A

Major Project

On

BLIGHT DIAGNOSIS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

Suri Dheeraj Sai	(177R1A05B2)
Alex Abrahim	(177R1A0561)
Ritwik Bojja	(177R1A0568)

Under the Guidance of

G.ASHWINI KUMARI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2017-21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**BLIGHT DIAGNOSIS**" being submitted by **SURI DHEERAJ SAI (177R1A05B2), ALEX ABRAHIM(177R1A0561) & BOJJA RITWIK (177R1A0568)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2020-21.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

INTERNAL GUIDE
Mrs. G. Ashwini Kumari
Assistant Professor

DIRECTOR
Dr.A. Raji Reddy

HOD
Dr. K. Srujan Raju

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ______________

ACKNOWLEGDEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide.

Mrs. G. Ashwini Kumari, Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane and Mr. B. Ramji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

SURI DHEERAJ SAI (177R1A05B2)

ALEX ABRAHIM(177R1A0561)

BOJJA RITWIK(177R1A0568)

ABSTRACT

The early detection of diseases is important as it makes social, ecological and economical impacts on agriculture .Automatic methods for classification of crop diseases also help taking action after detecting the symptoms of crop diseases as it plays a vital role interms of ruralemployment and GDP of the country. The objective of this work is to develop a model which can predict those diseases so that the farmers can take an appropriate action .This work can be done using a Convolution Neural Networks model using a dataset with a 10 fold cross-validation technique ,to identify common crop diseases . With this model we overcome the problem of the crop diseases.

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
Figure 3.1	Project Architecture	6
Figure 3.2	Use case diagram	7
Figure 3.3	Sequence diagram	8
Figure 3.4	Activity Diagram	9

LIST OF SCREENSHOTS

SCREENSHOT NO.	SCREENSHOT NAME	PAGE NO.
Screenshot 5.1	Main Page	16
Screenshot 5.2	Disease Selection	17
Screenshot 5.3	History	18
Screenshot 5.4	Disease Prediction	19

TABLE OF CONTENTS

ABSTRACT	i
LIST OF FIGURES	ii
LIST OF SCREENSHOTS	iii
1. INTRODUCTION	**1**
1.1 PROJECT SCOPE	1
1.2 PROJECT PURPOSE	1
1.3 PROJECT FEATURES	1
2. SYSTEM ANALYSIS	**2**
2.1 PROBLEM DEFINITION	2
2.2 EXISTING SYSTEM	2
2.2.1 LIMITATIONS OF THE EXISTING SYSTEM	3
2.3 PROPOSED SYSTEM	3
2.3.1 ADVANTAGES OF PROPOSED SYSTEM	3
2.4 FEASIBILITY STUDY	4
2.4.1 ECONOMIC FESIBILITY	4
2.4.2 TECHNICAL FEASIBILITY	4
2.4.3 SOCIAL FEASIBILITY	4
2.5 HARDWARE & SOFTWARE REQUIREMENTS	5
2.5.1 HARDWARE REQUIREMENTS	5
2.5.2 SOFTWARE REQUIREMENTS	5
3. ARCHITECTURE	**6**
3.1 PROJECT ARCHITECTURE	6
3.2 DESCRIPTION	6
3.3 USECASE DIAGRAM	7
3.4 SEQUENCE DIAGRAM	8
3.5 ACTIVITY DIAGRAM	9
4. IMPLEMENTATION	**10**
4.1 SAMPLE CODE	10
5. SCREENSHOTS	**16**
6. TESTING	**20**

6.1 INTRODUCTION TO TESTING 20
6.2 TYPES OF TESTING 21
6.2.1 UNIT TESTING 21
6.2.2 INTEGRATION TESTING 21
6.2.3 FUNCTIONAL TESTING 22
6.3 TEST CASES 22
6.3.1 UPLOADING IMAGES 22
6.3.2 CLASSIFICATION 22
7. CONCLUSION & FUTURE SCOPE 23
7.1 PROJECT CONCLUSION 23
7.2 FUTURE SCOPE 23
7.3 GITHUB LINK 24
8. REFERENCES 25
8.1 REFERENCES 25

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as "Blight Diagnosis". This software provides facility to upload the images and get to know the disease on the crop. This project uses machine-learning methods and computer vision with android integration to identify disease from pictures. First, we use a convolutional neural networks to classify crop image points for each image. We then compare a number of classification algorithms that use certain features to predict the crop disease shown in the image.

1.2 PROJECT PURPOSE

This has been developed to facilitate the identification, retrieval of the items and information. System is built with manually exclusive features. In all cases system will specify object which are physical or on performance characteristics. They are used to give optimal distraction and other information. Data are used for identifying, accessing, storing and matching records. The data ensures that only one value of the code with a single meaning is correctly applied to give entity or attribute as described in various ways.

1.3 PROJECT FEATURES

The main features of this project are that the designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

.For detecting the crop diseases, the conventional methods are human vision based approaches. In these cases seeking the expert advice is time consuming and very expensive. The human vision based methods suffer many drawbacks. The accuracy and precision of human vision approach is dependent on the eyesight of the person or expert hired. Machine learning based method enables to identify the types of diseases, make the right decision and to select proper treatment. One of the advantages of using machine learning based method is that it performs tasks more consistently than human experts. Therefore, to overcome the drawbacks of conventional methods there is a need for a new machine learning based classification approach

2.2 EXISTING SYSTEM

There is a conventional method which is human vision based approach which is time consuming.Testing the crop takes lot of time .But detecting the disease on the crop is not 100% accurate .If exact diseases are not detected and required actions are not taken than it may lead to other issues which may destroy not only the crop but the crops surrounding it.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Much time is being consumed
- The results have low accuracy
- Traditional Approaches are used
- Includes high cost

2.3 PROPOSED SYSTEM

As Machine Learning is now considered to be the biggest innovation it can be used to solve the problem of farmers in real world .Machine learning help us live happier, healthier and more productive lives. The main idea is to identify the crop diseases using CNN model.The entire process is partitioned into different stages: beginning with the preparation ofa novel training dataset,development of a novel CNN model,deep feature extraction for training the model and finally ,classification of the crop diseases .

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Performs the task consistently.
- Provides better accuracy.
- Simple and easy to use .
- Embracing sophisticated technologies is cost-effective & a viable choice for farmers

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

2.3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.3.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following aresome hardware requirements.

- Processor : Snapdragon 600
- Ram Capacity : 1gb
- Storage : 100mb

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface andof the system. The following are some software requirements,

- Operating System : Android 8
- Programming Language : Python 3,Kotlin,Java
- Library : Tensorflow lite ,OpenCV,keras

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCITECTURE

This project architecture shows the procedure followed for crop disease detection using machine learning along with android integration, starting from input to final prediction.

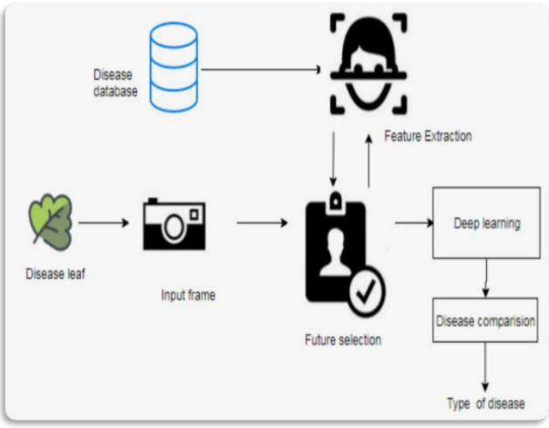


Figure 3.1: Project Architecture of Blight Diagnosis

3.2 DESCRIPTION

Image Acquisition: In this phase, images of plant leaves are gathered using digital media like camera, mobile phones etc. with desired resolution and size. The formation of database of images is completely dependent on the application system developer.

Image Segmentation: This phase aims at simplifying the representation of an image such that it becomes more meaningful and easier to analyze.

Feature Extraction: In this step the features from this area of interest need to be extracted. These features are needed to determine the meaning of a sample image.Features can be based on colour, shape, and texture. Recently, most of the researchers are intending to use texture features for detection of plant diseases.

Training and test data: Training data is passed to train the model. Test data is used to test the trained model whether it is making correct predictions or not.

Classification : The classification phase implies to determine if the input image is healthy or diseased. If the image is found to be diseased, some existing works have further classified it into a number of diseases.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the farmer and the app. The farmer takes the picture and upload it along with the input Id. Whereas the app does the preprocessing ,segmentation,detects the effected area and then extracts the features ,scans it and gives the remedy.

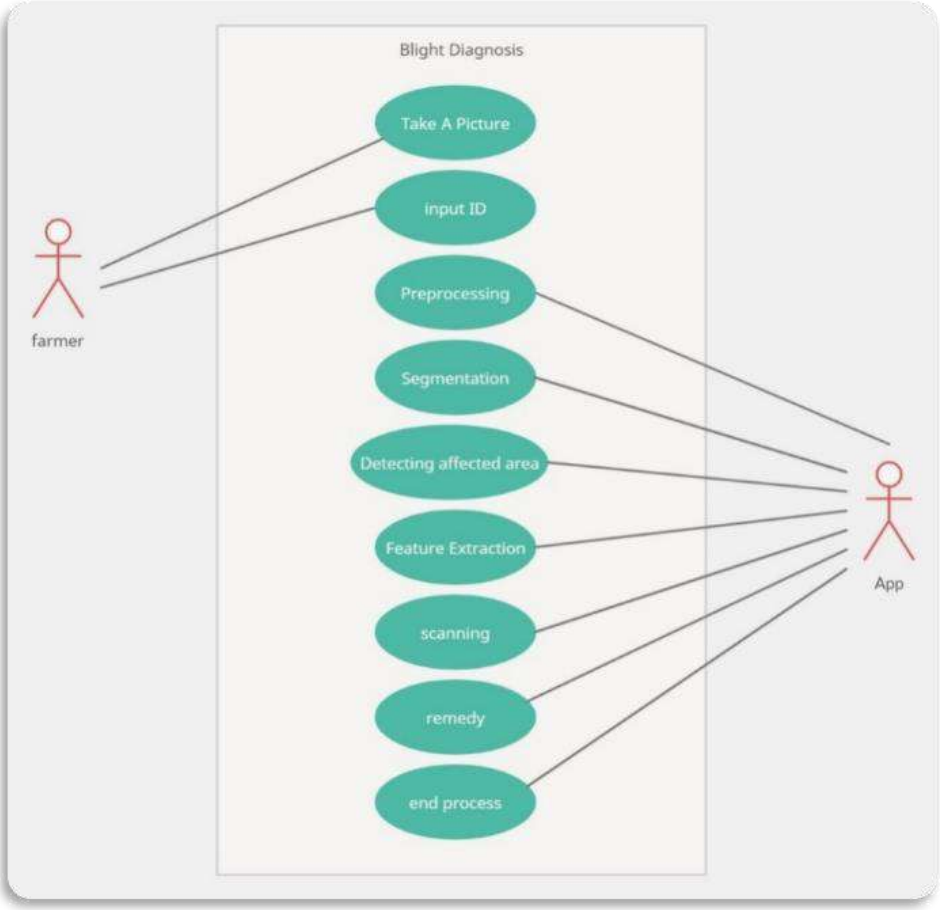


Figure 3.2: Use Case Diagram for user Crop Disease Detection

3.4 SEQUENCE DIAGRAM

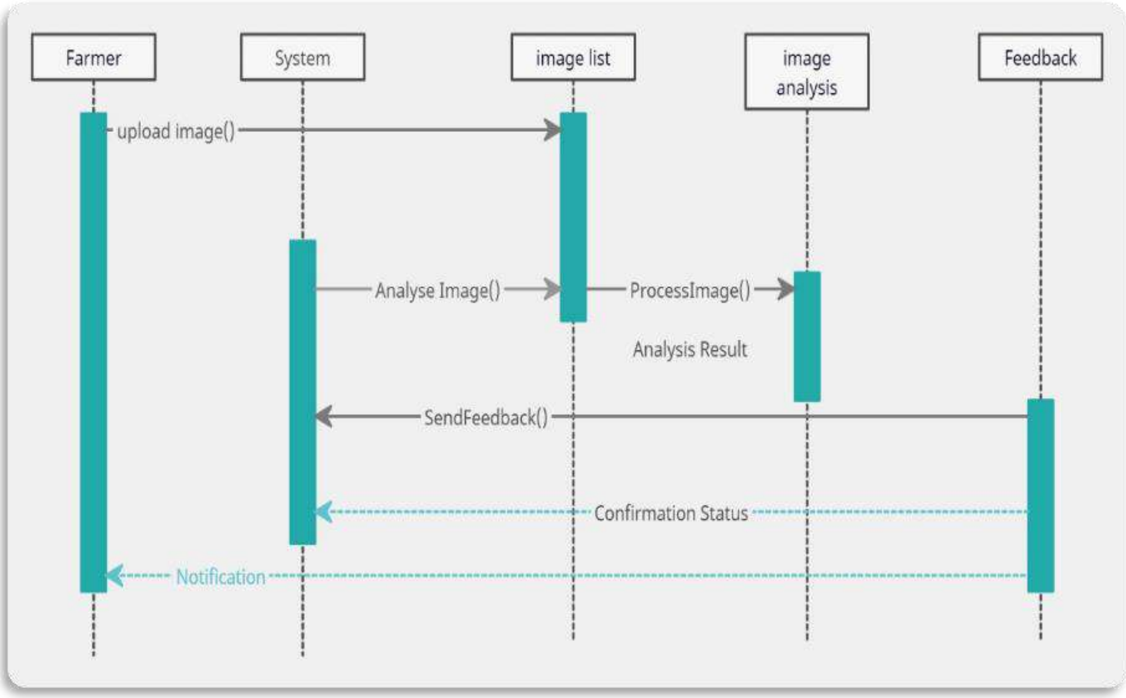


Figure 3.3: Sequence Diagram for Crop Disease Detection

3.4 ACTIVITY DIAGRAM

It describes about flow of activity states.

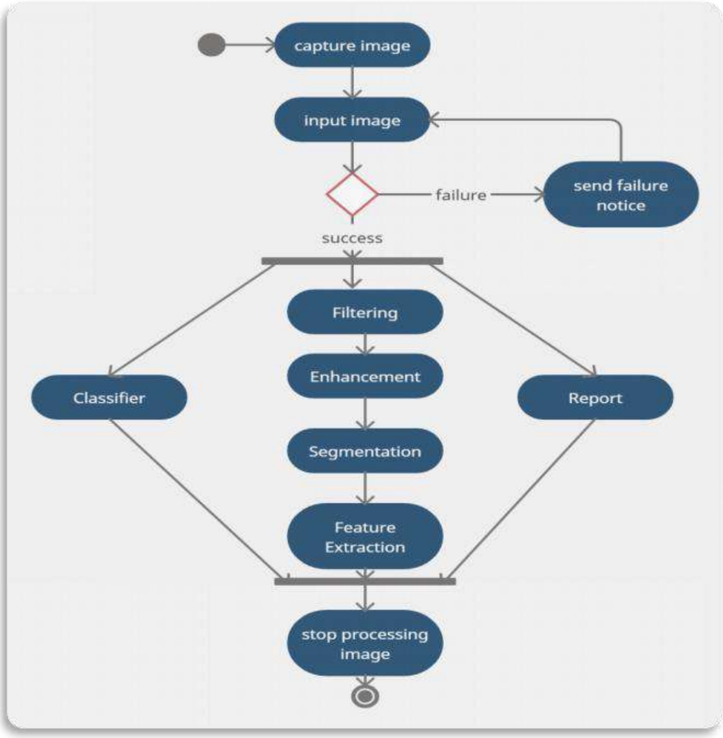


Figure 3.5: Activity Diagram for Crop Disease Detection

4.IMPLEMENTATION

4.IMPLEMENTATION

4.1 SAMPLE CODE

```
import numpy as np
import pickle
import cv2 from os
import listdir from sklearn.preprocessing
import LabelBinarizerfrom keras.models
import Sequentialfrom keras.layers.normalization
import BatchNormalizationfrom keras.layers.convolutional
import Conv2Dfrom keras.layers.convolutional
import MaxPooling2Dfrom keras.layers.core
import Activation, Flatten, Dropout, Densefrom keras
import backend as Kfrom keras.preprocessing.image
import ImageDataGeneratorfrom keras.optimizers
import Adamfrom keras.preprocessing
import imagefrom keras.preprocessing.image
import img_to_arrayfrom sklearn.preprocessing
import MultiLabelBinarizerfrom sklearn.model_selection
import train_test_split
import matplotlib.pyplot as plt

EPOCHS = 25
INIT_LR = 1e-3
BS = 32
default_image_size = tuple((256, 256))
image_size = 0
directory_root = '../input/plantvillage/'
width=256
height=256
depth=3
image_list, label_list = [], []try:
print("[INFO] Loading images ...")
root_dir = listdir(directory_root)
for directory in root_dir :
if directory == ".DS_Store" :
root_dir.remove(directory)
for plant_folder in root_dir :
plant_disease_folder_list = listdir(f"{directory_root}/{plant_folder}")
for disease_folder in plant_disease_folder_list :
if disease_folder == ".DS_Store" :
plant_disease_folder_list.remove(disease_folder)
for plant_disease_folder in plant_disease_folder_list:
print(f"[INFO] Processing {plant_disease_folder} ...")
```

```
plant_disease_image_list = listdir(f"{directory_root}/{plant_folder}
/{plant_disease_folder}/")
for single_plant_disease_image in plant_disease_image_list :
if single_plant_disease_image == ".    DS_Store" :
plant_disease_image_list.remove(single_plant_disease_image)
for image in plant_disease_image_list[:200]:
image_directory = f"{directory_root}/{plant_folder}/{plant_disease_folder}
/{image}"
if image_directory.endswith(".jpg") == True or image_directory.endswith
(".JPG") == True:
image_list.append(convert_image_to_array(image_directory))
label_list.append(plant_disease_folder)
print("[INFO] Image loading completed")  except Exception as e:
print(f"Error : {e}")
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer,open('label_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
np_image_list = np.array(image_list, dtype=np.float16) / 255.0
print("[INFO] Spliting data to train, test")
x_train, x_test, y_train, y_test = train_test_split(np_image_list,
image_labels, test_size=0.2, random_state = 42)
model = Sequential()
inputShape = (height, width, depth)
chanDim = -1if
K.image_data_format() == "channels_first":
inputShape = (depth, height, width)
chanDim = 1model.add(Conv2D(32, (3, 3)), padding="same",
input_shape=inputShape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Dropout(0.25))model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())model.add(Dense(1024))
model.add(Activation("relu"))
```

```
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(n_classes))
model.add(Activation("softmax"))

opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,metrics=["accuracy"])
print("[INFO] training network...")
history = model.fit_generator(aug.flow(x_train, y_train, batch_size=BS),
  validation_data=(x_test, y_test),
  steps_per_epoch=len(x_train) // BS,
 epochs=EPOCHS, verbose=1
  )
  acc = history.history['acc']
  val_acc = history.history['val_acc']
  loss = history.history['loss']
  val_loss = history.history['val_loss']
  epochs = range(1, len(acc) + 1)
  plt.plot(epochs, acc, 'b', label='Training accurarcy')
  plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
  plt.title('Training and Validation accurarcy')plt.legend()
  plt.figure()
  lossplt.plot(epochs, loss, 'b', label='Training loss')
  plt.plot(epochs, val_loss, 'r', label='Validation loss')
  plt.title('Training and Validation loss')
  plt.legend()
  plt.show()
  image_array = convert_image_to_array(target_img_file)
  image_array = np.array(image_array, dtype=np.float16) / 225.0
  image_array = np.expand_dims(image_array, axis=0)
  prediction = model.predict(image_array)
  result_rate = prediction[0].tolist()
  total_class = (label_binarizer.classes_).tolist()

  target_img_file = directory_root + 'PlantVillage/Potato___Early_blight
  /0a8a68ee-f587-4dea-beec-79d02e7d3fa4___RS_Early.B 8461.JPG
  'print(target_img_file)
  result_class = dict()for i in range(len(total_class)):
  result_class[total_class[i]] = result_rate[i]
  result_class = sorted(result_class.items(), key=lambda x:x[1], reverse=True)
  print(np.array(result_class))
```

ANDROID CODE :

```
import android.content.Context;
import android.content.Intent;
import android.content.res.AssetManager;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.net.Uri;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.provider.MediaStore;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import org.jetbrains.annotations.NotNull;
import org.jetbrains.annotations.Nullable;
import java.io.Closeable;
import java.io.IOException;
import java.io.InputStream;
import kotlin.TypeCastException;
import kotlin.Unit;
import kotlin.collections.CollectionsKt;
import kotlin.jvm.internal.Intrinsics;

/**
 * A simple {@link Fragment} subclass.
 * Use the {@link BlankFragment#newInstance} factory method to
 * create an instance of this fragment.
 */
public class BlankFragment extends Fragment {
    private Classifier mClassifier;
    private Bitmap mBitmap;
    private final int mCameraRequestCode=0;
    private final int mGalleryRequestCode = 2;
    private final int mInputSize = 224;
    private final String mModelPath = "plant_disease_model.tflite";
    private final String mLabelPath = "plant_labels.txt";
    private final String mSamplePath = "soybean.JPG";
    Closeable closeable;
    // TODO: Rename parameter arguments, choose names that match
```

```
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";

// TODO: Rename and change types of parameters
private String mParam1;
private String mParam2;

public BlankFragment() {
    // Required empty public constructor
}

/**
 * Use this factory method to create a new instance of
 * this fragment using the provided parameters.
 *
 * @param param1 Parameter 1.
 * @param param2 Parameter 2.
 * @return A new instance of fragment BlankFragment.
 */
// TODO: Rename and change types and number of parameters
public static BlankFragment newInstance(String param1, String param2) {
    BlankFragment fragment = new BlankFragment();
    Bundle args = new Bundle();
    args.putString(ARG_PARAM1, param1);
    args.putString(ARG_PARAM2, param2);
    fragment.setArguments(args);
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        mParam1 = getArguments().getString(ARG_PARAM1);
        mParam2 = getArguments().getString(ARG_PARAM2);
    }
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View view=inflater.inflate(R.layout.fragment_blank, container, false);
    AssetManager assetManager = getActivity().getAssets();
    Intrinsics.checkExpressionValueIsNotNull(assetManager, "assets");
    try {
        this.mClassifier = new Classifier(assetManager, this.mModelPath,
  this.mLabelPath, this.mInputSize);
    } catch (IOException e) {
```

```
            e.printStackTrace();
        }
        Resources resources = this.getResources();
        Intrinsics.checkExpressionValueIsNotNull(resources, "resources");
        try {
            closeable = (Closeable)resources.getAssets().open(this.mSamplePath);
        } catch (IOException e) {
            e.printStackTrace();
        }
 Intrinsics.throwUninitializedPropertyAccessException("mBitmap");
        }
        var10001 = Bitmap.createScaledBitmap(var10001, this.mInputSize,
       this.mInputSize, true);
        Intrinsics.checkExpressionValueIsNotNull(var10001, "Bitmap.create
  ScaledBitma…utSize, mInputSize, true)");
        this.mBitmap = var10001;
      //  ImageView var13 = (ImageView)view.findViewById(R.id.mPhotoImageView);
        var10001 = this.mBitmap;
        if (var10001 == null) {
            Intrinsics.throwUninitializedPropertyAccessException("mBitmap");
        }

//        var13.setImageBitmap(var10001);
        Unit var12 = Unit.INSTANCE;
        (view.findViewById(R.id.mCameraButton)).setOnClickListener
    ((View.OnClickListener)(new View.OnClickListener() {
            public final void onClick(View it) {
                Intent callCameraIntent = new Intent("android.media.action
    .IMAGE_CAPTURE");
                startActivityForResult(callCameraIntent, mCameraRequestCode);
            }
        }));
```

5.SCREENSHOTS

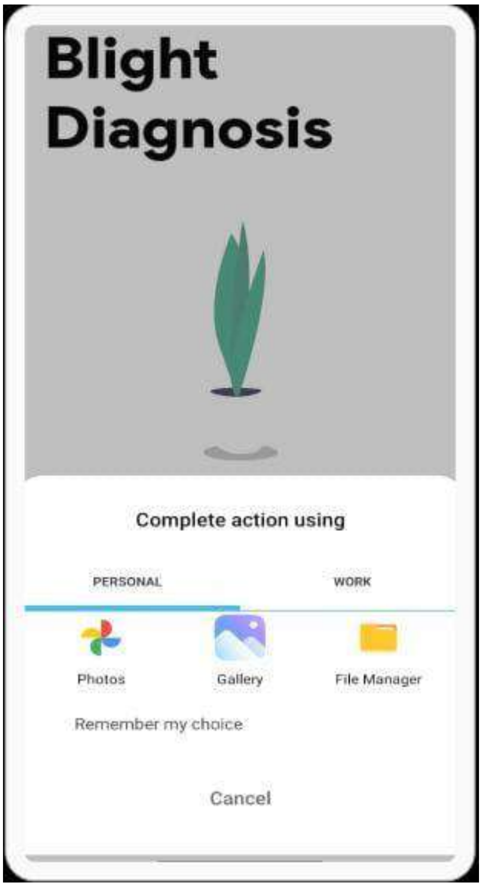
5.1 MAIN PAGE

After opening the application, The below page is displayed. There we can see two sections at the bottom which are named as Classify and Historyand it automatically processes the image.

Screenshot 5.1: Main Page

5.2 DISEASE SELECTION

We can see two buttons named as Camera and Gallery. The users can select their preferred option.Whenever user clicks Gallery button it prompts to select among various options like Photos, Gallery and File Manager. The user can select any option and can choose the image of the diseased plant from his storage. The user can also select Camera option and it redirects him to his phone camera where he can manually capture the picture of the crop .



Screenshot 5.2: Disease Selection

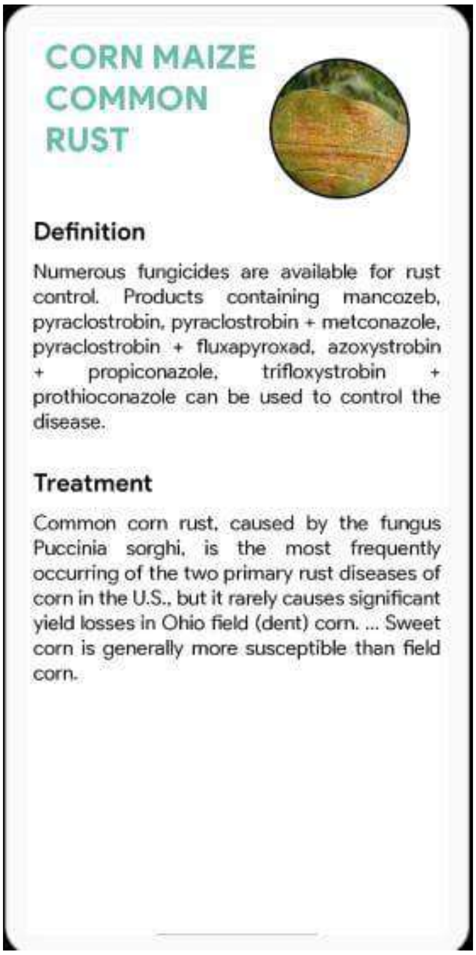
5.3 HISTORY

The application has a feature which saves the user past activity so that he can view it whenever needed. So that it can save his time.

Screenshot 5.3 History

5.4 DISEASE DETECTION

Whenever user selects the diseased image the application processes it and identifies the plantddisease and gives the output



Screenshot 5.4: Disease Detection

6.TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discoverevery conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct & consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

6.3.1 UPLOADING IMAGES

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads image	Use it for identification	The user uploads disease image.	Uploaded successfully
2	User uploads 2nd image	Use it for identification	The user uploads the a disease image.	Uploaded successfully

6.3.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the classifier performs its task	An image with apple disease is given.	Apple Disease is predicted.
2	Classification test 2	To check if the classifier performs its task	An image with tomato disease is given	Tomato Disease is predicted.
3	Classification test 3	To check if the classifier performs its task	An image with potato disease is given	Potato Disease is predicted.

7.CONCLUSION

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Recent studies on crop diseases show how they harm the plants. For detecting the crop diseases, people are using the conventional method, human vision-based approach which is time consuming .In this paper, we have proposed a custom CNN-based model that can classify common crop diseases which are commonly found in leaves.

Our new system is cost effective , simple and error free. With very less computational efforts the optimum results were obtained which also shows the efficiency of algorithm in recognition and classification of the leaf diseases. Another advantage of using these methods is that the crop diseases can be identified at early stage or the initial stage. Detecting the diseases from the crops helps in protecting the leaves surrounding it . Practical results might help in identifying other crop diseases.Therefore, there is a lot that can still be done in this field for enhancement of the existing works.

7.2 FUTURE SCOPE

In future research, we will use the neural network to generate zero initial set corresponding to different leaves, which will increase the end of calculation limit for the iterative process of Chan–Vese algorithm, speed up the training speed, and end the iteration ahead of time.

7.3 GITHUB LINK :

https://github.com/dh703/Major-Project.git

8.BIBILOGRAPHY

8. BIBILOGRAPHY

8.1 REFERENCES

- Dhiman Mondal, Dipak Kumar Kole, Aruna Chakraborty, D. Dutta Majumder" Detection and Classification Technique of Yellow Vein Mosaic Virus Disease in Okra Leaf Imagesusing Leaf Vein Extraction and Naive Bayesian Classifier., 2015, International Conference on Soft Computing Techniques and Implementations- (ICSCTI) Department of ECE, FET, MRIU, Faridabad, India, Oct 8-10, 2015.
- Pranjali B. Padol, Prof. AnjilA.Yadav, "SVM Classifier Based Grape Leaf Disease Detection" 2016 Conference on Advances in Signal Processing(CAPS) Cummins college of Engineering for Women, Pune. June 9-11, 2016.
- Detecting jute plant disease using image processing and machine learning 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT) .
- Tejoindhi M.R, Nanjesh B.R, JagadeeshGujanuru Math, AshwinGeetD'sa "Plant Disease Analysis Using Histogram Matching Based On Bhattacharya's Distance Calculation" International Conference on Electrical, Electroniocs and Optimization Techniques(ICEEOT)-2016.
- Detection of unhealthy plant leaves using image processing and genetic algorithm with Arduino2018 International Conference on Power, Signals, Control and Computation (EPSCICON).
- Tanvimehera, vinaykumar,pragyagupta "Maturity and disease detection in tomato using computer vision" 2016 Fourth international conference on parallel, distributed and grid computing(PDGC).

Blight Diagnosis

G.Ashwini Kumari[1], Suri Dheeraj Sai[2], Alex Abrahim[3], Bojja Ritwik[4]

[1]*Assistant Professor, CSE, CMR Technical Campus, Hyderabad, India*

[2]*Student, CSE, CMR Technical Campus, Hyderabad, India*

[3]*Student, CSE, CMR Technical Campus, Hyderabad, India*

[4]*Student, CSE, CMR Technical Campus, Hyderabad, India*

***Abstract*: The early detection of diseases is important as it has social, ecological, and economic impacts on agriculture. Automatic methods for classification of crop diseases also help with taking action after detecting the symptoms of crop diseases as it plays a vital role in terms of rural employment and GDP of the country. The objective of this work is to develop a model which can predict those diseases so that the farmers can take appropriate action. This work can be done using a Convolution Neural Networks model using a dataset with 10 fold cross-validation technique to identify common crop diseases. With this model, we overcome the problem of crop diseases. The early detection of diseases is important as it has social, ecological, and economic impacts on agriculture. Automatic methods for classification of crop diseases also help with taking action after detecting the symptoms of crop diseases as it plays a vital role in terms of rural employment and GDP of the country. The objective of this work is to develop a model which can predict those diseases so that the farmers can take appropriate action. This work can be done using a Convolution Neural Networks model using a dataset with 10 fold cross-validation technique to identify common crop diseases. With this model, we overcome the problem of crop diseases.**

***Keywords*: RNN ML Algorithm, IOT software, Real Time GPS, TensorFlowLite.**

1. Introduction

India is a developed country and about 70% of the populace relies upon agribusiness. Farmers have an enormous scope of variety for choosing different reasonable yields and tracking down the appropriate pesticides for plants. Infection on plants prompts a critical decrease in both the quality and amount of farming items. The investigations of plant sickness allude to the investigations of outwardly perceptible examples on the plants. Checking of wellbeing and infection on plants assumes a significant part in the fruitful development of yields in the ranch. In the early days, the observing and investigation of plant illnesses were done physically by the aptitude individual around there. This requires a colossal measure of work and furthermore requires unreasonable handling time. The picture preparing procedures can be utilized in the plant sickness discovery. In a large portion of the cases, sickness manifestations are seen on the leaves, stem, and natural products. The plant leaf for the location of infection is viewed as which shows the sickness manifestations. This paper gives the prologue to the picture preparing strategy utilized for plant sickness recognition.

2. Proposed System

In the proposed framework from the outset the pictures are procured from the farmer. The pictures are gotten from the farmer by means of the Android Application assistance of the farmer. At that point picture will be resized in suitable arrangement then it will be transferred on worker on which a calculation is carried out utilizing Convolutional Neural Network Each Convolutional Neural Network engineering is isolated into two sections initially is include extraction and second is grouping and has four primary segments.

1. Convolutional activity.

2. Max-pooling (Down examining)

3. ReLu (Non Linearity)

4. Grouping (completely associated layer)

When picture is reached to worker it is prepared with calculation here we separate the element of picture with convolutional activity by convolving the channel over picture which delivers the element guides like edges, surface, spots, openings, shading. These highlights maps are down tested so it very well may be passed to completely associated layer for example classifier after each layer we apply ReLu for example non linearity so tackle complex issue like order .After this the machine learning code is converted in tflite file which is kept in the asserts folder in android studio and a text file is created . The Image which is captured by the user is taken in the form of bitmap and it is compared with the images in tflite file and the result is computed accordingly .The proposed model was evaluated based on Plant VillageDataset, which achieved around 98% accuracy.

This ML model consists of three major parts : Building and creating a machine learning model using TensorFlow with Keras Deploying the model to an Android application using TFLite. Documenting and open-sourcing the development process.

The references are mentioned at the end of the paper.

3. Methodology

A.. System Architecture

This part portrays the means associated with making and conveying the classifier. Arrangement by CNN is partitioned into three stages that tackle separate undertakings. They are convolutional layers, pooling layers, and enactment capacities, ordinarily Rectified Linear Units (ReLUs). The number of layers utilized, their plan, and the presentation of other handling units differ starting with one design then onto the next, deciding their particularity.

Fig. 1: Cnn architecture

Below are the layers used in CNN along with activation function and the total params are 58,102,671 the trainable params are 58,099,791 And non-trainable params are 2,880.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 256, 256, 32)	896
activation_1 (Activation)	(None, 256, 256, 32)	0
batch_normalization_1 (Batch	(None, 256, 256, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 85, 85, 32)	0
dropout_1 (Dropout)	(None, 85, 85, 32)	0
conv2d_2 (Conv2D)	(None, 85, 85, 64)	18496
activation_2 (Activation)	(None, 85, 85, 64)	0
batch_normalization_2 (Batch	(None, 85, 85, 64)	256
conv2d_3 (Conv2D)	(None, 85, 85, 64)	36928
activation_3 (Activation)	(None, 85, 85, 64)	0
batch_normalization_3 (Batch	(None, 85, 85, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 42, 42, 64)	0
dropout_2 (Dropout)	(None, 42, 42, 64)	0
conv2d_4 (Conv2D)	(None, 42, 42, 128)	73856
activation_4 (Activation)	(None, 42, 42, 128)	0
batch_normalization_4 (Batch	(None, 42, 42, 128)	512
conv2d_5 (Conv2D)	(None, 42, 42, 128)	147584
activation_5 (Activation)	(None, 42, 42, 128)	0
batch_normalization_5 (Batch	(None, 42, 42, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 21, 21, 128)	0
dropout_3 (Dropout)	(None, 21, 21, 128)	0
flatten_1 (Flatten)	(None, 56448)	0
dense_1 (Dense)	(None, 1024)	57803776
activation_6 (Activation)	(None, 1024)	0
batch_normalization_6 (Batch	(None, 1024)	4096
dropout_4 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 15)	15375
activation_7 (Activation)	(None, 15)	0

b.

14

among 38Plants 2021, 10, 28 4 of 15crop-infection sets. This dataset contains clear pictures of plant leaves and each picture contains just one leaf. It likewise accompanies preset preparing/testing subsets which we continue in this investigation. We utilize this dataset in two distinct manners, first to arrange sickness crop pair and second to group infection paying little heed to the influenced crop. In the first configuration,32 classes (C7 to C38) are utilized as the source area Ds and 6 classes (C1 to C6) as the objective space Dt-like [38]. For this situation, the source area has 43,444 examples while the objective/novel space has 10,861 examples. In the subsequent arrangement, we revamp the dataset as indicated by the basic sickness name as in, bringing about 20 illness classes and one solid class. For infections that influence more than one plant, all examples are consolidated in one class under the name of this illness, and for the sound class, around 5000 examples were gathered from all accessible plants. The three infections with the least pictures were chosen from the adjusted dataset as target (space classes and the rest as the source area classes. The quantity of tests in the source area Ds is 44,081 and in the objective space, Dt is 1278.

Class	Crop	Disease	Samples	
			Train	Test
C23	Potato	Late blight	800	200
C24	Raspberry	Healthy	297	74
C25	Soybean	Healthy	4072	1018
C26	Squash	Powdery mildew	1468	367
C27	Strawberry	Healthy	364	92
C28	Strawberry	Leaf scorch	887	222
C29	Tomato	Bacterial spot	1702	425
C30	Tomato	Early blight	800	200
C31	Tomato	Healthy	1273	318
C32	Tomato	Late blight	1527	382
C33	Tomato	Leaf mold	761	191
C34	Tomato	Septoria leaf spot	1417	354
C35	Tomato	Spider mites Two-spotted spider mite	1341	335
C36	Tomato	Target spot	1123	281
C37	Tomato	Tomato mosaic virus	299	74
C38	Tomato	Tomato Yellow Leaf Curl Virus	4286	1071

Class	Crop	Disease	Samples	
			Train	Test
C23	Potato	Late blight	800	200
C24	Raspberry	Healthy	297	74
C25	Soybean	Healthy	4072	1018
C26	Squash	Powdery mildew	1468	367
C27	Strawberry	Healthy	364	92
C28	Strawberry	Leaf scorch	887	222
C29	Tomato	Bacterial spot	1702	425
C30	Tomato	Early blight	800	200
C31	Tomato	Healthy	1273	318
C32	Tomato	Late blight	1527	382
C33	Tomato	Leaf mold	761	191
C34	Tomato	Septoria leaf spot	1417	354
C35	Tomato	Spider mites Two-spotted spider mite	1341	335
C36	Tomato	Target spot	1123	281
C37	Tomato	Tomato mosaic virus	299	74
C38	Tomato	Tomato Yellow Leaf Curl Virus	4286	1071

Fig 3 : Dataset diseases

C.MACHINE LEARNING

1) Stage one:-

Stage one means to explore the impact that picture size has on model execution. Altogether, five pictures estimated are tried going from 150 x 150 to 255 x 255. As a default of move learning, all layers with the besides of the last two layers are frozen. These contain new loads and are explicit to the plant illness characterization task. Freezing permits these layers to be illness independently prepared, without back-propagating the inclinations. In precisely thusly, the 1cycle approach is utilized to prepare the last layers. With this total, the leftover layers are delivered. To help the calibrating interaction, a plot showing learning rate versus misfortune is produced and investigated. From this, appropriate learning is chosen, and the model is run. With results recorded, the model is re-made to the extra four picture sizes.

2) Phase two:-

Utilizing the most appropriate picture size, the Cnn model is advanced. To additionally improve the model's exhibition, extra expansion settings are added. Tasks incorporate brilliance changes (0.4,0.7) and twists (0.5). Then, the last two layers are disengaged and prepared at the default learning rate. With this total, tweaking is performed, running numerous preliminaries to test a progression of learning rates and a number of epochs.

The convolutional layer is the fundamental structure square of the convolutional neural organization. The layer's boundaries are included a bunch of learnable pieces which have a little responsive field yet reach out through the full profundity of the information volume.

Each convolutional layer has M maps of equivalent size, Mx and My , and a piece of size kx, and ky is moved over the specific district of the info picture. The skipping factors Sx and Sy characterize the number of pixels the channel/portion avoids in x- and y- heading between ensuing convolutions. The size of the yield guide could be characterized as

$$M_x^n = \frac{M_x^{n-1} - K_x^n}{S_x^n + 1} + 1,$$

$$M_y^n = \frac{M_y^{n-1} - K_y^n}{S_y^n + 1} + 1,$$

where n demonstrates the layer. Each map in layer Ln is associated with most maps Mn-1 in layer. Corrected Linear Units (ReLU) are utilized as an alternative for soaking nonlinearities. This enactment work adaptively learns the boundaries of rectifiers and improves precision at the insignificant extra computational expense. It is characterized as

$$F(z_i) = \max(0,z_i),$$

where zi addresses the contribution of the nonlinear initiation function f on the ith channel. Profound CNN with ReLUs trains a few times quicker. This technique is applied to the yield of each convolutional and completely associated layer. Regardless of the yield, the info standardization isn't needed; it is applied after ReLU nonlinearity after the first and second convolutional layer since it decreases top-1 and top-5 mistake rates. In CNN, neurons inside a secret layer are divided into "include maps." The neurons inside a component map share a similar weight and predisposition. The neurons inside the element map look for a similar component. These neurons are extraordinary since they are associated with various neurons in the lower layer. So for the primary secret layer, neurons inside a component guide will be associated with various districts of the information picture. The secret layer is sectioned into highlight maps where every neuron in an element map searches for a similar component yet at various places of the information picture. Essentially, the component map is the consequence of applying convolution across a picture. Each layer's highlights are shown in an alternate square, where perception addresses the most grounded initiation for the gave include map, beginning from the first convolutional layer, where highlights go from singular pixels to straight lines, to the fifth convolutional layer where learned highlights like shapes and certain pieces of leaves are shown.

Another significant layer of CNNs is the pooling layer, which is a type of nonlinear downsampling. Pooling activity gives the type of interpretation invariance it works freely on each profundity cut of the info and resizes it spatially. Covering pooling is advantageously applied to decrease overfitting. Additionally for decreasing overfitting, a dropout layer is utilized in the initial two completely associated layers. Yet, the weakness of dropout is that it builds preparing time 2-3 times contrasting with a standard neural organization of the specific design. Bayesian enhancement tests likewise demonstrated that ReLUs and dropout have collaboration impacts, which implies that it is invaluable when they are utilized together.

The development of CNNs allude to their capacity to learn rich mid-level picture portrayals instead of hand-planned low-level highlights utilized in other picture grouping strategies

Figure 4 shows the separated yield pictures after each convolutional and pooling layer of the profound organization. Yield pictures are marked with the name of the relating layer at the base right corner of each picture.

Fig. 4. Output Layers

D Performed tests:-

The basic methodology in estimating the execution of artificial neural organizations is parting information into the preparation set and the test set and afterward preparing a neural network on the training set and utilizing the test set for prediction. Accordingly, since the first results for the testing set and our model anticipated results are known, the exactness of our expectations can be determined. Various tests were performed with .50000 images from the database.

For the precision test, 10-fold cross-validation predictive was utilized to evaluate a prescient model. The cross-validation technique was repeated after each thousand preparing cycles. In generally assessed consequence of the test is graphically addressed as top-1, to test if the top class (the one having the most noteworthy likelihood) is equivalent to the objective marker. The top 5 error rate is there to test if the target label is one of the top 5 predictions, the ones with 5 of the greatest probabilities.

E Fine-tuning

Fine-tuning assists with expanding the exactness of expectation by making little adjustments to improve or upgrade the result. The most appropriate model for plant disease detection will be accomplished through the cycle of exploratory change of the boundaries. The green line in the diagram in Figure 5 shows the organization's prosperity on the validation test set, through preparing iterations. After each 10 thousand preparing iterations, the depiction of the model was gotten. The blue line in the diagram addresses the misfortune during the preparation stage. Through Training iterations, the loss was quickly decreased.

Fig 5 : Traning and validation accuracy

Fig 6 : Training and validation accuracy

4. Results and Discussions

The cnn architecture pictures the hidden layer output for each layer and its generated intermediate outputs are yields are summarized. In our trained model, a portion of the intermediate outputs in the shallow layers (Conv1, Conv5) feature the yellow and earthy colored injuries that are evident inside the picture . Be that as it may, in the more profound layer , attributable to the convolution and pooling layers, the picture size is too little to even think about interpretting whether such removed highlights have been held. Additionally, the worldwide normal pooling layer changes pictures over to a component vector that disposes of the spatial data, making it profoundly hard to see how the highlights are taken care of in continuing layers. It is hard to recognize whether the extricated includes decidedly add to the grouping of the info picture to the right infection class or are utilized for motivation to deny different potential outcomes Hence, understanding what the CNN has realized by just investigating the halfway yield is lacking.

Fig. 7. Main Page

After opening the application, The above page is displayed. There we can see two sections at the bottom which are named as Classify and History.

We can see two buttons named as Camera and Gallery. The users can select their preferred option.

Fig 8: Disease Selection

Whenever user clicks Gallery button it prompts to select among various options like Photos, Gallery and File Manager. The user can select any option and can choose the image of the diseased plant from his storage. The user can also select Camera option and it redirects him to his phone camera where he can manually capture the picture of the crop and it automatically processes the image.

Fig. 9. Disease Identification

Whenever user selects the diseased image the application processes it and identifies the plant ddisease and gives the output as shown in Fig 9.

Fig. 10. History

The application has a feature which saves the user past activity so that he can view it whenever needed. So that it can save his time.

5. Conclusion

Recent studies on crop diseases show how they harm the plants. For detecting the crop diseases, people are using the conventional method, human vision-based approach which is time consuming .In this paper, we have proposed a custom CNN-based model that can classify common crop diseases which are commonly found in leaves. Our new system is cost effective , simple and error free. Detecting the diseases from the crops helps in protecting the leaves surrounding it . Practical results might help in identifying other crop diseases.

6. References

[1] Neural Networks: H.A. Rowley , Dept. of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA.

[2] Deep Residual Learning for tomato plant leaf disease identification – Habibollah agh atabay.Department of computer, Gonbad Kavous university , Gonbad kavous Iran.

[3] Identifying two of tomatoes leaf viruses using support vector machine, in: Information Systems Design and Intelligent Applications : U. Mokhtar, M.A. Ali, A.E. Hassanien, H. Hefny.

[4] Large-scale machine learning on heterogeneous distributed systems : M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin.

[5] Deep Residual Learning for Image Recognition : Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

[6] Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks : Alec Radford, Luke Metz, Soumith Chintala.

[7] Learning Activation Functions to Improve Deep Neural Networks : Forest Agostinelli, Matthew Hoffman, Peter Sadowski, Pierre Baldi

[8] Applications of Convolutional Neural Networks : Ashwin Bhandare, Maithili Bhide, Pranav Gokhale, Rohan Chandavarkar .

[9] Gradient basel learning applied to document recognition: Yann LeCunn, Leon Bottou, Yoshua Bengio, Patrick Haffner.

[10]"PlantVillage", Plantvillage.psu.edu, 2020.

[11]D. Klauser, "Challenges in monitoring and managing plant diseases in developing countries".

[12]A.Muimba-Kankolongo, Food crop production by smallholder farmers in Southern Africa. Elsevier, 2018, pp. 23-27.

[13]J. Boulent, S. Foucher, J. Théau and P. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases", Frontiers in Plant Science, vol. 10, 2019.

[14]"iNaturalist", iNaturalist, 2020.

[15]I. Ltd., "Success-stories - PlantSnap: Training the world's largest plant recognition classifier. | Imagga Technologies Ltd.", Imagga.com, 2020. .

[16]"Smartphone users worldwide 2020 | Statista", Statista, 2020.

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Suri Dheeraj Sai

In recognition of the publication of the paper entitled

Blight Diagnosis

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 5 , May-2021 | Date of Publication: 2021-05-31

Parin P
EDITOR

EDITOR IN CHIEF

ISSN 2349-5162 Journal of Emerging Technologies and Innovative Research

JETIR2105880 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2105880** **Registration ID : 310098**

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal Peer-reviewed, Refereed Journal
www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Alex Abrahim

In recognition of the publication of the paper entitled

Blight Diagnosis

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 5 , May-2021 | Date of Publication: 2021-05-31

Parin P

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162 Journal of Emerging Technologies and Innovative Research

JETIR2105880 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2105880** **Registration ID : 310098**

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Bojja Ritwik

In recognition of the publication of the paper entitled

Blight Diagnosis

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 5 , May-2021 | Date of Publication: 2021-05-31

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

Journal of Emerging Technologies and Innovative Research

JETIR2105880 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2105880** **Registration ID : 310098**

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

G.Ashwini Kumari

In recognition of the publication of the paper entitled

Blight Diagnosis

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 5 , May-2021 | Date of Publication: 2021-05-31

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2105880 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2105880** **Registration ID : 310098**