Potato Disease Predictor

Submitted in partial fulfilment of the requirements of the degree of

T.E in Computer Engineering

By

Roll no. 40 Shivdhan Nikam

Roll no. 48 Rushikesh Patil

Roll no. 67 Soham Shinde

Supervisor:

Prof. Ranjit Mane

(Department of Computer Engineering)

Bharati Vidyapeeth College of Engineering, Navi Mumbai

Project Synopsis Report Approval for T.E.

This the project entitled **“Potato Disease Predictor”** by **“Shivdhan Nikam (40)”, “Rushikesh Patil (48)”, “Soham Shinde (67)”** is approved for the degree of **Bachelor of Engineering in Computer Engineering.**

Date: ________________

____

Prof. Ranjit Mane

Project Guide

___________________ ___________________

Dr. D. R. Ingle **Dr. Sandhya Jadhav**

Head of the Department **Principal**

___________________ ___________________

Internal Examiner **External Examiner**

Declaration

We declare that this written submission represents our ideas in our words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_______________	_______________	________________
(Shivdhan Nikam)	**(Rushikesh Patil)**	**(Soham Shinde)**
Roll no. 40	**Roll no. 48**	**Roll no. 67**

Date: ___________________

Table of Contents

Chapter – One

Introduction

1.1 Abstract

Potato is one of the most used crops in the world and 2^{ND} most important crop in India. Our economy is largely affected by the production of potato. In India, potato is cultivated in almost all states and under very diverse agro climate conditions. About 85 per cent of potatoes are cultivated in Indo-Gangetic plains of North India. The states of Uttar Pradesh, West Bengal, Punjab, Bihar, and Gujarat accounted for more than 80 per cent share in total production. But its production is hampered due to different diseases of potato leaves. These diseases decrease production and increase the price of potatoes. Our objective is to develop an automated system which will predict the potato disease and helps farmers to take necessary steps. In this work, we implemented a model based on Convolutional Neural Network (CNN) which provides 98.33% accurate result in predicting different diseases of potatoes. This is the maximum accuracy gained for only potato disease prediction to the best of our understanding. The system is cost effective, less time consuming and provides an efficient way of predicting potato diseases from leaves. This will help the farmers and lead our country towards a digital agricultural system.

1.2 Proposed Problem

The Farmers who grow potatoes suffer from serious financial standpoint losses each year which cause several diseases that affect potato plants. To suffice the demand for food, at least 70 % increase from the present global yield must be produced. The presence of plant diseases imposes great challenges for farmers to meet the demand. The diseases Early Blight and Late Blight are the most frequent. Early blight is caused by fungus and late blight is caused by specific micro-organisms and if farmers detect this disease early and apply appropriate treatment then it can save a lot of waste and prevent economical loss. The treatments for early blight and late blight are a little different so it's important that you accurately identify what kind of disease is there in that potato plant. Behind the scenes, we are going to use Convolutional Neural Network – Deep Learning to diagnose plant diseases. Here, we'll develop an end-to-end Deep Learning project in the field of agriculture. We will create a simple Image Classification Model that will categorize Potato Leaf Disease using a simple and classic Convolutional Neural Network Architecture.

1.3 Definition and explanation

CNN: A CNN is a kind of network architecture for deep learning algorithms and is specifically used for image recognition and tasks that involve the processing of pixel data. There are other types of neural networks in deep learning, but for identifying and recognizing objects, CNNs are the network architecture of choice.

TensorFlow: The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining. Using production-level tools to automate and track model training over the lifetime of a product, service, or business process is critical to success.

NumPy: NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project, and you can use it freely.

React.js: The React. js framework is an open-source JavaScript framework and library developed by Facebook. It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

Fast API: Fast API is a Python framework and set of tools that enables developers to use a REST interface to call commonly used functions to implement applications. It is accessed through a REST API to call common building blocks for an app. In this example, the author uses FastAPI to create accounts, login, and authenticate.

Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update. Customize visual style and layout.

1.4 Aim and Scope

Our aim is to develop an automated system which will predict the potato disease and helps farmers to take necessary steps. In this work, we implemented a model based on Convolutional Neural Network (CNN) which provides 98.33% accurate result in predicting different diseases of potatoes. Now a days it is very important to detect disease in a plant in the budding stage so that productivity and quality of the yield can be upgraded. Since disease detection needs a lot of expertise so it would be very beneficial if we could implement this system on the website in which farmers can click a picture of the leaf and send it to the server. The server will automatically identify and classify the type of disease and send results back to the website.

Chapter – Two

Literature Survey

2.1 Published Papers

Author	Title	Published at	Abstract
Hassan Afzaal	Detection of a Potato Disease (Early Blight) Using Artificial Intelligence	25 January 2021	This study evaluated the potential of using machine vision in combination with deep learning (DL) to identify the early blight disease in real-time for potato production systems.
M. C. M. Pérombelon	Potato diseases caused by soft rot erwinias: an overview of pathogenesis	06 February 2002	Three soft rot erwinias, Erwinia carotovora ssp. carotovora, E. carotovora ssp. atroseptica and E. chrysanthemi are associated with potatoes causing tuber soft rot and blackleg
Javed Rashid	Multi-Level Deep Learning Model for Potato Leaf Disease Recognition	26 August 2021	Potato leaf disease detection in an early stage is challenging because of variations in crop species, crop diseases symptoms and environmental factors. These factors make it difficult to detect potato leaf diseases in the early stage.

Anh Dinh	Detection of potato diseases using image segmentation and multiclass support vector machine	2017 IEEE 30th CCECE	Modern phenotyping and plant disease detection provide promising step towards food security and sustainable agriculture. Imaging and computer vision-based phenotyping offers the ability to study quantitative plant.

2.2 Study of Existing System

Company Name	Type of Service	Description
Crop Diagnosis	Mobile App	Crop Diagnosis is a mobile application aiming to improve pest management decisions by making crop diagnosis more accurate, selection of chemicals error-free and application assisted by personalised instructions.
Bit Refine Group	Mobile App	A plant disease recognition system can work as a universal detector, recognizing general abnormalities on the leaves, such as scorching or mold.
Plantix Vision API	Website and Mobile App	Provide the best support to your farmers with accurate crop diagnosis and customised solutions anytime and anywhere.

Chapter – Three

Methodology

To achieve the objectives of this project, in this section, the methodology and procedures applied are discussed.

3.1 Technologies used:

1. Frontend:

ReactJS:

- The React.js framework is an open-source JavaScript framework and library developed by Facebook.
- It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.
- In web applications, all the data you show on the page should reside somewhere, for example, cache, database, storage account, etc. All the data can be accessed through APIs nowadays and most of the time the format would be in the JSON format. You need to fetch the data from the APIs and do some processing and then render the data on the UI.

TensorFlow:

- TensorFlow is an open-source library developed by Google primarily for deep learning applications therefore Google open-sourced it
- TensorFlow accepts data in the form of multi-dimensional arrays of higher dimensions called tensors.
- TensorFlow architecture works in three significant steps: Data pre-processing, Building the model, Training, and estimating the model.
- TensorFlow works based on data flow graphs that have nodes and edges. As the execution mechanism is in the form of graphs, it is much easier to execute TensorFlow code in a distributed manner across a cluster of computers while using GPUs.

2. **Backend:**

Fast API:

- Fast API is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. Its performance can be compared with NodeJS and Go, and it is rated as one of the fastest Python frameworks available.
- Tech giants like **Microsoft, Netflix, Uber** amongst many other corporations is already started building their APIs with the **Fast API** library.
- The framework is designed to optimize the experience so that production-ready APIs with best practices by default. It has a data validation system that can detect **any invalid data type at the runtime** and returns the reason for bad inputs to the user in the JSON format only which frees developers from managing this exception explicitly.

3.2 Functionalities:

UI for user:

- The proposed system provides a UI with drop and drop box which is a React.JS component that accepts image as input and performs its analysis at a backend where the Fast API server is working.
- It gives two parameters in output. First the detected leaf condition, whether the leaf is healthy or diseased (Late blight or early blight). Second is confidence (meaning the accuracy of detected class).

Easy to use API:

- Each deep learning/TensorFlow model requires platform to serve them. In this project we have built an API based platform provided to classification model.
- Using the Swagger UI (one of component of Fast API) one can easily classify the image.

3.3 Methodology / Project Workflow

Conceptual Framework:

In this section, Figure -- depicts the conceptual framework that will used to in the study. The framework is the basis for proposing a concept that can be implemented or utilized in the development of a website for potato disease classification. The framework shows the interaction between elements: user, software and hardware. The framework shows an API based website which uses the systems camera or images saved in system for the detection of potato disease which is the main module of the web application. The captured image will be analysed by the convolutional neural network algorithm to determine the status or condition of the potato leaf. Further, based on the analysis using the trained model, results of the analysis will be displayed. Results include the status of the plant whether it is healthy or pathogenic (early blight or late blight). If the plant leaf shows signs of disease, the system will suggest the detected disease with the confidence.

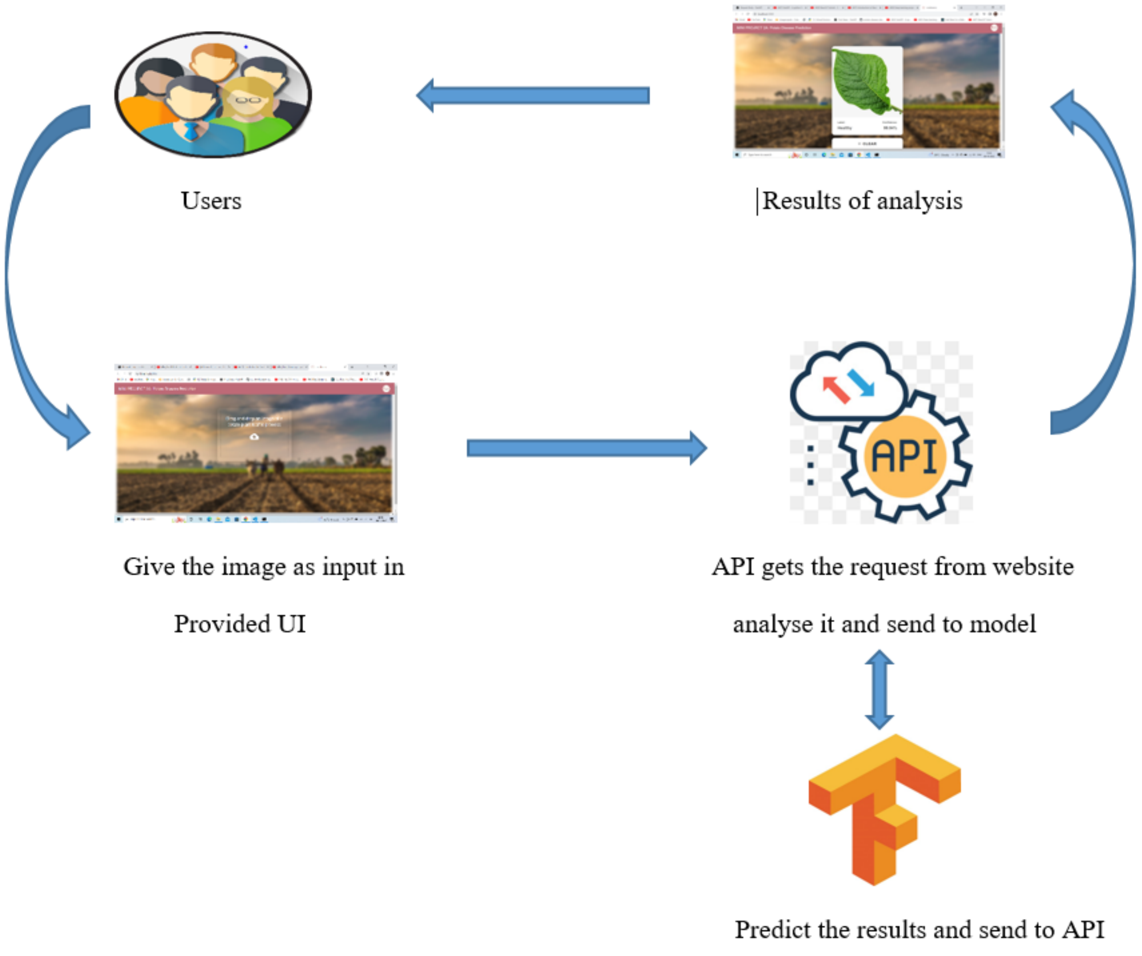
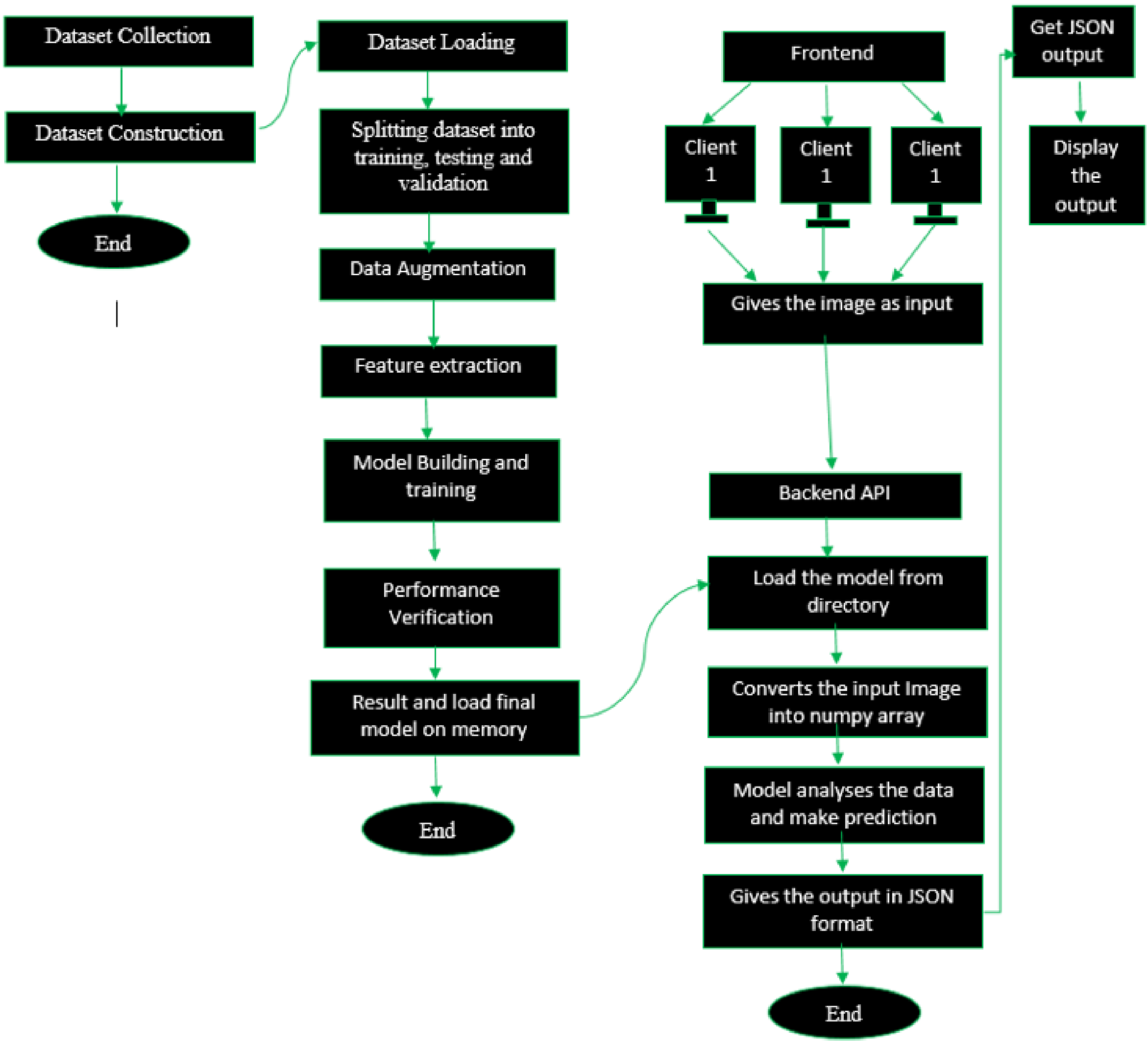


Fig. (a): Conceptual Framework

Project Workflow:

Plants are susceptible to various disease-related disorders and seizures. There are various causes which can be characterized by their effect on plants, disturbances due to environmental conditions such as temperature, humidity, excessive or insufficient food, light and the most common diseases such as bacterial, viral and fungal diseases. In the proposed system, we use the CNN algorithm to detect disease in plant leaves because with the help of CNN the maximum accuracy can be achieved if the data is good.



System Architecture

Fig. (b): System Architecture

1. **Dataset Collection:** Kaggle is a data science and artificial intelligence platform. It provides the various facilities and one of its major uses is it provides the large datasets on variety of the domains. We use Plant Village Dataset more specifically Potato disease. The Potato disease dataset consists of total 2252 leaf images divided into 3 categories by healthy and disease. We analysed more than 2252 images of plant leaves with distributed labels from 3 classes.
2. **Data Processing and Augmentation:** Image augmentation plays a key role in building an effective image classifier. Though datasets may contain anywhere from hundreds to a couple of thousand training examples, the variety might still not be enough to build an accurate model. Some of the many image augmentation options are flipping the image vertically/horizontally, rotating through various angles and scaling the image. These augmentations help increase the relevant data in a dataset. The size of each image in the Plant Village dataset is found to be 256 x 256 pixels. The data processing and image augmentation are done using the Keras deep-learning framework. The augmentation options used for training are as follows:
    - Random Rotation - To rotate a training image randomly over various angle
    - Random flip - To flip a training image randomly over horizontal and vertical direction.

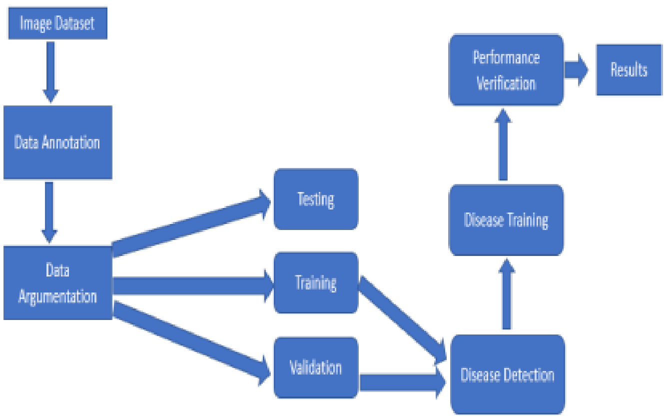
Steps related to image processing to detect plant diseases

The whole process is divided into three stages:

1. Input images are first loaded in the program.

2. Segmentation pre-processing includes the process of image segmentation, image enhancement and colour space conversion. First, the digital image of the image is enhanced with a filter. Then convert each image into an array. each image name is converted to a binary field.

3. CNN classifiers are trained to identify diseases in each plant class. Level 2 results are used to call up a classifier, which is trained to classify various diseases in that plant. If not present, the leaves are classified as "healthy".



Chapter – Four

Results and Discussion

4.1 Current Outcomes

- The performance of convolutional neural networks in object recognition and image classification has made tremendous progress in the past few years. Previously, the traditional approach for image classification tasks has been based on hand-engineered features, such as SIFT, HoG, SURF, etc., and then to use some form of learning algorithm in these feature spaces. The performance of these approaches thus depended heavily on the underlying predefined features. Feature engineering itself is a complex and tedious process which needs to be revisited every time the problem at hand or the associated dataset changes considerably. This problem occurs in all traditional attempts to detect plant diseases using computer vision as they lean heavily on hand-engineered features, image enhancement techniques, and a host of other complex and labour-intensive methodologies.

- In addition, traditional approaches to disease classification via machine learning typically focus on a small number of classes usually within a single crop. Examples include a feature extraction and classification pipeline using thermal and stereo images to classify Potato Early blight & Late blight against healthy Potato leaves; the detection of Potato Early blight & Late blight in uncontrolled environments using RGB images.

- Our approach is based on recent work which showed for the first time that end-to-end supervised training using a deep convolutional neural network architecture is a practical possibility even for image classification problems with a very large number of classes, beating the traditional approaches using hand-engineered features by a substantial margin in standard benchmarks. The absence of the labour-intensive phase of feature engineering and the generalizability of the solution makes them a very promising candidate for a practical and scalable approach for computational inference of plant diseases.

- Using the deep convolutional neural network architecture, we trained a model on images of plant leaves with the goal of classifying both crop species and the presence and identity of disease on images that the model had not seen before. Within the Potato Leaf data set of 2100 images containing 3 classes of 1 crop species and 2 diseases (or absence thereof), this goal has been achieved as demonstrated by the top accuracy of 99.35%. Thus, without any feature engineering, the model correctly classifies crop and disease from 3 possible classes. Importantly, while the training of the model takes a lot of time (multiple hours on a high-performance GPU cluster computer), the classification itself is very fast (less than a second on a CPU) and can thus easily be implemented on a smartphone. This presents a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.

- However, there are several limitations at the current stage that need to be addressed in future work. First, when tested on a set of images taken under conditions different from the images used for training, the model's accuracy is reduced substantially to just above 31%. It's important to note that this accuracy is much higher than the one based on random selection of 3 classes (2.6%), but nevertheless, a more diverse set of training data is needed to improve the accuracy. Our current results indicate that more (and more variable) data alone will be sufficient to substantially increase the accuracy, and corresponding data collection efforts are underway.

- The second limitation is that we are currently constrained to the classification of single leaves, facing up, on a homogeneous background. While these are straightforward conditions, a real-world application should be able to classify images of a disease as it presents itself directly on the plant. Indeed, many diseases don't present themselves on the upper side of leaves only (or at all), but on many different parts of the plant. Thus, new image collection efforts should try to obtain images from many different perspectives, and ideally from settings that are as realistic as possible.

- At the same time, by using 3 classes that contain both crop species and disease status, we have made the challenge harder than ultimately necessary from a practical perspective, as growers are expected to know which crops, they are growing. Given the very high accuracy on the Potato Leaf dataset, limiting the classification challenge to the disease status won't have a measurable effect. However, on the real-world datasets, we can measure noticeable improvements in accuracy. Overall, the presented approach works reasonably well with many different crop species and diseases and is expected to improve considerably with more training data.

- Finally, it's worth noting that the approach presented here is not intended to replace existing solutions for disease diagnosis, but rather to supplement them. Laboratory tests are ultimately always more reliable than diagnoses based on visual symptoms alone, and oftentimes early-stage diagnosis via visual inspection alone is challenging. Nevertheless, given the expectation of more than 5 billion smartphones in the world by 2020—of which almost a Billion in Africa (GSMA Intelligence, 2016)—we do believe that the approach represents a viable additional method to help prevent yield loss. What's more, in the future, image data from a smartphone may be supplemented with location and time information for additional improvements in accuracy. Last but not least, it would be prudent to keep in mind the stunning pace at which mobile technology has developed in the past few years and will continue to do so. With ever improving number and quality of sensors on mobiles devices, we consider it likely that highly accurate diagnoses via the smartphone are only a question of time.

Chapter – Five

Conclusion

5.1 Conclusion

With the advancement of technology and development on agricultural research there have been recent research providing prototypes and models for plant leaf recognition and disease detection. However, these models are not optimized for small platforms due to heavy computational operations.

Moreover, this study only uses a total of 2152 images for the retrained model. For future research, this model can be developed and improved by using a larger quantity of retraining images to achieve accurate results. Compared with traditional image processing methods, which deal with plant diseases detection tasks in several steps and links, plant diseases detection methods based on deep learning unify them into end-to-end feature extraction, which has a broad development prospects and great potential. Although plant diseases detection technology is developing rapidly, it has been moving from academic research to agricultural application, there is still a certain distance from the mature application in the real natural environment, and there are still some problems to be solved.

5.2 Scope for Future Development

In future it is very important to detect disease in a plant in the budding stage so that productivity and quality of the yield can be upgraded. Since disease detection needs a lot of expertise so it would be very beneficial if we could implement this system on the smartphone in which farmers can click a picture of the leaf and send it to the server. The server will automatically identify and classify the type of disease and send results along with prescribed medicines back to the smartphone.

Chapter – Six

Appendices

6.1 Screenshots

In [9]:

```
plt.figure(figsize=(10, 10))
for image_batch, lable_batch in dataset.take(1):
    for i in range(12):
        ax = plt.subplot(3,4,i+1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(class_names[lable_batch[i]])
        plt.axis("off")
```

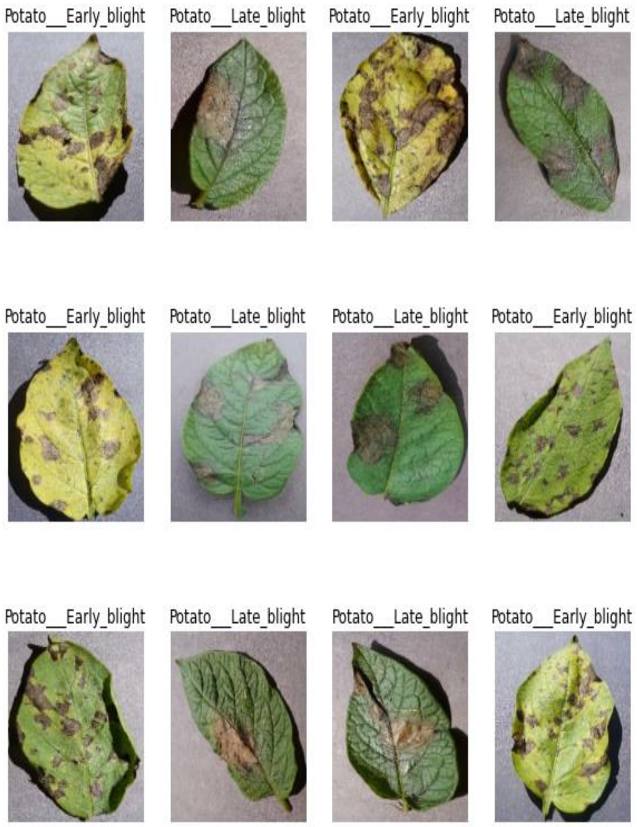


Fig (i) Data Visualization

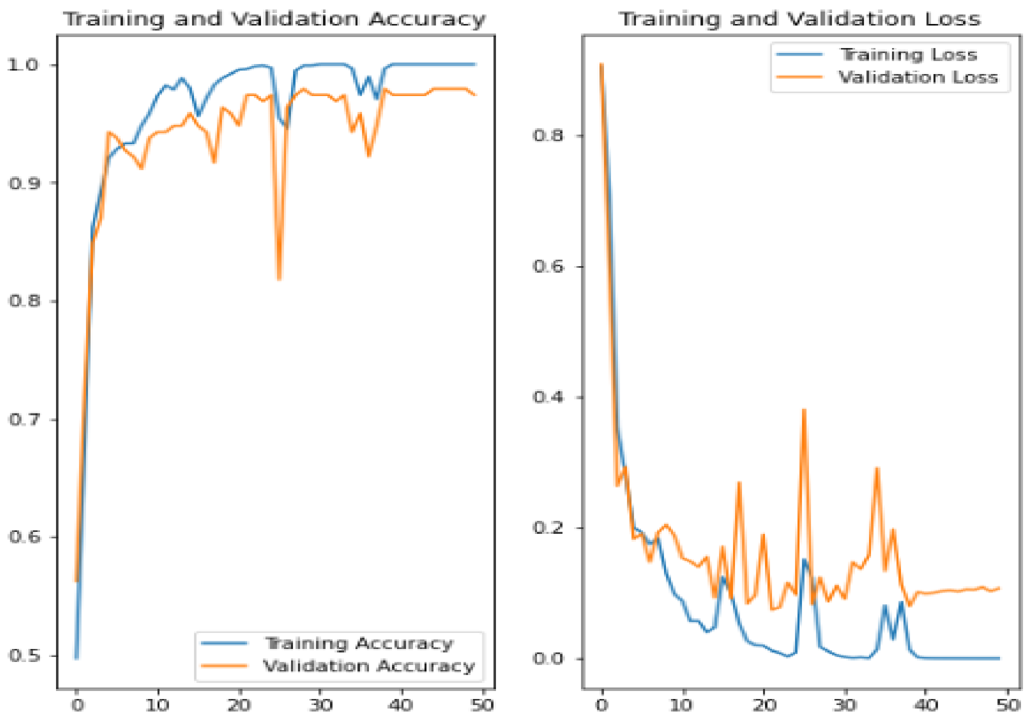
In [39]:
```
model.summary()
```

```
Model: "sequential_2"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 256, 256, 3)         0

 conv2d (Conv2D)             (32, 254, 254, 32)        896

 max_pooling2d (MaxPooling2D  (32, 127, 127, 32)       0
 )

 conv2d_1 (Conv2D)           (32, 125, 125, 64)        18496

 max_pooling2d_1 (MaxPooling  (32, 62, 62, 64)         0
 2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_2 (MaxPooling  (32, 30, 30, 64)         0
 2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_3 (MaxPooling  (32, 14, 14, 64)         0
 2D)

 conv2d_4 (Conv2D)           (32, 12, 12, 64)          36928

 max_pooling2d_4 (MaxPooling  (32, 6, 6, 64)           0
 2D)

 conv2d_5 (Conv2D)           (32, 4, 4, 64)            36928

 max_pooling2d_5 (MaxPooling  (32, 2, 2, 64)           0
 2D)

 flatten (Flatten)           (32, 256)                 0

 dense (Dense)               (32, 64)                  16448

 dense_1 (Dense)             (32, 3)                   195

=================================================================
Total params: 183,747
Trainable params: 183,747
Non-trainable params: 0
_________________________________________________________________
```

Fig (ii) Model Summary, CNN Description

```
In [44]: model.compile(
             optimizer='adam',
             loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
             metrics=['accuracy']
         )

In [45]: history = model.fit(
             train_ds,
             batch_size=BATCH_SIZE,
             validation_data=val_ds,
             verbose=1,
             epochs=50,
         )

         Epoch 1/50
         54/54 [==============================] - 86s 2s/step - loss: 0.9043 - accuracy: 0.4971 - val_loss: 0.9077 - val_accuracy: 0.
         5625
         Epoch 2/50
         54/54 [==============================] - 76s 1s/step - loss: 0.7084 - accuracy: 0.6707 - val_loss: 0.5978 - val_accuracy: 0.
         7188
         Epoch 3/50
         54/54 [==============================] - 78s 1s/step - loss: 0.3531 - accuracy: 0.8623 - val_loss: 0.2635 - val_accuracy: 0.
         8490
         Epoch 4/50
         54/54 [==============================] - 74s 1s/step - loss: 0.2815 - accuracy: 0.8924 - val_loss: 0.2924 - val_accuracy: 0.
         8698
         Epoch 5/50
         54/54 [==============================] - 74s 1s/step - loss: 0.1998 - accuracy: 0.9207 - val_loss: 0.1823 - val_accuracy: 0.
         9427
         Epoch 6/50
         54/54 [==============================] - 72s 1s/step - loss: 0.1921 - accuracy: 0.9282 - val_loss: 0.1908 - val_accuracy: 0.
         9375
         Epoch 7/50

In [46]: scores = model.evaluate(test_ds)

         8/8 [==============================] - 3s 393ms/step - loss: 0.1107 - accuracy: 0.9698

In [47]: scores

Out[47]: [0.11070696264505386, 0.9698275923728943]
```

Fig(iii) Model Training



Fig(iv) Accuracy and Loss Graph of Model

```
first image to predict
actual label: Potato___Early_blight
1/1 [==============================] - 0s 445ms/step
predicted label: Potato___Early_blight
```

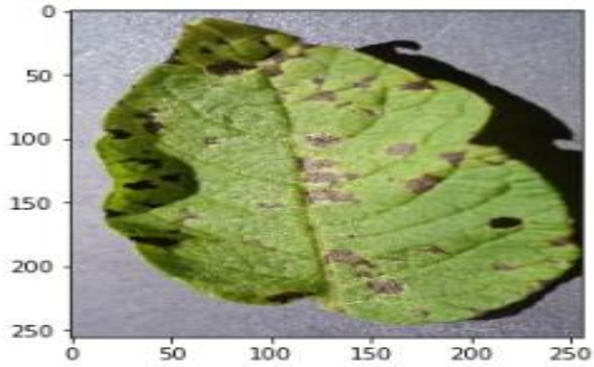


Fig (v) Test Result

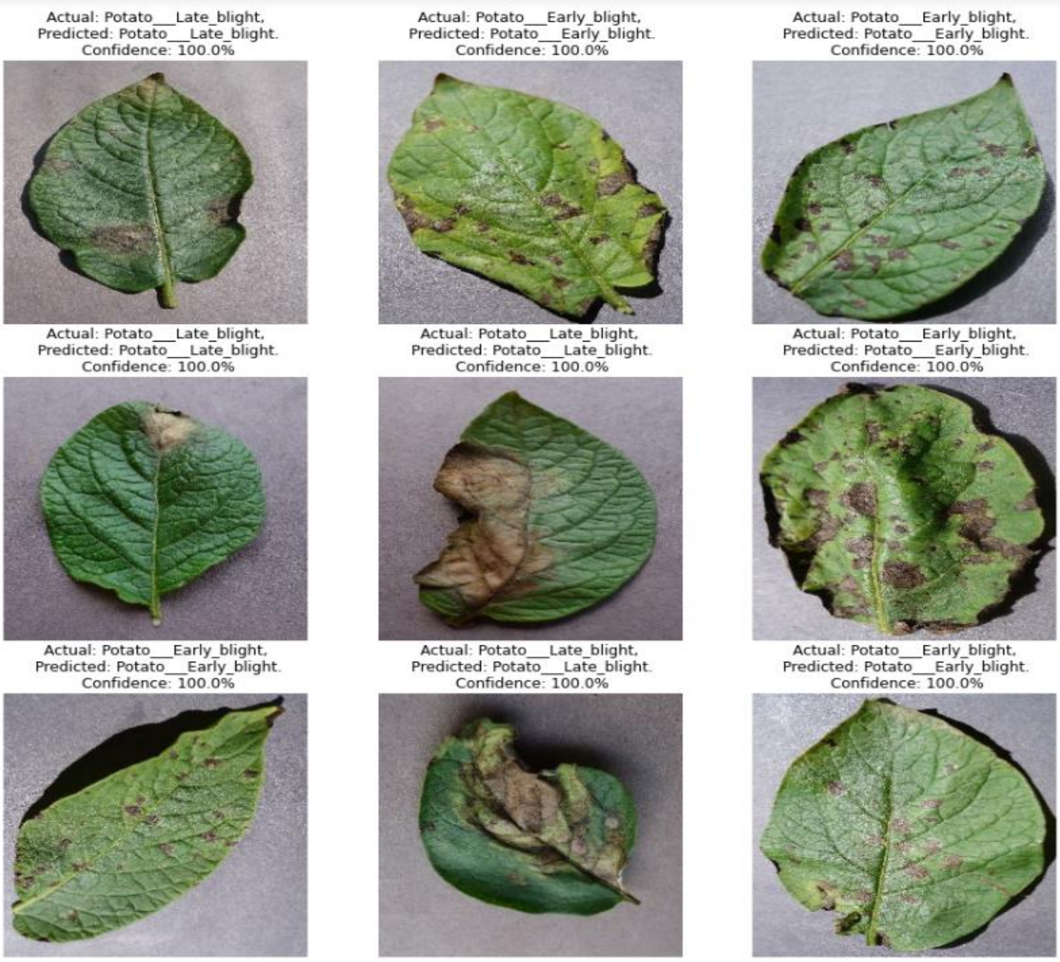


Fig (vi) Test result of various images

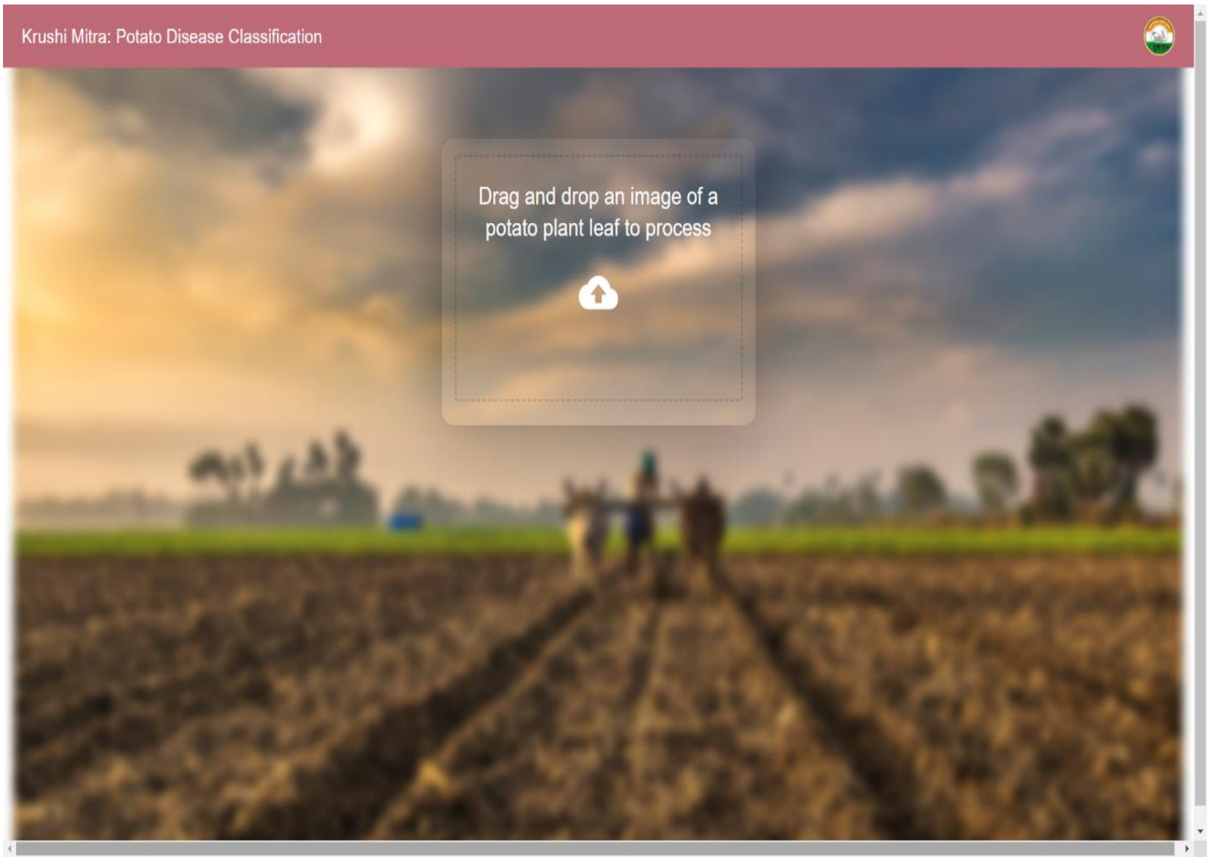


Fig (vii) Krushi Mitra Website

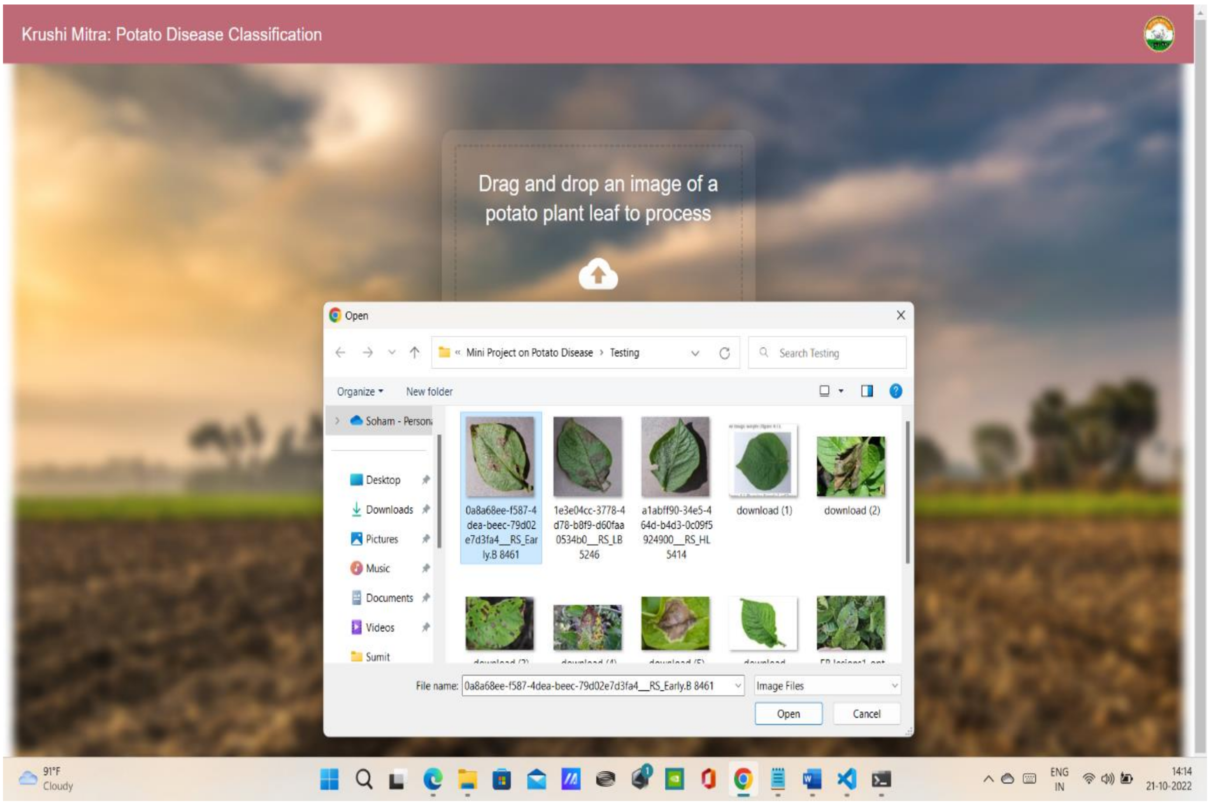


Fig (viii) Drag and drop of the image for testing

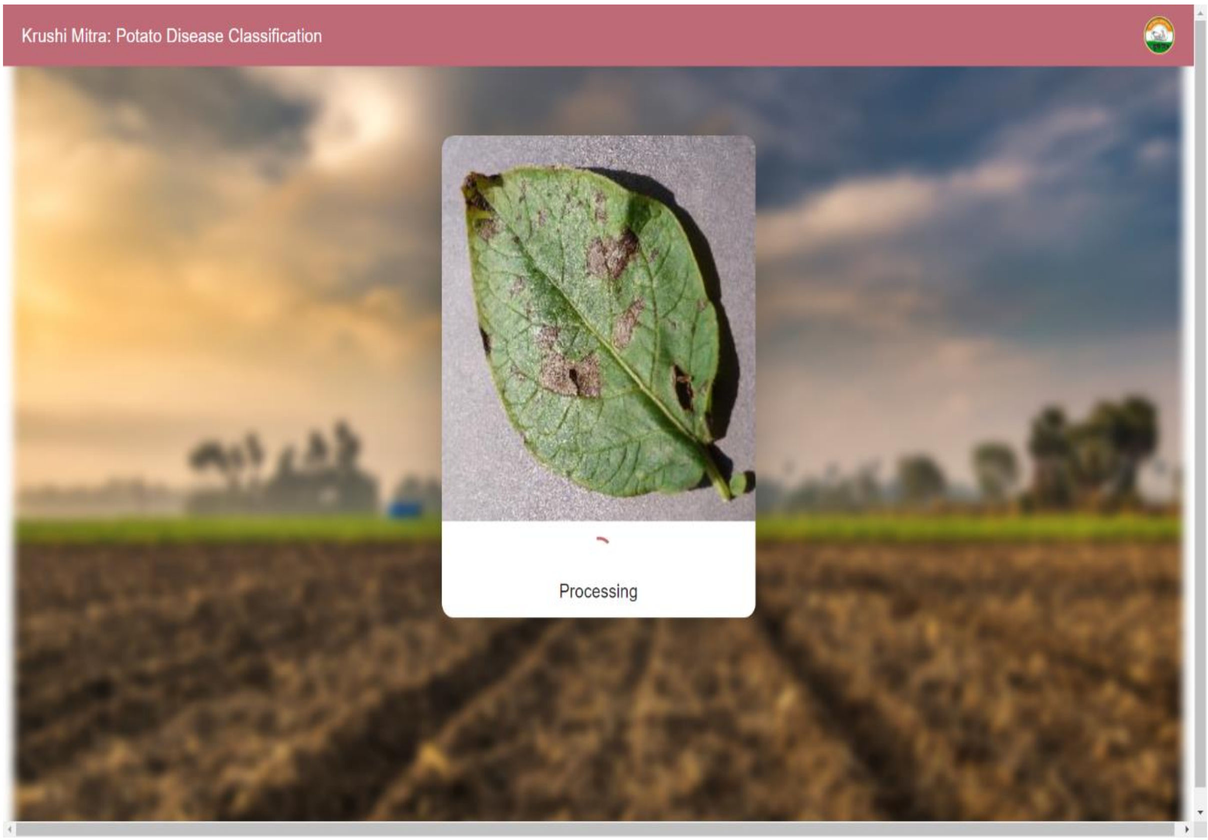


Fig (ix) Image is in the process of testing

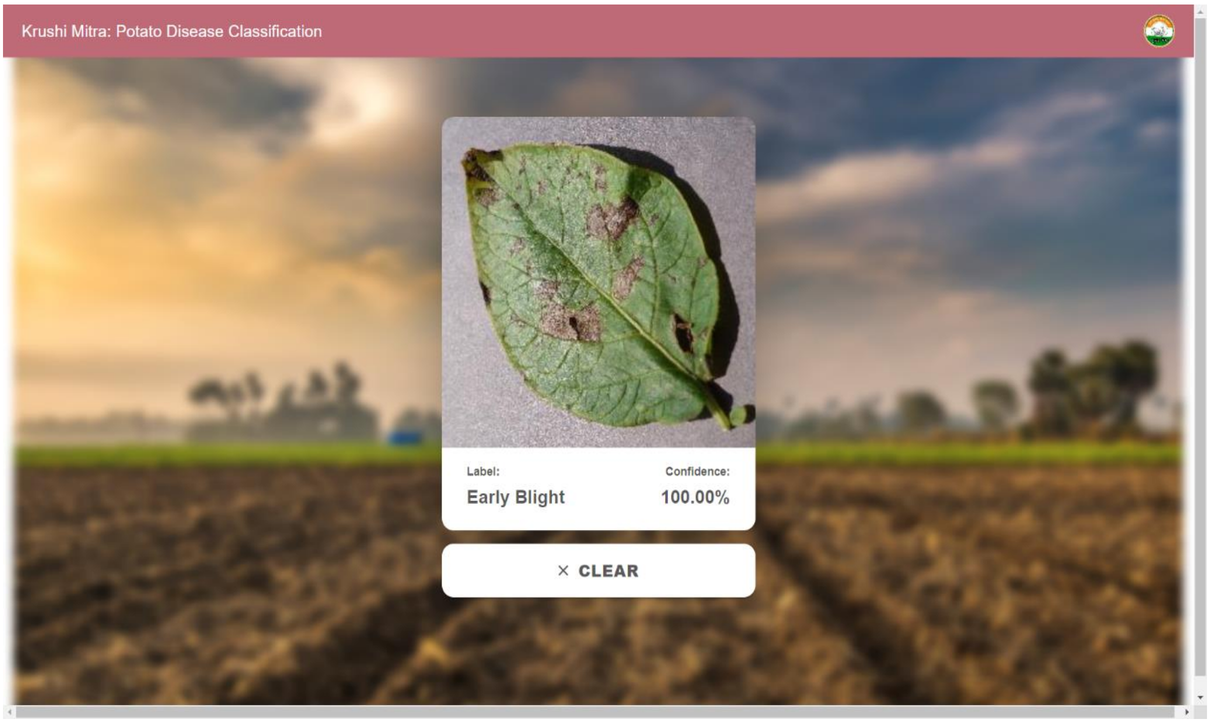


Fig (x) Predicted result of the image

Chapter – Seven

References

IEEE Publications

1. Lightweight Model for Leaf Classification and Disease Detection using TensorFlow Richard Hortizuela School of Advanced Studies Saint Louis University Baguio City, Philippines 2206385@slu.edu.ph
2. Plant Disease Detection Using CNN 1Sumit Kumar, 2Veerendra Chaudhary, 3Ms. Supriya Khaitan Chandra 1,2,3Dept. of Computer Science and Engineering, Glagotias University, India, Uttar Pradesh
3. Plant Leaf Disease Detection and Classification using Image Processing Author(s): *1 Yin Min Oo, 2Nay Chi Htun, Affiliation(s): 1 Department of Information Technology, Pyay Technological University 2Department of Information Technology, Pyay Technological University *Corresponding Author: yinminoo.it.85@gmail.com,
4. Potato Leaf Disease Classification Using Deep Learning Approach (1) Rizqi Amaliatus Sholihati. (2) Eny Kusumawati. (3) Anhar Risnumawan. (4) Indra Adji Sulistijono Politeknik Elektronika Negeri Surabaya.
5. Detection of a Potato Disease (Early Blight) Using Artificial Intelligence, 25 January 2021, by Hassan Afzzaal
6. Potato Disease caused by soft rot erwinias: an overview of pathogenis , 06 February 2002, by M.C.M Perombelon
7. Multi-level deep learning model for Potato Leaf Disease Recognition, 26 August 2021, by Javed Rashid
8. Detection of potato diseases using image segmentation and multiclass support vector machine, 2017 IEEE 30th CCECE, by Anh Dinh

Chapter – Eight
Acknowledgement

An endeavour over a long period can be advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them. In performing our project, we had to take help and guidance from some professors, who deserve our greatest gratitude. The completion of this project gives us immense pleasure. Firstly, we would like to show our gratitude to **Prof. Ranjit Mane, Project Guide, Mumbai University** for giving us the opportunity to widen our vision towards technology building, for giving us the best advice and a good guideline for this mini project through numerous consultations. We would also like to thank and expand our deepest gratitude to all the remaining faculty for being good supporters and the ones who have directly or indirectly guided us in this due course of time.

Many people, especially our classmates, have made valuable comment suggestions on this proposal which gave us inspiration to improve. Lastly, we would like to acknowledge by saying that this project has helped us and given us a better understanding of development.

Project Guide – Prof. Ranjit Mane
Head of Department – Dr. D. R. Ingle
Principal – Dr. Sandhya Jadhav
Supervisor – Prof. Ranjit Mane