*A Progress Report*

*on*

**Plant Disease detection using Deep Learning and image processing**

*carried out as part of the course CSE CS3270 Submitted by*

***Shivansh Syal, Sarthak Ahuja***

***199301241, 199301246***

***VI-CSE***

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

In

**Computer Science & Engineering**

Text

Description automatically generated

**Department of Computer Science & Engineering,**

**School of Computing and IT,**

**Manipal University Jaipur,**

***April 2022***

**Abstract**

The latest generation of convolutional neural networks (CNNs) has achieved impressive results in the field of image classification. By the means of this minor project, we aim to classify the plant diseases by assessing the images of the leaves with the application of Deep learning and image processing and finding the most optimal machine learning algorithm to achieve the same, and create an interface to classify the provided image and generate the appropriate results using the model which may include various types of disease and their respective treatments.

Plant diseases have turned into a major problem as it can cause significant reduction and losses in both quality and quantity of agricultural products.A vast majority of the growing national population depends on agriculture yields. Farmers have wide range of diversity to select suitable fruit or vegetable crops to grow. But the cultivation of these crops for optimum yield and quality produce is highly technical & challenging. It can be improved by the aid of technological support and mechanized farming.

A variety of neuron-wise and layer-wise visualization methods is applied using a CNN, trained with a publicly available plant disease image dataset and the developed model will be able to recognize and classify several different types of plant diseases out of healthy leaves, with the ability to distinguish plant leaves from their surroundings.

In this project our aim is to detect plants disease using a web app which is implemented using image processing. We are creating an web app through which we will capture image which will be analysed by the model and classified and based on its class, The data of the disease is retreived by the database. The captured image will again go through the image processing steps and will be compared with the haarcascade created using the dataset of images. If the images are matched then it will send the results to the user.

Automatic detection of plant diseases is a very important research topic as it may prove the benefits in monitoring large fields of crops, and thus automatically detect the symptoms of diseases as soon as they appear on plant leaves. Therefore looking for fast, automatic, less expensive and accurate method to detect plant disease cases is of great realistic significance Machine learning based detection and recognition of plant diseases can provide extensive clues to identify and treat the diseases in its very early stages

**Table of Contents**

1. Introduction 1

1.1 Scope of the Work 1

1.2 Product Scenarios 1

2. Requirement Analysis 2

2.1 Functional Requirements 2

2.2 Non-functional Requirements 3

* 1. Use Case Scenarios 3

3. System Design 4

3.1 Design Goals 4

* 1. System Architecture 4

3.3 Detailed Design Methodologies

4. Work Done 4

4.1 Development Enviroment 5

4.2 Results and Discussion 8

4.3 Individual Contribution of Project Members 9

5. Conclusion and Future Plan 9

**1. Introduction**

**1.1 Scope of the Work**

The project is aimed at being a web application to identify to identify plant disease using user inputted leaf images. This project will be available on GitHub as an open-source project for other people to improve this project. The application will also provide information that includes possible treatment of diseases and additional symptoms of the disease.

**1.2 Product Scenarios**

This project aims to improve agriculture work by providing detailed information on crops and diseases. This application will use user inputted images to identify and provide information about various plant diseases. The information will incudes possible treatment of diseases and additional symptoms of the disease

**2. Requirement Analysis**

**Hardware Requirements**

For the hardware requirements, the logical characteristics of each interface are defined such as memory restrictions, the processor, RAM size etc.

Minimum Hardware Requirements

* + Operating System: Windows 7/8/8.1/10/11
  + Processor: Intel i3 or higher or any equivalent AMD processor
  + Memory: 4GB or more

Recommended Hardware Requirements

* + Operating System: Windows 7/8/8.1/10/11
  + Processor: Intel i3 or higher or any equivalent AMD processor
  + Memory: 8GB or more

**Software Requirements**

Any operating system with browser support is the primary requirement for software development. The system must be connected via LAN and an active internet connection is necessary.

**2.1 Functional Requirements**

**2.1.1 Data Sets**

* The leaf images of peach plants have been extracted from the [New Plant Disease Dataset](https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset)
* The datasets image data for many plant leaves diseases and each of them has a healthy class contains many leaf images, and the diseased class comprise of diseased leaf images.
* This dataset consists of about 87K rgb images of healthy and diseased crop leaves which is categorized into 38 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purpose.

**2.1.2 Database**

* Database will be used to store information about diseases and their remedies which will be displayed with results.
* We are using firebase’s cloud Firestore for storing and indexing the **Leaf type, Leaf diseases, symptoms, specifications and diseases prevention** and **cure.**

**2.1.3 Image processing and labelling**

* Images intended to be used as datasets for deep neural network classifiers are to be preprocessed. Preprocessing involves cropping all the images, in order to highlight the region of interest, and using CV2, adjusting the size and colors of the image and making more changes according to the dataset.

**2.1.4 Augmentation Process**

* The main purpose of applying augmentation is to increase the dataset and introduce slight distortion to the images which helps in reducing overfitting during the training stage. The image augmentation contained one of several transformation techniques including affine transformation, perspective transformation, and simple image rotations. Affine transformations were applied to express translations and rotations (linear transformations and vector addition, resp.) where all parallel lines in the original image are still parallel in the output image.

**2.1.5 Neural Network Training**

* To train the model, the leaf images of plants will be randomly divided such that 70% of them form the training dataset, and 30% form the testing dataset.
* In order to get better feature extraction, final images intended to be used as dataset for deep neural network classifier were preprocessed in order to gain consistency. Procedure of image preprocessing involved cropping of all the images manually, making the square around the region of interest (plant leaves). Images with smaller resolution and dimension less than 500 px were not considered as valid images for the dataset.

**2.1.6 Web Application**

* Web Application created using react with redux to develop the front-end of the application where a user can upload the image of the plant (or use the camera), which is then processed by the developed model to find out any disease in the plant, and as a result, return the ailments and treatment of the unhealthy plant. We are using firebase as the backend service provider and use its hosting, cloud storage and firestore SDKs

**2.2 Non-functional Requirements**

**2.2.1 Google authentication**

Authentication with google implemented which can be later used to collect user data to further improve the project.

**2.2.2 Sample Images**

Sample images included in the web application to make it easier to observe the working of the application.

**2.2.3 Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. There is a customer friendly system which can be accessed by people around the world, and it should work for 24 hours. In case of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the organizer. Then the service will be restarted.

**2.2.2 Reliability**

The reliability of the overall project depends on the reliability of the separate components. The mam pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also, the system will be functioning inside a container. Thus, the overall stability of the system depends on the stability of container and its underlying operating system.

**2.3 Use Case Scenarios**

1. **Farmers-** This application can be used to make it easier for farmer to identify and get detailed information on various diseases using pictures of plant leaves This application can be used to make it easier for farmer to identify and get detailed information on various diseases using pictures of plant leavesThis application can be used to make it easier for farmer to identify and get detailed information on various diseases using pictures of plant leaves
2. **Botanical Research-** this application can also be used by researchers as an quick way to identify different plant diseases.
3. **Gardeners-** Gardeners can use this application to identify and diagnose different plant diseases.

**3. System Design**

**3.1 Design Goals**

**3.1.1 Performance** : The end-user software should be able to run on low-end hardware with extremely high confidence metrics

**3.1.2 Technology**: The Algorithms used to develop this project are extremely sophisticated and have gone through numerous iterations to yield best possible attributes.

**3.1.3 Portability** : The Project can be installed and run on most major operating systems and potentially be ported over to IoT devices that can help with reaching out to a huge number of nodes that can operate over a network if so needed.

**3.2 System Architecture**

Design

**4. Work Done**

**4.1 Development Environment**

Most of the Project’s development has been done on Google Collab platform. It provides complementary access to a cutting edge Graphics Processing Unit (GPU) and a succinct integrated development environment with Jupiter notebook support.

All the dependencies, modules and version controlling is taken care of by the Collaboratory and it makes the process of Training our model more reliable, safer and faster.

Design software – Figma

IDE- VSCode

Frontend framework- React

Libraries – Redux, React router, Material UI, Firebase, React google login,keras, tensorflow. Matplotlib, cv2, sklearn

Database- Firebase firestore

**DataSet**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Screenshots from the CNN model**

**Text

Description automatically generated** **Text

Description automatically generated**

**Chart, line chart

Description automatically generated**

**A picture containing diagram

Description automatically generated**

**Screenshots from the Application**

**Graphical user interface, website

Description automatically generated**

**A picture containing graphical user interface

Description automatically generated**

Sample Plant images

A picture containing grass, different

Description automatically generated

Results

Text

Description automatically generated

**4.2 Results and Discussion**

After fine-tuning the parameters of the network, an overall accuracy of 97.3% was achieved, after the 100th training iteration (96.8% without fine-tuning). Even after the 30th training iteration high accuracy results were achieved with exceedingly reduced loss, but after the 60th iteration, the balance in accuracy and loss was carried out in high accuracy.

it is observed that the trained model’s accuracy was slightly less for classes with lower number of images in the training dataset. Achieved accuracy was in range from 91.11% for peach, powdery mildew, up to 97.21% for background images. High accuracy of model’s prediction of background images allows good separation of plants leaves and the surroundings.

**Chart, bar chart

Description automatically generated**

CNNs provide unparalleled performance in tasks related to the classification and detection of crop diseases. They can manage complex issues in difficult imaging conditions. Their robustness may now allow them to emerge from the research environment and become part of operational tools. However, before tools for expertise assistance and automatic screening become a reality, a few steps still need to be tested and integrated. In this section, we first discuss the best practices to adopt all along the development chain so that trained models can handle the real-world complexities of agricultural and phytosanitary problems. We then identify the elements to be further addressed to make such tools fully operational, including possible research directions.

**4.3 Individual Contribution of Project Members**

Shivansh has so far worked on the front end user interface (UI) design of the web application using a widely popular design tool Figma. Moreover, Shivansh has also taken in consideration the user experience (UX), has also done user research, competitive design analysis, and created design style guides. Sarthak will be finishing off the front-end web development and database and model integration of the application soon.

We both worked on research and ideation which involves all the features and the use cases of the web platform, and brainstorming the same, and building and deployment of the CNN model with TensorFlow and Keras.

**5. Conclusion and Future Plan**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Months | | | | | | | | |
|  | January | February | | March | | April | | May | |
| **Schedule for Project Work** | 15-30 | 1 - 15 | 15 - 30 | 1-15 | 15-30 | 1-15 | 15-30 | 1-15 | 15-30 |
| 1 | Initial Research |  |  |  |  |  |  |  |  |  |
| 2 | Synopsis submission |  |  |  |  |  |  |  |  |  |
| 3 | Setting up basic requirements |  |  |  |  |  |  |  |  |  |
| 4 | Image pre-processing |  |  |  |  |  |  |  |  |  |
| 5 | Preparing a trainable dataset |  |  |  |  |  |  |  |  |  |
| 6 | Mid Term Presentation |  |  |  |  |  |  |  |  |  |
| 7 | Develop the web application |  |  |  |  |  |  |  |  |  |
| 8 | Using Model on test dataset |  |  |  |  |  |  |  |  |  |
| 9 | Project Report |  |  |  |  |  |  |  |  |  |
| 10 | Final Presentation |  |  |  |  |  |  |  |  |  |

Based on the investigation, grayscale pictures are simple to process and execute. They have superior clarity and suited for investigation than RGB pictures. These types of pictures

will be utilized to dissect and determination the plant leaves illnesses and decides the maladies level of the plant takes off. The point of this proposition was to create a user friendly computerized framework for the agriculturists that will help

them in deciding location illnesses of takes off without bringing an master to the field.

A web application for detection and classification of plant disease using deep learning was developed in this study. The web application uses CNN model to accurately and efficiently detect diseases using leaf images. The proposed web application is designed to run as a standalone application on a smartphone. Experiments on leaf images show that the application is able to achieve high accuracy on detecting various types of common leaf diseases.

Deep learning techniques for crop diseases identification are promising for the development of new agricultural tools that could contribute to a more sustainable and secure food production. This lack of conformity may lead to poor generalization capabilities for unfamiliar data samples and/or imaging conditions, which lowers the practical use of the trained models.

In the future we will focus on gathering images for enriching the database and improving accuracy of the model using different techniques of fine-tuning and augmentation

In addition, we will include expanding the model's use by training it for plant disease recognition on larger land areas, merging aerial photographs of orchards and vineyards obtained by drones with object detection using convolution neural networks. We hope that by expanding this study, they will be able to have a positive influence on long-term development and crop quality for future generations.

***References***

1. R. Balodi, S. Bisht, A. Ghatak, and K. H. Rao, “Plant disease diagnosis: Technological advancements and challenges,” *Indian Phytopathology*, vol. 70, no. 3, pp. 275–281, 2017.
2. F. Martinelli, R. Scalenghe, S. Davino et al., “Advanced methods of plant disease detection. A review,” *Agronomy for Sustainable Development*, vol. 35, no. 1, pp. 1–25, 2015.
3. J. S. West, C. Bravo, R. Oberti, D. Lemaire, D. Moshou, and H. A. McCartney, “The potential of optical canopy measurement for targeted control of field crop diseases,” *Annual Review of Phytopathology*, vol. 41, pp. 593–614, 2003.
4. A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, “Machine learning for high-throughput stress phenotyping in plants,” *Trends in Plant Science*, vol. 21, no. 2, pp. 110–124, 2016.
5. A. Johannes, A. Picon, A. Alvarez-Gila et al., “Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case,” *Computers and Electronics in Agriculture*, vol. 138, pp. 200–209, 2017.
6. S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
7. J. Amara, B. Bouaziz, and A. Algergawy, “A deep learning-based approach for banana leaf diseases classification,” in *Proceedings of the Datenbanksysteme für Business, Technologie und Web (BTW '17) - Workshopband*, 2017.
8. K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
9. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3289801, 11 pages, 2016.
10. G. Wang, Y. Sun, and J. Wang, “Automatic image-based plant disease severity estimation using deep learning,” *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 2917536, 8 pages, 2017.
11. A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, “Deep learning for image-based cassava disease detection,” *Frontiers in Plant Science*, vol. 8, p. 1852, 2017.
12. A. Fuentes, S. Yoon, S. Kim, and D. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition,” *Sensors*, vol. 17, no. 9, p. 2022, 2017.
13. E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, “Basic investigation on a robust and practical plant diagnostic system,” in *Proceedings of 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 989–992, 2016.
14. K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
15. Y. LeCun, B. Boser, J. S. Denker et al., “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.