LEAF DISEASE DETECTION USING DEEP LEARNING MODELS

A MINOR PROJECT REPORT

Submitted by

KARNA RUTHWIK REDDY(RA2011026010232)
GUNTI MEGHANATH(RA2011026010219)

Under the guidance of
Mr. S JOSEPH JAMES
ASSISTANT PROFESSOR, DEPARTMENT OF CINTEL

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE & ENGINEERING WITH SPECIALIZATION IN AI&ML
of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M.Nagar, Kattankulathur, Chengalpattu District

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that 18CSP107L minor project report titled "**LEAF DISEASE DETECTION USING DEEPLEARNING MODELS**" is the bonafide work of "KARNA RUTHWIK REDDY (RA2011026010232), GUNTI MEGHANATH(RA201026010219)" who carried out the minor project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mr. S JOSEPH JAMES
GUIDE
ASSISTANT PROFESSOR
Dept of Computational Intelligence

SIGNATURE

Dr.R. Annie Uthra
HEAD OF THE DEPARTMENT
PROFESSOR
Dept of Computational Intelligence

SIGNATURE

Dr. Lakshmi C
PANEL HEAD
PROFESSOR
Dept of Computational Intelligence

ABSTRACT

The economy plays a pivotal role in agricultural productivity. Plant diseases are a common concern in agriculture, but their detection has become more viable due to economic advancements. Presently, the scrutiny of plant disease detection is expanding, especially in the surveillance of vast and diverse crop fields. Farmers often encounter challenges when switching from one disease management strategy to another. The standard approach for disease detection involves identifying or spotting tomato leaf diseases, which requires the expertise of surveillance and monitoring professionals. Failing to take proper control measures can significantly impact plant health, affecting both crop quality and quantity.

Keyword: Plant leaf disease images, deep learning, Machine Learning, Dense Net, ANN, CNN, Resnet50.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 Motivation:

The adoption of deep learning for leaf disease detection in the agricultural sector is driven by its transformative potential. Through the utilization of sophisticated algorithms and extensive datasets, deep learning models have the capacity to provide precise and rapid identification of diseases affecting plant leaves. This technology offers the promise of early and accurate disease detection, facilitating timely interventions that can effectively mitigate crop losses, optimize resource allocation, and bolster food security. Furthermore, the automation of disease detection processes reduces the need for extensive manual labor and enhances scalability. As a result, deep learning becomes a pivotal tool in modern agriculture, contributing to increased yields, the promotion of sustainable farming practices, and ultimately playing a role in global food production and the well-being of farmers and communities.

1.2 Problem Statement:

The difficulty in applying deep learning to the identification of leaf diseases is in creating an accurate and effective system that can recognize plant illnesses from photos of leaves. This solution is designed to elevate agricultural productivity by enabling the early detection and diagnosis of diseases, facilitating timely interventions to prevent crop losses. Leveraging deep learning algorithms, the system's objectives include distinguishing between healthy and diseased leaves, classifying various diseases, and furnishing farmers with actionable insights for effective disease management and the maintenance of crop health. In the end, the primary goal is to enhance agricultural methods and play a substantial role in ensuring global food security by promptly and accurately identifying plant diseases.

1.3 Objective of the Project:

The primary goal of the "Leaf Disease Detection Using Deep Learning" project is to establish a highly accurate and efficient automated system for the early detection of diseases in plant leaves. Utilizing cutting-edge deep learning methods, the central goal of the project is to create a model with the ability to analyze leaf images and precisely categorize them according to the existence of diseases. This system is envisioned as a valuable tool for farmers and agricultural professionals, facilitating timely disease identification, which, in turn, enables targeted treatments and minimizes potential crop damage.

1.4 Scope:

Deep learning-based leaf disease detection entails training models to closely inspect leaf photos and quickly identify abnormalities or diseases. Within this field, deep learning algorithms, specifically Convolutional Neural Networks (CNNs), assume a crucial role. These algorithms are adept at learning intricate patterns and features from a comprehensive dataset that encompasses both healthy and diseased leaves. Their ability to automate the detection process ensure high level of accuracy. The applications of such technology extend to precision agriculture, where it allows for timely interventions to prevent crop damage and ultimately augments agricultural productivity. Consequently, it plays a significant role in fostering food security and promoting sustainable farming practices. The continuous progress in research and the evolution of deep learning techniques further refine and expand the capabilities of leaf disease detection systems, promising even greater levels of accuracy and efficiency.

1.5 Project Introduction:

Welcome to our ground breaking initiative, "Leaf Disease Detection Using Deep Learning," a project with the vision to transform agricultural practices and secure sustainable crop yields. Agriculture is the cornerstone of our society, and its optimization is paramount for global food security. However, the presence of plant diseases presents a significant challenge to crop health and yield.

Deep learning, a subset of artificial intelligence, has exhibited immense promise across various domains, including image recognition. With this potential in mind, we are committed to constructing a robust model with the capability to scrutinize images of plant leaves and accurately pinpoint disease symptoms. Through this endeavor, our objective is to empower farmers with the ability to identify diseases in their early stages, take swift and informed actions, and effectively curtail crop damage.

2.1 LITERATURE REVIEW

2.1 Related Work:

[1] Maniyath, S. R., P V, V., M, N., R, P., N, P. B., N, S., & Hebbar: Crop diseases pose a significant threat to food security, particularly in regions lacking the necessary infrastructure for their rapid detection. The advent of precise techniques in leaf-based image classification has yielded promising outcomes. This paper utilizes the Random Forest algorithm to distinguish between healthy and diseased leaves, leveraging datasets specifically crafted for this purpose. The presented paper involves various implementation phases, encompassing dataset creation, feature extraction, classifier training, and classification. The datasets, containing both diseased and healthy leaves, are collectively trained using Random Forest to categorize the images accordingly. Feature extraction is conducted utilizing the Histogram of Oriented Gradients (HOG) method.

Summary: In the initial phase, the image processing workflow commences by converting the RGB image into a grayscale image. This transformation is a prerequisite because certain shape descriptors like Hu moments and Haralick features are designed to be computed over a single channel. Hence, converting the RGB image to grayscale is essential before the subsequent computation of these features. Furthermore, the calculation of a histogram necessitates a preliminaryconversion of the image to HSV (hue, saturation, and value) format. This HSVconversion is performed to facilitate the histogram analysis.

[2] Hassan, S. M., Maji, A. K., Jasiński, M., Leonowicz, Z., & Jasińska, E.: Detecting and preventing crop diseases in a timely manner is of utmost importance for enhancing agricultural production. This paper introduces a novel method using deep Convolutional Neural Network (CNN) models for identifying and diagnosing plant diseases through leaf images. While CNNs have achieved substantial success in machine vision, conventional CNN models typically entail a large number of parameters and considerable computational expenses. In this research, the authors have chosen to employ depth-separable convolution instead of standard convolution to decrease the number of parameters and alleviate computational burdens.

Summary:

On average, the training time for the MobileNetV2 and EfficientNetB0 architectures was comparatively shorter than other deep learning models, with training durationsof 565 and 545 seconds per epoch, respectively, for color images. In comparison to alternative deep learning approaches, our implemented model demonstrated superior predictive capability, exhibiting enhanced accuracy and loss performance. Additionally, the MobileNetV2 architecture is optimized for efficiency, minimizing parameters and operations, making it well-suited for deployment on mobile devices. This research underscores the potential of deep learning models in addressing the pressing issue of plant disease detection, offering enhanced accuracy, efficiency, and practicality for real-world agricultural applications.

[3] Muhammad E.H. Chowdhury, Tawsifur Rahman, Amith Khandakar, Nabil Ibtehaz,: Plants serve as a crucial primary food source for the global population, and the repercussions of plant diseases causing production losses are a substantial concern. Tackling this challenge requires ongoing monitoring, yet manual plant disease monitoring is labor-intensive and prone to errors. Early disease detection in plants, achieved through the application of computer vision.

Summary: The journey through the various stages of applying machine learning to agriculture is an exploration of infinite possibilities, accompanied by illustrative case studies. Notably, within this landscape, the utilization of state-of-the-art pre-trained convolutional neural network (CNN) models, such as ResNet, MobileNet, DenseNet201, and InceptionV3, has demonstrated remarkable efficiency in disease classification based on plant leaf images.

Moreover, the suggested framework can be incorporated with a feedback system, offering invaluable insights, treatment recommendations, disease prevention strategies, and effective control techniques.This holistic approach is envisioned to result in improved crop yields, fostering the goal of enhanced agricultural productivityand food security.

[4] Zhang, Y., Song, C., & Zhang., D. In the quest to improve the accuracy of recognition models for identifying crop disease leaves and precisely localizing diseased leaves, this paper introduces an enhanced Faster R-CNN (Region-based Convolutional Neural Network) approach for detecting healthy tomato leaves and four specific diseases: powdery mildew, blight, leaf mold fungus, and ToMV. The proposed method integrates several key innovations to enhance the disease detection process.

Initially, the conventional VGG16 model, employed for image feature extraction, is substituted with a deeper depth residual network, facilitating the extraction of more intricate disease-related features.This deeper feature extraction process contributes to more accurate disease identification.

Secondly, the paper utilizes the k-means clustering algorithm to cluster the bounding boxes, a pivotal aspect of object detection.The clustering results are utilized to enhance the anchor boxes, making them align more closely with the actual bounding boxes in the dataset. Crop disease detection is fundamental to ensuring crop quality and preventing disease-related losses. Conventional detection methods frequently depend on manual observation, leading to low detection efficiency and reliability.

Summary: The paper introduces the Faster R-CNN algorithm as a means to detect diseased tomato leaves, offering a dual capability to recognize tomato diseases and pinpoint the locations of affected tomato leaves. To enhance the algorithm's performance and ensure close alignment between the anchors and the ground truth of the dataset, the k-means clustering algorithm is employed to cluster the bounding boxes of tomato disease images. The anchor boxes are then refined based on the clustering results.In the process of feature extraction, ResNet101 is selected as a replacement for VGG16, enabling the extraction of deeper and more meaningful features associated with tomato diseases.

The experimental results corroborate the effectiveness of this method in detecting and recognizing tomato diseases, demonstrating superior detection accuracy compared to the original Faster R-CNN. It is crucial to highlight that the dataset utilized in this study comprises laboratory data, enabling the detection of a single leaf disease in each image.Future research endeavors will encompass the collection of images from natural plant environments, enabling more comprehensive detection.

[5] Aakanksha Rastogi; Ritika Arora; Shanu Sharma:

The proposed system consists of two distinct phases. In the first phase, the plant is identified based on the characteristics of its leaves. This process encompasses preprocessing of leaf images, extracting relevant features, and employing Artificial Neural Network (ANN) for training and classification to recognize the type of plant based on its leaves.

In the second phase, the system identifies and classifies the disease affecting the leaf. This process involves employing K-Means-based segmentation to isolate the affected area, extracting features from the diseased portion, and utilizing an Artificial Neural Network (ANN) for classifying the specific disease. Subsequently, the disease is graded based on the extent of its presence on the leaf.

Summary: (ANN) for classifying the specific disease. Subsequently, the disease is graded based on the extent of its presence on the leaf.

[6] Nithish Kannan E.; Kaushik M.; Prakash P.; Ajay R.; Veni S.: The proposed system comprises two distinct phases: the initial stage involves identifying plants by analyzing their leaf characteristics. This phase encompasses preprocessing leaf images, extracting pertinent features, and using an Artificial Neural Network (ANN) for training and classification to identify plant types based on their unique leaf attributes. The subsequent phase focuses on identifying and classifying leaf diseases, utilizing K-Means-based segmentation to isolate affected areas, extracting features from these regions, and employing an ANN for disease classification. Additionally, the system grades the severity of the disease based on its extent of impact on the leaf.

Networks (CNNs), a subset of deep neural networks. Initially, the dataset is organized to isolate tomato leaves. Transfer learning is employed by importing a pre- trained model, ResNet-50, which is then fine-tuned for our specific classification task. To improve the model's quality and bring it closer to real-world disease detection, data augmentation techniques are applied. With these steps in mind, a tomato leaf disease detection model is created using PyTorch, leveraging deep CNNs.

Summary: This project aims to detect diseases in tomato leaves using Convolutional Neural Networks (CNNs). It begins by preprocessing the dataset to isolate tomato leaves. Transfer learning is employed by fine-tuning a pre-trained model, ResNet-50, to address the classification problem. Data augmentation techniques are used to enhance the model's performance. The resulting tomato leaf disease detection model, implemented in PyTorch with deep CNNs, is evaluated on testing dataset. The model achieves a high accuracy rate of 97% and can identify six prevalent tomato crop diseases.

[6] Nitesh Agrawal; Jyoti Singhai; Dheeraj K. Agarwal: In today's technology- driven era, many fields are seeking efficient and cost-effective software solutions to replace manual decision-making processes. Support Vector Machine (SVM) was originally designed for binary classification but can be adapted for multi-class problems with some modifications. This project aims to enhance the classification of leaf diseases.

Traditionally, most efforts have focused on extracting statistical features from RGB signals converted into LAB color space. However, this project introduces a novel approach by incorporating the unique properties of the HSI (Hue, Saturation, Intensity) color model. HSI is known for its ability to maintain consistent hue information even when the lighting conditions change in an image. Therefore, certain HSI image properties are added to the database. SVM is then utilized for classification, effectively working in a higher-dimensional space with the added properties from the HSI images to improve the accuracy of leaf disease classification.

Summary: In the modern age of technology, various fields are looking for software- based solutions to replace manual decision-making processes and reduce costs. This project focuses on enhancing the classification of leaf diseases using Support Vector Machines (SVM). While SVM was originally designed for binary classification, it can be adapted for multi-class scenarios. The project introduces a novel approach by incorporating the unique properties of the HSI (Hue, Saturation, Intensity) color model, known for its ability to maintain consistent hue information despite changes in lighting conditions. These HSI properties are added to the database, and SVM is employed for classification in a higher-dimensional space, improving the accuracy of leaf disease classification.

[7] Deepalakshmi P., Prudhvi Krishna T., Siri Chandana S., Lavanya K., Parvathaneni Naga Srinivasu: Agriculture plays a central role in India's economic development due to factors like fertile soil, favorable weather conditions, and the economic value of crops. Farmers carefully choose suitable crops for each season to meet the growing population's needs. To address the increasing demand for food production, the agricultural industry seeks innovative methods that can enhance yields while reducing investment. Precision agriculture is an emerging technology that holds promise in improving farming practices. Notable applications of precision agriculture include pest and weed detection, as well as the identification of plant leaf diseases.

Summary:

Precision agricultures one such technology, with applications in pest and weed detection and the identification of plant leaf diseases.

This research paper focuses on using a CNN algorithm to identify diseased and healthy leaves of different plants from input images. The CNN extracts features from these images, enabling accurate classification of images into their respective classes. The system achieves an impressive accuracy rate of over 94.5% and typically takes around 3.8 seconds to identify image classes.

[8] Vijayakumar Ponnusamy; Amrith Coumaran; Akhash Subramanian Shunmugam; Kritin Rajaram; Sanoj Senthilvelavan: In today's fast-paced world, the ability to keenly observe and recognize patterns in small details is a challenging task. These patterns often hold valuable information for humans. To harness these regularities and predict future activities, various Artificial Neural Network architectures with high levels of accuracy are available. However, a significant drawback is that these architectures require high-performance GPUs, which can increase the overall system's size. Current systems capable of processing Machine Learning algorithms are either costly or lack portability.

To address this, we introduce a Smart Glass device that combines mobility with the advanced You Only Look Once (YOLOv3) object detection system. This Smart Glass is designed for highly accurate real-time binary classification of data. By training this architecture with agricultural data, the wearable device becomes capable of identifying Healthy and Unhealthy plant leaves in real-time. Researchers can also adapt the architecture by training it with different datasets to provide solutions for a wide range of problems across various industries, including Agriculture, Healthcare, and the Automobile Industry.

Summary: In today's fast-paced world, recognizing important patterns in small details can be challenging. Artificial Neural Networks offer high accuracy but often require powerful GPUs, making systems expensive and non-portable. To address this, a Smart Glass device is introduced, integrating mobility with the advanced YOLOv3 object detection system.

This device can perform real-time binary classification, such as identifying Healthy and Unhealthy plant leaves after training with agricultural data. Researchers can adapt the architecture for various industries, including Agriculture, Healthcare, and the Automobile Industry, by training it with different datasets. This innovation offers a practical and versatile solution for pattern recognition and data classification.

[9] Dinesh Kumar R, Prema V, Radhika R, Queen Mercy C.A, Ramya S: Green plants play a crucial role in maintaining environmental sustainability and long-term ecosystem health. This project proposes a system that utilizes a Raspberry Pi to detect the health status of plants and send alerts to farmers via email. The primary objective is the early detection of plant diseases, with a particular focus on employing image processing techniques. The process involves several steps, starting from capturing leaf images and culminating in disease identification using Raspberry Pi. This device serves as an interface for the camera and display, with data storage in the cloud.

Summary: This project focuses on the importance of green plants in maintaining environmental sustainability and proposes a system that uses a Raspberry Pi for early detection of plant diseases. The system captures leaf images, processes them through various stages, and identifies diseases, sending alerts to farmers via email. Raspberry Pi facilitates camera interfacing, data display, and cloud-based data storage, allowing for real-time monitoring of crops. This approach reduces labor requirements, lowers costs, and enhances productivity. The automatic disease symptom detection contributes to agricultural product quality and chemical application efficiency, making it a valuable advancement in agriculture.

[10] **Shima Ramesh; Ramachandra Hebbar; Niveditha M.; Pooja R.; Prasad Bhat N.; Shashank N.; Vinod P.V.:** Crop diseases pose a significant threat to food security, but their swift identification remains a challenge in many regions due to inadequate infrastructure. Recent advancements in leaf-based image classification techniques have yielded promising results. This paper employs the Random Forest algorithm to distinguish between healthy and diseased leaves using datasets createdfor this purpose. The paper outlines several key phases, including dataset construction, feature extraction, classifier training, and classification.

Summary: Crop diseases pose a substantial threat to food security, particularly in regions lacking the necessary infrastructure for quick identification. Recent advances in leaf-based image classification techniques offer promising solutions. This paper utilizes the Random Forest algorithm to differentiate between healthy and diseased leaves, using datasets specially created for this purpose. The approach involves several critical phases, including dataset construction, feature extraction, classifier training, and image classification.

The datasets consist of images of both healthy and diseased leaves, and the Random Forest algorithm is employed to collectively train and classify these images. Feature extraction relies on the Histogram of Oriented Gradients (HOG) technique. In summary, the application of machine learning to train on extensive publicly available datasets provides an effective method for the large-scale detection of plant diseases.

[11] **Iftikhar Ahmad,[1]Muhammad Hamid,[1]Suhail Yousaf,[1]Syed Tanveer Shah,[2]and:** The cultivation of vegetables and fruits is vital for sustaining the global population of 7.5 billion, playing a crucial role in maintaining life on Earth. However, the widespread use of chemicals like fungicides and bactericides to combat plant.

The prevalence of crop diseases on a large scale negatively impacts both the quantity and quality of production. To address this issue, our research focuses on developing a reliable and swift method for the early identification and diagnosis of tomato leaf diseases using convolutional neural network (CNN) techniques.

We explore four CNN architectures—VGG-16, VGG-19, ResNet, and Inception V3—employing feature extraction and parameter tuning to effectively classify and identify tomato leaf diseases. Our evaluation involves testing these models on two datasets: one from a controlled laboratory environment and another consisting of data collected from real-world field conditions. Interestingly, we find that all architectures exhibit superior performance on the laboratory-based dataset compared to the field- based data, with performance metrics showing a variance in the range of 10%–15%. Inception V3 emerges as the top-performing algorithm on both datasets, showcasing its effectiveness in disease identification and classification.

Summary: Our research focuses on using convolutional neural network (CNN) techniques to identify and classify tomato leaf diseases, addressing the negative impact of chemical use on crops. We explore four CNN architectures and test them on both laboratory and field datasets. While all architectures perform better on the laboratory-based dataset, Inception V3 stands out as the best-performing algorithm on both datasets. The study highlights the need for reliable methods to detect plant diseases early and the challenges in adapting models to real-world field conditions.

[12] Manpreet Kaur; Rekha Bhatia: Detecting plant leaf diseases at an early stage is crucial for the Indian economy, as 10-30% of crops are damaged without early detection. Various methods are employed for different crops, and this study focuses on using a pretrained Deep Learning Model to detect and classify Tomato Leaf diseases. The dataset is sourced from the plant village repository, categorized into six diseased and one healthy class. The implementation is carried out in MATLAB, utilizing features extracted from the Fully Connected Layer of the pre-trained ResNet model. Training is conducted separately, employing a linear learner of the Error- Correcting Output Codes (ECOC) for classification, resulting in a highly accurate model.

Summary: This study focuses on the early detection of plant leaf diseases in India, where 10-30% of crops are damaged before being identified. Using a pretrained Deep Learning Model, specifically ResNet, the research utilizes a dataset of Tomato Leaf images with six diseased and one healthy category. Implemented in MATLAB®,the model extracts features from the Fully Connected Layer, employing a linear learner of the ECOC for classification. The trained model exhibits higher accuracy, precision, F-score, specificity, and a lower false positive rate compared to the base article, showcasing its effectiveness in disease classification.

[13] Robert G. de Luna; Elmer P. Dadios; Argel A. Bandala: The integration of smart farming infrastructure is a groundbreaking technology that enhances agricultural production, including tomatoes, by considering variables like environment, soil, and sunlight. Despite these considerations, the presence of diseases in tomato plants remains inevitable. Leveraging recent advancements in computer vision enabled by deep learning, this study introduces an innovative solution for efficient disease detection in tomato plants.

The system employed Convolutional Neural Network (CNN) to identify the presence of tomato diseases. The F-RCNN trained anomaly detection model achieved an 80% confidence score, while the Transfer Learning disease recognition model demonstrated an accuracy of 95.75%. In real-world implementation, the automated image capturing system achieved a recognition accuracy of 91.67% in identifying tomato plant leaf diseases.

Summary: This study introduces a smart farming system using innovative technology to enhance tomato plant farming. By leveraging deep learning and computer vision, the research developed a motor-controlled image capturing system to efficiently detect and recognize leaf diseases in tomato plants.

Specifically focusing on Phoma Rot, Leaf Miner, and Target Spot diseases in the Diamante Max tomato breed, the system utilized a dataset of 4,923 images for training a convolutional neural network (CNN). The automated image capturing system achieved a notable 91.67% accuracy in real-world implementation, showcasing the effectiveness of the proposed solution for tomato plant disease detection and diagnosis.

[14] **Majji V Applalanaidu**; **G. Kumaravelan:** This study aims to identify recent advancements in plant disease detection and classification systems utilizing MachineLearning (ML) and Deep Learning (DL) models. Over 45 papers published between 2017 and 2020 were gathered from peer-reviewed journals in databases like Scopus and Web of Science, focusing on keywords such as plant disease identification, recognition, and classification with ML and DL algorithms. The analysis presents a structured overview of various plant disease classification models in well-organized tables.

The study serves as a valuable resource for researchers seeking to identify plant diseases through data-driven approaches. Additionally, the application of the studied ML/DL approaches in developing mobile-based applications is expected to contribute to increased agricultural productivity.

Summary: This study explores recent advancements in plant disease detection and classification systems, utilizing Machine Learning (ML) and Deep Learning (DL) models. Analyzing over 45 papers published between 2017 and 2020, the research focuses on keywords like plant disease identification and classification with ML .

The findings serve as a valuable resource for researchers employing data-driven approaches to identify plant diseases. Furthermore, the application of ML/DL approaches in mobile-based applications is highlighted as a potential contributor to increased agricultural productivity.

3 SYSTEM ANALYSIS

3.1 EXISTING METHOD:

This model primarily focuses on an existing method that utilizes deep learning algorithms. Deep learning, a type of transfer learning, is used in this approach to carry out the procedure.However, it is worth noting that this approach didnot yield the desired level of high accuracy.

3.1 DISADVANTAGES:

Limited Dataset and Generalization: Deep learning models rely on extensive and varied datasets for robust training and the ability to generalize effectively. Nevertheless, compiling a comprehensive dataset encompassing all potential variations and disease types affecting plant leaves is a daunting task. This limitation can impact the model's capacity to accurately identify less prevalent or recently emerged diseases.

Sensitivity to Environmental Factors and Image Quality: Models designed for leaf disease detection are susceptible to disruptions caused by fluctuations in lightingconditions, the presence of background clutter, and variations in image quality. Elements like lighting intensity, camera angles, and the specific environmental conditions under which the images are captured can introduce noise into the dataset. This noise can significantly impact the model's performance and reduce its robustness when deployed in real-world applications.

Interpretability and Explain ability: Deep learning models, especially those with intricate, multi-layered architectures, are frequently regarded as "black boxes" because of their inherent lack of interpretability and explainability. Deciphering the decision-making mechanisms of these models can prove to be a daunting task. This issue is particularly significant in critical domains like agriculture, where farmers and researchers rely on comprehending the model's rationale for accurate disease diagnosis and informed decision-making.

3.2 PROPOSED SYSTEM:

The suggested approach uses densenet-121, a convolutional neural network (CNN) from deep learning, in conjunction with a customized Centernet framework to classify plant leaf diseases.

3.3 ADVANTAGES:

High Accuracy and Reliability:

In image recognition tasks, deep learning models—particularly convolutional neural networks (CNNs)—have demonstrated impressive performance. These models excel at learning intricate patterns and features from leaf images, ultimately leading to high levels of accuracy and reliability in disease detection. Their proficiency in discerning subtle variations in leaf textures and colors contributes to precise and dependable disease diagnosis, providing a valuable tool for both farmers and researchers in the field of agriculture.

Automated and Efficient:

The proposed deep learning system serves as an automated solution for leaf disease detection, eliminating the necessity for manual inspection and analysis. This automation introduces a substantial enhancement in efficiency, enabling the swift screening of a considerable number of plants. Such rapid screening is pivotal for timely disease detection and intervention, facilitating early treatment and mitigating the potential spread of diseases throughout crop populations.

Scalability and Adaptability:

Deep learning models possess the ability to undergo training on diverse datasets and exhibit adaptability in the context of various leaf diseases affecting different plant species. This inherent scalability and versatility endow the proposed system with the capability to effectively address a broad spectrum of agricultural scenarios. Furthermore, as new data becomes accessible, the model can be retrained, thereby continually enhancing its performance and ensuring its readiness to accommodate emerging diseases in the agricultural domain.

Cost-Effectiveness and Resource Optimization:

Integrating a deep learning-based leaf disease detection system holds the potential for long-term cost-effectiveness. Following the initial phases of model training and deployment, the system can be employed without the recurring expenses linked to human labor for manual inspections.

Furthermore, ongoing advancements in hardware and optimization techniques have led to a reduction in the computational demands for running deep learning models. This progress translates into a more resource-efficient and cost- effective system, offering sustainable benefits over time.

3.4 POSSIBLE SYSTEM'S WORK FLOW:

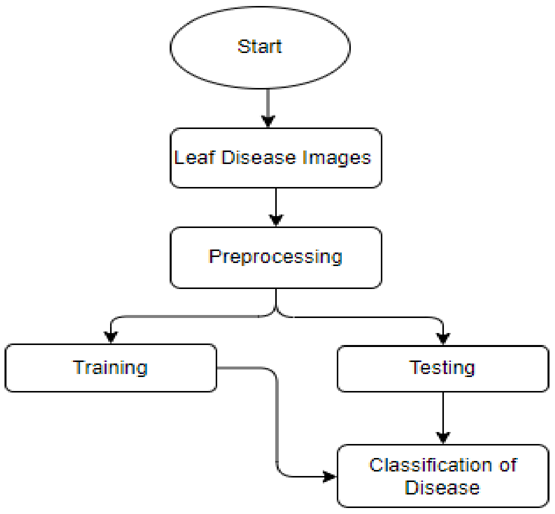


Fig 1. Block schematic of suggested procedure

4. REQUIREMENT ANALYSIS

4.1 Both necessary and non-necessary functions

One essential step in determining whether a system or software project is successful is requirements analysis. Generally, there are two categories of requirements: non-functional requirements and functional requirements.

Functional Requirements: The end user specifically requests these features as essential functions that the system must fulfill. It is a requirement of the contract that all of these features be included in the system. Input to be supplied to the system, action carried out, and anticipated result are how they are expressed or presented. Unlike non-functional needs, these requirements are essentially stated by the user and can be seen in the finished product.

1) Upon each user's login, the user's identity is verified.
2) A cyber attack triggers an automated system shutdown.
3) After a user registers for the first time in a software system, an email verification is sent to them.

Non-functional requirements: As stipulated in the project contract, these are basically the quality constraints that the system needs to meet. Project-to-project variations may occur in the importance or level of implementation of these elements. Non-behavioral requirements is another name for them.

In essence, they handle problems such as:

- Versatility
- Protection
- The ability to sustain
- Trustworthy
- Abilityto grow
- Achievement
- Sturdiness
- The pliability

Examples of non-functional requirements:

1) Sending emails should happen no more than 12 hours after the relevant activity.
2) A maximum of 10 seconds should pass between each request's processing.
3) When there are more than 10,000 users on the site at once, the page should load in three seconds.

4.2 Hardware Requirements

Processor	- I3/Intel Processor
Hard Disk	- 160GB
Key Board	- Standard Windows Keyboard Mouse
	- Two or Three Button Mouse
Monitor	- SVGA
RAM	- 8GB

4.3 Software Requirements:

Operating System	: Windows 7/8/10
Server side Script	: HTML, CSS, Bootstrap & JS

Programming Language	: Python
Libraries	: Flask, Pandas, Mysql.connector, Os, Smtplib, Numpy
IDE/Workbench	: PyCharm
Technology	: Python 3.6+
Server Deployment	: Xampp Server
Database	: MySQL

4.4 ARCHITECTURE

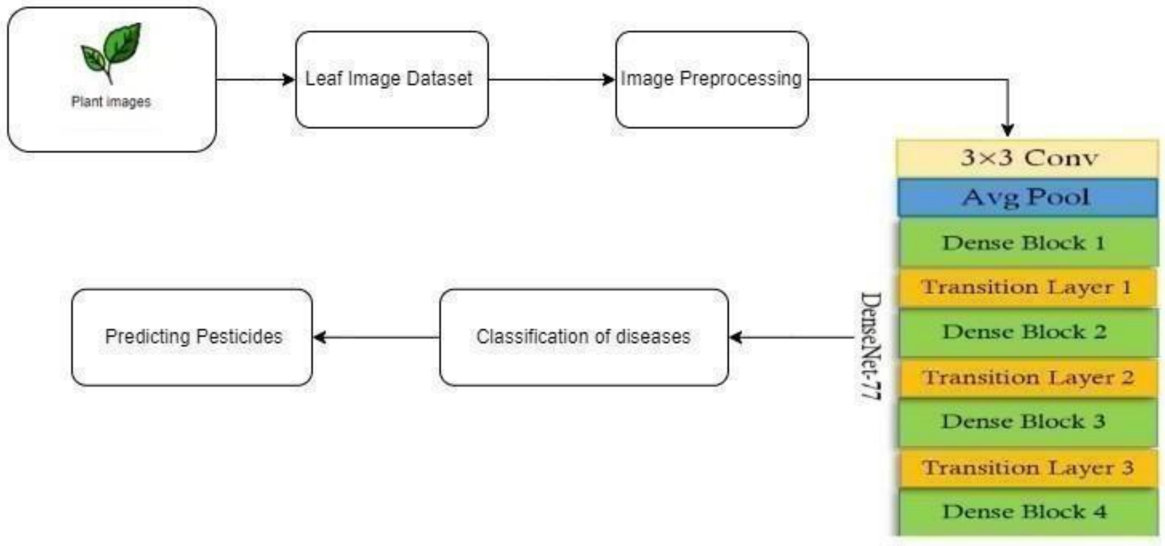


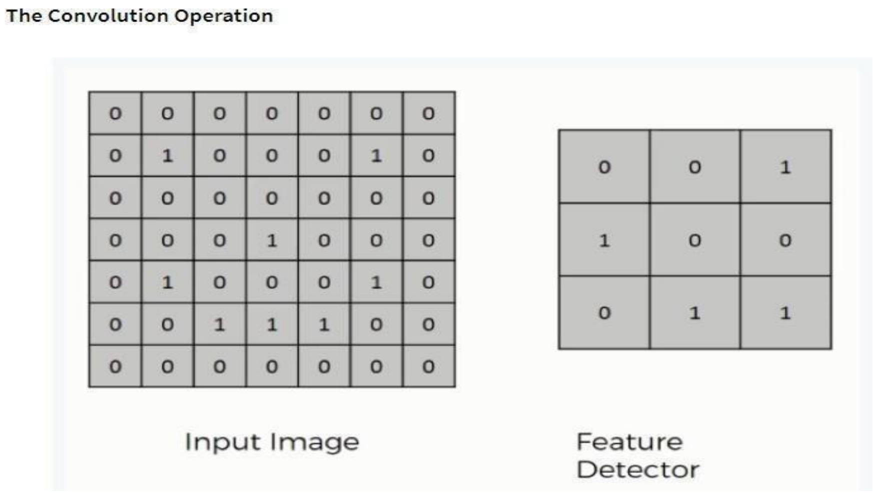
Fig 2:Architecture

5. METHODOLOGY

5.1. CONVOLUTIONAL NEURALNETWORK

Step1: Convolutional operation

First, the convolution operation is the fundamental building block of our strategy. In this stage, we study the idea of feature detectors, which are the filters of the neural network in essence. We delve into the field of feature maps, comprehending the parameter learning procedure linked to these maps, the workings of pattern recognition, the level by level detection, and the mapping of the resulting insights.



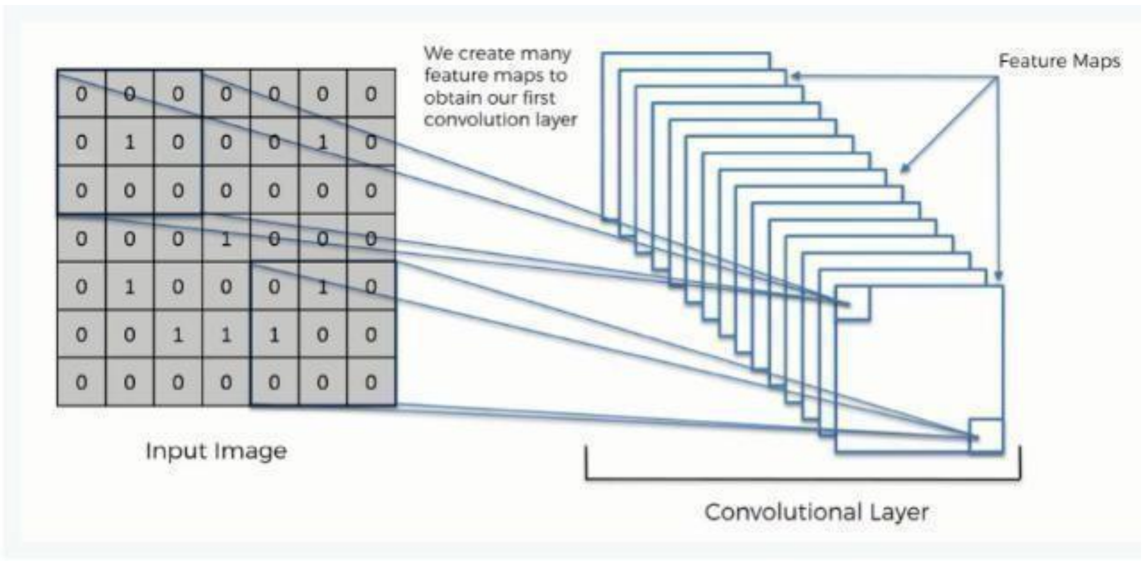

We create many feature maps to obtain our first convolution layer
Feature Maps
Input Image
Convolutional Layer

Fig 3: CNN Architecture

Step (1b): ReLU Layer

This stage's second component is the Rectified Linear Unit, or ReLU. We explore the operation of linearity in the context of Convolutional Neural Networks and the workings of Re LU layers. A quick lesson in this area can improve your knowledge and skill set, but i t 's not necessary to understand CNNs.

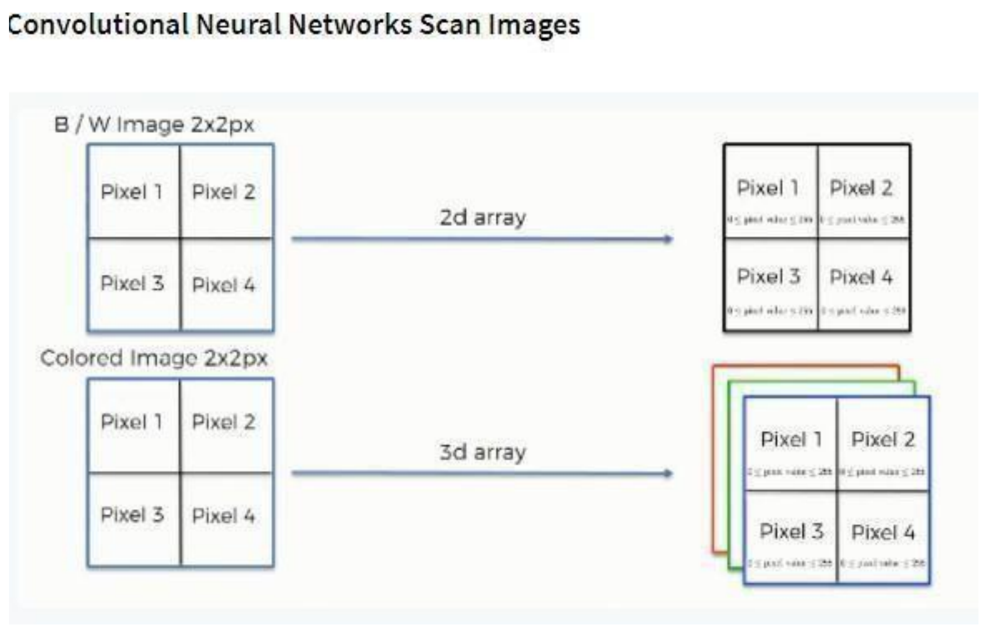


Fig 4: CNN Images

Step 2: Conv2D

A convolution kernel is created by the Keras Conv2D layer, a 2D convolution layer, and applied to the layer's input. As a result of this procedure, the outputs are combined to create a tensor.

The term "kernel," within the realm of image processing, refers to a convolution matrix or mask. Kernels serve a crucial role in various image processing operations, including but not limited to blurring, sharpening, embossing, and edge detection. Convolution between a kernel and an image is used to carry out these operations, which enable the enhancement and manipulation of visual features contained in the image data.

Step 3: Flattening

The flattening procedure and the change from pooled to flattened layers when using convolutional neural networks will be briefly reviewed here.

Step 4: Full Connection

Everything we've discussed so far will be summarized in this section. Comprehending this will help you gain a deeper comprehension of how Convolutional Neural Networks function, particularly how the generated "neurons" determine which images to classify.

Summary

Additionally, if you're inclined to enhance your understanding further, there's an optional tutorial available that delves into Softmax and Cross-Entropy. While not mandatory for this course, familiarity with these concepts can prove valuable, particularly when working with Convolutional Neural Networks. Being well-versed in Softmax and Cross-Entropy can empower you with a deeper comprehension of key aspects in the field of CNNs, making your journey in this domain more insightful and productive.

Convolutional neural network (CNN):

The layers of an input, hidden, and output convolutional neural network are as follows. Because the activation function and the final convolution hide the inputs and outputs of intermediate layers in any feed-forward neural network, these layers are referred to as "hidden".

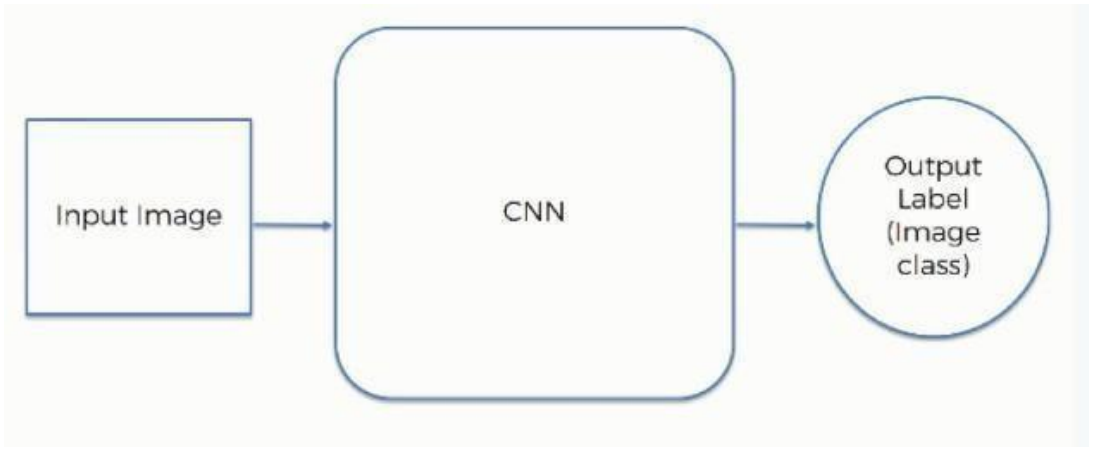


Fig 5: CNN flow chart

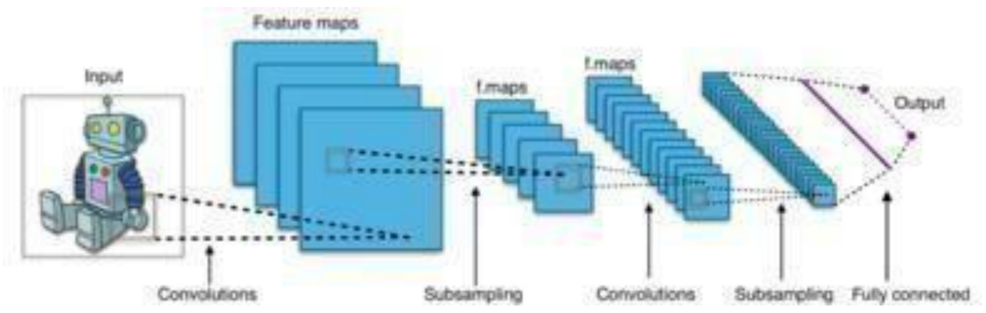


Fig 6. CNN Architecture

5.2 ARTIFICIAL NEURAL NETWORK (ANN):

The design and operation of artificial neural networks (ANNs) are influenced by biological neural networks. Much like neurons in the human brain, ANNs comprise interconnected neurons organized into different layers. A widely used type of Artificial Neural Network (ANN) is the feed forward neural network. The structure comprises of three layers: an input layer responsible for recognizing patterns in external data, an output layer that offers a solution to a particular issue, and one or more hidden layers that function as a mediator layer between the input and output layers. Neurons in adjacent layers are connected through acyclic arcs, enabling the unidirectional flow of information. The error rate between the intended target output and the actual output is used by ANNs' training algorithms to learn from datasets and modify the weights of their neurons. Back propagation algorithm is commonly used for this purpose, enabling ANNs to learn and adapt to the provided data. The general structure of ANN is shown in Fig.

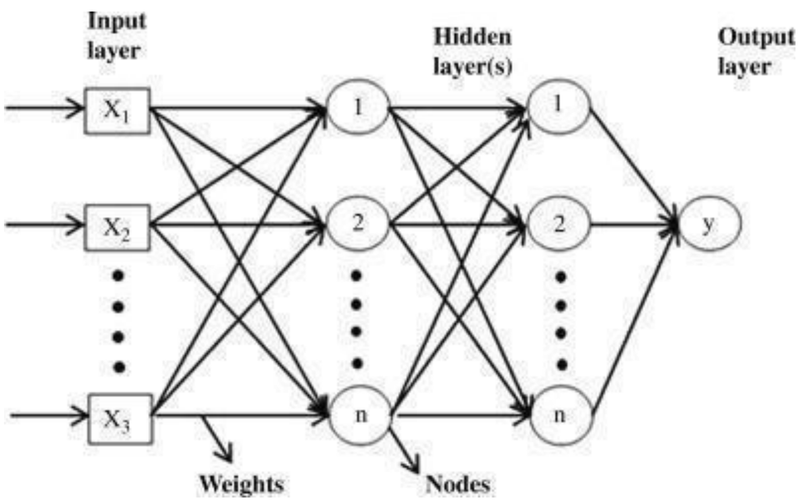


Fig 7: ANN Architecture

5.3 RESNET50:

With fifty layers deep, ResNet50 is a convolutional neural network. In their paper "Deep Residual Learning for Image Recognition," Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun describe how they developed and trained the model in 2015. From the ImageNet database, more than a million photos were used to train this model. The network was trained on colored images with 224 x 224 pixels, and it can classify up to 1000 objects, much like VGG-19. Its size and performance can be briefly summarized as follows:

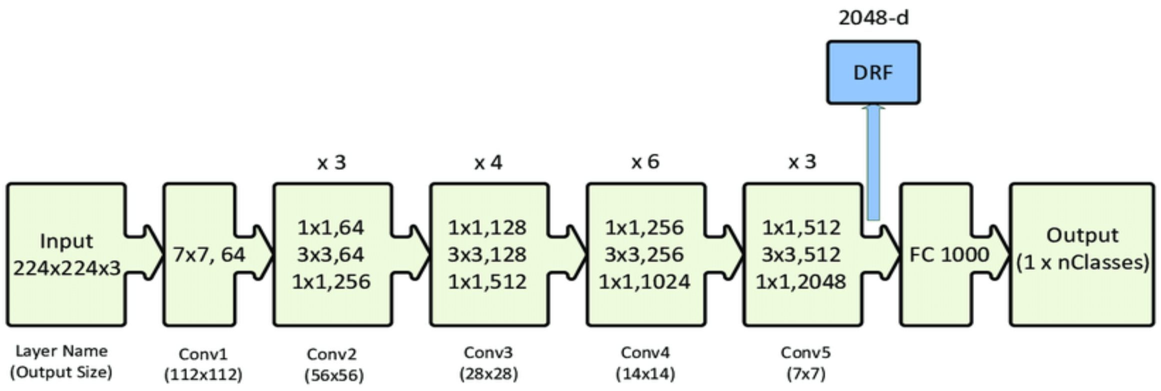


Fig 8. ResNet50 Architecture

To sum up, the development of the residual network, or ResNet, marked a substantial advancement in the training of deep convolutional neural networks, particularly for computer vision applications. The foundation for this innovation was laid by the original ResNet, which had 34 layers and 2-layer blocks. In order to decrease training time and increase accuracy, more advanced versions, such as ResNet50, incorporated 3-layer bottleneck blocks. The popular deep learning API Keras is renowned for its approachable model building methodology. It makes a number of pre-trained models, such as ResNet50, easily obtainable for testing. For tasks like image classification, building a residual network in Keras is therefore a relatively simple process that only requires a few basic steps.

5.4 DenseNet201:

Particularly in the area of visual object recognition, DenseNet marks a substantial breakthrough in neural networks. Although it and ResNet have certain commonalities, they differ greatly. ResNet employs an additive strategy (+), combining the identity layer's output from the previous layer with the next layer; DenseNet, on the other hand, uses a concatenation strategy (.), combining the output of the previous layer with the forthcoming layer. With deep neural networks, the vanishing gradient problem is known to cause accuracy to decline. This method was created expressly to solve this problem. In other words, information tends to disappear before it reaches its destination when it travels a longer path from the input layer to the output layer.

The idea of using a composite function operation to link the input of the subsequent layer to the output of the previous layer is the fundamental component of DenseNet. A convolution layer, pooling layer, batch normalization, and non-linear activation layer are all included in this composite operation. DenseNet stands out for having an architecture that is densely connected, guaranteeing that the network has L (L+1)/2 direct connections, where L is the total number of layers in the architecture. DenseNet comes in different versions, called DenseNet-121, DenseNet-160, and DenseNet-201, depending on how many layers it includes. For example, DenseNet-121 has 121 layers.

Notably, whether through addition or concatenation, these layer connections are feasible when feature map dimensions are consistent. In cases where the feature map dimensions differ, DenseNet divides the architecture into DenseBlocks.

6. SYSTEM DESIGN

6.1 Introduction of Input Design:

The raw data that is processed to produce output is referred to as input in an information system. During the input design stage, developers need to consider input devices like PCs, OMR (Optical Mark Recognition), MICR (Magnetic Ink Character Recognition), etc.

Thus, the system output's quality is determined by the quality of its intake. These are the characteristics of well-designed input forms and screens: −

- Information storage, recording, and retrieval are only a few of the functions it should efficiently fulfill.
- It guarantees precise and correct completion.
- Filling it out should be simple and uncomplicated.
- Attention, consistency, and simplicity for the user should be its main priorities.
- The understanding of fundamental design concepts is used to accomplish all of these goals. −
    - What are the inputs that the system requires?
    - The responses of end users to different form and screen elements.

Objectives for Input Design:

The goals of input design include : −

- To create input and data entry processes
- To lower the input loudness
- To create screens for data entry, user interface, input data records, etc.
- To implement efficient input controls and validate data.

Output Design:

For every system, producing output is an essential duty. Developers identify the kinds of outputs that are needed, as well as the prototype report layouts and output controls, during the output design phase.

Objectives of Output Design:

The following are the aims of input design::

- Developing an output design in accordance with the needs of the final user..
- Delivering the proper quantity of work.
- Appropriately formatting the result and sending it to the right person.
- Assuring prompt delivery of results so that decisions may be made with knowledge.

6.2 UML Diagrams:

USE CASE DIAGRAM

► It is meant to provide a visual depiction of the functionality that a system offers by showing the actors, their objectives (expressed as use cases), and any interdependencies among those use cases.

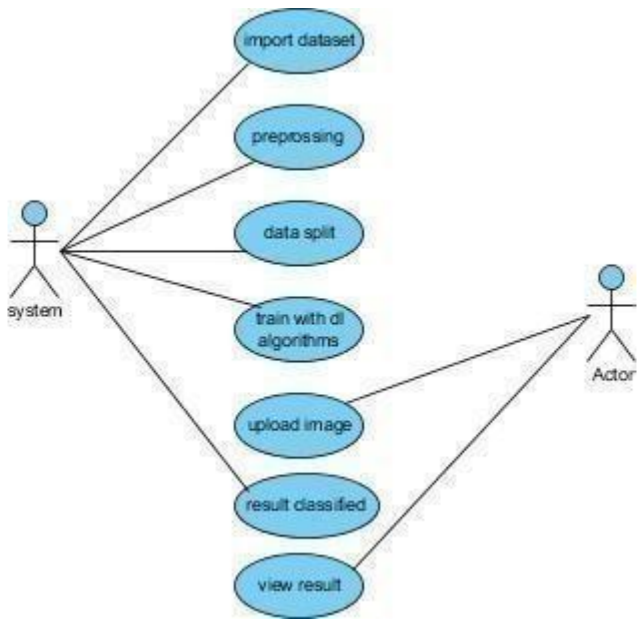


Fig 9:Use Case Diagram

CLASS DIAGRAM

A class diagram in software engineering is a form of static structural diagram that shows a system's structure according to the Unified Modeling Language (UML). In order to do this, it provides illustrations of the classes in the system, along with information on their properties, operations, and relationships between them. This diagram helps identify which class has information in it.

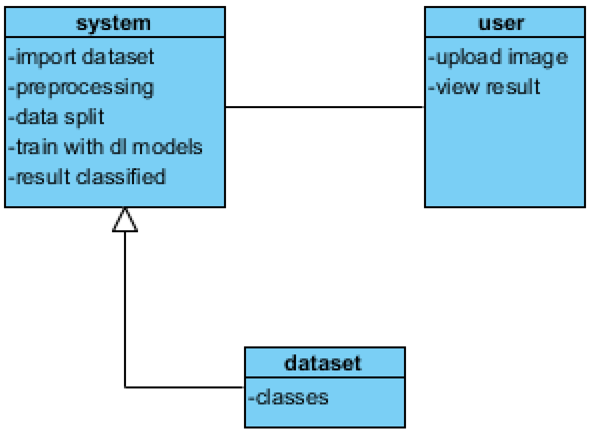


Fig 10: UML Diagram

SEQUENCE DIAGRAM

- ► The structure was created using a message sequence chart as a model. Sometimes, event diagrams, event situations, and timing diagrams are used interchangeably with sequence diagrams.

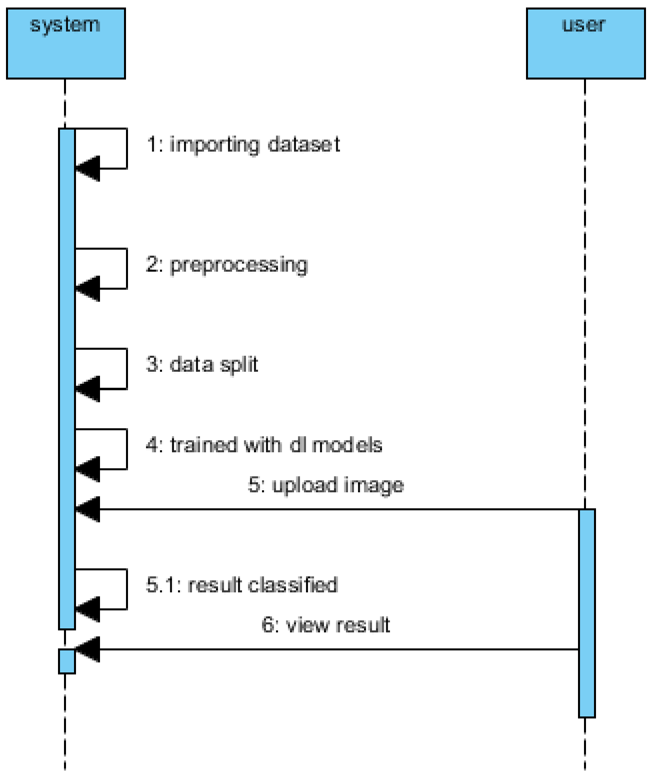


Fig 11: Sequence Diagram

COLLABORATION DIAGRAM:

Here's an example of how to use a numbering strategy in a collaboration diagram to show the order of method calls. The sequence of methods' calls is indicated by the number. To show the cooperation diagram, we have used the same order management system. It looks like a sequence diagram with the method calls. The cooperation diagram clearly illustrates the object organization, but the sequence diagram does not. This is where the differences lie.

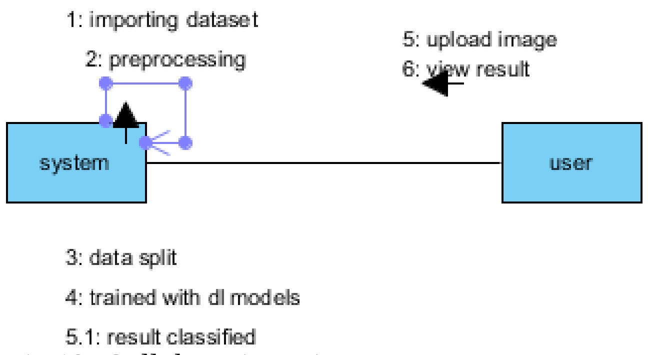


Fig 12: Colloboration Diagram

DEPLOYMENT DIAGRAM

Given that deployment diagrams are used to deploy components, they are closely connected to component diagrams. A deployment diagram depicts the deployment perspective of a system. The actual hardware that is utilized to deploy an application is represented by nodes in a deployment diagram.

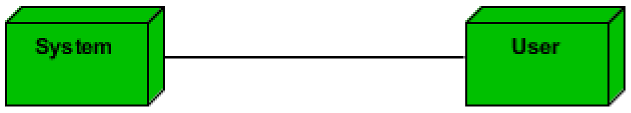


Fig 13: Deployment Diagram

ACTIVITY DIAGRAM:

Using choices, iteration, concurrency, and other supporting aspects, activity diagrams are graphic representations of sequential processes. Activity diagrams, as defined by the Unified Modeling Language (UML), are used to show how business and operational components of a system function sequentially together. The general control flow is shown in these schematics.

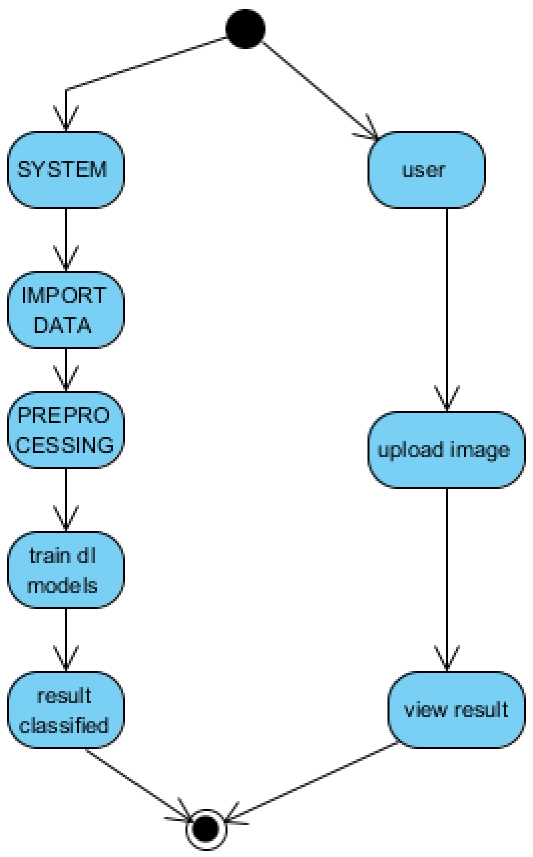


Fig 14: Activity Diagram

COMPONENT DIAGRAM:

The organization and linkages between the physical parts of a system are shown in a component diagram, often known as a UML component diagram. To guarantee that all facets of the necessary functionality of the system are covered in the intended development, these diagrams are often made to simulate implementation specifics.

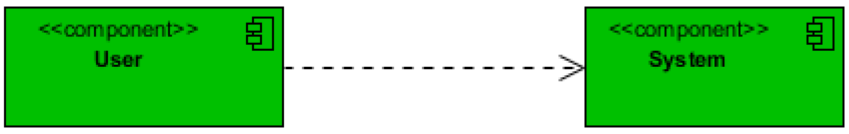


Fig 15: Component Diagram

ER DIAGRAM:

A database's organizational structure is shown by an Entity-Relationship model (ER model), which is also known as an Entity-Relationship Diagram (ER Diagram). An implementation of a database may be facilitated by using the ER model as a design or blueprint. Relationship sets and entity sets are the two main parts of the E-R model.

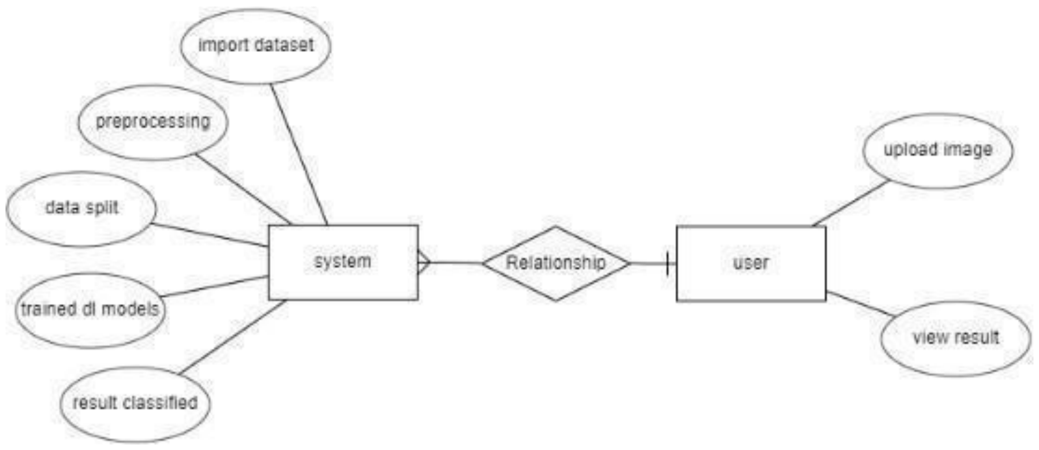


Fig 16 :ER Diagram

DFD DIAGRAM:

The standard technique for illustrating the information flow within a system is a data flow diagram (DFD). Significant parts of the system requirements may be visually represented by a clear and well-organized DFD. It has two possible modes of operation: automatic and manual. Information entry and exit points, information alteration mechanisms, and information storage locations are all shown in the figure. Illustrating the limits and scope of a system as a whole is the main goal of a DFD. As the foundation for system redesign, it functions as a communication tool between a systems analyst and other parties engaged in the system.

Zero level Diagram:

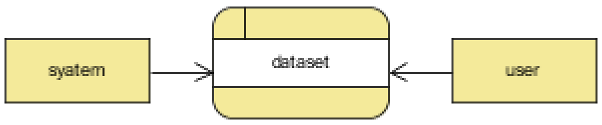


Fig 17: Zero Level Diagram

Level 1 Diagram:

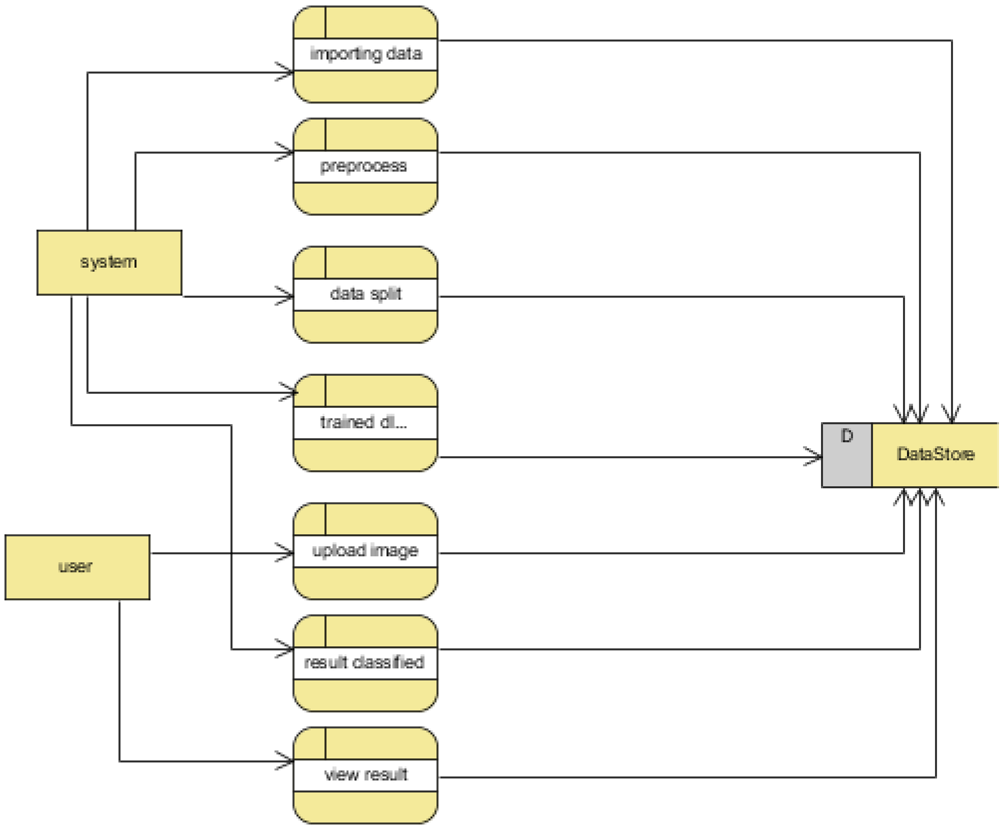


Fig 18:Level 1 Diagram

Level 2 Diagram:

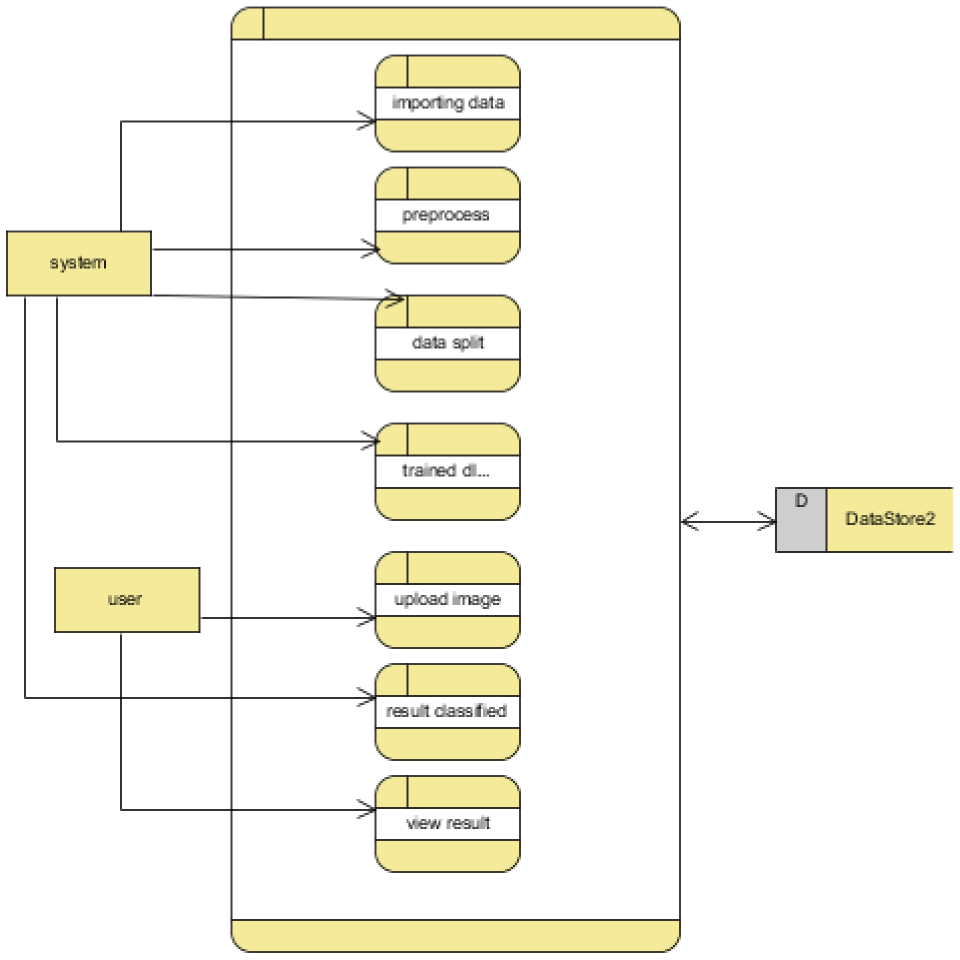


Fig 19:Level 2 Diagram

CODING AND TESTING:

```
import os
import pandas as pd
import numpy as np
import tensorflow as tf
from flask import Flask, request, render_template, send_from_directory
from tensorflow.keras.preprocessing import image
from keras.models import load_model

app = Flask(_name_)

Plants = ['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot', 'Corn_(maize)___Common_rust_',
'Corn_(maize)___healthy', 'Corn_(maize)___Northern_Leaf_Blight', 'Potato___Early_blight',
'Potato___healthy', 'Potato___Late_blight', 'Tomato___Bacterial_spot', 'Tomato___healthy',
'Tomato___Late_blight', 'Tomato___Tomato_Yellow_Leaf_Curl_Virus']
```

```
@app.route("/about")
def about():
    return render_template("about.html")

@app.route("/upload/<filename>")
def send_image(filename):
    return send_from_directory("images",filename)

@app.route("/upload",methods=["POST","GET"])
def upload():
    if request.method=='POST':
        print("hdgkj")
        m = int(request.form["alg"])
        acc = pd.read_csv("Accuracy.csv")

        myfile = request.files['file']
        fn = myfile.filename
        mypath = os.path.join("images/", fn)
        myfile.save(mypath)

        print("{} is the file name", fn)
        print("Accept incoming file:", fn)
        print("Save it to:", mypath)

        if m == 1:
            print("bv1")
            new_model = load_model(r'models/SVC.h5')
            test_image = image.load_img(mypath, target_size=(128, 128))
            test_image = image.img_to_array(test_image)
            a = acc.iloc[m - 1, 1]

        elif m == 2:
            print("bv2")
            new_model = load_model(r'models/ANN.h5')
            test_image = image.load_img(mypath, target_size=(128, 128))
            test_image = image.img_to_array(test_image)
            a = acc.iloc[m - 1, 1]

        elif m == 3:
            print("bv2")
            new_model = load_model(r'models/CNN.h5')
            test_image = image.load_img(mypath, target_size=(128, 128))
            test_image = image.img_to_array(test_image)
            a = acc.iloc[m - 1, 1]

        else:
            print("bv3")
            new_model = load_model(r'models/ResNet50.h5')
            test_image = image.load_img(mypath, target_size=(128, 128))
            test_image = image.img_to_array(test_image)
            a = acc.iloc[m - 1, 1]

        test_image = np.expand_dims(test_image, axis=0)
        result = new_model.predict(test_image)
        preds = Plants[np.argmax(result)]

        if preds == "Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot":
            msg="Foliar fungicides can be used to manage gray leaf spot outbreaks"
        elif preds =="Corn_(maize)___Common_rust_":
            msg = "Use resistant varieties like DHM 103, Ganga Safed - 2 and avoid sowing of suceptable varieties
like DHM 105"

        elif preds =="Corn_(maize)___healthy":
            msg = "Plant is Good no treatment required"

        elif preds == "Corn_(maize)___Northern_Leaf_Blight":
            msg = "Integration of early sowing, seed treatment and foliar spray with Tilt 25 EC (propiconazole) was
the best combination in controlling maydis leaf blight and increasing maize yield"


        elif preds == "Potato___Early_blight":
            msg = "Mancozeb and chlorothalonil are perhaps the most frequently used protectant fungicides for early
blight management"

        elif preds =="Potato___healthy":
            msg = "Plant is Good no treatment required"

        elif preds == "Potato___Late_blight":
            msg = "Effectively managed with prophylactic spray of mancozeb at 0.25% followed by
cymoxanil+mancozeb or dimethomorph+mancozeb at 0.3% at the onset of disease and one more spray of
mancozeb at 0.25% seven days"

        elif preds == "Tomato___Bacterial_spot":
            msg = "When possible, is the best way to avoid bacterial spot on tomato. Avoiding sprinkler irrigation
and cull piles near greenhouse or field operations, and rotating with a nonhost crop also helps control the
disease"

        elif preds =="Tomato___healthy":
            msg = "Plant is Good no treatment required"

        elif preds == "Tomato___Late_blight":
            msg = "Ungicides that contain maneb, mancozeb, chlorothanolil, or fixed copper can help protect plants
from late tomato blight"

        else:
            msg = "Homemade Epsom salt mixture. Combine two tablespoons of Epsom salt with a gallon of water
and spray the mixture on the plant"

        return render_template("template.html", text=preds,msg=msg ,image_name=fn,a=round(a*100,3))
    return render_template("upload.html")

if __name___== '_main_':
    app.run(debug=True)

<?xml version="1.0" encoding="UTF-8"?>
<module type="PYTHON_MODULE" version="4">
  <component name="NewModuleRootManager">
    <content url="file://$MODULE_DIR$">
      <excludeFolder url="file://$MODULE_DIR$/venv" />
    </content>
    <orderEntry type="jdk" jdkName="Python 3.8 (CODE)" jdkType="Python SDK" />
    <orderEntry type="sourceFolder" forTests="false" />
  </component>
  <component name="TestRunnerService">
    <option name="PROJECT_TEST_RUNNER" value="pytest" />
```

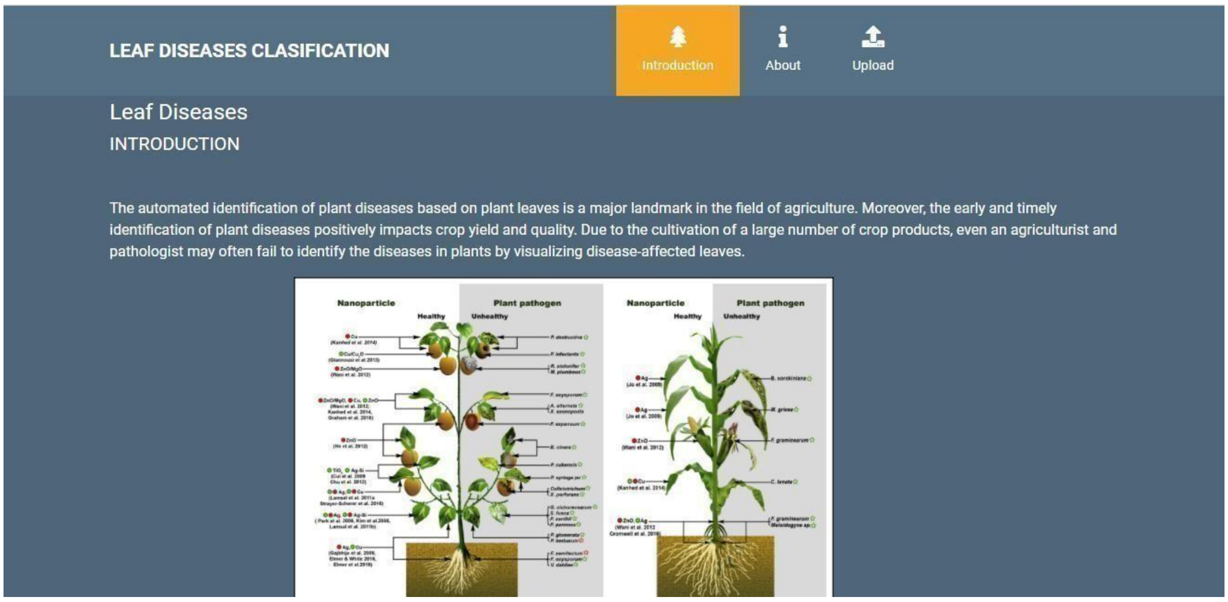
```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ProjectRootManager" version="2" project-jdk-name="Python 3.8 (CODE)" project-jdk-
type="Python SDK" />
  <component name="PyCharmProfessionalAdvertiser">
    <option name="shown" value="true" />
  </component>
</project>

<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ProjectModuleManager">
    <modules>
      <module fileurl="file://$PROJECT_DIR$/.idea/CODE.iml" filepath="$PROJECT_DIR$/.idea/CODE.iml"
/>
    </modules>
  </component>
</project>

<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ChangeListManager">
    <list default="true" id="3c2aed5c-35e0-47b6-908c-60ca465f86d6" name="Default Changelist" comment=""
/>
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="ProjectId" id="2WqWOUUS6Tyc5FrD33aypU0ld2r" />
  <component name="ProjectViewState">
    <option name="hideEmptyMiddlePackages" value="true" />
    <option name="showExcludedFiles" value="true" />
    <option name="showLibraryContents" value="true" />
  </component>
  <component name="PropertiesComponent">
    <property name="RunOnceActivity.ShowReadmeOnStart" value="true" />
    <property name="last_opened_file_path" value="$PROJECT_DIR$" />
  </component>
  <component name="ServiceViewManager">
    <option name="viewStates">
      <list>
        <serviceView>
          <treeState>
            <expand />
            <select />
          </treeState>
        </serviceView>
      </list>
    </option>
  </component>
  <component name="SvnConfiguration">
    <configuration />
  </component>
  <component name="TaskManager">
    <task active="true" id="Default" summary="Default task">
      <changelist id="3c2aed5c-35e0-47b6-908c-60ca465f86d6" name="Default Changelist" comment="" />
      <option name="number" value="Default" />
      <option name="presentableId" value="Default" />
      <updated>1697460559549</updated>
    </task>
    <servers />
  </component>
  <component name="WindowStateProjectService">
    <state x="470" y="234" key="com.intellij.ide.util.TipDialog" timestamp="1697463570001">
      <screen x="0" y="0" width="1536" height="816" />
    </state>
    <state x="470" y="234" key="com.intellij.ide.util.TipDialog/0.0.1536.816@0.0.1536.816"
timestamp="1697463570001" />
  </component>
</project>

<component name="InspectionProjectProfileManager">
  <settings>
    <option name="USE_PROJECT_PROFILE" value="false" />
    <version value="1.0" />
  </settings>
</component>
```

RESULTS AND DISCUSSIONS



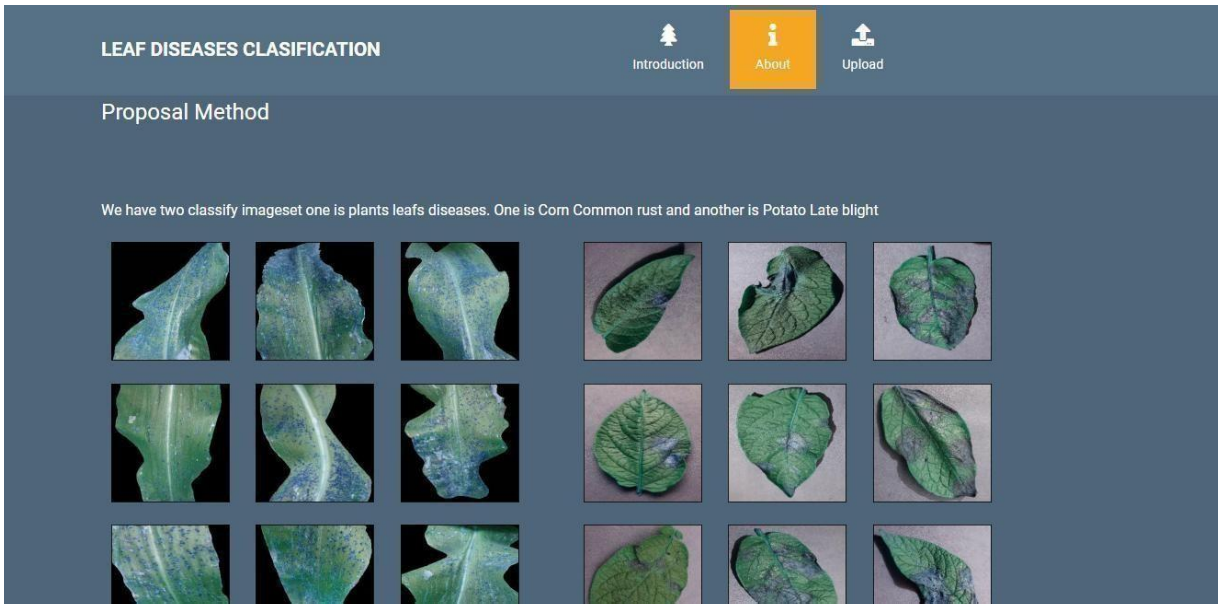
Leaf Diseases

INTRODUCTION

The automated identification of plant diseases based on plant leaves is a major landmark in the field of agriculture. Moreover, the early and timely identification of plant diseases positively impacts crop yield and quality. Due to the cultivation of a large number of crop products, even an agriculturist and pathologist may often fail to identify the diseases in plants by visualizing disease-affected leaves.

Fig20: Home

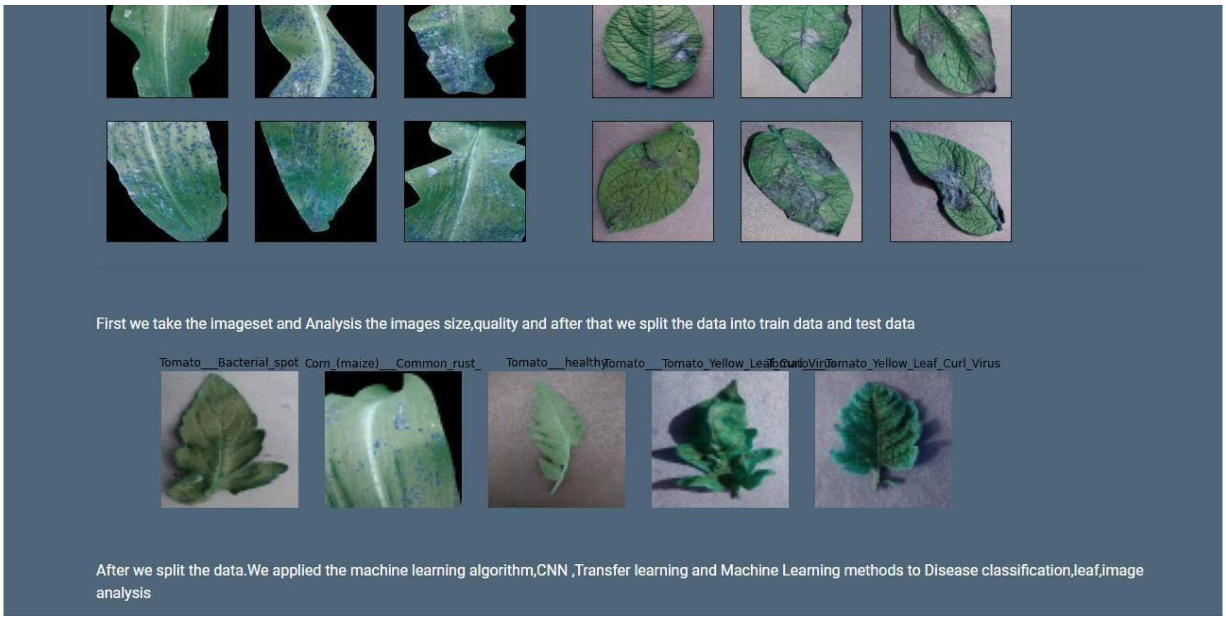
About Project: Here the user will get a breif idea about the project.



Proposal Method

We have two classify imageset one is plants leafs diseases. One is Corn Common rust and another is Potato Late blight

Fig21: About Project


Tomato__Bacterial_spot
Corn_(maize)__Common_rust_
Tomato__healthy
Tomato__Tomato_Yellow_Leaf_Curl_Virus
Tomato_Yellow_Leaf_Curl_Virus

First we take the imageset and Analysis the images size,quality and after that we split the data into train data and test data

After we split the data.We applied the machine learning algorithm,CNN ,Transfer learning and Machine Learning methods to Disease classification,leaf,image analysis

Fig22: About Project

Upload Image:

Here the images can be uploaded those which are to be classified.

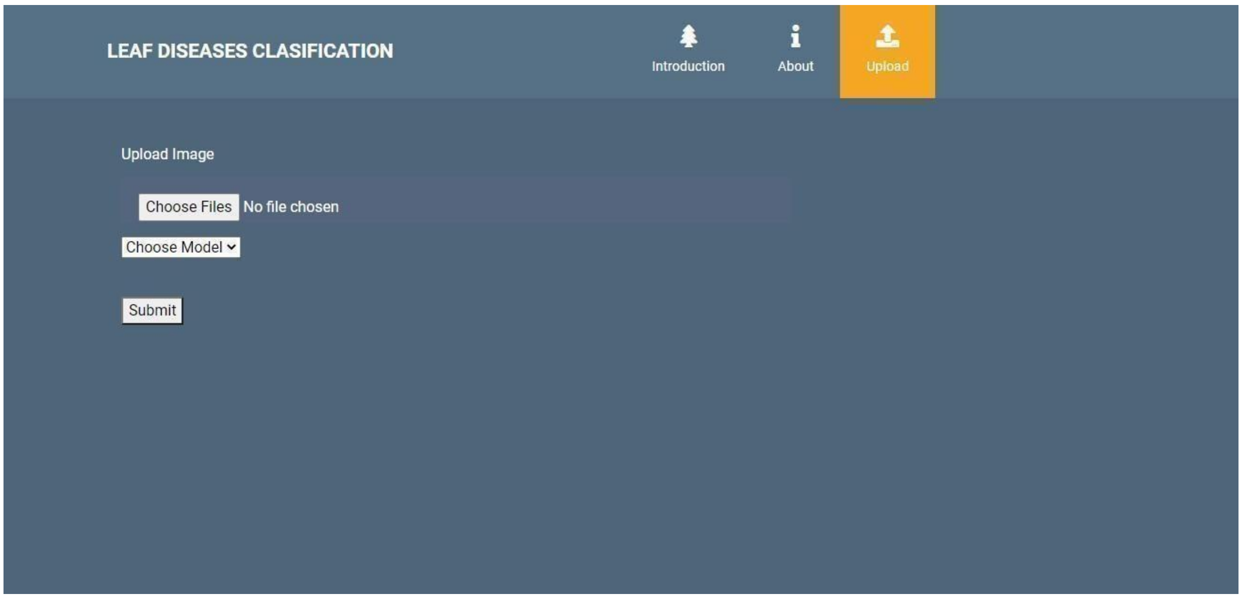


Fig23: Image Uploading

Model choosing:

Here the model can be selected, by which the image is to be classified.

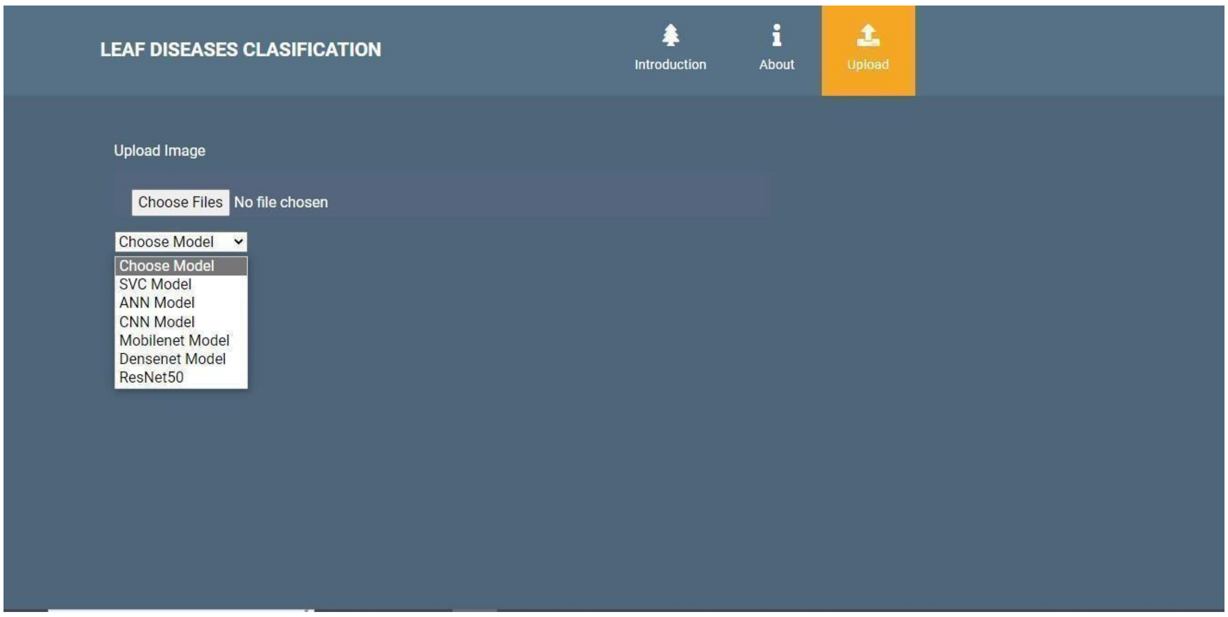
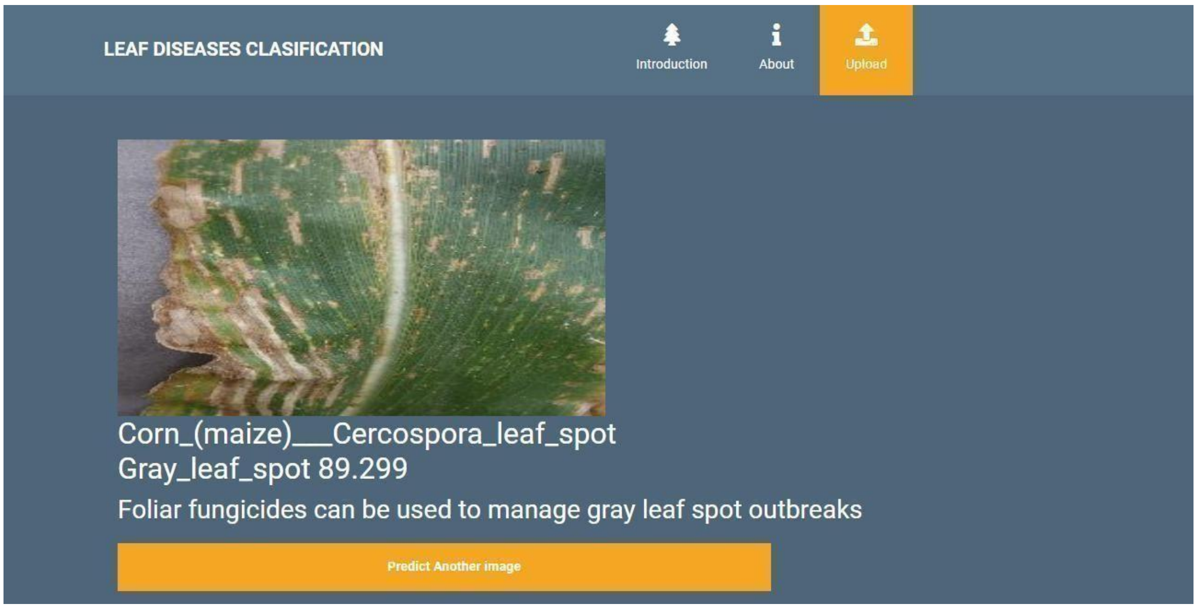


Fig4: Model choosing

Classified output:

The uploaded image is classified as the Corn_(maize)___________Cercospora_leaf_spot Gray_leaf_spot.



Corn_(maize)___Cercospora_leaf_spot
Gray_leaf_spot 89.299
Foliar fungicides can be used to manage gray leaf spot outbreaks

Fig24: Classified output

TEST CASES:

Input	Output	Result
Input text	Tested for the classification of Plant Disease Classification	Success

TEST CASES MODEL BUILDING:

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Go over the dataset.	Dataset path.	Need to read the dataset with success.	Dataset fetched successfully.	P
2	applying pre-processing to the dataset	Pre-processing part takes place	Preprocessing has to be done on the dataset.	Pre-processing successfully completed.	P
3	Model Building	Model Building for the clean data	Model creation is necessary using the necessary algorithms	Model Created Successfully.	P
4	Classification	Input image provided.	Identifying and classifying plant diseases should be the output.	Effective model classification	P

CONCLUSION:

In this project, our primary achievement lies in the successful classification of images related to the identification of plant leaf diseases.

Our dataset encompasses a diverse range of images associated with Plant Leaf Diseases Classification, featuring various types of diseases and plants in different health conditions. To accomplish this classification task, we harnessed the power of Support Vector Classifier (SVC), Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and further leveraged transfer learning methods like automated plant disease detection and classification, contributing to the field of agriculture.

FUTURE SCOPE:

The outcomes of this project hold substantial promise for future applications, particularly in the efficient classification of various types of plant diseases. This advancement paves the way for the early prediction of diseases in plants, enabling proactive treatment in the initial stages. By swiftly identifying and classifying plant diseases, we can take timely measures to initiate the necessary treatment, safeguarding the health of individual plants, and preventing the spread of diseases toneighboring plants. This proactive approach not only ensures the well-being of the affected plants but also contributes to the overall health and productivity of agricultural crops. Ultimately, our work signifies a significant step towards enhancing plant disease management and promoting sustainable agricultural practices.

REFERENCES

1) V. S. Babu, R. Satheesh Kumar and R. Sunder, "A comparative Study on Disease Detection of Plants Using Machine Learning Techniques", 2021 7 th International Conference on Advanced Computing and Communicating Systems (ICACCS 2021) .
2) Yang Zhang, Chenglong Song and Dongwen Zhang, Deep Learning-Based Object Detection Improvement for Tomato Disease, IEEE, 2020.
3) Jun Sun, Yu Yang, Xiaofei He and Xiaohong Wu, Northern Maize Leaf Blight Detection under Complex Field Environment Based on Deep Learning, IEEE, 2020.
4) Peng Jiang, Yeuhan Chen, Bin Liu, Dongiian He and Chunquan Liang, Real- Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks, IEEE, 2019.

5) Nikita Jadhav, Himali Kasar, Sumita Chandak and Shivani Machha, "Crop Leaf Disease Diagnosis using Convolutional Neural Network", Published 2020 Biology International Journal of Trend in Scientific Research and Development.

6) G. Geetha, S. Samundeswari, Saranya Gangadhara Moorthy, K. Meenakshi and M. Nithya, "Plant Leaf Disease Classification and Detection System Using Machine Learning", December 2020 Journal of Physics Conference Series (ICCPET 2020).

7) M. Francis and C. Deisy, "Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks", A Visual Understanding Computer Science 2019 6 th International Conference on Signal Processing and Integrated Networks (SPIN) , 2019.

8) S. Hari, M. Siva Kumar, P. Renuga, S. Karthikeyan and S. Suriya, "Detection of Plant Disease by Leaf Image Using Convolutional Neural Network", Computer Science 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN) 2019.

9) Md Rahmat Ullah, Nagifa Anjum and Abdus Sattar, "Plant Disease Recognition Using Machine Learning", 2019 8 th International Conference System Modeling and Advancement in Research Trenda (SMART 2019) .

10) Melike Sardogan, Adem Tuncer and Yunnus Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm", 3 rd International Conference on Computer Science and Engineering (UBMK) Posted , 2018.

11) Shima Ramesh Maniyath, P V Vinod, M Niveditha, P. R, P. N, N Shashank, et al., "Plant Disease Detection Using Machine Learning", 2018 InternationalConference on Design Innovations for 3C's Compute Communicate Control (ICDI3C).

12) P. Maheswari, P. Raja and N. M. Ghangoankar, "Intelligent Disease Detection System for Early Blight of Tomato Using Foldscope: A Pilot Study", 2018 IEEE 4th International Symposium in Robotics and Manufacturing Automation (ROMA).

13) Yow-Wen Tian, Peng-Hui Zheng and Rui-Yao Shi, "The Detection System for Greenhouse Tomato Disease Degree Based on Android Platform", 3 rd

International Conference on Information Science and Control Engineering Posted , 2016.

14) C Ravi, Jibu Shinde, C Mathew and C Y Patil, "Segmentation Technique for Soybean Leaves Disease Detection", International Journal of Advanced Research, vol. 3, no. 5, 2015.

15) M Ramakrishna, A Sahaya and Anselin Nisha, "Groundnut Leaf Disease Detection and Classification by using Nack Propagation Algorithm", IEEE International Conference, 2015.

16) U. Mokhtar, Mona A. S. Ali, Aboul Ella Hassenian and H. Hefny, "Tomato Leaves Diseases Detection Approach Based on Support Vector Machines", 2015 11 th International Computer Engineering Conference(ICENCO 2015)

17) A. Camargo and J.S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," Biosystems Engineering, vol.102, pp.9–21, January 2009.

18) J.S. Cope, D. Corney, J.Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: A review," Expert Systems with Applications, vol.39, pp.7562–7573, June 2012.

19) J. Garcia and A. Barbedo, "Using digital image processing for counting whiteflies on soybean leaves," Journal of Asia-Pacific Entomology, vol.17, pp.685–694, December 2014.

20) A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," Computers and Electronics in Agriculture, vol.116, pp.8–19, August 2015.

leaf_disease-report-final-Plg

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		