Aquatic Plants Disease Detection using Deep Learning

Submitted in partial fulfillment
of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

FROILA STEPHANIE P. A (Reg.No – 39110301)
DHANUSHA (Reg.No - 39110219)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with ―A‖ grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Froila stephanie P.A(39110301) and Dhanusha(39110219)** who carried out the Project Phase-2 entitled **"RTSC- DVM: A NOVEL METHODOLOGY FOR REAL-TIME SIREN CALLTO HALT DISTRACTED VEHICLE MISHAPS"** under my supervision from January 2023 to April 2023.

Internal Guide

Ms. R. YOGITHA M.E., (Ph.D)

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva-voce Examination held on 20.04.2023

Internal Examiner **External Examiner**

DECLARATION

We**,** hereby declare that the Project Phase-2 Report entitled **RTSC-DVM: A NOVEL METHODOLOGY FOR REAL-TIME SIREN CALL TO HALT DISTRACTED VEHICLE MISHAPS"** done by me under the guidance of **Ms. R.Yogitha M.E.,(Ph.D)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 20.04.2023

PLACE: Chennai **Froila Stephanie P.A**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. R.Yogitha M.E.,(Ph.D),** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

ABSTRACT

Accurate detection of plant diseases is a complicated task that farmers encounter throughout the plant development and production phases.We will be utilizing Tensorflow for Deep Learning and the CNN architecture.Existing solutions are ANN-based and have been shown to be less accurate and nearly incompatible with the majority of datasets and applications.We present a real-time, automated method for detecting plant diseases based on the CNN architecture. It was trained using transfer learning on Kaggle dataset to categorize the input images. The suggested system's advantages include security, interpretability, high accuracy, a lightweight model, and rapid processing, all of which are desirable in plant research, and its performance does not degrade with time. It has the potential to be utilized in the future for real-time disease detection on plants and is easily accessible to botanists and researchers for research purposes. Another possible application for the technology is in the realm of pest control.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The detection of plant disease is a precondition for efficiently and accurately preventing plant disease in a complicated environment.The fast growth of smart farming has resulted in the digitization and data-driven diagnosis of plant disease, enabling enhanced decision support, smart analysis, and planning.The ongoing development of profound deep learning methods has made a significant contribution to the detection of plant diseases, providing a robust tool with exceptionally precise results. However, the accuracy of deep learning models is highly dependent on the volume and quality of labelled data used for training. The agriculturist in provincial regions may think that it’s hard to differentiate the malady which may be available in their harvests. It's not moderate for them to go to agribusiness office and discover what the infection may be. Our principle objective is to distinguish the illness introduce in a plant by watching its morphology by picture handling and machine learning. Pests and Diseases results in the destruction of crops or part of the plant resulting in decreased food production leading to food insecurity. Also, knowledge about the pest management or control and diseases are less in various less developed countries. Toxic pathogens, poor disease control, drastic climate changes are one of the key factors which arises in dwindled food production. Various modern technologies have emerged to minimize postharvest processing, to fortify agricultural sustainability and to maximize the productivity. Various Laboratory based approaches such as polymerase chain reaction, gas chromatography, mass spectrometry, thermography and hyper spectral techniques have been employed for disease identification. However, these techniques are not cost effective and are high time consuming. In recent times, server based and mobile based approach for disease identification has been employed for disease identification. Several factors of these technologies being high resolution camera, high performance processing and extensive built in accessories are the added advantages resulting in automatic disease recognition. Modern approaches such as machine learning and deep learning algorithm has been employed to increase the recognition rate and the accuracy of the results. Various researches have taken place under the field of machine learning for plant disease detection and diagnosis, such traditional machine learning approach being random forest, artificial neural network, support vector machine(SVM), fuzzy logic, K-means method, Convolutional neural networks etc.… Random forests are as a whole, learning method for classification, regression and other tasks that operate by constructing a forest of the decision trees during the training time. Unlike decision trees, Random forets overcome the disadvantage of over fitting of their training data set and it handles both numeric and categorical data. The histogram of oriented gradients (HOG) is an element descriptor utilized as a part of PC vision and image processing for the sake of object detection. Here we are making utilization of three component descriptors: 1. Hu moments 2. Haralick texture 3. Color Histogram Hu moments is basically used to extract the shape of the leaves. Haralick texture is used to get the texture of the leaves and color Histogram is used to represent the distribution of the colors in an image.

CHAPTER 2

LITERATURE SURVEY

[1] India's population and food needs depend on agriculture. Boost agriculture. Infections hurt agriculture. Identifying plant diseases helps. Machine learning will identify plant illnesses since it analyses data. Data-prioritizing machine learning algorithms may identify illnesses. This AI-based approach diagnoses plant illnesses. Deep learning enhances plant disease detection. Deep learning enhanced ML and CV. Deep learning surpasses machine learning. Deep learning can identify leaf diseases in photos to reduce losses. The advantages are It's easy to use and acceptable for real-life scenarios and the disadvantages are Extremely sensitive to noise.

[2] Plant diseases harm sustainable agriculture and cost money. Identifying new agricultural illnesses requires a mobile plant disease detection device. Deep learning techniques are limited in mobile applications because they need more memory and processing power. We provide a mobile-based, lightweight deep learning model with a small footprint and processing capability to solve this problem. This method was 97% accurate on apple, citrus, and tomato leaf datasets. One of our models got 93.33% accuracy on images of apple leaves in the field, confirming the framework's applicability. The proposed model is lighter than comparable ones.

[3] Predicting plant diseases manually is expensive, time-consuming, and requires expertise. These challenges are solved by machine learning and deep learning. Accurate models help make predictions and decisions. Plant species and disease prediction may be investigated together. We propose and evaluate a novel multi-task learning approach that uses

shared representations. We use a multi-input network to predict plant species and disease. We design a CNN-and-transferred-characteristics multitask model. We test using public data. Our study demonstrated that our Multi-Input Multi-Task Neural Network model enhances efficiency and learning speed.

[4] Climate change affects Latin American and Caribbean farming. This impacts food safety, production, and quality. Deep learning might identify plant diseases. Unseen samples aren't well-predicted. Ineffective disease detection algorithms hinder crop management. Sample selection bias hinders scalability of plant disease detection systems. We use Bayesian deep learning and uncertainty to construct a probabilistic programming technique for plant disease detection. Bayesian inference performs similarly to deep learning model optimization. The recommended approach estimates plant disease posterior density and out-of-sample prediction uncertainty.

[5] Early sickness diagnosis affects 7 billion people's food security. Most plant disease deep learning algorithms fail on independent data. This study trains CNN models using segmented image data. S-CNN model trained using segmented photos doubles F-CNN model's accuracy to 98.6% with 10 sickness diagnoses. We show that S-CNN model self-classification confidence is greater than F-CNN using tomato plant and target spot disease. This breakthrough allows non-experts to use automated sickness detection. The advantages are model performance on independent data increases and the disadvantages are reduced performance when applied to real world images

[6] Convolutional neural network models were constructed to identify and diagnose plant diseases using basic photos of healthy and sick leaves. The models were trained using 87,848 photos of 25 different plants in 58

classes of [plant, illness] pairings, including healthy plants. The optimal model architecture had a 99.53 percent success rate in detecting [plant, illness] combinations (or healthy plant). The model's high success rate makes it a valuable advising or early warning tool, and it might be developed to support a real-world plant disease diagnosis system. The disadvantages are plants checking condition(season) is not mentioned

[7] This research examines AlexNet, VGG16, ResNet50, and DenseNet121 as edge solutions. PlantVillage data are used. Image processing and downsampling reduced class imbalance. DenseNet121 was implemented because it was most accurate (96.4%). The model is validated on numerous endpoint devices (CPU, GPU, VPU) in different programming environments (standard PyTorch and OpenVINO). VPU has high recall, accuracy, and F1 scores. The model's successful implementation on VPU shows its potential as a plant disease edge solution/embedded application.

[8] Plant diseases and pests threaten agriculture. Early detection and diagnosis are crucial. Deep learning techniques have substantially aided in the identification of plant diseases, providing a powerful tool with exact results. However, the effectiveness of deep learning models relies on the amount and quality of labelled data for training. In this study, we present a deep learning-based technique for tomato disease diagnosis that uses C-GAN to produce synthetic leaf pictures. A DenseNet121 model trained on simulated and real pictures classifies tomato leaf images into 10 disease groups. The suggested model was trained and tested using PlantVillage data. The suggested approach classified tomato leaf images accurately into 5 classes, 7 classes, and 10 classes. The suggested methodology trumps current methods. The advantages are Better diagnosis methodology and the disadvantages are accuracy is missing

[9] Plant diseases and pernicious insects are a considerable threat in the agriculture sector. Therefore, early detection and diagnosis of these diseases are essential. The ongoing development of profound deep learning methods has greatly helped in the detection of plant diseases, granting a vigorous tool with exceptionally precise outcomes but the accuracy of deep learning models depends on the volume and the quality of labeled data for training. In this paper, we have proposed a deep learning-based method for tomato disease detection that utilizes the Conditional Generative Adversarial Network (C-GAN) to generate synthetic images of tomato plant leaves. Thereafter, a DenseNet121 model is trained on synthetic and real images using transfer learning to classify the tomato leaves images into ten categories of diseases. The proposed model has been trained and tested extensively on publicly available PlantVillage dataset. The proposed method achieved an accuracy of 99.51%, 98.65%, and 97.11% for tomato leaf image classification into 5 classes, 7 classes, and 10 classes, respectively. The proposed approach shows its superiority over the existing methodologies. The advantages are highly accurate the disadvantages are accuracy will be question apart from sample dataset

[10] Agriculture faces serious danger from plant diseases and pest insects. As a result, it is crucial to identify and diagnose such conditions early. Accuracy of deep learning models is dependent on the quantity and quality of labelled data for training, however the continual development of deeper deep learning techniques has tremendously aided in the identification of plant diseases, offering a powerful instrument with extremely exact results. We present a deep learning-based strategy for tomato disease diagnosis in this study, one that makes use of the Conditional Generative Adversarial Network (C-GAN) to produce synthetic pictures of the leaves of tomato plants. Next, we use transfer learning

using a DenseNet121 model that was trained on both synthetic and real pictures to categorise the tomato leaves images into 10 types of illness. Using the freely accessible PlantVillage dataset, the suggested model has been trained and tested extensively. Tomato leaf images were successfully classified into 5 classes, 7 classes, and 10 classes with an accuracy of 99.51, 98.65, and 97.11 percent, respectively, using the suggested technique. Compared to the current methods, the suggested methodology clearly excels. The advantages are faster output driven the disadvantages are accuracy will be driven low when rare disease cases

2.1 INFERENCES FROM LITERATURE SURVEY

From the above-mentioned literature works, it is clear that there has been effective research on this topic has been done and many models have been proposed. It is evident that the above-mentioned systems have their own pros and cons. While some of the recent works involve hybrid technologies and provide better accuracies, they are still far from what is needed. With higher accuracy, comes the need for low computational costs, high processing speed, and most of all, the convenience of use.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

The existing systems are based on ANN and are found to be less accurate and almost incompatible with most datasets and applications. Although prediction results achieved are promising, these traditional approaches are still far from being highly accurate and efficient. The existing systems are simple and effective but are extremely vulnerable to impact. Moreover, state-of-the-art methods can predict outputs pertaining to certain parts of the plant only while some pest infestations may go completely undetected. Also in the existing system, the plant disease is detected using leaves only.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

FEASIBILITY STUDY

The feasibility of the project is server performance increase in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Operational feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

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

OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Hardware specifications:

- Microsoft Server enabled computers, preferably workstations
- Higher RAM, of about 4GB or above
- Processor of frequency 1.5GHz or above

Software specifications:

- Python 3.6 and higher
- Anaconda software

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Import of necessary libraries. We utilised a downloadable dataset to train our neural network model and expanded it to include real-time picture testing through Kaggle.The method begins with preprocessing the data from a huge dataset that has been provided. At this step, the data is cleaned and pre-processed, with missing and null value records being deleted.We removed all null values from our dataset and verified that all data types were valid.Preprocessing is mostly used to discover and eliminate or substitute missing values in a dataset that represent a very tiny percentage of the total data, in order to assure an accurate output.EDA is then performed. The primary goal of EDA is to uncover hidden patterns in a dataset and to detect previously unknown values.

Building the algorithm

The dataset's input photos are transformed to vector form.
The Embedding Layer converts the characteristics to a new representation known as word embedding.The output of the Embedding layer is sent to the CNN layer for feature extraction.Max-pooling was utilized in this model. Furthermore, the batch normalization layer was utilized since it required the training of a very deep neural network.As a result, the approach modifies the scaling and activation in order to normalize the input layer and accelerate the learning process between hidden units.A flatting layer at the end of the CNN utilized in the research converts the output of convolutional

layers into a single-dimensional feature vector.Retraining the model would enhance accuracy**.**

Prediction of Output

The last module of the system is a projection of the result.To begin, a camera is used to acquire an image of a mazie crop plant part, and then algorithms will be used to determine the disease classified.The image or picture is fed into the network architecture.If the system detects any indicators of any disease, it provides the output and names the disease of the plant.The evaluation of the model is a critical step in the model-creation process.With its assistance, we can assess which model best describes our data and how well it will function in the future.Accuracy is defined as the percentage of accurately anticipated values given testing data.Our detection system's capabilities may be accessed via agricultural management systems, desktop applications, and websites.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

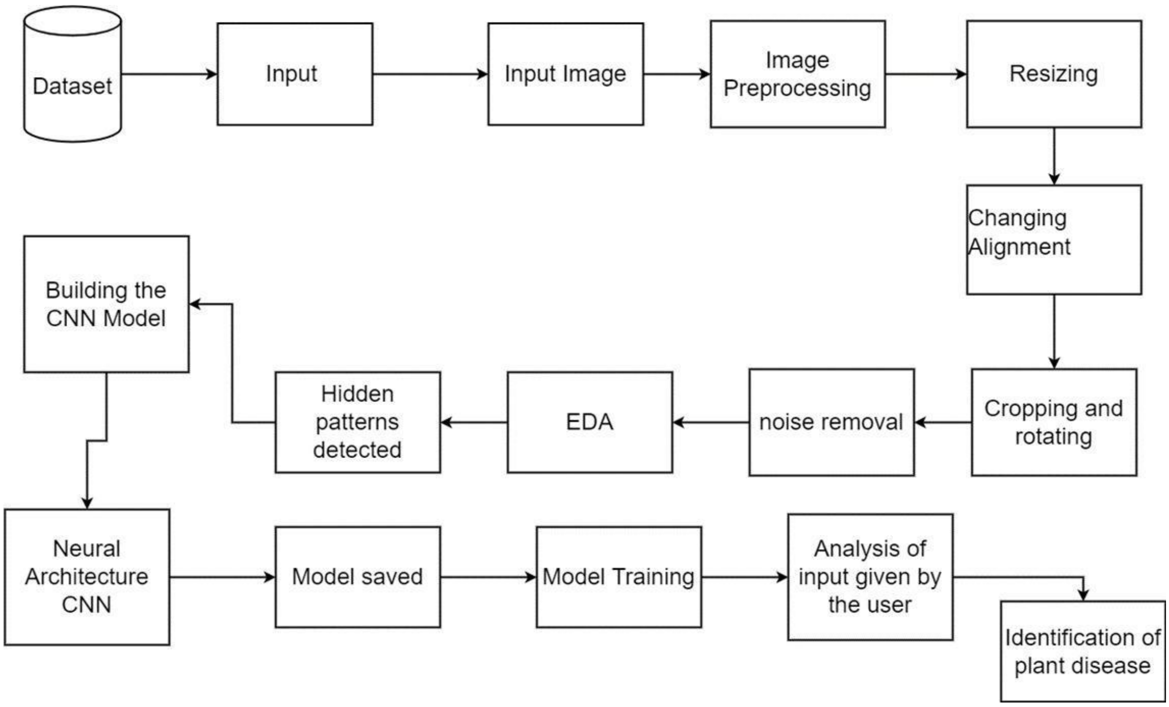

Fig 4.2: System Architecture

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field.Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project with a few clicks when you’re using Anaconda.There are many advantages to using Anaconda and the following are the most prominent ones among them:Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don't face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn't be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues.It gives you a seamless environment which simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it's a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let's discuss some fundamental commands you can use to start using this package manager.

Listing All Environments

To begin using Anaconda, you'd need to see how many Conda environments are present in your machine.

conda env list

It will list all the available Conda environments in your machine.

Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

conda create -n <your_environment_name>

You can replace <your_environment_name> with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

conda create -n <your_environment_name> python=3.6

Similarly, if you want to create an environment with a particular package, you can use the following command:

conda create -n <your_environment_name>pack_name

Here, you can replace pack_name with the name of the package you want to use.

If you have a .yml file, you can use the following command to create a new Conda environment based on that file:

conda env create -n <your_environment_name> -f <file_name>.yml

We have also discussed how you can export an existing Conda environment to a .yml file later in this article.

Activating an Environment

You can activate a Conda environment by using the following command:

conda activate <environment_name>

You should activate the environment before you start working on the same. Also, replace the term <environment_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

conda deactivate

Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

conda install <pack_name>

Replace the term <pack_name> with the name of the package you want to install in your Conda environment while using this command.

Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

conda update

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

conda install <package_name>=<version>

Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can create an environment configuration .yml file and share it with that person. Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you'll first have to activate the same and run the following command:

conda env export ><file_name>.yml

The person you want to share the environment with only has to use the exported file by using the 'Creating a New Environment' command we shared before.

Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

conda remove -n <env_name><package_name>

On the other hand, if you want to uninstall a package from an activated environment, you'd have to use the following command:

conda remove <package_name>

Deleting an Environment

Sometimes, you don't need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

conda env remove –name <env_name>

The above command would delete the Conda environment right away.

4.4 PROJECT MANAGEMENT PLAN

Introduction:	**September 1-30**
Literature Survey:	**October 1-31**
System Design:	**November 1-30**
System Implementation:	**December 1-31**

Testing:	Januray 1-30

.

REFERENCES:-

[1] C Jackulin, S. Murugavalli,A comprehensive review on detection of plant disease using machine learning and deep learning approaches,Measurement: Sensors,2022

[2]Sivasubramaniam Janarthan, Selvarajah Thuseethan, Sutharshan Rajasegarar, John Yearwood,P2OP—Plant Pathology on Palms: A deep learning-based mobile solution for in-field plant disease detection,Computers and Electronics in Agriculture,2022

[3]Ali Seydi Keceli, Aydin Kaya, Cagatay Catal, Bedir Tekinerdogan,Deep learning-based multi-task prediction system for plant disease and species detection,Ecological Informatics,2022

[4]S. Hernández, Juan L. López,Uncertainty quantification for plant disease detection using Bayesian deep learning,Applied Soft Computing,2020

[5] Parul Sharma, Yash Paul Singh Berwal, Wiqas Ghai,Performance analysis of deep learning CNN models for disease detection in plants using image segmentation, Information Processing in Agriculture,2020

[6] Konstantinos P. Ferentinos,Deep learning models for plant disease detection and diagnosis,

Computers and Electronics in Agriculture,2018

[7] Aanis Ahmad, Dharmendra Saraswat, Aly El Gamal,A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools,
Smart Agricultural Technology,2022

[8] Soo Jun Wei, Dimas Firmanda Al Riza, Hermawan Nugroho,
Comparative study on the performance of deep learning implementation in the edge computing: Case study on the plant leaf disease identification,Journal of Agriculture and Food Research,2022

[9] Amreen Abbas, Sweta Jain, Mahesh Gour, Swetha Vankudothu,Tomato plant disease detection using transfer learning with C-GAN synthetic images,Computers and Electronics in Agriculture,2021

[10] Nirmal Raj, Senthil Perumal, Sanjay Singla, Girish Kumar Sharma, Shamimul Qamar, A. Prabhu Chakkaravarthy,Computer aided agriculture development for crop disease detection by segmentation and classification using deep learning architectures,Computers and Electrical Engineering,2022