Tuesday | Meeting held on data preparation |
|  | 23/03/22 | Wednesday | Finalized the data |

|  |  |  |  |
| --- | --- | --- | --- |
| **4th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 24/03/22 | Thursday | Review Meeting held on data with guide |
| 25/03/22 | Friday | Division of Team |
| 26/03/22 | Saturday | Listing out algorithm for project |
| 27/03/22 | Sunday | Decided on the 4 Algorithms |
| 28/03/22 | Monday | Study on Algorithm |
| 29/03/22 | Tuesday | Referring websites for in-depth knowledge about algorithm |
|  | 30/03/22 | Wednesday | Presentation Work |

|  |  |  |  |
| --- | --- | --- | --- |
| **5th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 31/03/22 | Thursday | Meeting for Presentation work |
| 01/04/22 | Friday | Presenting PPT on project problem statement |
| 02/04/22 | Saturday | Uploaded the Research Paper |
| 03/04/22 | Sunday | Overall information about algorithm |
| 04/04/22 | Monday | Referring YouTube videos about the algorithm |
| 05/04/22 | Tuesday | Team meeting on algorithm |
|  | 06/04/22 | Wednesday | Working on Tools presentation |

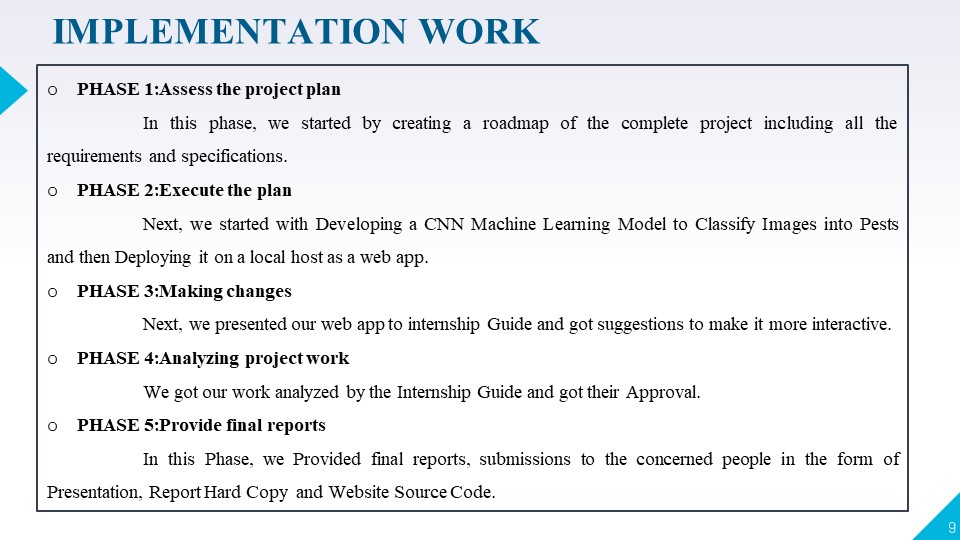
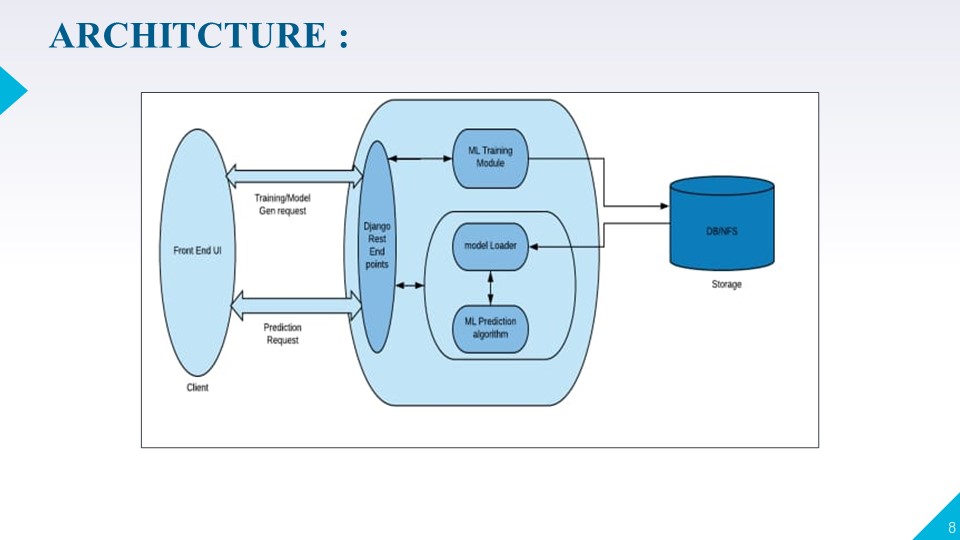
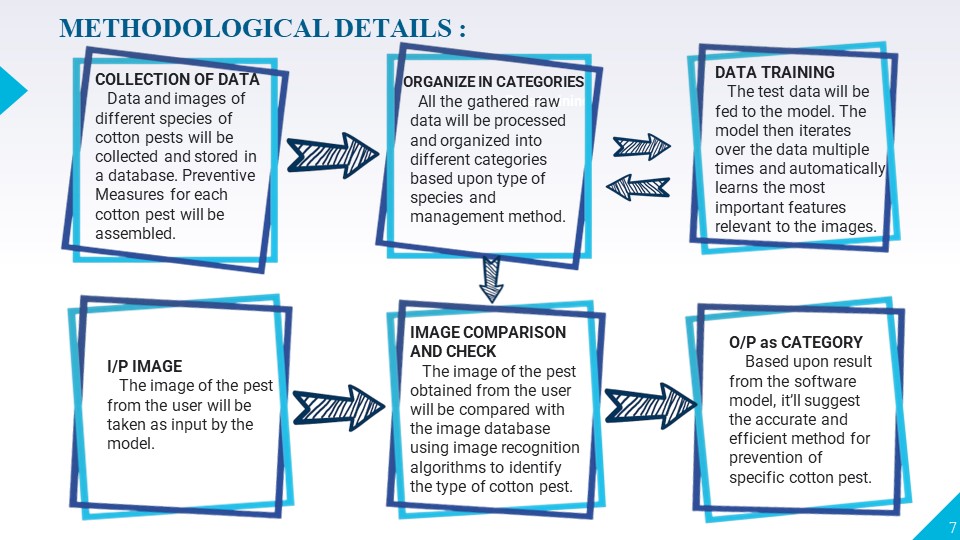
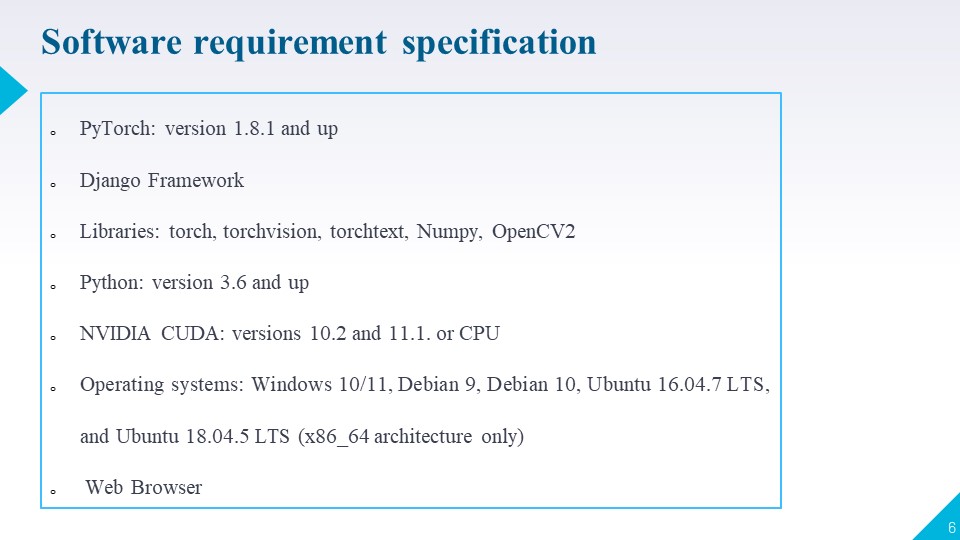
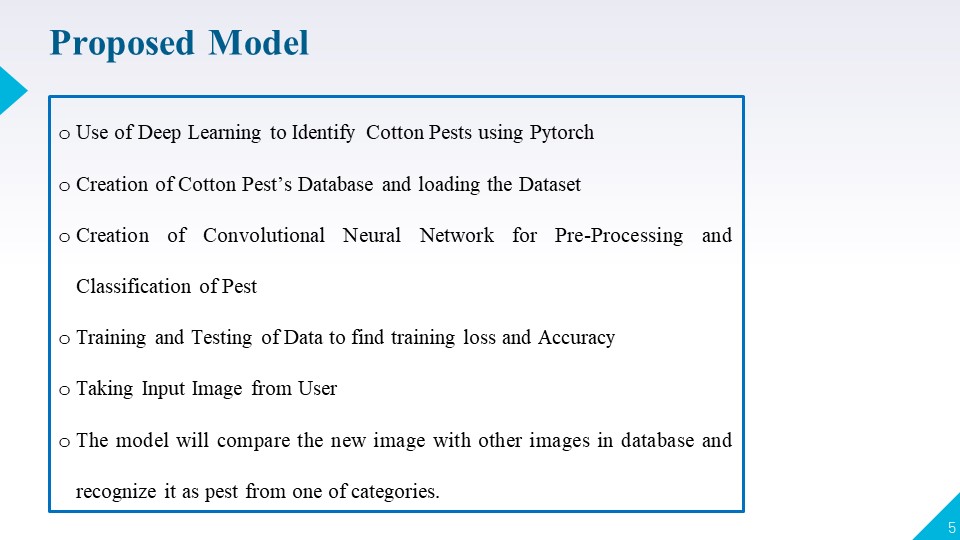
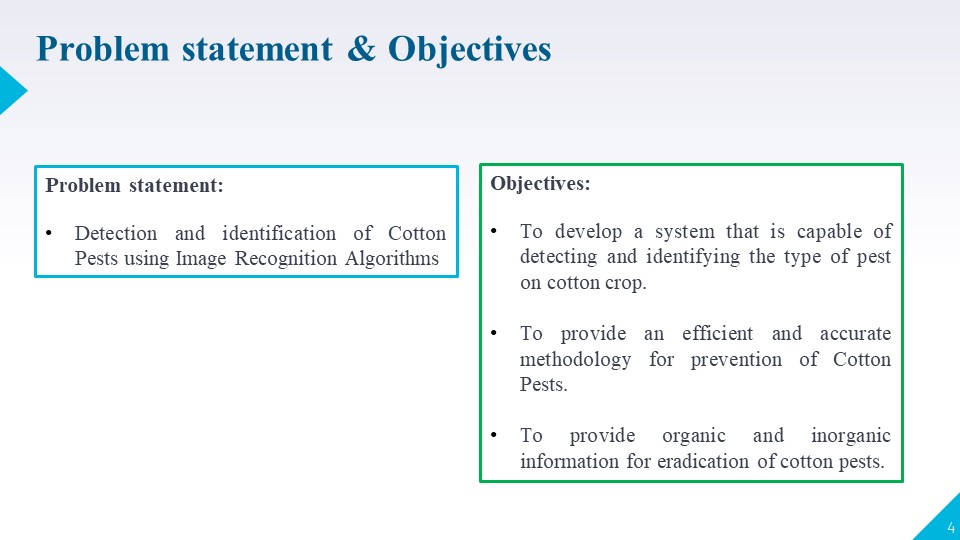
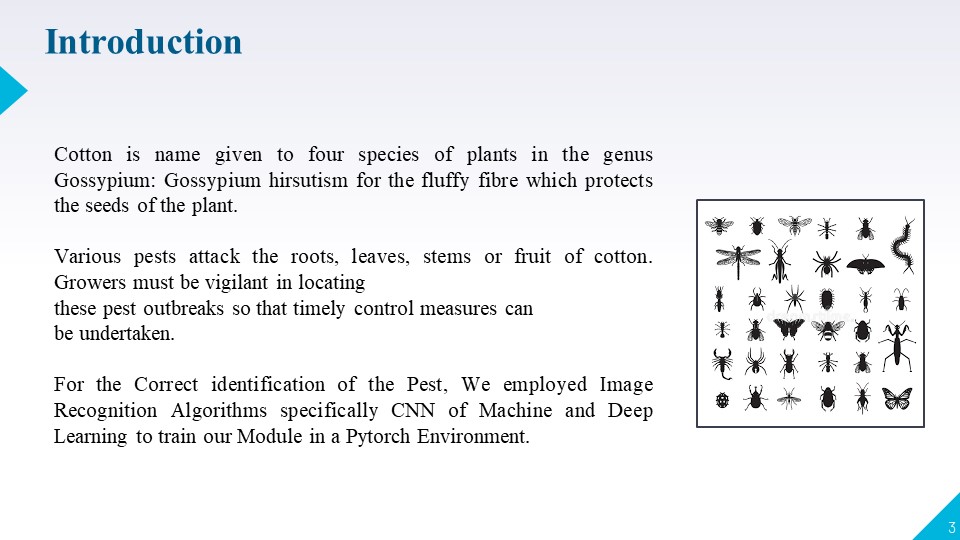
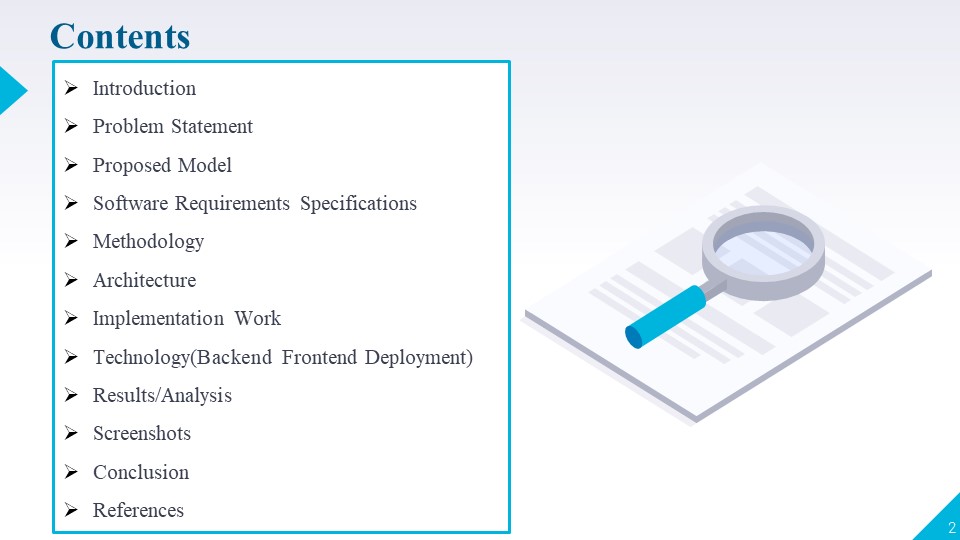
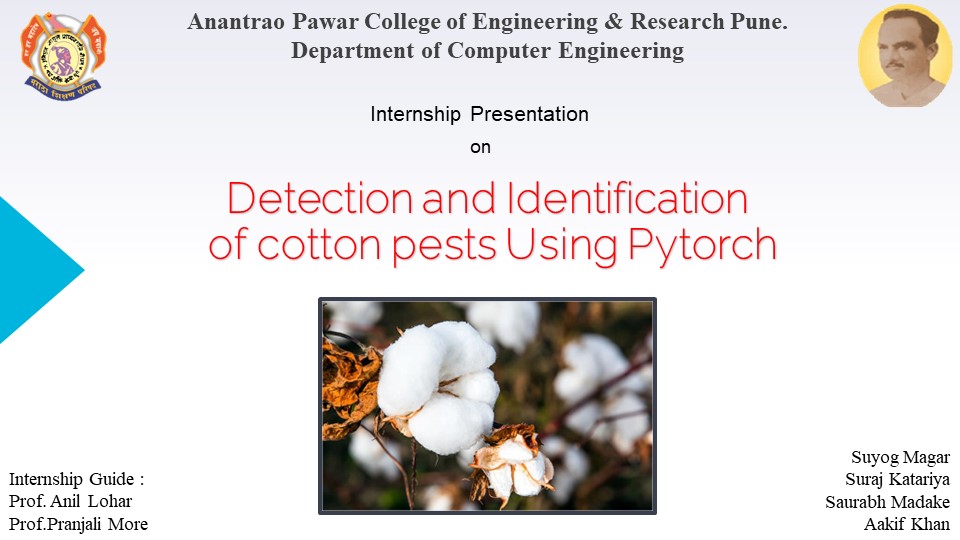
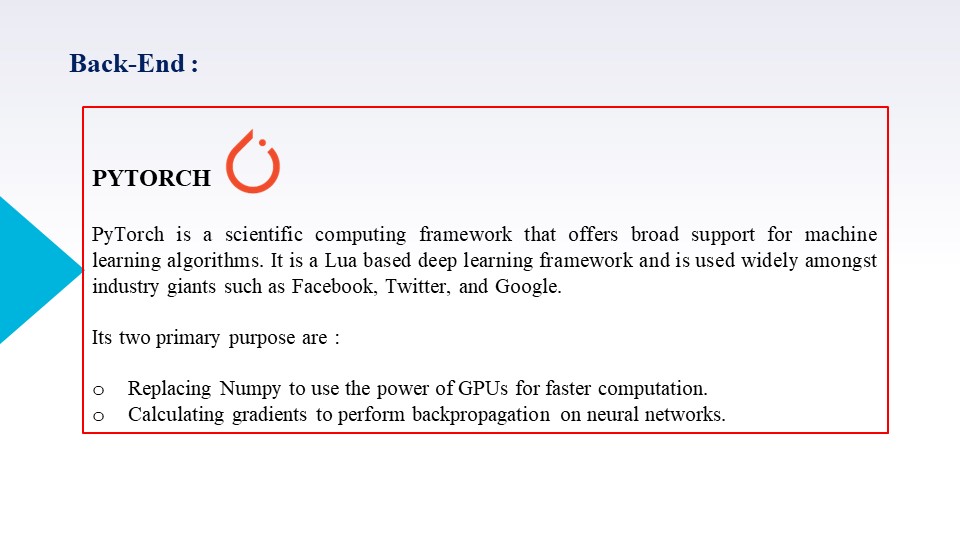
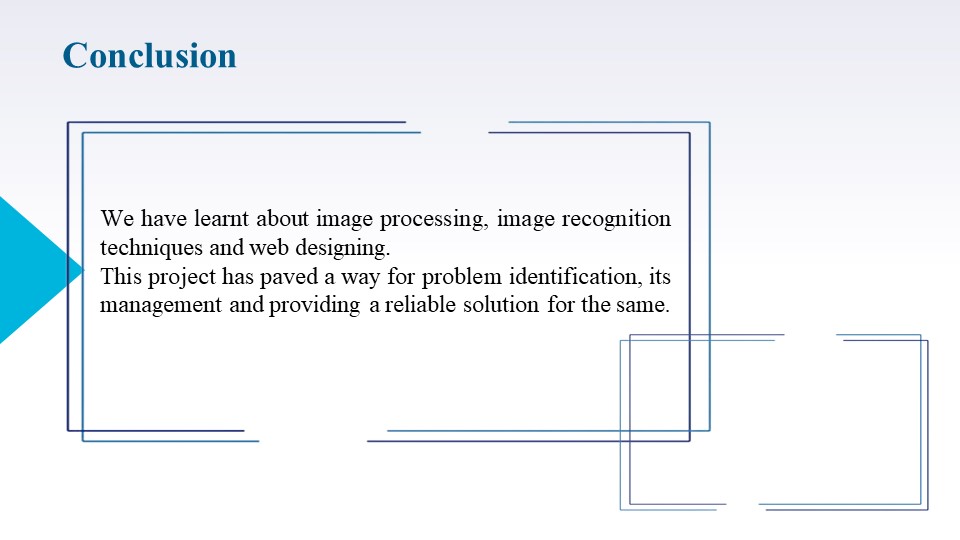
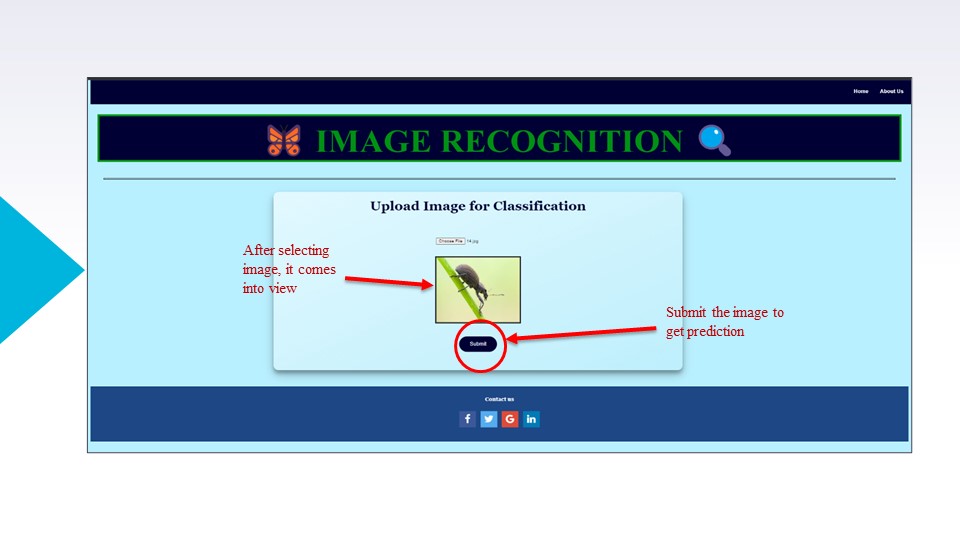
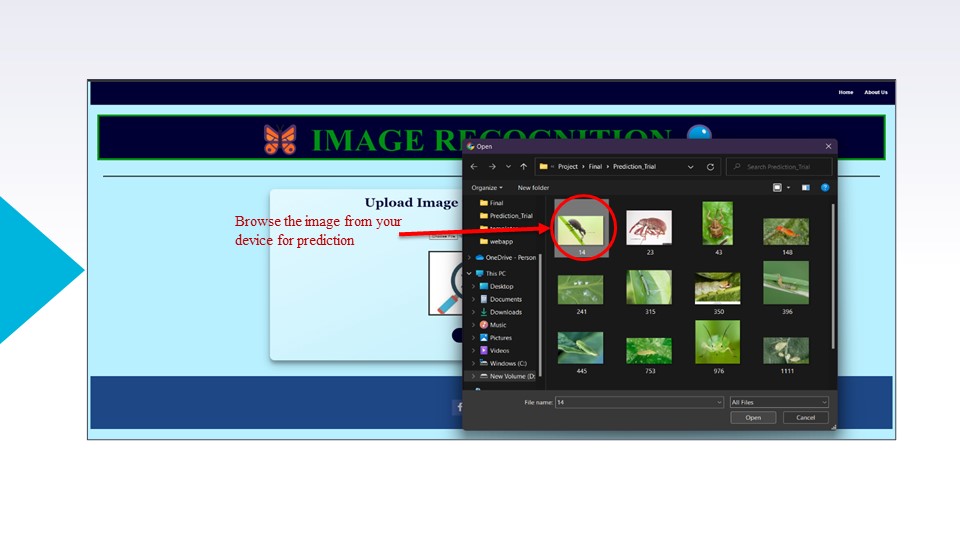
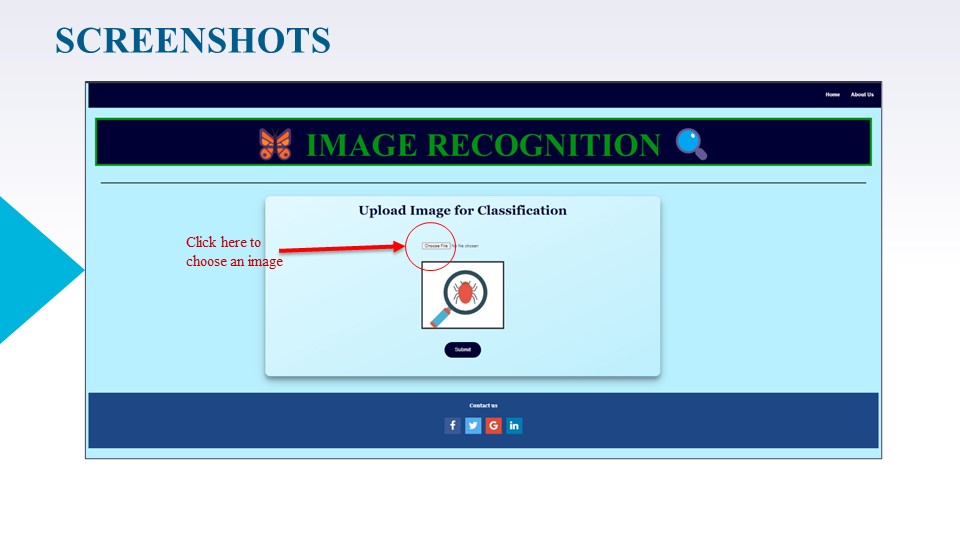
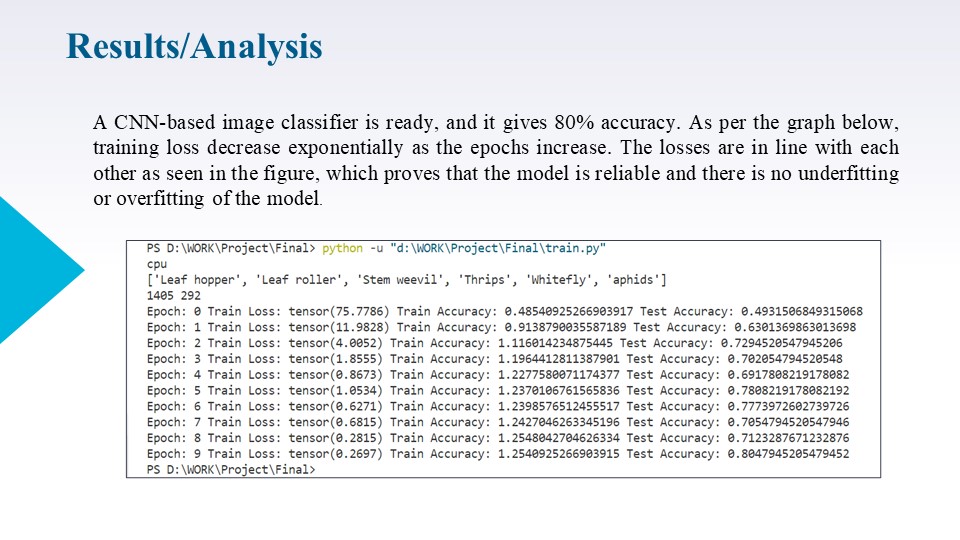
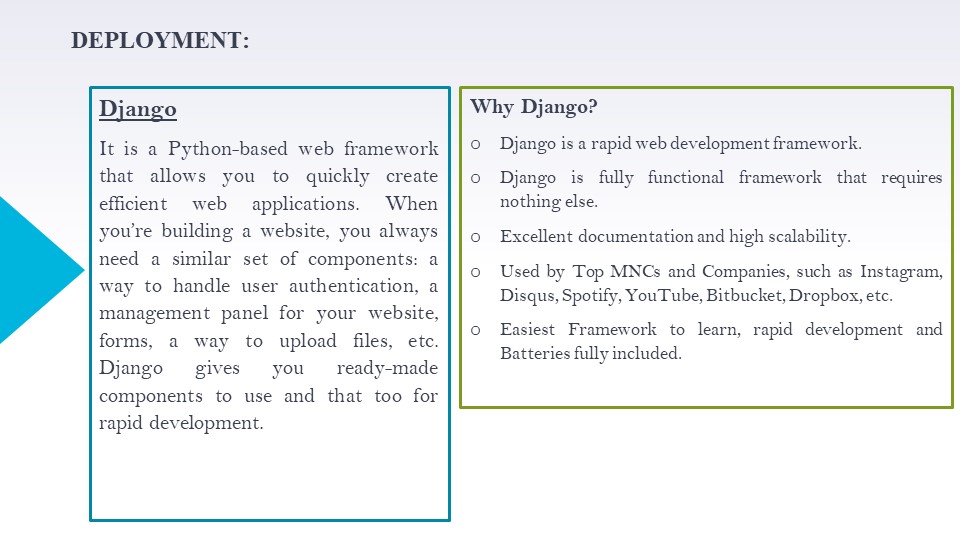
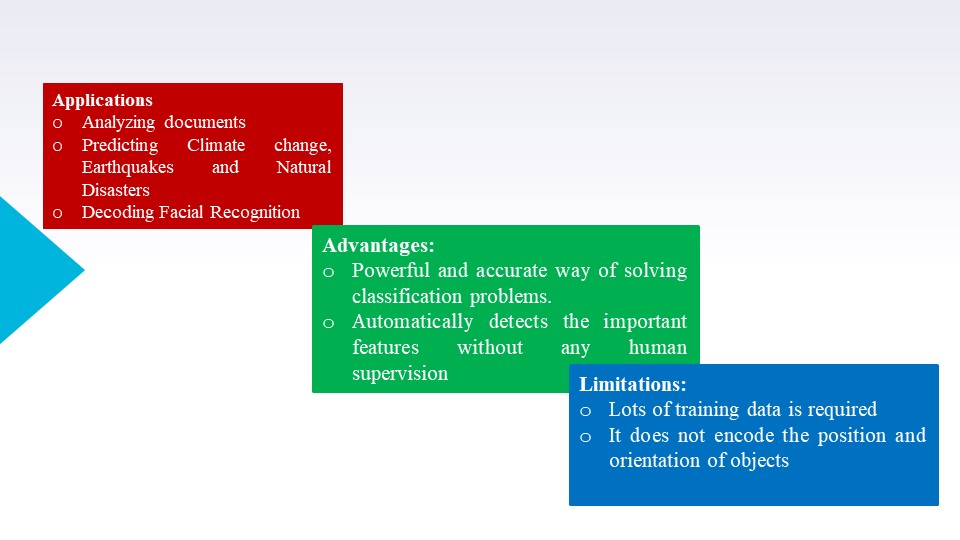
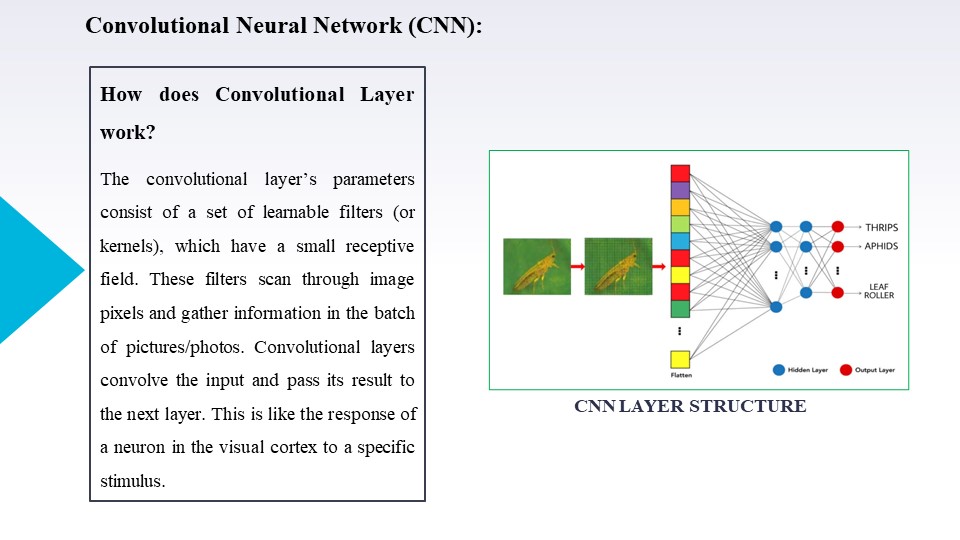
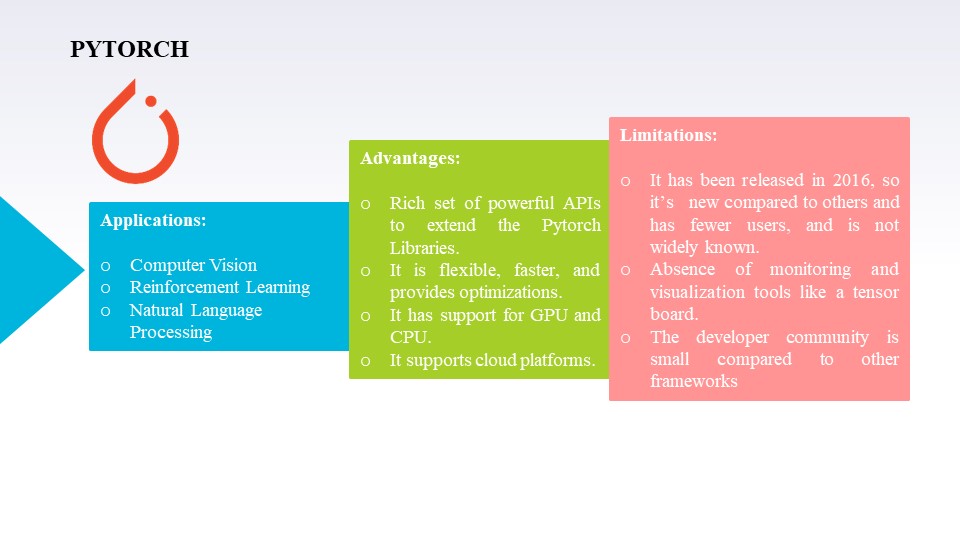
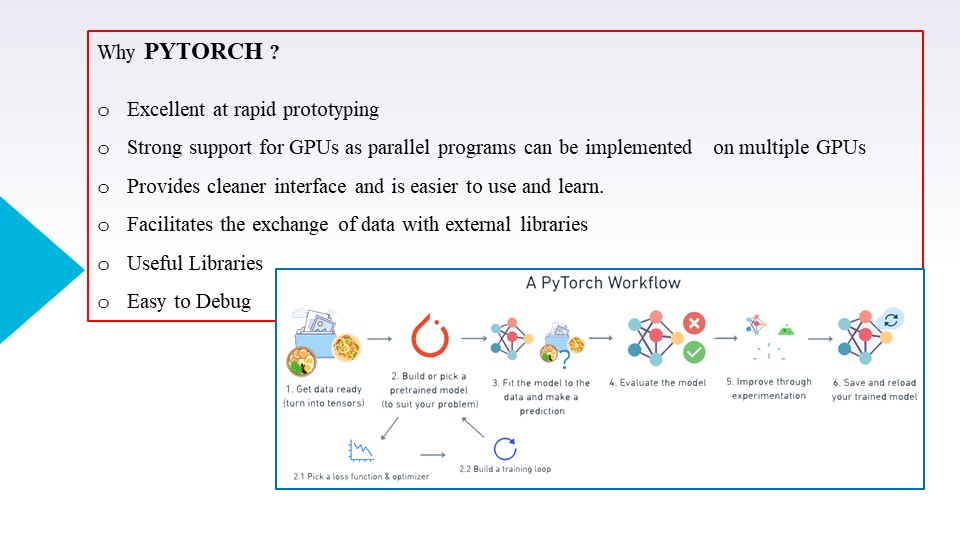
|  |  |  |  |
| --- | --- | --- | --- |
| **6th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 07/04/22 | Thursday | Collecting hardcopy of Research paper |
| 08/04/22 | Friday | Review meeting held on Algorithm & Research paper |
| 09/04/22 | Saturday | Presentation on tools techniques |
| 10/04/22 | Sunday | Changes in tools PPT |
| 11/04/22 | Monday | Finalizing tool PPT |
| 12/04/22 | Tuesday | Meeting held on tool PPT with guide |
|  | 13/04/22 | Wednesday | Started in depth learning study |

|  |  |  |  |
| --- | --- | --- | --- |
| **7th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 14/04/22 | Thursday | Team discussion on the Algorithm |
| 15/04/22 | Friday | Finalizing CNN Algorithm |
| 16/04/22 | Saturday | Collecting all information about CNN |
| 17/04/22 | Sunday | Working of CNN is properly understood |
| 18/04/22 | Monday | Discussion on Django Framework |
| 19/04/22 | Tuesday | Features of Django Framework |
|  | 20/04/22 | Wednesday | Facilities provided by Django Framework |

|  |  |  |  |
| --- | --- | --- | --- |
| **8th WEEK** | **DATE** | **DAY** | **NAME OF THE TOPIC/MODULE COMPLETED** |
| 21/03/22 | Thursday | Listing out all Library for project |
| 22/04/22 | Friday | Installing the libraries in the respective system |
| 23/04/22 | Saturday | Discussion on Pytorch |
| 24/04/22 | Sunday | Features of Pytorch |
| 25/04/22 | Monday | Facilities of Pytorch |
| 26/04/22 | Tuesday | Team meeting held regarding Tools |
|  | 27/04/22 | Wednesday | Discussion about the Tools |

**Chapter 10**

**PRESENTATION**

********