TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KALANKI, KATHMANDU

A MAJOR PROJECT PROGRESS DEFENSE REPORT ON

"Tomato Leaf Disease Detection"

[CT707]

SUBMITTED BY:

Aakash Khanal (ACE076BCT001)

Aavash Gyawali (ACE076BCT003)

Bigyan Dahal(ACE076BCT025)

Deekshit Bhattrai(ACE076BCT035)

UNDER SUPERVISION OF:

Er. Laxmi Prasad Bhatt

A Major project progress report submitted to the department of Electronics and Computer Engineering in the partial fulfillment of the requirements for degree of Bachelor of Engineering in Computer Engineering

Kathmandu, Nepal

December 23, 2023

ACKNOWLEDGEMENT

We take this opportunity to express our deepest and most sincere gratitude to our Project Coordinator **Er. Laxmi Prasad Bhatt**, Department of Electronics and Computer Engineering for his insightful advice, motivating suggestions, invaluable guidance, help, and support in the partial fulfillment of this project, and for his/her constant encouragement and advice throughout our Bachelor's program.

We express our deep gratitude to **Er. Prem Chandra Roy**, Head of the Department of Electronics and Computer Engineering, **Er. Bikash Acharya and Er. Dhiraj Pyakurel**, Deputy Head, Department of Electronics and Computer Engineering for their support, cooperation, and coordination.

The in-time facilities provided by the department throughout the Bachelor's program are also equally acknowledgeable.

We would like to thank the teaching and non-teaching staff of the Department of Electronics & Communication and Computer Engineering, ACEM for their invaluable help and support throughout their Bachelor’s Degree. We also thank all our classmates for their service, encouragement, and valuable suggestions.

Finally, yet, more importantly, I would like to express our deep appreciation to my parents for their perpetual support and encouragement throughout the Bachelor’s degree period.

Aakash Khanal (ACE076BCT001)
Aavash Gyawali (ACE076BCT003)
Bigyan Dahal(ACE076BCT025)
Deekshit Bhattrai(ACE076BCT035)

ABSTRACT

Automated detection and segmentation of plant diseases have emerged as pivotal tools in agricultural technology, enabling early and accurate diagnosis for effective crop management. In this study, we propose a novel approach utilizing MASK R-CNN, a state-of-the-art deep learning architecture, for the identification and segmentation of tomato leaf diseases. Leveraging a meticulously annotated dataset comprising diverse instances of healthy and diseased tomato leaves, our model learns to discern and segment the affected regions with high precision. The methodology involves training the MASK R-CNN architecture on annotated images, and fine-tuning the model to achieve optimal performance. By harnessing the power of convolutional neural networks (CNNs) and instance segmentation techniques, our system demonstrates robustness in detecting multiple types of diseases, including but not limited to Tomato_Healthy, Tomato_Leaf_Spot and Tomato_Leaf_Mold. The evaluation of our model showcases promising results, with high accuracy and efficiency in disease identification and segmentation. Furthermore, the system's ability to provide precise delineation of infected areas on tomato leaves aids in early intervention strategies, facilitating timely disease management and crop preservation. This research contributes to the domain of agricultural technology by offering an automated solution for rapid and accurate detection of tomato leaf diseases. The proposed approach holds significant potential for real-time implementation in smart farming systems, assisting farmers in making informed decisions for better crop yield and sustainable agricultural practices.

Keywords: *Mask R-CNN, Convolution Neural Network, segmentation, disease management, smart farming system.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
GUI	Graphical User Interface
k-NN	K-Nearest Neighbors
RNN	Recurrent Neural Network
R-CNN	Region-based Convolutional Neural Network
SIFT	Scale-Invariant Features
SVM	Support Vector Machine

CHAPTER 1
INTRODUCTION

1.1 Background

Tomato plants are widely cultivated and highly valued for their nutritious fruits. However, they are susceptible to various diseases that can severely impact their growth and yield. The early detection and accurate identification of these diseases are crucial for implementing timely control measures and minimizing crop losses. With the advancements in machine learning and artificial intelligence, these technologies have gained attention as potential tools for automated disease detection in agriculture. In this project, we aim to identifying tomato leaf diseases.

CNN is a deep learning algorithm that has shown exceptional performance in image recognition tasks. Its ability to automatically learn hierarchical features from raw image data makes it well-suited for analyzing visual patterns and detecting complex disease symptoms in tomato leaves.

Mask R-CNN stands as a transformative tool in the realm of image segmentation, especially when it comes to detecting diseases within tomato leaves. Developed as an extension of the Faster R-CNN architecture, Mask R-CNN revolutionizes the field by integrating instance segmentation capabilities. By leveraging its prowess in object detection, Mask R-CNN takes a quantum leap forward by delineating not just objects but also precisely outlining their contours through pixel-level segmentation masks.

In the context of tomato leaf disease detection, Mask R-CNN operates as a sophisticated solution capable of discerning subtle anomalies or infections within the leaf's intricate structure. By meticulously identifying and segmenting affected regions, it offers a precise understanding of the disease's extent, enabling swift and accurate diagnosis.

This methodology is instrumental in agricultural contexts, aiding in the early detection and targeted treatment of diseases that afflict tomato plants. Mask R-CNN's ability to delineate the boundaries of lesions or affected areas on the leaf surface plays a pivotal role in facilitating rapid intervention strategies, thereby safeguarding crop yield and quality.

Employing convolutional neural networks (CNNs) and a multi-stage architecture, Mask R-CNN's fusion of region-based detection and pixel-level segmentation empowers it to not just recognize the presence of disease but to precisely outline and isolate the infected areas. Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. [1]

In summary, Mask R-CNN stands as a beacon of innovation in image segmentation, particularly in the critical domain of identifying diseases in tomato leaves. Its prowess in delineating precise boundaries of anomalies within the foliage has redefined the landscape of agricultural diagnostics, promising enhanced crop health and yield through swift, accurate, and proactive measures against diseases.

1.2 Motivation

Detecting diseases in tomato plants using CNN and Mask R-CNN is a pursuit fueled by the convergence of technological innovation and agricultural sustainability. By employing Mask R-CNN's sophisticated image segmentation capabilities, the project endeavors to revolutionize the way we safeguard crop health and yield. This endeavor is not merely about leveraging advanced AI for disease detection; it encapsulates a broader mission encompassing economic stability, environmental consciousness, and food security. Through early disease identification, this initiative aims to empower farmers with timely interventions, reduce reliance on harmful pesticides, bolster scientific research in agricultural diagnostics, and ultimately fortify global efforts towards sustainable farming practices. At its core, the motivation lies in fostering resilience within agricultural communities, preserving crop yields, and contributing to a more sustainable and secure food future for generations to come.

1.3 Statement of the Problem

The detection and accurate identification of diseases in tomato plants pose significant challenges in modern agriculture. Manual inspection techniques are time-consuming and prone to human error, hindering timely interventions.

1.4 Significance of the study

This study holds significant value as it provides a comparative evaluation of machine learning algorithms for tomato leaf disease identification, contributing to the development of an automated system that can effectively detect diseases, minimize crop losses, optimize crop management practices, and enhance the sustainability of agricultural operations.

1.5 Project objective

- Tomato Leaf Disease identification using Mask R-CNN and CNN

CHAPTER 2
LITERATURE REVIEW

Qimei Wang et al, proposed the object detection models with deep CNN architectures. The, mask R-CNN with ResNet-101 shows a higher detection rate and performance of 99.64%mAP. Merits include accurate and quick performance. Demerits include Mask-RCNN training time is longer than faster R-CNN. [2]

H.Al-Hiary et al, proposed a plant disease detection method, using the k-means clustering algorithm with Neural Network. Both the detection and classification of plant diseases can be identified by this trained model. It provides a precise accuracy between 83% and 94%. Merits include precise disease detection with less computational effort. Demerits are recognition rate is found to be declined. [3]

Yang Zhang et al, proposed a faster RCNN with res101 based model. An accuracy of 98.54% mAP is obtained. Merit includes, this proposed crop disease detection technique shows faster detection speed compared to original faster RCNN. Demerits include only a solitary leaf disease in the given image is being detected. [4]

Robert G.de Luna et al, proposed a CNN and F-RCNN based trained model. The proposed method acquires a 91.67% performance. Merits include, transfer learning recognizing model gains high accuracy. Demerits include retraining the CNN for high performance is required. [5]

Mehmet Metin Qzguven et al, proposes a faster R-CNN architecture. The proposed model is time-consuming in disease detection rates. A maximum and overall classification accuracy of 99.25% is achieved. Merits include time spent in identifying diseases and human errors are reduced in disease identification. Demerits include large data inputs the disease detection accuracy is poor. [6]

Mehra et al. focused on identifying the presence of fungal infection in leaves using k-means clustering. Their approach utilized k-means for disease detection and categorization, addressing the challenges of clustering algorithms in determining cluster counts and parameter adjustment. [7]

Dandawate and Kokare explored the use of scale-invariant features (SIFT) for disease identification in plants. They employed SIFT in combination with the support vector machine (SVM) to identify the presence of diseases in plant images. In order to achieve effective disease classification, Dandawate and Kokare combined SIFT features with SB distribution for detecting diseases in tomato leaves. This combination of features improved the accuracy of disease detection and classification. [8]

Rothe et al. proposed a pattern recognition technique using snake segmentation to assess the vitality of infected parts in plant leaves. This method achieved a classification accuracy of 85.52% for disease identification. [9]

Rastogi et al. used k-means clustering to segment the affected parts of plant leaves and applied an artificial neural network (ANN) for classification. The proposed approach helped identify the severity of diseased leaves. [10]

Hall et al. employed random forest and convolutional neural network (CNN) for leaf classification among 32 different species. Their approach achieved a classification accuracy of 97.3% using a large dataset. [11]

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Software Requirement

- Python
- React
- Fast API

3.2 Hardware Requirement

- RAM : 4 GB
- Hard Disk : minimum of 1GB free space

3.3 Functional Requirements

- The user must be able to upload images of leaves.
- The user must be able to get predicted output as result.

3.4 Non-Functional Requirements

- The system should be flexible to changes.
- The system will be extensible to the latest changes.
- The GUI of the system will be user-friendly.
- The system should easy to recover if any crashes occur.

3.5 Feasibility Study

3.5.1 Economic Feasibility

The development of this application is highly economically feasible. The only thing to be done is making an environment with effective supervision. It is cost-effective in the sense that it doesn’t require any kind of additional paperwork or supervision of a human. The system is also time effective since necessary data is provided in real-time.

3.5.2 Technical Feasibility

The technical requirement for the system is economical and it does not use any other additional Hardware and software. A technical evaluation must also assess whether the existing systems can be upgraded to use the new technology.

3.5.3 Operational Feasibility

The system working is quite easy to use and learn due to its simple but attractive interface. The user requires no special training for operating the system. Any user who has the basic knowledge to operate a web browser can operate the system.

3.5.4 Social Feasibility

Our system will be very socially friendly. Everyone will be able to use this system for the identification and classification of diseases. This will be highly cost-effective and easy to adapt to in society and farmer will be able to find different kinds of diseases.

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

4.1 Block Diagram

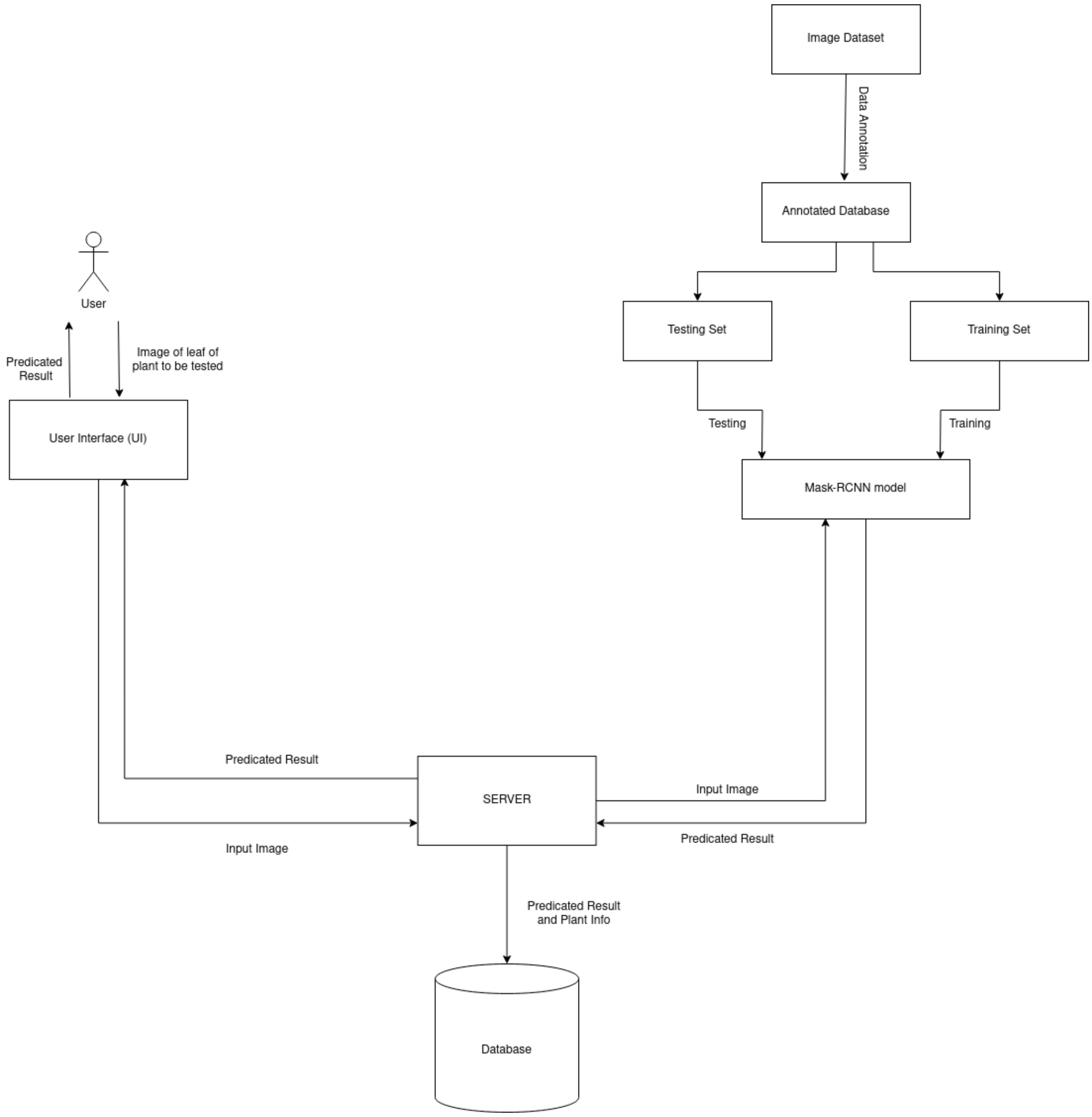

Figure 4.1: Block Diagram of Tomato Leaf Disease identification

4.2 DFD

4.2.1 DFD level 0

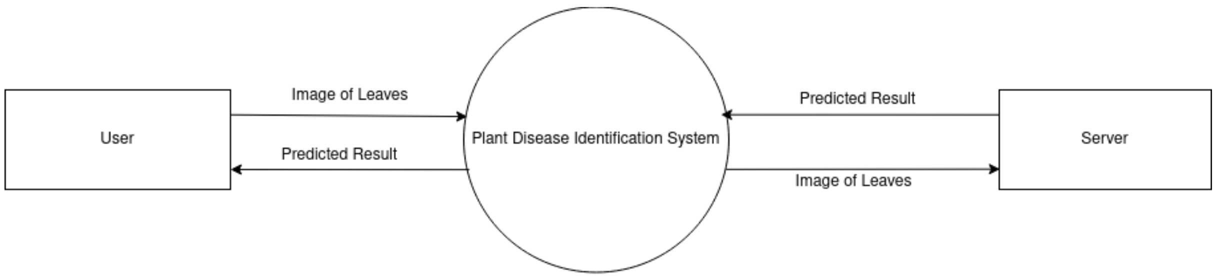

Figure 4.2.1: DFD Level 0

4.2.2 DFD level 1

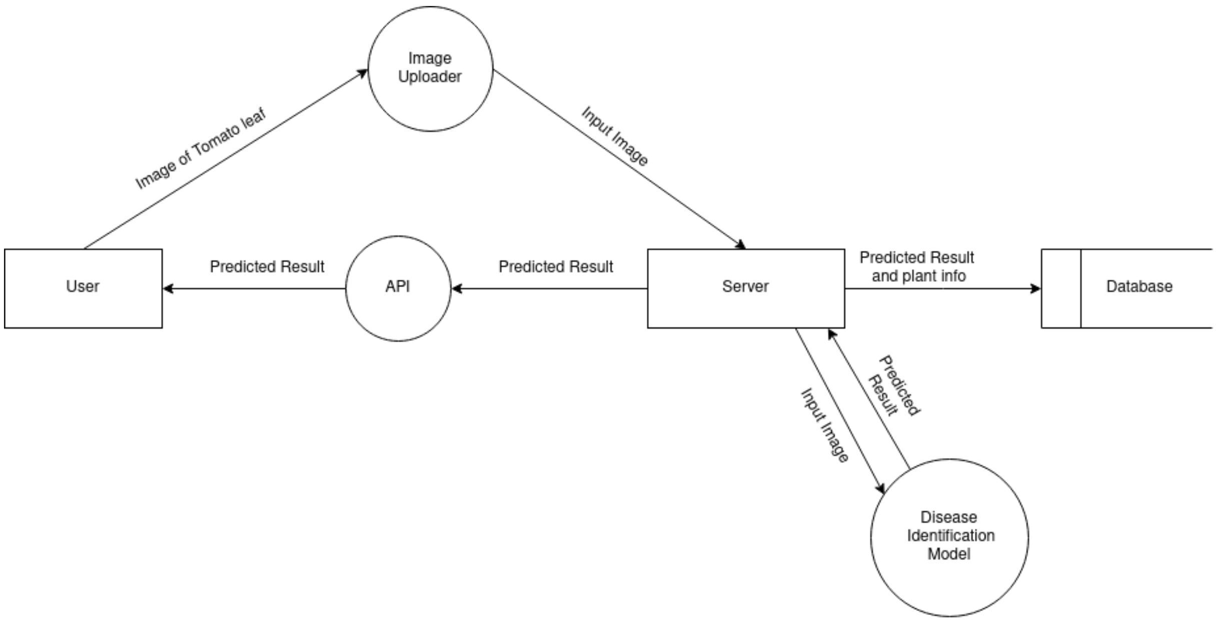

Figure 4.2.2: DFD Level 1

4.3 Usecase Diagram

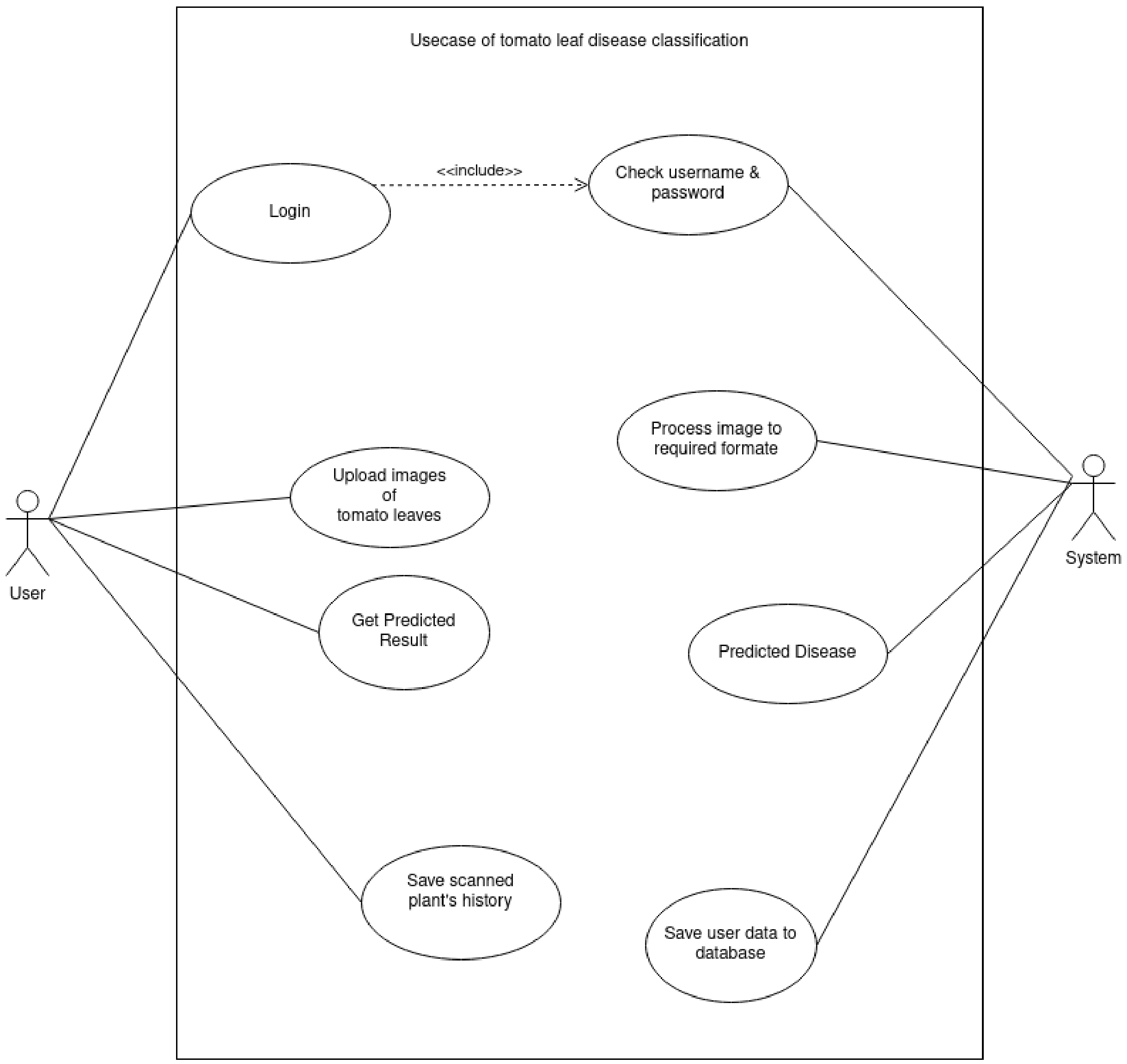

Figure 4.3: Usecase Diagram

CHAPTER 5
METHODOLOGY

5.1 Iterative and Incremental Model

The iterative and incremental model is a software development approach that emphasizes the progressive and cyclical nature of the development process. It involves breaking down the development of a software system into smaller, manageable portions and then repeatedly refining and enhancing these portions through iterations.

In the iterative aspect of the model, the development process is divided into iterations, each of which involves a complete cycle of designing, implementing, testing, and reviewing a subset of the system's functionality. Each iteration produces a working version of the software, which can be tested and evaluated by stakeholders. Feedback and insights gained from each iteration are used to inform subsequent iterations, allowing for continuous improvement and refinement of the system.

The incremental aspect of the model involves gradually adding new features or functionality to the software system over multiple iterations. With each iteration, the software evolves and grows, incorporating new features and enhancements while building upon the existing functionality. This incremental approach allows for regular delivery of functioning software, enabling early feedback, and facilitating the adaptation of requirements based on changing needs.

The iterative and incremental model promotes flexibility, collaboration, and adaptability throughout the development process. It allows for the early detection and correction of issues, reduces risk by addressing potential problems early on, and facilitates a more responsive approach to changing requirements. Additionally, it encourages stakeholder involvement and feedback at various stages, ensuring that the final software product aligns closely with user expectations.

Overall, the iterative and incremental model is well-suited for complex projects where requirements are not fully known or may change over time. It supports a dynamic and responsive development process, fostering continuous improvement and delivering value to stakeholders at regular intervals. [12]

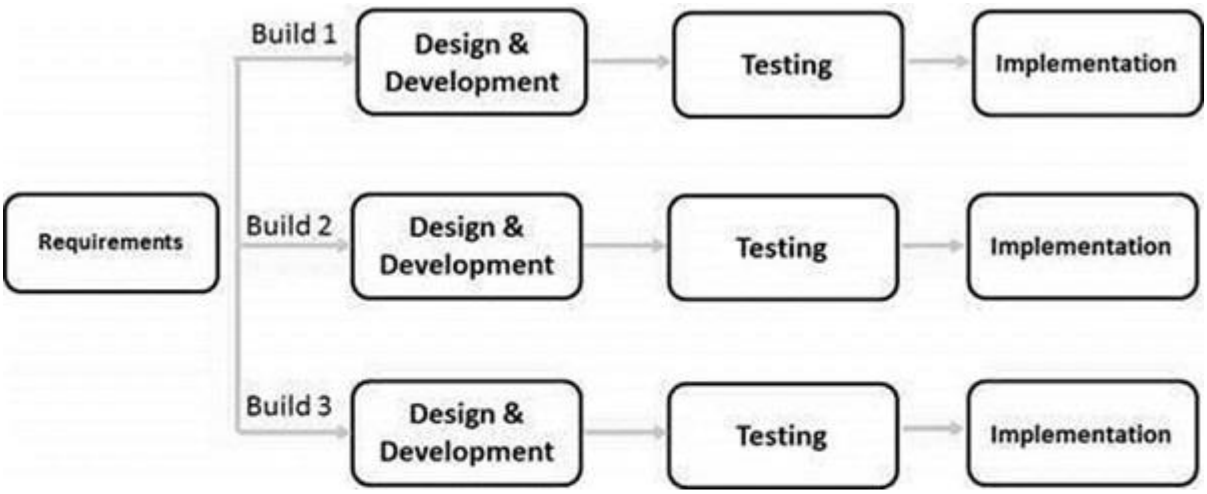

Figure 5.1: Iterative and Incremental Model

[Source: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.html/ accessed Jun 13 2023]

5.2 Algorithms used

5.2.1 CNN

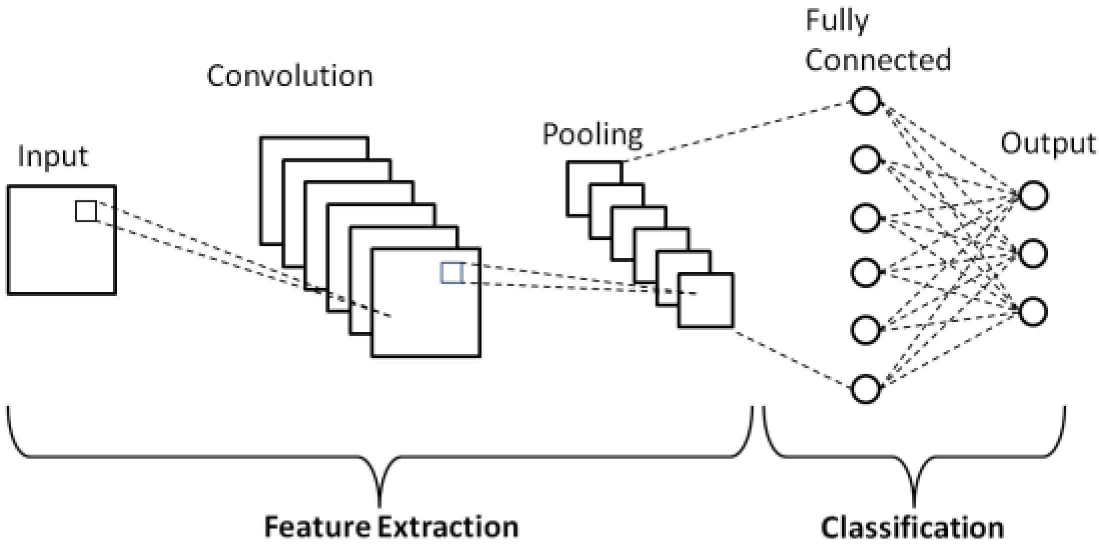

Figure 5.2.1: CNN Architecture
[Source: https://www.upgrad.com/blog/basic-cnn-architecture/ accessed Jun 22 2023]

In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series.

Biological processes inspired convolutional networks in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

5.2.2 Mask R-CNN

Mask R-CNN, or Mask RCNN, is a Convolutional Neural Network (CNN) and state-of-the-art in terms of image segmentation and instance segmentation. Mask R-CNN was developed on top of Faster R-CNN, a Region-Based Convolutional Neural Network. Faster R-CNN and YOLO are good at detecting the objects in the input image. They also have very low detection time and can be used in real-time systems. However, there is a challenge that can't be dealt with object detection, the bounding box generated by YOLO and Faster R-CNN does not give any indication about the shape of the object.

The computer vision task Image Segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). This segmentation is used to locate objects and boundaries (lines, curves, etc.).

There are 2 main types of image segmentation that fall under Mask R-CNN:

- Semantic Segmentation
- Instance Segmentation

Semantic Segmentation

Semantic segmentation classifies each pixel into a fixed set of categories without differentiating object instances. In other words, semantic segmentation deals with the identification/classification of similar objects as a single class from the pixel level. As shown in the image above, all objects were classified as a single entity (person). Semantic segmentation is otherwise known as background segmentation because it separates the subjects of the image from the background.

Instance Segmentation

This segmentation identifies each instance (occurrence of each object present in the image and colors them with different pixels). It basically works to classify each pixel location and generate the segmentation mask for each of the objects in the image. This approach gives more idea about the objects in the image because it preserves the safety of those objects while recognizing them.

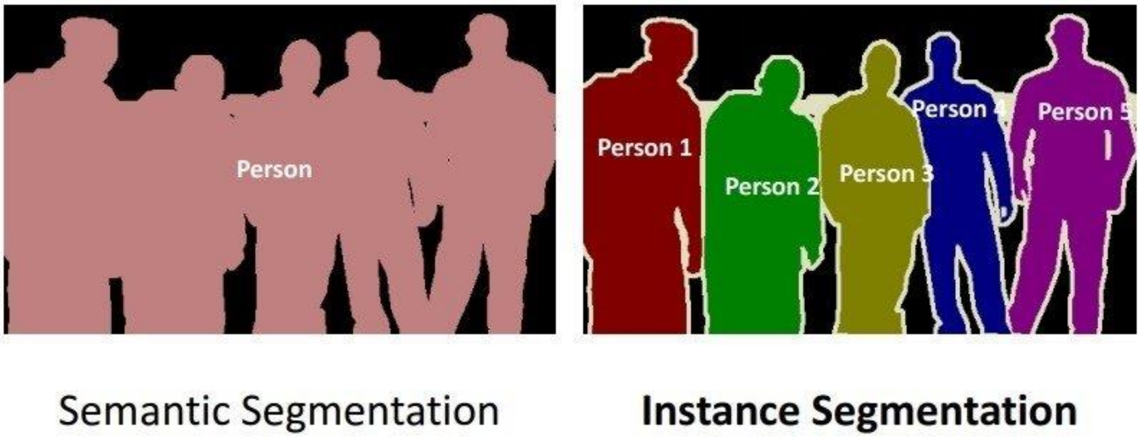

Figure 5.2.2.1: Semantic Segmentation Vs Instance Segmentation

[Source: https://viso.ai/deep-learning/mask-r-cnn /

accessed Jun 13 2023]

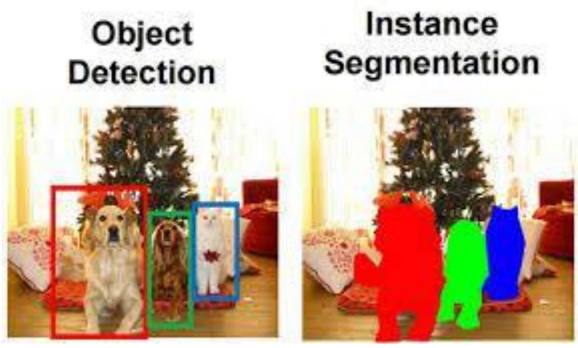

Figure 5.2.2.2: Object Detection Vs Instance Segmentation

[Source: https://www.geeksforgeeks.org/mask-r-cnn-ml /

accessed Jun 13 2023]

Mask R-CNN architecture

Mask R-CNN was proposed by Kaiming He et al. in 2017. It is very similar to Faster R-CNN except there is another layer to predict segmented. The stage of region proposal generation is the same in both the architecture the second stage which works in parallel predicts the class generates a bounding box as well as outputs a binary mask for each RoI.

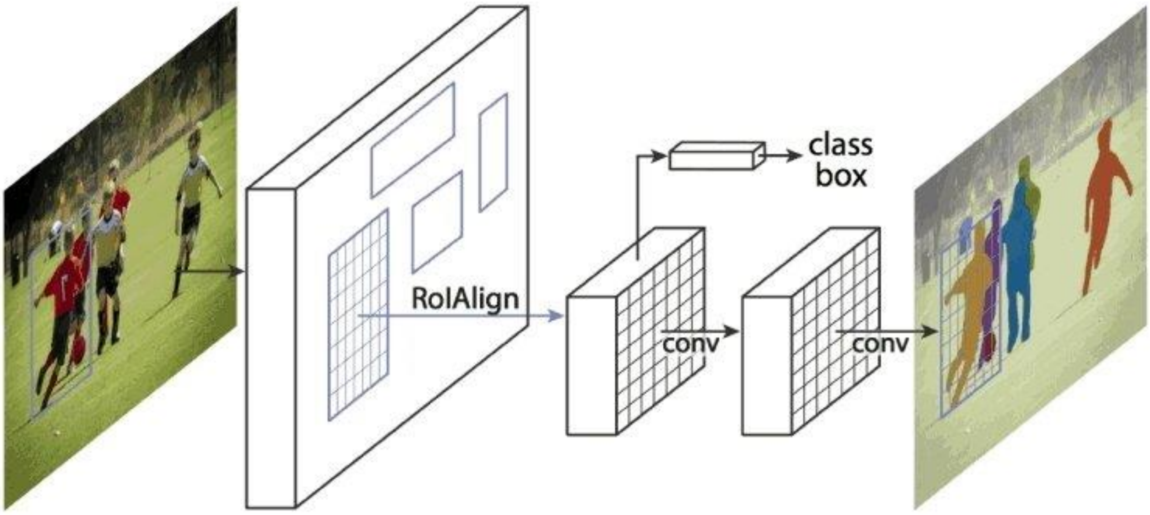

Figure 5.2.2.3: Mask R-CNN Architecture

[Source: https://viso.ai/deep-learning/mask-r-cnn/

accessed Jun 13 2023]

It comprises of

- Backbone Network
- Region Proposal Network
- Mask Representation
- RoI Align

Backbone Network

The authors of Mask R-CNN experimented with two kinds of backbone networks. The first is standard ResNet architecture (ResNet-C4) and another is ResNet with a feature pyramid network. The standard ResNet architecture was similar to that of Faster R-CNN but the ResNet-FPN has proposed some modification. This consists of a multi-layer RoI generation. This multi-layer feature pyramid network generates RoI of different scale which improves the accuracy of previous ResNet architecture. At every layer, the feature map size is reduced by half and the number of feature maps is doubled. We took output from four layers (layers – 1, 2, 3, and 4). To generate final feature maps, we use an approach called the top-bottom pathway. We start from the top feature map (w/32, h/32, 256) and work our way down to bigger ones, by upscale operations. Before sampling, we also apply the 1*1 convolution to bring down the number of channels to 256. This is then added element-wise to the up-sampled output from the previous iteration. All the outputs are subjected to 3 X 3 convolution layers to create the final 4 feature maps (P2, P3, P4, P5). The 5th feature map (P6) is generated from a max pooling operation from P5.

Region Proposal Network

All the convolution feature map that is generated by the previous layer is passed through a 3*3 convolution layer. The output of this is then passed into two parallel branches that determine the objectness score and regress the bounding box coordinates. Here, we only use only one anchor stride and 3 anchor ratios for a feature pyramid (because we already have feature maps of different sizes to check for objects of different sizes).

Mask Representation

A mask contains spatial information about the object. Thus, unlike the classification and bounding box regression layers, we could not collapse the output to a fully connected layer to improve since it requires pixel-to-pixel correspondence from the above layer. Mask R-CNN uses a fully connected network to predict the mask. This ConvNet takes an RoI as input and outputs the m*m mask representation. We also upscale this mask for inference on the input image and reduce the channels to 256 using 1*1 convolution. In order to generate input for this fully connected network that predicts mask, we use RoIAlign. The purpose of RoIAlign is to use convert different-size feature maps generated by the region proposal network into a fixed-size feature map. Mask R-CNN paper suggested two variants of architecture. In one variant, the input of mask generation CNN is passed after RoIAlign is applied (ResNet C4), but in another variant, the input is passed just before the fully connected layer (FPN Network). This mask generation branch is a full convolution network and it output a K * (m*m), where K is the number of classes (one for each class) and m=14 for ResNet-C4 and 28 for ResNet_FPN.

RoI Align

RoI align has the same motive as of RoI pool, to generate the fixed size regions of interest from region proposals. It works in the following steps:

- Given the feature map of the previous Convolution layer of size h*w, divide this feature map into M * N grids of equal size (we will NOT just take integer value).
- The mask R-CNN inference speed is around 2 fps, which is good considering the addition of a segmentation branch in the architecture.

CHAPTER 6

WORK PROGRESS

6.1 Work Progress

We have created a model using CNN and are improving the model created using Mask R-CNN

6.1.1 Model Building Using CNN:

The initial phase of the project focused on implementing a CNN-based model for tomato leaf disease classification. Convolutional Neural Networks have proven highly effective in image recognition tasks. The process involved:

Dataset: The dataset utilized for this major project has been sourced from Kaggle and is aptly named "Plant Village." Comprising a comprehensive collection of 16,011 images, this dataset encompasses a diverse range of tomato leaf diseases, meticulously categorized into 10 distinct classes. These classes encapsulate the various ailments that can affect tomato plants, such as Tomato Bacterial Spot, Tomato Early Blight, Tomato Late Blight, Tomato Leaf Mold, Tomato septoria Leaf Spot, Tomato Spider Mites (Two-Spotted Spider Mite), Tomato Target Spot, Tomato Yellow Leaf Curl Virus, Tomato Mosaic Virus, and lastly, Tomato Healthy.

Data Preprocessing: In preparation for the major project, the dataset underwent essential preprocessing steps to optimize its suitability for machine learning. The images were rescaled to pixel values between 0 and 1, ensuring consistent normalization. Augmentation techniques like a rotation range of 10 degrees and horizontal flipping were applied to enhance diversity and robustness. Additionally, all images were resized to a uniform 256 by 256 pixel size. These steps collectively ensure the dataset's quality and readiness for training accurate predictive models in the identification and classification of tomato leaf diseases

Training and Evaluation: The CNN model was trained on the dataset and evaluated using various metrics such as accuracy, loss. The model exhibited a accuracy level of 92.3% and error of 0.27, indicating the potential for improvement in subsequent iterations.

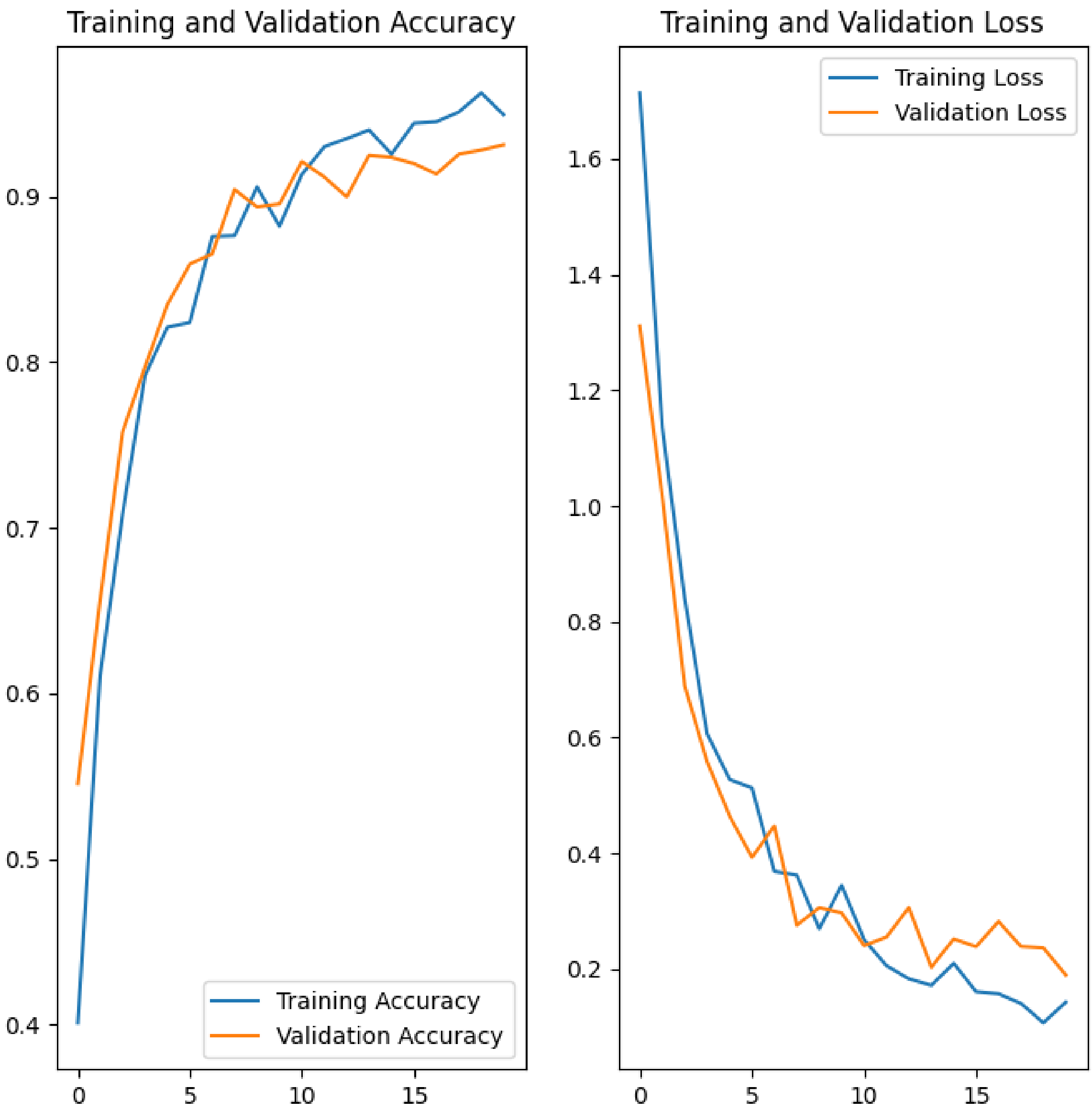

Figure 6.1.1: Training and validation Accuracy and Loss graph

6.1.2 Model Building Using Mask R-CNN:

The initial phase of the project focused on implementing a Mask R-CNN-based model for tomato leaf disease identification. The process involved:

- **Dataset with Annotation:** Comprising a comprehensive collection of 1646(training) and 410(validation) images of Tomato leaf diseases with 3 classes Tomato_Healthy, Tomato_Leaf_Spot and Tomato_Leaf_Mold with annotated json file.

6.1.3 User Interface

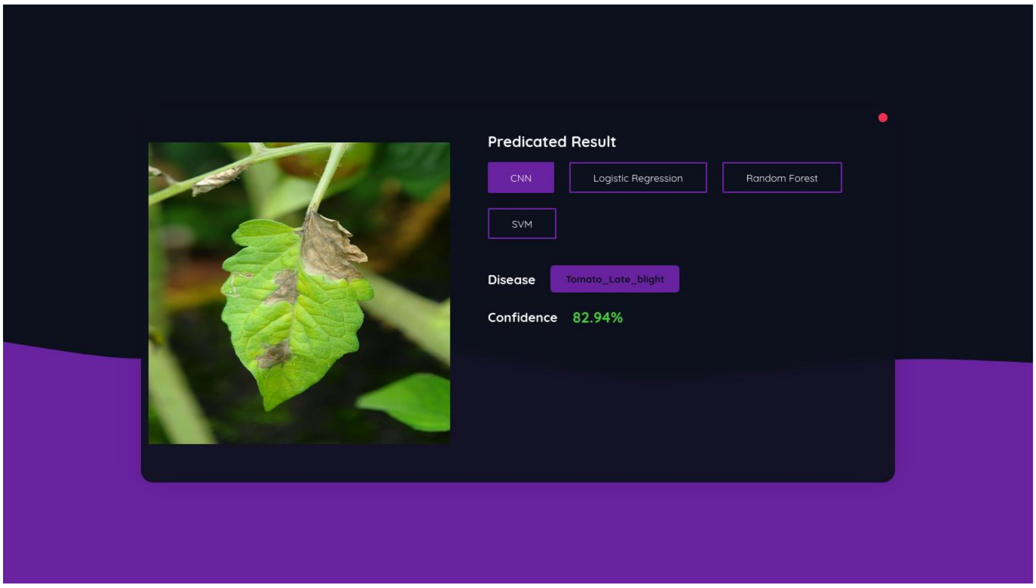

Figure 6.1.3: User Interface

6.2. Remaining Work

1. **Identification of remaining classes of Tomatoes Leaf disease.**
   The next phase requires the identification of remaining 7 classes.

2. **Annotating the Remaining classes dataset.**
   Annotation of the remaining classes is needed to be done

3. **Improving the Output of Mask R-CNN**

4. **Responsive**

5. **web design:**
   The pending tasks for implementing responsive web design encompass adapting layout elements, employing media queries, optimizing navigation menus, refining spacing and visuals, thorough cross-device testing

6.3. Challenges in Model Implementation

Dataset Size and Quality:

- Small Dataset: Collecting a sufficiently large dataset for deep learning can be challenging, especially for multiclass problems. Limited data can lead to overfitting.
- Imbalanced Classes: Having imbalanced class distributions can bias the model towards the majority class, resulting in poor performance on minority classes.
- Noisy Data: Low-quality or mislabeled images can affect model performance. Cleaning and preprocessing the dataset is crucial.

Deployment and Inference:

- Model Deployment: Deploying the model for real-world applications might involve converting it to formats like Tensor flow Lite or deploying on cloud platforms like AWS or Azure.
- Latency and Resource Constraints: Ensuring that the deployed model meets latency requirements and works efficiently on various devices can be tricky.

Debugging and Troubleshooting:

- Error Analysis: Debugging errors or low accuracy requires thorough analysis of model predictions, inputs, and outputs.
- Runtime Errors: Debugging runtime errors, such as shape mismatches or data type issues, can be challenging.

REFERENCES

[1] H. Kaiming, G. Gkioxari, P. Dollár and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE international conference on computer vision,* pp. 2961-2969, 2017.

[2] W. Qimei, F. Qi, M. Sun, J. Qu and J. Xue, "Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolution Neural Networks and Object Detection Techniques," *Computational Intelligence and Neuroscience 2019,* 2019.

[3] A.-H. Heba, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. Alrahamneh, "Fast and accurate detection and classification of plant diseases," *International Journal of Computer Applications,* pp. 31-38, 2011.

[4] Z. Yang, C. Song and D. Zhang, "Deep learning based object detection improvement for tomato disease," *IEEE Access 8,* pp. 56607-56614, 2020.

[5] L. D. RG, E. P. Dadios and A. A. Bandala, "Automated image capturing system for deep learning-based tomato plant leaf disease detection and recognition," *TENCON 2018-2018 IEEE Region 10 Conference,* pp. 1414-1419, 2018.

[6] M. M. Ozguven and A. Kemal, "Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms," *2019 10th International conference on computing, Communication and Networking Technologies (ICCCNT),* 2019.

[7] T. Mehra, V. Kumar and P. Gupta, "Maturity and disease detection in tomato using computer vision.," *Fourth international conference on parallel, distributed and grid computing (PDGC).,* pp. 399-403, 2016.

[8] Y. Dandawate and R. Kokare, "An automated approach for classification of plant diseases towards development of futuristic Decision Support System in Indian perspective," *2015 International conference on advances in computing, communications and informatics (ICACCI),* pp. 794-799, 2015.

[9] P. R. Rothe and R. V. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," *2015 International conference on pervasive computing (ICPC),* pp. 1-6, 2015 .

[10] A. Rastogi, R. Arora and S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic.," *2015 2nd international conference on signal processing and integrated networks (SPIN),* pp. 500-505, 2015.

[11] D. Hall, C. McCool, D. Dayoub, N. Sunderhauf and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," *2015 IEEE Winter Conference on Applications of Computer Vision*, p. 797–804, 2015.