# PRECISE IDENTIFICATION OF DISEASED LEAF REGIONS USING DEEP LEARNING-BASED SEMANTIC SEGMENTATION TECHNIQUES

**Thesis Submitted**
**In Partial Fulfillment of the Requirements for the degree of**

## MASTER OF TECHNOLOGY

**in**

**Artificial Intelligence**

**by**

**Anurag Nandaghale**

**(Roll No. 2K23/AFI/21)**

**Under the Supervision of**

**Prof. Rahul Katarya**

**Department Of Computer Science and Engineering**

**DELHI TECHNOLOGICAL UNIVERSITY**
**(Formerly Delhi College of Engineering)**
**Shahbad Daulatpur, Main Bhawana Road, Delhi-110042, India**

**May 2025**

# ACKNOWLEDGEMENTS

I wish to express my sincerest gratitude to **Prof. Rahul Katarya** for being my guide and supervisor on this journey. He has been my great inspiration for achieving perfection, and he always motivates us to focus on good and fruitful Research.

I am also grateful to all the researchers who worked hard and shared their great findings via research papers.

Also, I would like to thank my college Delhi Technological University for this opportunity and giving me the requisite environment.

Anurag Nandaghale
31 May 2025

**DELHI TECHNOLOGICAL UNIVERSITY**
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bhawana Road, Delhi-42

## CANDIDATE'S DECLARATION

I **Anurag Nandaghale** hereby certify that the work which is being presented in the thesis entitled **“Precise Identification of Diseased Leaf Regions Using Deep Learning-Based Semantic Segmentation Techniques”** in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of Computer Science and Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from August 2023 to Jun 2025 under the supervision of Prof. Rahul Karatya, Prof, Dept of Computer Science and Engineering.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

**Candidate’s Signature**

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge

**Signature of Supervisor** **Signature of External Examiner**

**DELHI TECHNOLOGICAL UNIVERSITY**
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bhawana Road, Delhi-42

## CERTIFICATE BY THE SUPERVISOR

Certified that **Anurag Nandaghale** (Roll No. 2K23/AFI/21) has carried out the research work presented in the thesis titled **"Precise Identification of Diseased Leaf Regions Using Deep Learning-Based Semantic Segmentation Techniques"**, for the award of Degree of Master of Technology from Department of Computer Science and Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies result of original work and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree for the candidate or submit else from the any other University /Institution.

Prof. Rahul Katarya
(Supervisor)
Department of CSE
Delhi Technological University

Date:

# ABSTRACT

Plant leaf disease segmentation has been a very important topic for research over a decade. Segmentation helps to calculate the spread of infection, and proper image preprocessing leads to better results in disease detection and identification. In this thesis we have proposed our work, findings and potential of AI models in segmenting image accurately based on recently proposed architectures. Making models too complex makes it computationally expensive. We must focus on finding the model that has less trainable parameters. This is achieved when we focus more on making architecture lossless. It can be achieved in many ways, we have proposed propagating the learnings of each previous layer to propagate in subsequent layer to avoid any information loss and up sample using indices to reduce loss. Model maintains good accuracy with very a smaller number of parameters.

Our model uses a few dense layers to trap the information within the encoder decoder architecture. This reduces exponential increment in number of channels and stills make network deep and more lossless.

# TABLE OF CONTENTS

# List Of Tables

# List Of Figures

# List Of Abbreviations

| | |
|---|---|
| MHA | Multi Head Attention |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| SL | Supervised Learning |
| GDA | Gradient Descent Algorithm |
| DL | Deep Learning |
| VIT | Vision Transformer |
| LSTM | Long Short-Term Memory |
| LRT | Local Reversible Transformer |
| GA | Genetic Algorithms |
| IoU | Intersection over Union |

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Agriculture is the basic and most important sector, being the primary sector, it affects many sectors higher in hierarchy. It affects the economy, stability, and sustainability in many countries. In the field of AI, the detection of plant disease is an important task that is used to automate the detection of pests and implement the necessary measures. Data preprocessing plays a crucial role in improving the performance of AI models [1]. Hence, it is obvious to improve those techniques individually. Here, we propose a Semantic Segmentation technique to separate background, leaf, and infected area in given leaf image. Better segmentation results in better prediction [1].

## 1.2 Motivation

As the global population grows, addressing agricultural production challenges is essential to enhance food security and efficiency. Smart farming is a promising approach that boosts agricultural productivity by utilizing advancements with the help sensors attached to iot devices, monitoring the farms using AI.

Key areas within smart farming include planning which crop to plant where, monitoring all the activities if they are performed properly, and predicting the yield, which are crucial for improving performance and expanding the economic power of modern approach of agriculture. Yield prediction vital for agricultural businesses. It helps buyers to make fresh plan based on sales estimates or customer orders.

If buyers anticipate a shortage or surplus in production, they can take proactive steps, such as sourcing extra items from retailers or redistributing over supply. Ac curate yield predictions enable better stock management, helping buyers reduce costs while keeping prices favourable for farmers. This improves sourcing strategies, meets customer needs more effectively, and fosters closer collaboration between farmers and buyers, ultimately benefiting both parties.

## 1.3 Overview

Our ultimate objective is plant disease detection from the image of leaf or leaves. To increase accuracy of deep learning models we focus on better segmentation that leads towards improvement in terms of accuracy and computational advantage.

## 1.4 Challenges

Segmentation methods have lot of problems.

- They have limited generalization; models are experimented on limited number of crops and datasets.
- Various algorithms are computationally expensive and heavy to run on normal de vices

- Data set we have for training models are manually or synthetically created, but while using models in real world, it can undergo diverse backgrounds, weather and lighting conditions

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 Literature Survey

Recently lots of efforts are made for effective leaf image segmentation. That include lot of traditional and modern deep learning methods.

In 2023, X. Zhang et al. [2] presented a model for segmenting grape disease. Its goal is to achieve precise automatic image segmentation for small diseased areas in grape leaves, which is challenging due to complex backgrounds and rich edge textures in grape diseases. The segmentation model used is the Cross-Resolution Transformer (CRFormer). It employs attention mechanism that encodes channel and spatial information, that handles tiny, infected areas in complex image conditions.

The integration of artificial intelligence in agriculture, particularly for plant disease detection and classification, has shown immense promise. Among various AI methods, deep learning-based segmentation and classification techniques have demonstrated superior accuracy, especially in identifying diseases from plant leaf images. Recent research has shifted from traditional convolutional neural networks (CNNs) to more advanced hybrid models, incorporating Transformers and optimization strategies to enhance performance and efficiency. This review synthesizes key contributions from recent studies to outline the state-of-the-art in this domain.

A significant breakthrough came with the introduction of the CRFormer model, which leverages multi-resolution attention mechanisms to segment grape leaf diseases effectively [2]. By integrating cross-resolution features, the model captures both local and global context, significantly improving its robustness to variable disease patterns and complex background textures. This approach not only enhances segmentation accuracy but also demonstrates better generalization to unseen samples, making it suitable for real-world agricultural deployment.

In a similar vein, researchers have explored multiscale convolutional designs for tomato leaf disease segmentation. A notable example is the multiscale U-Net model proposed for detecting and isolating tomato leaf lesions [3]. By incorporating convolutional filters of varying kernel sizes and hierarchical feature learning, the model achieves improved precision in identifying disease boundaries. The use of residual blocks and enhanced skip connections further boosts the model's ability to retain fine-grained details during up sampling, which is critical for accurate segmentation.

While CNNs have been the backbone of most segmentation tasks, some researchers have generalized their work to a wider range of plant species and environmental conditions. One such study proposed a robust deep learning model with an emphasis on edge-aware segmentation [4]. By enhancing boundary localization and incorporating attention mechanisms, the model achieved accurate segmentation under varying lighting and occlusion scenarios. This adaptability is vital for field conditions, where controlled environments are rare.

In addition to advanced neural architectures, some studies have taken a more systems-based approach. One such work proposed a complete pipeline for tomato leaf disease detection that spans from image preprocessing to disease treatment recommendations [5]. The integration of traditional image processing with deep learning techniques made the system more interpretable and applicable in resource-limited settings. It highlights the importance of not only developing accurate models but also ensuring they can be deployed in real agricultural ecosystems.

The U-Net architecture remains a popular baseline due to its encoder-decoder structure and ability to preserve spatial details. An improved version of U-Net, which modifies the decoder and incorporates enhanced skip connections, was shown to better handle complex and irregular lesion shapes [6]. The refinement in up sampling paths helps maintain spatial coherence, ensuring that segmented outputs align closely with ground truth masks. These improvements underline how slight architectural changes can yield significant gains in segmentation quality.

A growing trend in this domain involves integrating optimization techniques into deep learning pipelines. For instance, a hybrid framework that combines metaheuristic optimization with CNNs has been developed for the classification and segmentation of tomato leaf diseases [7]. The optimization layer fine-tunes model parameters dynamically, leading to better convergence and improved classification accuracy. This approach offers a balance between high performance and computational efficiency.

Transformers, initially designed for natural language processing, have now found applications in plant disease segmentation due to their ability to model long-range dependencies. LRT specifically adapted for grape leaf segmentation, introduces a memory-efficient reversible block structure [8]. This design reduces memory overhead during training, allowing for deeper architectures without increasing resource requirements. LRT's performance is especially noteworthy in scenarios requiring high-resolution processing.

Beyond segmentation, some works have emphasized the creation of domain-specific datasets to facilitate research. A study focusing on morning glory plant segmentation introduced a dedicated dataset and used a semantic segmentation model tailored for selective harvesting applications [9]. The availability of such benchmark datasets enables fair comparison among models and supports the development of more generalized solutions across plant species.

Efficiency remains a crucial factor for real-world application, especially in mobile or edge devices. To address this, the EAIS-Former model was introduced, combining Transformer modules with efficient attention mechanisms to segment fruit leaf diseases accurately while keeping computation low [10]. The model's architecture allows for faster inference and reduced latency, making it ideal for handheld agricultural diagnostic tools.

Further exploring lightweight architectures, the Reformer model modifies traditional convolutional kernels and applies re-parameterization to reduce redundancy in network weights [11]. This approach helps compress the model without sacrificing segmentation performance. The reduction in computational burden makes Reformer a practical choice for remote sensing applications and drone-based monitoring systems.

While deep learning dominates current literature, traditional algorithmic methods still offer value. One study employed genetic algorithms to optimize segmentation using various color spaces [12]. Although it lacks the adaptability of neural networks, this approach is particularly effective for preprocessing or enhancing image quality before feeding data into deep models. It underscores the potential of hybrid systems combining classical and modern methods.

Another lightweight Transformer-based approach, UPFormer, adopts a U-shaped architecture tailored for grape leaf disease segmentation [13]. The model prioritizes computational efficiency and generalization, making it suitable for deployment in uncontrolled field environments. Its attention layers focus on disease regions, ensuring that irrelevant background noise is minimized during segmentation.

In addition to grape and tomato plants, research has also been directed toward other crops. For example, a deep CNN system was developed for paddy plant disease identification [13]. The model balances lightweight design and high classification accuracy, with an emphasis on real-time disease detection in smallholder farming contexts. The research stresses the importance of building inclusive and adaptable tools to address the diversity of crops and farming practices worldwide.

Across all these efforts, several key themes emerge. Firstly, there is a noticeable shift from traditional CNNs to Transformer-based models due to their superior feature extraction capabilities and ability to capture long-range spatial relationships. Secondly, many recent models focus on computational efficiency, either through model compression, lightweight architectures, or hybrid approaches, ensuring practical usability in the field. Finally, the integration of real-world constraints, such as varying lighting conditions, occlusion, and background complexity, reflects a growing interest in deploying these models in practical, real-time agricultural scenarios.

To conclude, the literature demonstrates rapid advancements in deep learning-based segmentation and classification methods for plant leaf disease detection. Innovations in model design, attention mechanisms, optimization strategies, and dataset development collectively contribute to more accurate, efficient, and field-ready diagnostic tools. As the field continues to evolve, future work is likely to focus on cross-species generalization, self-supervised learning, and integration with Internet of Things (IoT) platforms to create fully autonomous crop health monitoring systems.

**Table 2.1** Literature Survey.

| Author(s) | Paper Reference | Year | Advantages | Limitations |
|---|---|---|---|---|
| Xinxin Zhang, Chaojun Cen, Fei Li, Meng Liu, Weisong Mu | [2] | 2023 | less computation than competitors specifically on the Field-PV dataset | Down sampling is irreversible and leads to loss of information. |
| Kahkashan Perveen, Sandip Debnath, Brijesh Pandey, Sumanta Prasad Chand, Najat A. Bukhari, Pradipta Bhowmick, Najla A. Alshaikh, Shaista Arzoo, Shanzeh Batool | [3] | 2023 | Decrease wastage of time on annotating dataset for with high accuracy, at very tiny dots of pixels. | dataset is unequal/biased across disease categories and is from single geographical source. |
| Olfa Mzoughi,<br>Itheri Yahiaoui | [4] | 2023 | helps to recognize new species holding diseases that were previously learnt | Biased because as related to bias in PlantVillage dataset. |
| Rahman, S.U., Alam, F., Ahmad, N. et al | [5] | 2023 | For tomato crops it is fast and effective in terms of cost. | Limited to tomato crop |
| Shanwen Zhang, Chuanlei Zhang | [6] | 2023 | Resolves overfitting, which leads to being a better model. Overpowers underfitting issue with the help of residual connection. | Model is not properly optimized and not tuned properly. |
| Manjunatha Badiger, Jose Alex Mathew | [7] | 2023 | Successfully segment tomato leaf images more accurately and effectively | Dataset was limited to specific conditions and hence model is not properly validated as per scope of generalization. |
| Jared Cervantes Canales, Jair Cervantes Canales, Farid Garcıa-Lamont, Arturo Yee-Rendon, Jose Sergio Ruiz Castilla, Lisbeth Rodriguez Mazahua | [12] | 2024 | Easily choose best segmentation techniques for any new domain using genetic algo. | experiment do not include segmentation techniques that focus on separating damage regions on leaves |
| Jiangwen Lu, Bibo Lu, Wanli Ma, Yang Sun | [10] | 2024 | Huge increment in accuracy compared to U-Net model. | High computational Cost. |

| | | | | |
|---|---|---|---|---|
| Xinxin Zhang, Fei Li, Haibin Jin, Weisong Mu | [8] | 2023 | Use reversible approach to compress data while passing through MHA. | In terms of performance, it has less Dice compared to other segmentation models specially for segmenting leaf area. |
| Jingxuan Su, Sean Anderson, Mahed Javed, Charoenchai Khompatraporn, Apinanthana Udomsakdigool, Lyudmila Mihaylova | [9] | 2023 | Reduce exponential increase in number of channels in next successive layers, leads to lower computation. Also, no layer does not have to suffer any information loss from any of the previous layers of a block. | Experimentation is not performed to segment diseased areas. |
| Xinxin Zhang, Zibo Feng, Weisong Mu | [11] | 2025 | Model works very effectively, with very high computational efficiency | Model has poor generalization ability, works only for grape leaves |
| Xinxin Zhang, Fei Li, Haiying Zheng, Weisong Mu | [13] | 2024 | Architecture supports model highly in terms of reduction of cost. | Model has poor generalization ability, works only for grape leaves. |

## 2.2 Datasets

Below is list of publicly available datasets.

**Table 2.2** List of publicly available dataset.

| Dataset Name | No of images | Year Released |
|---|---|---|
| Plant Village Dataset | 61486 | 2019 |
| Field Plant Village [7] | 665 | 2023 |
| Syn-Plant Village [7] | 54306 | 2023 |
| Rice Leaf disease dataset | 120 | - |
| Tomato leaf detection dataset | 11000 | 2022 |
| PlantDoc | 2598 | 2020 |
| Leaf disease segmentation dataset | 588 images including mask | 2021 |

## 2.3 Research Gaps and future scope

Deep learning and image processing technologies have significantly advanced the automatic detection and segmentation of plant leaf diseases, offering promising tools for precision agriculture. However, despite impressive progress, there are several intrinsic challenges that limit the current models' real-world applicability and robustness. In this section, we discuss these limitations in detail and propose potential research directions to overcome them, aiming to push the boundaries of automated plant disease diagnostics.

### 2.3.1 Limited Generalization Across Crops and Datasets

One of the critical limitations faced by existing leaf disease segmentation models is their limited generalization capability. Many deep learning systems are trained and evaluated on datasets comprising only a handful of crop types and disease classes. These models tend to overfit the specific characteristics of their training data, making them less effective when applied to new crops or regions with different disease manifestations.

The agricultural domain is highly diverse; different crops show distinct leaf structures, disease symptoms vary widely, and environmental factors further affect disease presentation. Consequently, a model trained on a narrow dataset may fail to recognize or accurately segment diseases on other crops.

To address this, novel algorithmic frameworks such as Genetic Algorithms (GAs) offer promising potential. GAs operates by evolving a population of candidate solutions through iterative processes mimicking natural selection. As proposed in recent research [12], multiple segmentation algorithms and their associated hyperparameters can be encoded as candidate solutions in a GA framework. By evaluating these candidates on a new dataset using accuracy metrics like (IoU), the GA can identify the optimal combination of segmentation methods and parameter settings. This approach enables a flexible, data-driven selection process that adapts to different datasets and crop types, improving model generalization.

Furthermore, assembling large, diverse datasets spanning multiple crops and geographic regions is crucial. Initiatives for collaborative data sharing and the creation of standardized benchmarks would significantly accelerate the development of more robust, generalizable models.

### 2.3.2 Accurate Separation of Infected Leaf Areas

Accurately segmenting the infected regions within a leaf remains a significant challenge. Many existing segmentation models focus primarily on separating the leaf from the background, which is important but insufficient for disease diagnosis that requires localizing affected portions.

- Traditional image processing methods often rely on multi-stage filtering: one filter isolates the leaf itself, and a subsequent filter attempts to identify disease spots or lesions based on color, texture, or shape cues. While somewhat effective, this process is limited by handcrafted feature design and sensitivity to varying lighting and leaf appearance.
- Deep learning-based segmentation models, particularly those using multi-class masks, offer a more precise solution. These models can be trained to predict distinct labels for healthy leaf tissue and various disease classes simultaneously. Creating such detailed training data requires manual annotation with tools like LabelMe [3], which allows fine-grained pixel-wise labeling of infected and healthy regions. This enables the model to learn subtle visual differences and improve localization accuracy.

Moreover, enhancing the dataset with diverse examples of disease progression stages and symptom variations can help the model generalize better across different infection levels and environmental conditions.

### 2.3.3 Computational Efficiency and Hardware Constraints

A practical limitation in deploying leaf disease segmentation models is the high computational cost associated with many state-of-the-art architectures [10]. Large models with millions of parameters often require powerful GPUs and extensive memory, which are not always available in agricultural settings, especially in resource-constrained environments like small farms or mobile devices.

To mitigate this, researchers have proposed several strategies:

- Region of Interest (ROI) Detection: Instead of processing the entire image at full resolution, an initial lightweight detector identifies candidate regions likely to contain diseased tissue. Subsequent segmentation is then performed only on these ROI patches, drastically reducing computation without sacrificing accuracy.
- Lightweight Model Design: Architectures such as MobileNet, ShuffleNet, or efficient transformer variants prioritize model compactness and speed [2]. Improving the accuracy of such lightweight models through techniques like knowledge distillation, quantization, and neural architecture search (NAS) can make disease segmentation feasible on edge devices.

Balancing computational efficiency with segmentation quality is critical to enable real-time disease diagnosis directly in the field, empowering farmers with accessible tools for crop health monitoring.

### 2.3.4 Dataset Bias and Class Imbalance

In many leaf disease datasets, some disease classes are underrepresented relative to others, creating a class imbalance problem. Training a model on biased data can lead to poor performance in minority classes and unreliable segmentation results [4].

Several strategies exist to counteract class imbalance:

- Data Augmentation: Artificially expanding the number of samples for rare classes by applying transformations such as rotation, flipping, cropping, and color jittering helps balance the dataset.
- Resampling Techniques: Oversampling underrepresented classes or under-sampling dominant classes during training can ensure more equitable representation.
- Loss Function Re-weighting: Assigning higher penalty weights to errors in minority classes guides the model to pay more attention to these classes.
- Feature Prioritization: Incorporating domain knowledge to prioritize specific features relevant to rare diseases during training can improve classification and segmentation accuracy.

Overall, addressing dataset bias is fundamental to creating fair and reliable disease segmentation models that perform consistently across all disease types.

### 2.3.5 Geographic and Environmental Limitations

Most disease datasets are collected from specific geographic locations, reflecting the local climate, crop varieties, and disease prevalence [3]. Consequently, models trained on such datasets often struggle to perform well when applied to new regions due to domain shifts caused by differing environmental conditions, leaf appearances, and disease symptom expressions.

Two main approaches can help bridge this gap:

- Transfer Learning: By starting with models pretrained on large, diverse datasets and fine-tuning them on smaller region-specific data, one can leverage existing learned representations and adapt to new domains more efficiently.
- Data Augmentation for Domain Diversity: Simulating a range of environmental factors, such as different lighting, weather, or leaf conditions, during training encourages models to learn invariant features robust to domain shifts.

Combining these methods with continued collection of diverse datasets from multiple regions is essential to building universal disease detection systems deployable worldwide.

### 2.3.6 Handling Diverse Lighting and Background Conditions

Agricultural images are often captured under varying lighting, weather, and background conditions, which introduces substantial visual variability. Models trained under controlled conditions or synthetically generated data may fail to generalize when deployed outdoors in real-world environments [12].

Improving robustness to such variability involves:

- Advanced Image Preprocessing: Techniques to normalize illumination, such as histogram equalization, gamma correction, or shadow removal, can standardize inputs before segmentation.
- Augmentation with Lighting Variations: Training on images augmented with different brightness, contrast, and color variations helps models learn to handle diverse lighting.
- Domain Adaptation Techniques: Methods that explicitly learn lighting-invariant features or use adversarial training to reduce domain discrepancy can further enhance performance.

These steps ensure models remain reliable in heterogeneous agricultural settings, increasing their practical utility.

### 2.3.7 Avoiding Information Loss During Downsampling

Many segmentation architectures rely on down sampling operations like pooling to reduce computational complexity and increase receptive fields. However, these operations can cause irreversible information loss, degrading the model's ability to accurately segment fine-grained details such as small disease spots.

Recent research [2, 8] highlights the benefits of using reversible down sampling techniques that preserve spatial information. Approaches like learnable pooling, dilated convolutions, and attention mechanisms help maintain detail while still achieving effective feature abstraction.

Incorporating these strategies into future model designs promises higher segmentation accuracy, particularly for diseases that manifest as subtle visual patterns.

# CHAPTER 3

# PROPOSED METHODOLOGY

## 3.1 Data Acquisition

The dataset used in this study is known as Field-PV, a publicly available dataset designed for plant visual disease analysis [8]. To tailor it to our research objectives, we refined the dataset by selecting only the images related to grape leaves. This focused approach allowed us to concentrate specifically on identifying and segmenting grape leaf diseases with greater accuracy and consistency. The initial raw dataset contained a broader variety of crops and plant species; hence, filtering it down to grape leaf images was a necessary preprocessing step to ensure domain-specific learning.

After selection, we manually annotated the images using the LabelMe annotation tool. This open-source tool allows for pixel-level segmentation of regions of interest in the image and is especially suitable for image segmentation tasks. Each leaf image was carefully labeled by drawing polygonal boundaries around infected and healthy regions. These annotations serve as the ground truth masks for supervised training, enabling the deep learning model to distinguish between different types of leaf conditions during training. Manual annotation, though time-consuming, ensures high-quality, accurate data labels, which are critical for semantic segmentation tasks.

The dataset included a total of 104 high-resolution images specific to grape leaves. These images covered four classes: three distinct types of grape leaf diseases and one class representing healthy leaf samples. The disease types included in the dataset exhibit visual differences in terms of color patterns, lesion shapes, and spread across the leaf surface. These visual variations play a key role in helping the deep learning model learn to distinguish between healthy and diseased areas. Ensuring all four classes were sufficiently represented in the dataset was crucial to developing a balanced and unbiased model.

One challenge encountered was the relatively small size of the dataset, as 104 images is considered a limited number for training deep neural networks. To address this issue and improve model generalization, we applied a set of data augmentation techniques. These augmentations were carefully chosen to simulate real-world variability while maintaining the semantic integrity of the original image content. The augmentation strategies applied included [1]:

1. Scaling: Images were resized to introduce size variation, which helps the model adapt to leaves appearing at different scales.
2. Rotation: Images were rotated at various angles, which is important because, in real field conditions, leaves are rarely aligned in the same orientation.
3. Zooming: Cropping and zooming in on different regions helped the model focus on localized infection patterns.
4. Brightness adjustment: Varying the brightness levels of images simulated different lighting conditions, making the model more robust to changes in illumination.

By augmenting the original dataset in these ways, we significantly increased the effective size and diversity of the training data. This not only improved the model's ability to generalize but also helped in reducing overfitting. These augmented images were used exclusively for training purposes and were not included in the validation or test sets to preserve the integrity of model evaluation [13].

Another critical consideration was ensuring class balance throughout the dataset. In many real-world agricultural datasets, certain disease types may be overrepresented while others are rare. This class imbalance can cause a model to perform well on majority classes but poorly on minority ones [14]. To mitigate this, we made sure that all four categories—three disease types and healthy leaves—were fairly represented across the training, validation, and test sets. This balance is particularly important in classification and segmentation tasks, where underrepresented classes can lead to skewed performance metrics.

The public availability of the Field-PV [8] dataset ensures that the results obtained using this data can be reproduced and benchmarked by other researchers in the field. By using an open dataset and clearly outlining our preprocessing steps, including annotation and augmentation, we support transparency and encourage further advancements in automated plant disease detection.

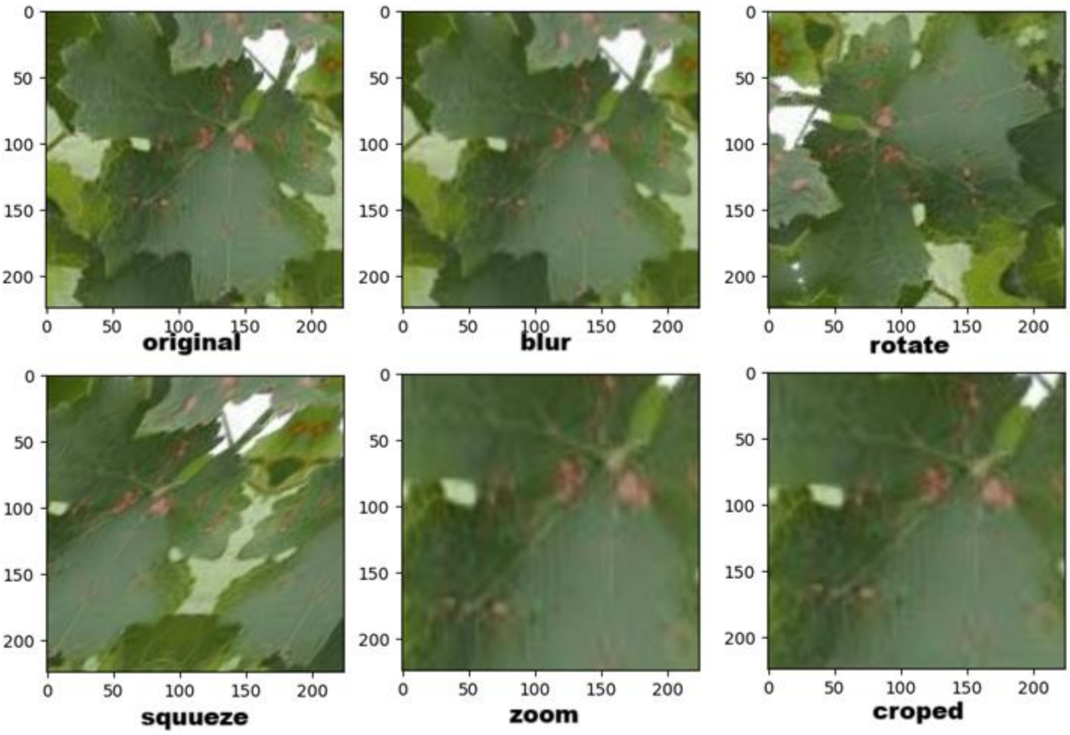


***Figure 3.1*** *Sample showing some of the data augmentation techniques we used in experiment.*

In summary, the Field-PV dataset served as a solid foundation for our grape leaf disease segmentation task. Through careful selection, manual annotation, and extensive data augmentation, we transformed a modest dataset into a robust training resource [15]. The inclusion of multiple disease classes and healthy samples ensured that the model could learn to recognize a wide range of visual patterns, ultimately enhancing its practical applicability in real-world scenarios such as precision agriculture and disease management.

## 3.2 Dataset Partitioning

Effective dataset partitioning is a critical step in developing robust deep learning models, especially for image-based plant disease detection systems. A well-structured split ensures that the model is exposed to diverse samples during training, can be reliably tuned using a separate subset, and ultimately evaluated on truly unseen data. In this study, we followed a commonly adopted partitioning strategy by dividing our dataset into three primary subsets: training, validation, and testing. This approach helps to reduce the risk of overfitting and ensures a fair assessment of the model's performance; perfect partition highly matters for better results [16]:

1. Training data: The largest portion of the dataset, comprising 70% of the original images (72 images), was allocated for training purposes. However, due to the limited number of original samples, we applied data augmentation techniques to artificially expand the training set. Through rotation, flipping, scaling, brightness adjustment, and translation transformations, the initial 72 training images were augmented to generate a total of 576 images. This augmentation process is crucial, particularly when working with relatively small datasets, as it helps the model generalize better by exposing it to varied versions of the same data.
   The training data serves as the foundation for model learning [16]. It is during this phase that the model adjusts its internal parameters to minimize prediction errors based on the input image-label pairs. Through iterative optimization using gradient descent, the network learns to distinguish between healthy leaf regions and those exhibiting disease symptoms. Augmentation not only increases data diversity but also acts as a regularization technique that prevents overfitting, ensuring that the model does not memorize specific patterns from the limited original dataset but instead learns to identify underlying features relevant across different images.

2. Validation data: To monitor the model's performance during the training process and fine-tune hyperparameters, we reserved 15% of the dataset (16 images) as the validation set. This subset was not included in the training process, ensuring that the model's performance on it provides a genuine measure of generalization to unseen data. During training, the validation set helps in early stopping decisions, hyperparameter tuning (such as learning rate, batch size, or number of layers), and in selecting the model checkpoint that generalizes best [17].
   A key role of the validation set is to prevent overfitting by acting as a feedback loop. For instance, if the training loss continues to decrease while the validation loss starts increasing, it indicates that the model is learning patterns that do not

generalize well to new data. Such insights allow us to intervene early—by adjusting training strategies or architecture—to maintain model performance across unseen inputs. Moreover, validation accuracy or Intersection over Union (IoU) scores serve as interim benchmarks to compare the effectiveness of different model variants or configurations.

3. Test data: The remaining 15% of the dataset (16 images) was allocated for final testing. This test set is entirely isolated from the training and validation phases and is only used after the model has been fully trained and fine-tuned. The purpose of this dataset is to provide an unbiased evaluation of the model's performance in real-world scenarios [17]. It acts as the ultimate benchmark for assessing segmentation accuracy, classification precision, and overall robustness.
   Performance metrics such as accuracy, precision, recall, F1-score, and Dice coefficient are computed on this test set to quantify how well the model performs on previously unseen data. A consistent performance across training, validation, and test sets is a strong indicator of a well-generalized model.

4. Balanced Representation: To ensure fair and reliable learning, care was taken to maintain a balanced distribution of all disease categories across the three subsets [14]. This is especially important in agricultural disease datasets, where class imbalance can significantly affect model performance. For example, if one disease type is overrepresented in the training set but underrepresented in the validation or test sets, the model may appear to perform well during training but fail to recognize underrepresented diseases in deployment. By evenly distributing images of different disease classes and healthy samples, the risk of such biases is minimized [15].
   Additionally, stratified sampling was considered during partitioning, which ensures that each subset contains a proportional representation of the various leaf disease conditions. This method increases the reliability of validation and test results, as each disease category receives sufficient representation in all phases of model development.

## 3.3 Image Segmentation

Our model is designed to segment the background, leaf and disease areas in the given image. For that we have proposed architecture that use Dense block [9] in CNN network and uses max unpooling for upsampling to get back the image. The purpose of max unpooling is to track back the mask using lossless manner. It saves the indices of maximum intensity pixel and uses that location to upsample more accurately.

Proposed architecture of the model:

1. Shrinking and expanding: For this we use encoder then decoder to first shrink then expand image back to desired mask. We have five layers to shrink the image and five layers to expand the image. At the shrinking phase we use pooling layer pf size 2 after each shrinking layer and we use max unpooling layer size 2 before each expanding layer, corresponding indices calculated at pooling phase. Each shrinking and expanding layer consists of a few Convo

layers of filter size 3, and stride 1, followed by batch norm and Relu layers. We have dense blocks after the first three expanding layers and before the last three shrinking layers. The first two shrinking layers and the last two expanding layers have two conv layers, and the remaining layers have three conv layers, based on VGG-16 model [9].

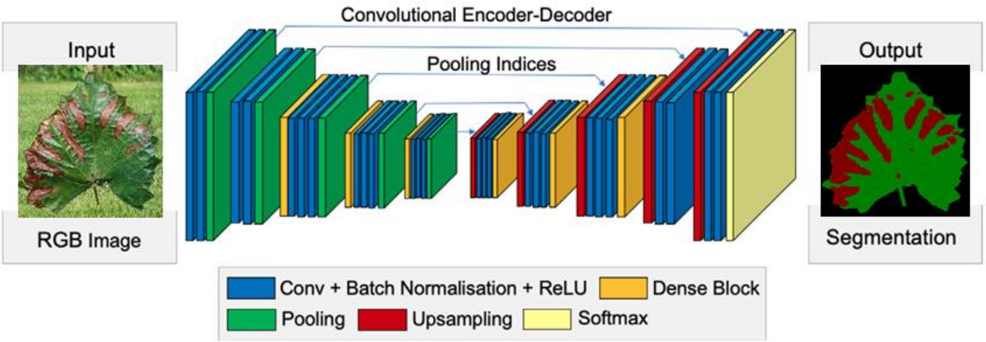


***Figure 3.2*** *Architecture of the network.*

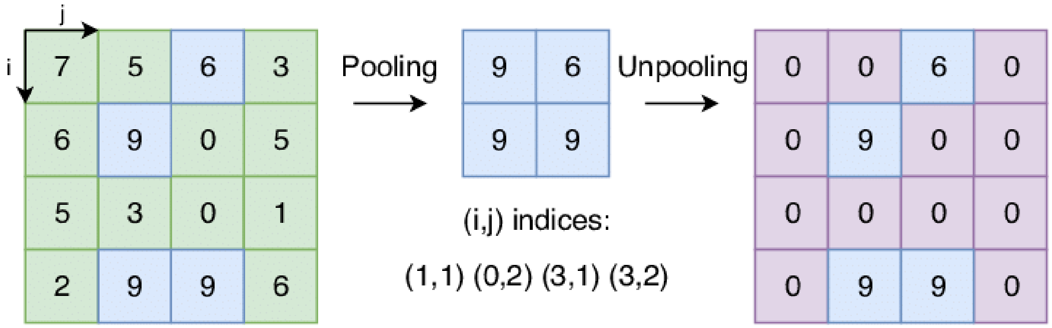


***Figure 3.3*** *Representing lossless preservation of max unpooling operation.*

2. Dense blocks: This are blocks that are fitted in both upsampling as well as downsampling phase. Its purpose is to reduce the exponential increment of the number of channels in successive layers. Also, it is much lossless. Here at each stage, we concatenate all the outputs of previous layers then pass through layers. So, the input size at each layer will be summed up of all the previous layers channel count. These blocks learn features so effectively with very few computational costs as number layer increases as per arithmetic addition [9]. It is also lossless even after making it so deep. We use a constant k, which is the number of filters at each layer of dense block. So, it tells the growth by which the number of input channels increases in each successive layer. Each layer has two conv layers, first is of kernel size one and the second is of kernel three.

The first kernel is to shrink the number of channels equal to k, then we actual use second filter to calculate the features.

So, we have a total of six blocks, three in the shirking phase a three in the expanding phase. The number of layers in each block is as follows, first: 12, second: 24, third: 16, fourth: 16, fifth: 8 and sixth: 4. And we keep k = 32.

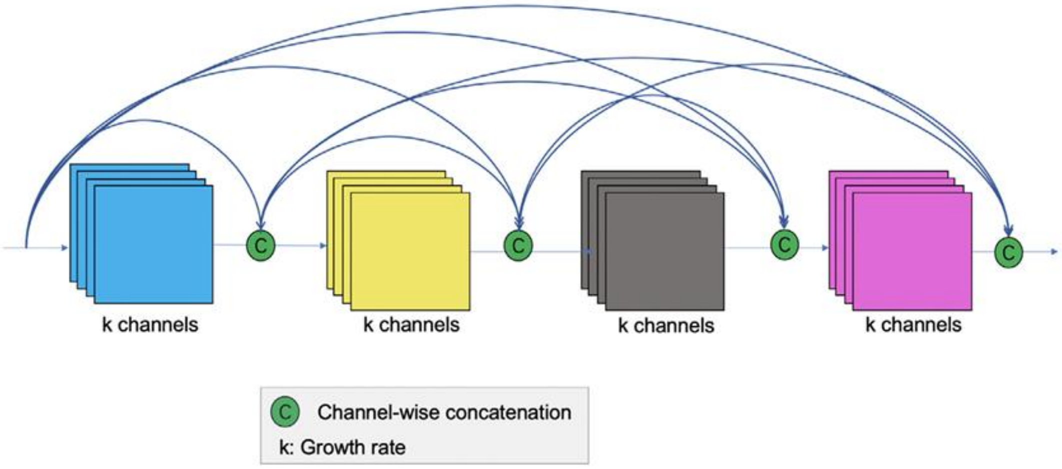


***Figure 3.4*** *Dense block represents connection of all the previous layers by each layer.*

3. Activation: We use SoftMax layer to calculate the category of each pixel. We have three different colors of output mask as background, leaf and infection. So, we must map the probability of color, so we have three categories for each pixel.

**Table 3.1** Table showing layers, input size and number of output channels at each layer.

| **Layers** | **Output size (height and width)** | **Number of channels** |
|---|---|---|
| Input | 224 | 3 |
| Shrinking layer 1 | 112 | 64 |
| Shrinking layer 2 | 56 | 128 |
| Dense layer1 | 56 | 512 |
| Shrinking layer 3 | 28 | 256 |
| Dense layer 2 | 28 | 1024 |

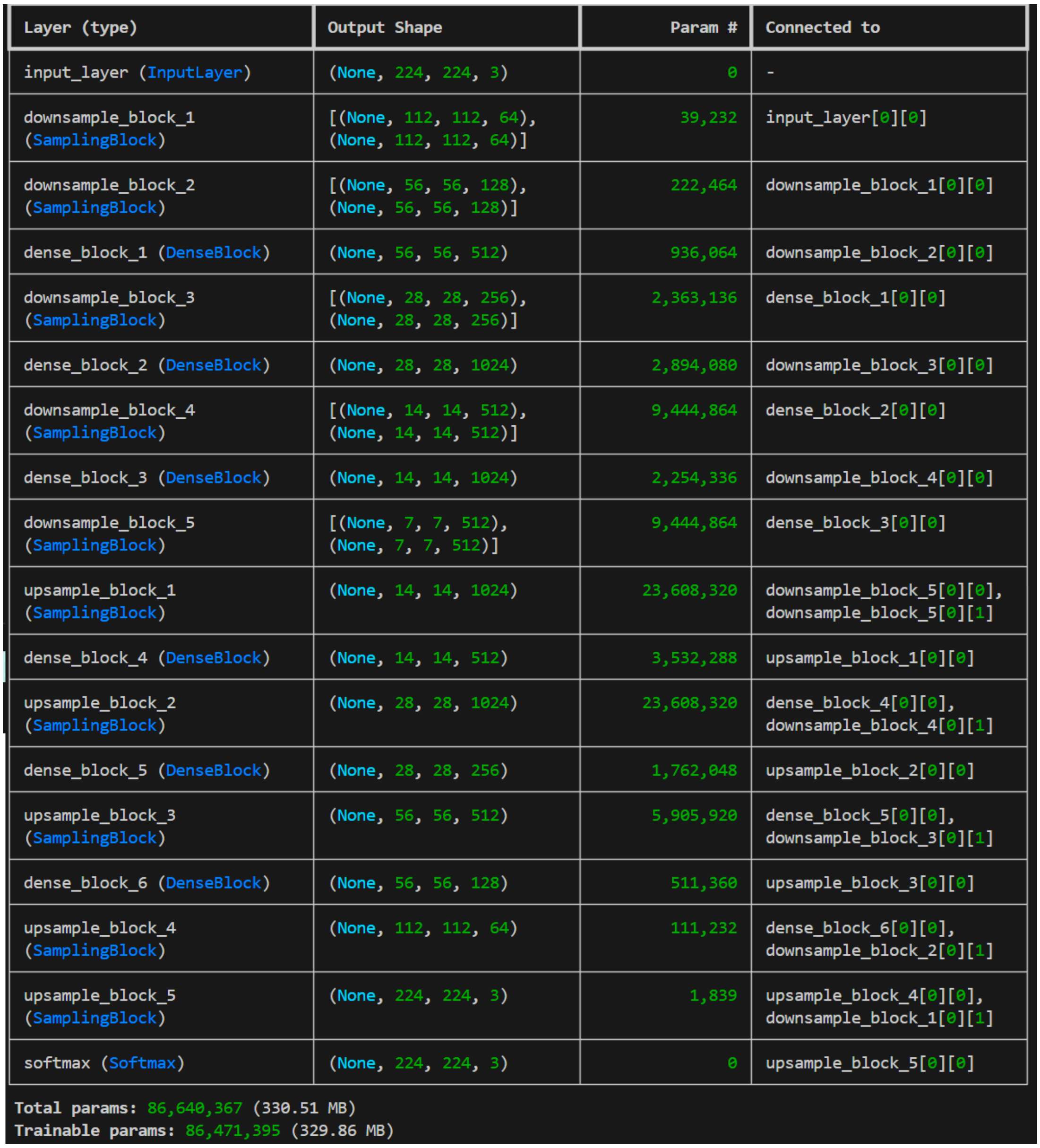
|  |  |  |
|---|---|---|
| Shrinking Layer 4 | 14 | 512 |
| Dense Layer 3 | 14 | 1024 |
| Shrinking Layer 5 | 7 | 512 |
| Expanding Layer 1 | 14 | 1024 |
| Dense Layer 4 | 14 | 512 |
| Expanding Layer 2 | 28 | 1024 |
| Dense Layer 5 | 28 | 256 |
| Expanding Layer 3 | 56 | 512 |
| Dense Layer 6 | 56 | 128 |
| Expanding Layer 4 | 112 | 64 |
| Expanding Layer 5 | 224 | 3 |
| Output layer | 224 | 3 |

4. Loss: Dimensions of output will be of size 4, batch size of output, channel size of mask, height of mask and width of mask. And channel size will be 3, height and width will be 224, output mask and input image will have same size, batch we kept 32. We use categorical cross entropy loss to train the model.

## 3.4 Train the model

Model training is most important part of ML; it helps to learn features from the training data [18]. First, we set random parameters in models. Then we must train the parameters as per the SL approach. For that we divide the data into the chunks of batches and pass complete batch through the model and take out the output for complete batch. Then we find the loss using the chosen or created loss function and we the calculate the gradient and propagate the loss backward through each layer and update the parameters using GDA.

After few epochs we stop the process, the new parameters we get we sat it is trained. When the new unseen data is passed it gives result with some accuracy.



```
┃ Layer (type)                ┃ Output Shape              ┃     Param # ┃ Connected to              ┃
│ input_layer (InputLayer)    │ (None, 224, 224, 3)       │           0 │ -                         │
│ downsample_block_1          │ [(None, 112, 112, 64),    │      39,232 │ input_layer[0][0]         │
│ (SamplingBlock)             │ (None, 112, 112, 64)]     │             │                           │
│ downsample_block_2          │ [(None, 56, 56, 128),     │     222,464 │ downsample_block_1[0][0]  │
│ (SamplingBlock)             │ (None, 56, 56, 128)]      │             │                           │
│ dense_block_1 (DenseBlock)  │ (None, 56, 56, 512)       │     936,064 │ downsample_block_2[0][0]  │
│ downsample_block_3          │ [(None, 28, 28, 256),     │   2,363,136 │ dense_block_1[0][0]       │
│ (SamplingBlock)             │ (None, 28, 28, 256)]      │             │                           │
│ dense_block_2 (DenseBlock)  │ (None, 28, 28, 1024)      │   2,894,080 │ downsample_block_3[0][0]  │
│ downsample_block_4          │ [(None, 14, 14, 512),     │   9,444,864 │ dense_block_2[0][0]       │
│ (SamplingBlock)             │ (None, 14, 14, 512)]      │             │                           │
│ dense_block_3 (DenseBlock)  │ (None, 14, 14, 1024)      │   2,254,336 │ downsample_block_4[0][0]  │
│ downsample_block_5          │ [(None, 7, 7, 512),       │   9,444,864 │ dense_block_3[0][0]       │
│ (SamplingBlock)             │ (None, 7, 7, 512)]        │             │                           │
│ upsample_block_1            │ (None, 14, 14, 1024)      │  23,608,320 │ downsample_block_5[0][0], │
│ (SamplingBlock)             │                           │             │ downsample_block_5[0][1]  │
│ dense_block_4 (DenseBlock)  │ (None, 14, 14, 512)       │   3,532,288 │ upsample_block_1[0][0]    │
│ upsample_block_2            │ (None, 28, 28, 1024)      │  23,608,320 │ dense_block_4[0][0],      │
│ (SamplingBlock)             │                           │             │ downsample_block_4[0][1]  │
│ dense_block_5 (DenseBlock)  │ (None, 28, 28, 256)       │   1,762,048 │ upsample_block_2[0][0]    │
│ upsample_block_3            │ (None, 56, 56, 512)       │   5,905,920 │ dense_block_5[0][0],      │
│ (SamplingBlock)             │                           │             │ downsample_block_3[0][1]  │
│ dense_block_6 (DenseBlock)  │ (None, 56, 56, 128)       │     511,360 │ upsample_block_3[0][0]    │
│ upsample_block_4            │ (None, 112, 112, 64)      │     111,232 │ dense_block_6[0][0],      │
│ (SamplingBlock)             │                           │             │ downsample_block_2[0][1]  │
│ upsample_block_5            │ (None, 224, 224, 3)       │       1,839 │ upsample_block_4[0][0],   │
│ (SamplingBlock)             │                           │             │ downsample_block_1[0][1]  │
│ softmax (Softmax)           │ (None, 224, 224, 3)       │           0 │ upsample_block_5[0][0]    │
 Total params: 86,640,367 (330.51 MB)
 Trainable params: 86,471,395 (329.86 MB)
```

***Figure 3.5*** *Total trainable parameters at each layer after compiling the model.*

# CHAPTER 4

# EXPERIMANTATION

## 4.1 Dataset Preparation

The dataset we used was publicly available [9] but not annotated, We, made the annotation in json format using label me and converted it into mask using python script. The name of the dataset is Field plant village, from that we have taken images of grape leaf that include four types of classification based on the name of disease. They are Black rot, Black Measles, Leaf Blight and healthy leaf images. We have 104 images in total. 70%, that is 72 images were used for training purposes. To increase the size of data we augmented each image with 8 variations, and we got 576 images in total.

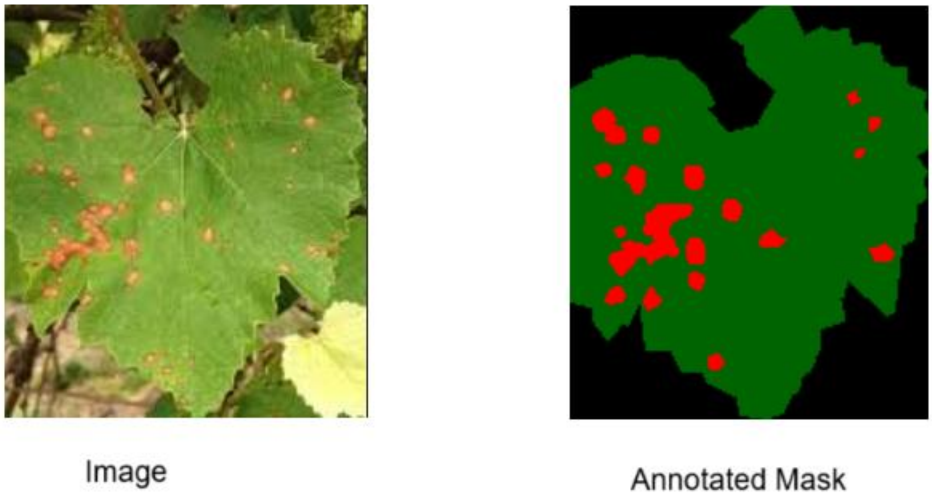


***Figure 4.1*** *Represents the generation of mask from the image.*

The size of this dataset is small and, to make it effective we need to increase this, making diverse images also helps better generalization. The color of mask we chose is black for background, green for leaves and red for infected areas. Images are of random size we converted it into height and width both of 224.

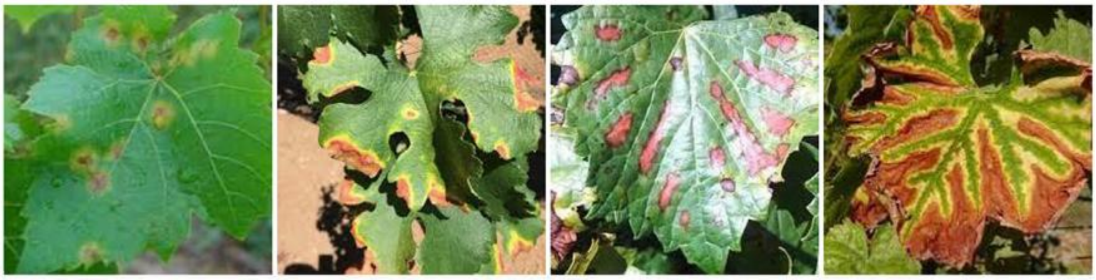


***Figure 4.2*** *Dataset is diverse has leaves with different background, lighting conditions and difference in quantity of infection spread.*

## 4.2 Implementation

Implementation process consists of following steps:

1. Preprocessing: The first step in our pipeline involved preparing the raw image data for use in a deep learning environment. All input images and their corresponding ground-truth segmentation masks were initially in standard image formats. To feed these into the neural network, we converted both the images and the masks into tensors, which are the fundamental data structures used by deep learning frameworks like TensorFlow.
   Next, we applied a series of data preprocessing and augmentation techniques [1] to enhance the quality and diversity of the dataset. This step is crucial, especially when working with a relatively small number of images. Augmentation techniques such as rotation, scaling, horizontal and vertical flipping, brightness adjustment, and zooming were applied to artificially increase the dataset size [1]. These augmentations not only helped introduce variations in the dataset, simulating different real-world conditions, but also improved the model's ability to generalize to unseen data.
   After preprocessing and augmentation, the dataset was divided into three distinct subsets: training, validation, and testing. The training set, which constituted most of the data, was used to fit the model. The validation set allowed fine-tuning hyperparameters and monitoring overfitting, while the test set was held back to evaluate the model's final performance [16]. Each subset was carefully constructed to ensure balanced representation across different grape leaf disease classes and healthy samples.

2. Model Training: Once the data was preprocessed, we initiated the training phase of our deep learning model. The prepared tensors were fed into a custom-built segmentation model designed for identifying diseased and healthy areas in grape leaves. The model was trained over a total of 50 epochs, which is a reasonable training duration that allows for sufficient learning without overfitting.
   We used a batch size of 32, meaning that during each iteration of training, the model processed 32 image-mask pairs at a time. This batch size provided a good balance between computational efficiency and memory constraints given our hardware limitations. For optimization, we employed the Adam optimizer, a widely used gradient descent variant known for its ability to adaptively adjust learning rates based on the complexity of the loss surface.
   The learning rate itself was not fixed but varied dynamically during training. This variable learning rate strategy helps the model converge more efficiently [19]. Typically, we start with a relatively higher learning rate to make quick progress during the early epoch and then gradually decrease it to fine-tune the weights with greater precision in later stages. This adaptive adjustment ensures that the model does not get stuck in suboptimal solutions and can continue improving steadily.

3. Hardware and software setup: Our experiments were conducted on a personal computing system equipped with an NVIDIA GTX 1650 GPU, which offers 4

GB of dedicated graphics memory. Although this GPU is relatively modest compared to high-end systems used in research institutions, it provided sufficient computational power to train our model within a reasonable time frame, especially given the augmented dataset and optimized training parameters.

In terms of software, the entire implementation was carried out using the Python programming language, which is widely adopted for machine learning and deep learning tasks due to its simplicity and extensive ecosystem of libraries. For the deep learning components of our project, we used TensorFlow, a powerful and flexible framework that supports the construction and training of neural networks with high efficiency.

Additional preprocessing and data analysis tasks were supported by other Python libraries. We used LabelMe, an open-source annotation tool, to manually label the disease-affected regions in grape leaf images. These annotations formed the ground-truth masks used during supervised training. For handling and organizing tabular data, we utilized the Pandas library, which offers convenient data manipulation functions and is well-suited for tasks such as class distribution analysis and data integrity checks.

## 4.3 Evaluation Metrics

We used following metrics for evaluation:

1. IoU: it stands for intersection over union to measure the performance of segmentation task. It tells how much area is common between original and predicted area over the total area covered by union of original and predicted area [9]. It is calculated by below formula.

$$\text{IoU} = \frac{\text{Ground Truth} \bigcap \text{Prediction}}{\text{Ground Truth} \bigcup \text{Prediction}} \quad (1)$$

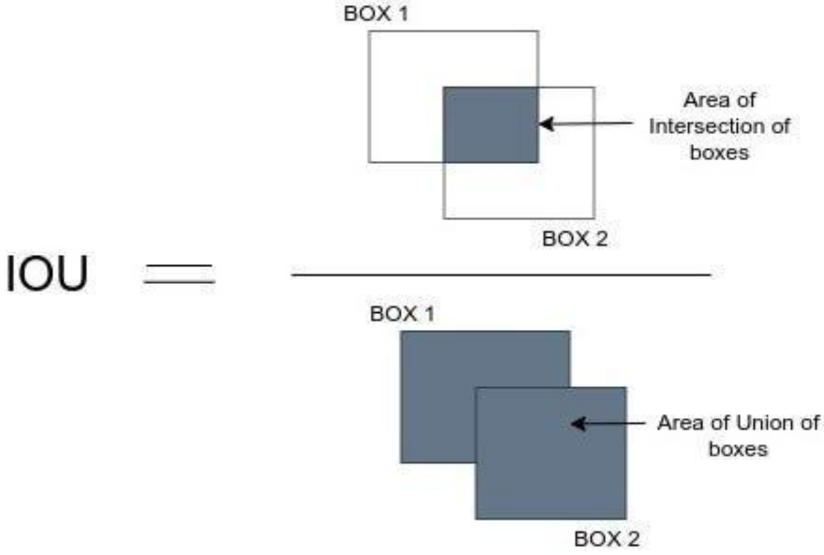


***Figure 4.3*** *Visualization of intersection over Union operation.*

2. Precision: It is used to calculate the ratio of the positive samples that are correctly predicted with the samples that are predicted as positive. Here we calculate average precision of per-class [9]. It is calculate using formula:

$$\text{Precision}_M = \frac{1}{l}\sum_{i=1}^{l}\text{Precision}_i \qquad (2)$$

Where $l$ is number of classes, here we have three class background, leaf and infection.

## 4.4 Result

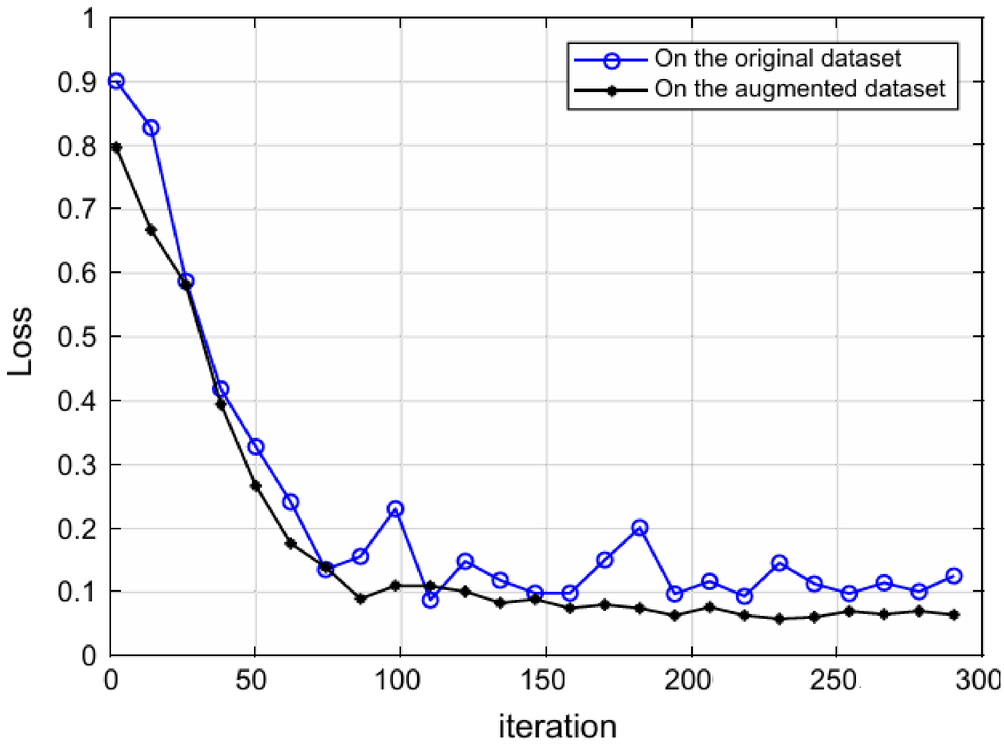


***Figure 4.4*** *Represents the decrease in loss with epochs in our model during the training phase.*

Irrespective of very less size of dataset, model has good performance. Our model gives IoU of 82.67 and Precision of 78.23. It is averaged when compared to relevant models in competition.

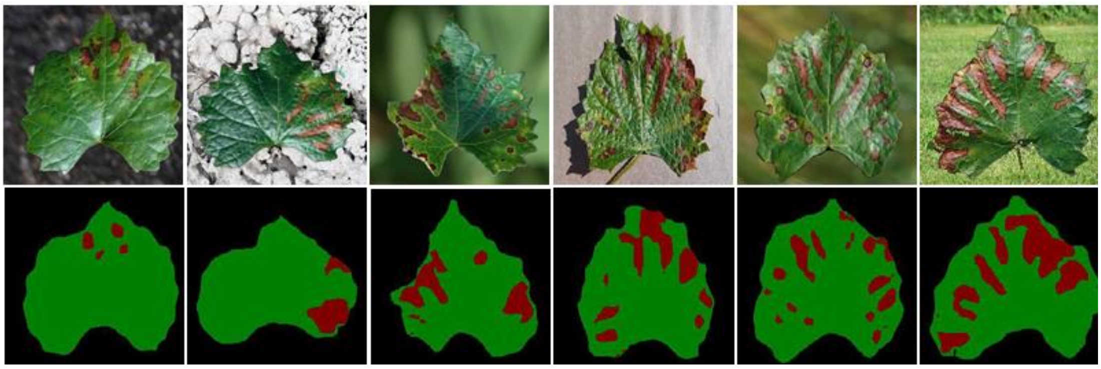


***Figure 4.5*** *Shows the output of model on sample of test data after training the model.*

# CHAPTER 5

# CONCLUSION AND FUTURE SCOPE

## 5.1 Conclusion

Increasing layers in arithmetically instead of exponentially reduces a lot of computational complexity [9]. Also, it leads to deeper network without losing and of the previous information at each stage of network. It can remove all the constraints of the data being so lengthy, unclear, because as the depth starts, it means increasing performance is at the next level. Usually, models forget all the previous learnings of previous layers since the more we propagate backward we realize that there is less impactful gradient that we are getting updated, and model and no good learning. The model seeks to suffer improper loss propagation at initial layers. We know how gradient starts to explode or vanish [20]. What difference does it make? As we continue the concatenation of all the previous input layers then in convo operation it simply has access to all the layers and its learning means a last layer of the block has access to its first layer of block also, to its learnings, all the previous blocks help learning the next layers not just the one of the previous subsequent layers [21]. In this way information is propagated a lot easily and learned easily without gradient exploding or vanishing.

We can easily conclude that by very few parameters it easily catches all the features there is no failure to any important feature coming from corresponding layer. When the convo layer takes the input and concatenates all the inputs into one, then while passing the filters it combines all channels into one, and the number of output channels depends on the number of output channels it has [22]. We can see all the information is merged into just one channel like, each of the single channel has all the information, all the learning considering the learnings of all the channels. This is how information is preserved, and we gain control over the loss of information.

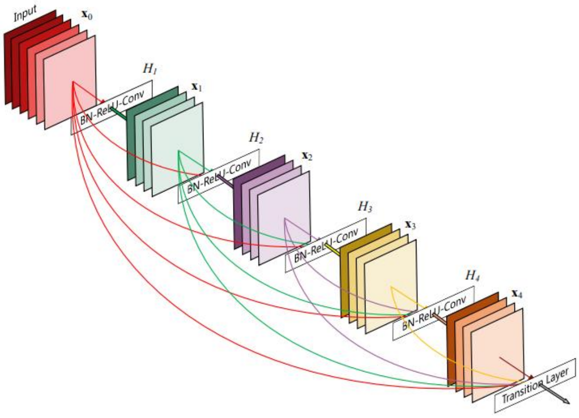


***Figure 5.1*** *Represents the arithmetic increment of the number of channels in dense blocks.*

We had a shrinking phase that successfully shrinks the data by downscaling operation, for that we use max pooling, and we also save its indices for future reference, for up sample operation [23]. Max Pooling is best because it catches high intensity pixel, the pixel that has more value than neighboring pixel, leads to finalizing strong pixels. In the shrinking phase models easily start getting compact, feature vector starts decreasing in height and width but increase in terms of channels [23], this what we call learning, considering that we are now having feature understanding of all features in data.

But we are doing segmentation, Compressed data is not what we finally need, we need complete mask that has size equivalent to original image. So, we must expand it back for that max un-pooling operation we used has high impact, because we have access over the location of high valued pixel and now, we are expanding based on that. This also leads to some degree of lossless approach, and we get to maintain the integrity of output and performance of model [23].

We successfully were able to make our model very light weight, efficient with capturing features, highly lossless, learning preserving, and effective training with no redundant layers.

Limitations: As the data is available publicly but not the annotations, we have taken and experimented on a very small dataset to avoid huge manual and computational costs. But we can expand the dataset and see how it behaves after sufficient training size, and how easily it catches to parameters and learning the sample.

## 5.2 Future Work

CNN are not that good in terms of capturing global dependencies [24], recent advancement in VIT has opened new dimension of image learning. Where we divide image into patches and use variations of MHA to calculate the relationship between. We can use combination of dense blocks and VIT to increase the performance by new type of learning. MHA has ability to known how much given part of image is connected to remaining part of the image [24]. It is calculated by below equation.

$$\text{Attention}^{R}\left(Q_i, K_i^R, V_i^R\right) = \text{Softmax}\left(\frac{Q_i K_i^{RT}}{\sqrt{d_{\text{head}}}}\right) V_i^R \tag{3}$$

Where Q is a query, K is a key, V is a value, *i* is the attention head, T refers to transpose, *d* is dimension of each head. We first calculate attention of each head individual then concatenate all to form MHA [27]. It can be calculated by the following equation:

$$\text{MultiHead}^{\mathbf{R}}\left(Q, K, V, X^r\right) = \text{Concat}\left(head_0, head_1, \ldots, head_{N_h}\right) W^O \tag{4}$$

MHA is just the combination attention of all the heads, then we can multiply with total weight.

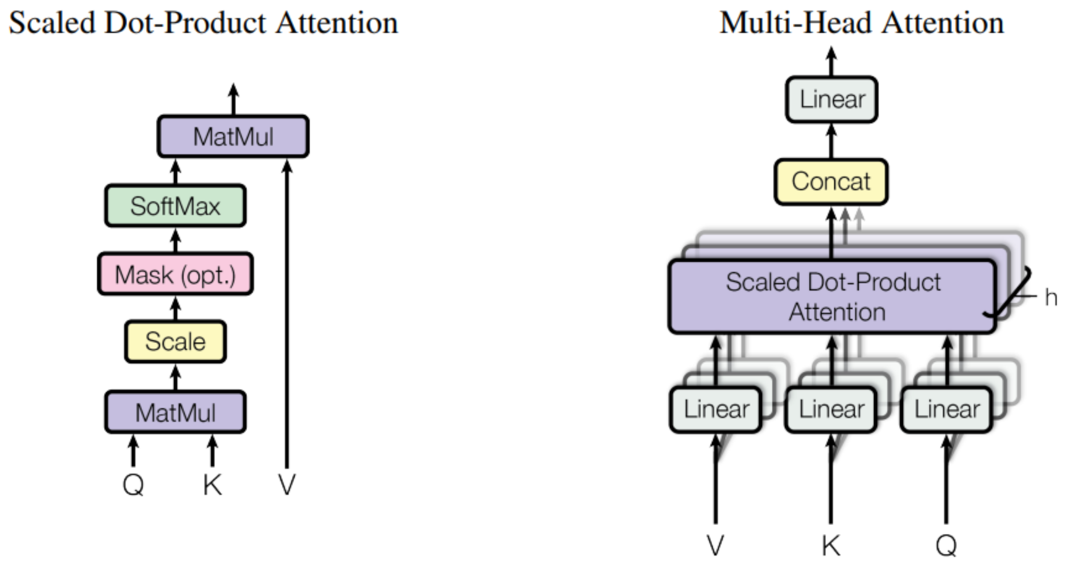


***Figure 5.2*** *Explains the architecture of MHA.*

Initially K, Q and V are the same input matrix, we can perform operations to compress them for increasing efficiency [25], like pooling or convolution operation [26]. Or we can also do reversible compressions [8, 26] before calculation of MHA and then reverse it back to remove loss. When we multiply Q and K we get a square matrix that has the highest values in diagonal elements, it gives the result that when a patch is multiplied with another patch what will the result be? Then that value gives us the hint of how much both the patches are interrelated. The complete matrix saves information on each pair of patches. It is called self-attention, we then divide by dimensional size of head to rescale the new tensor.

After taking SoftMax of that tensor we can scale it between zero and one, this brings it into probabilistic terms. Then multiplying it with value is actual attention score. It is a very basic idea of any transformer architecture [24].

Recent developments in VIT have opened the doors to many new possibilities, many attempts are being made to combine abilities of VIT and CNN. VIT is good for capturing global dependencies in complete image whereas CNN are good in capturing local dependencies within nearby pixels [24], this tradeoff is much intense and so the huge research is made to make a hybrid approach to that can capture abilities of both CNN and VIT [7].

Our model has Dense blocks for reducing sudden increment in channels, this ability of dense channels is very useful. We can use it to capture local features properly without much computation [22]. With each dense block we will use MHA in parallel and concatenate both dense and MHA (or any improved version of MHA) into one and then pass to shrinking or expanding layer.

How will it help? The advantage of dense block is to get local features [22] and MHA on other hand takes global features [24]. When combination of both is passed into next layer (shrinking or expanding) it will capture both the dependencies, far and near.

Usually, the leaf size is big, and infected regions can be small as well as large based on type of disease or quantity of disease spread [8]. This variable is needed for any generalization to happen. Our model must be sharp enough to handle both together at the same time is most important to increase performance to every type of image and disease. Scope of VIT is limited when we start caring for local properties [24], we are able reduce the size of patches, but it gives huge computational overhead that increase training time multiple times. Usually in modern approaches the size of patches is kept small in initial stages of encoder, and it is increased in next layers this helps to make pyramidal architecture like CNN [25], and the depth of tensors is increased.

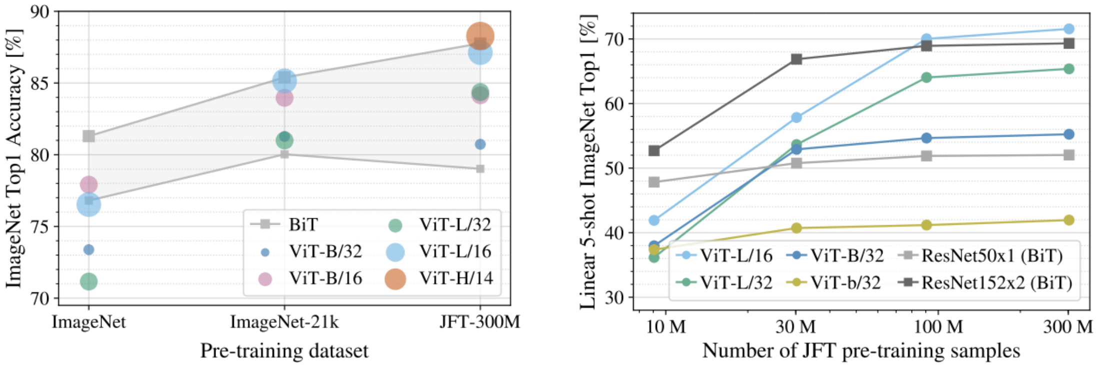


***Figure 5.3*** *Shows the behaviors of VIT. Image taken from the original paper of VIT. CNN and VIT work differently, capture different features (local can global), has their own way of achieving accuracy.*

We try to make VIT look like CNN, but VIT is VIT and CNN is CNN, just trying to make it look like each other is not sufficient. We must merge its abilities into one model to work in parallel and combinate the result based on input or forget to get parameters like LSTM [28] to manipulate which among MHA and Dense block which part must be forgotten and which to be inputted for next layers this makes model like to be more generalized.

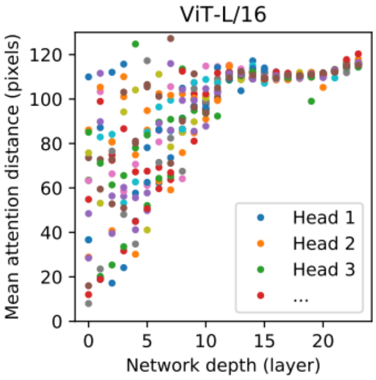


***Figure 5.4*** *Image taken from the original paper of VIT. Compared to CNN, VIT works poorly on smaller datasets as the dataset size increases VIT starts overpowering CNN.*

## 5.3 Social impact

Agriculture stands as one of the most vital pillars of human civilization. As the foundational sector of the economy, it supports not only the sustenance of life but also fuels the operations of industries and markets. Without a thriving agricultural base, the continuity of food production, economic stability, and social development would face significant threats [29]. In recent decades, especially with the rapid growth in global population, the demand for higher agricultural output has become more urgent than ever. This increase in demand must be met while also managing the challenges posed by plant diseases and minimizing the overall cost of farming.

To address these modern-day challenges, Artificial Intelligence (AI) has emerged as a transformative tool in the domain of smart farming. AI-based technologies are being used to automate various aspects of agriculture, including the early detection of plant diseases [29]. Traditionally, farmers relied heavily on manual observation and experience to identify symptoms of disease on crops, which often resulted in delayed detection and suboptimal treatment. However, with the integration of AI-powered systems, farmers can now utilize mobile devices, drones, or smart cameras equipped with disease detection models that rapidly analyse plant conditions and offer precise feedback [30].

One of the most critical components in this AI-driven workflow is image segmentation. Segmentation allows for the accurate delineation of diseased areas from healthy regions on plant leaves. This step is not only essential for visual understanding but also plays a central role in quantifying the severity of infections. With accurate segmentation, farmers can measure the exact portion of the crop that is infected, which in turn allows for targeted application of pesticides or treatment. This reduces wastage, ensures appropriate dosing, and minimizes environmental impact [1].

Moreover, segmentation serves as a fundamental stage in the data preprocessing pipeline of deep learning models. The quality of segmentation directly influences the overall performance of AI models in disease prediction [1]. When the infected areas are clearly and precisely segmented, the model can learn better features, resulting in higher accuracy during classification and prediction phases. This means fewer false diagnoses and more reliable decision support for farmers.

The social benefits of such technological integration are far-reaching [30]. First, it empowers farmers with actionable insights, reducing dependency on external agricultural experts. Second, it promotes sustainable farming practices by optimizing resource usage. Third, it contributes to food security by helping prevent large-scale crop failures. And lastly, it enhances rural development by introducing modern technological practices into traditionally manual sectors.

# REFERENCES

[1] Haridasan, A., Thomas, J. & Raj, E.D. Deep learning system for paddy plant disease detection and classification. *Environ Monit Assess* **195**, 120 (2023). https://doi.org/10.1007/s10661-022-10656-x.

[2] Xinxin Zhang, Chaojun Cen, Fei Li, Meng Liu, Weisong Mu, CRFormer: Cross Resolution Transformer for segmentation of grape leaf diseases with context mining, Expert Systems with Applications, Volume 229, Part A, 2023, 120324, ISSN 0957 4174, https://doi.org/10.1016/j.eswa.2023.120324.

[3] Kahkashan Perveen, Sandip Debnath, Brijesh Pandey, Sumanta Prasad Chand, Najat A. Bukhari, Pradipta Bhowmick, Najla A. Alshaikh, Shaista Arzoo, Shanzeh Batool, Deep learning-based multiscale CNN-based U network model for leaf disease diagnosis and segmentation of lesions in tomato, Physiological and Molecular Plant Pathology, Volume 128, 2023, 102148, ISSN 0885-5765, https://doi.org/10.1016/j.pmpp.2023.102148.

[4] Olfa Mzoughi, Itheri Yahiaoui, Deep learning-based segmentation for disease iden tification, Ecological Informatics, Volume 75, 2023, 102000, ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2023.102000.

[5] Rahman, S.U., Alam, F., Ahmad, N. et al. Image processing based system for the detection, identification and treatment of tomato leaf diseases, Multimed Tools Appl 82, 9431–9445 (2023). https://doi.org/10.1007/s11042-022-13715-0.

[6] Shanwen Zhang, Chuanlei Zhang, Modified U-Net for plant diseased leaf image seg mentation, Computers and Electronics in Agriculture, Volume 204, 2023, 107511, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2022.107511.

[7] Manjunatha Badiger, Jose Alex Mathew, Tomato plant leaf disease segmentation and multiclass disease detection using hybrid optimization enabled deep learn ing, Journal of Biotechnology, Volume 374, 2023, Pages 101-113, ISSN 0168-1656, https://doi.org/10.1016/j.jbiotec.2023.07.011.

[8] Xinxin Zhang, Fei Li, Haibin Jin, Weisong Mu, Local Reversible Transformer for semantic segmentation of grape leaf diseases, Applied Soft Computing, Volume 143, 2023, 110392, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2023.110392.

[9] Jingxuan Su, Sean Anderson, Mahed Javed, Charoenchai Khompatraporn, Apinanthana Udomsakdigool, Lyudmila Mihaylova, Plant leaf deep semantic segmentation and a novel benchmark dataset for morning glory plant harvesting, Neurocomputing, Volume 555, 2023, 126609, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2023.126609.

[10] Jiangwen Lu, Bibo Lu, Wanli Ma, Yang Sun, EAIS-Former: An efficient and accurate image segmentation method for fruit leaf diseases, Computers and Electronics in Agriculture, Volume 218, 2024, 108739, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2024.108739.

[11] Xinxin Zhang, Zibo Feng, Weisong Mu, Reformer: Re-parameterized kernel lightweight transformer for grape disease segmentation, Expert Systems with Applications, Volume 265, 2025, 125757, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2024.125757.

[12] Jared Cervantes Canales, Jair Cervantes Canales, Farid Garc´ıa-Lamont, Arturo Yee-Rendon, Jos´e Sergio Ruiz Castilla, Lisbeth Rodriguez Mazahua, Optimal seg mentation of image datasets by genetic algorithms using color spaces, Expert Systems with Applications, Volume 238, Part D, 2024, 121950, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023.121950.

[13] Xinxin Zhang, Fei Li, Haiying Zheng, Weisong Mu, UPFormer: U-sharped Perception lightweight Transformer for segmentation of field grape leaf diseases, Expert Systems with Applications, Volume 249, Part A, 2024, 123546, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2024.123546.

[14] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. ACM Comput. Surv. 54, 6, Article 115 (July 2021), 35 pages. https://doi.org/10.1145/345760.

[15] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023. Trustworthy AI: From Principles to Practices. ACM Comput. Surv. 55, 9, Article 177 (January 2023), 46 pages. https://doi.org/10.1145/3555803.

[16] K. E. Bennin, J. Keung, A. Monden, Y. Kamei and N. Ubayashi, "Investigating the Effects of Balanced Training and Testing Datasets on Effort-Aware Fault Prediction Models," 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, 2016, pp. 154-163, doi: 10.1109/COMPSAC.2016.144.

[17] Lu, J.; Tan, L.; Jiang, H. Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. Agriculture 2021, 11, 707. https://doi.org/10.3390/agriculture11080707.

[18] Anna M Derossis, Gerald M Fried, Michal Abrahamowicz, Harvey H Sigman, Jeffrey S Barkun, Jonathan L Meakins, Development of a Model for Training and Evaluation of Laparoscopic Skills 11This work was supported by an educational grant from nited States Surgical Corporation (Auto Suture Canada)., The American Journal of Surgery, Volume 175, Issue 6, 1998, Pages 482-487, ISSN 0002-9610, https://doi.org/10.1016/S0002-9610(98)00080-4.

[19] Giorgia Franchini, Federica Porta, Valeria Ruggiero, Ilaria Trombini, Luca Zanni, A stochastic gradient method with variance control and variable learning rate for Deep Learning, Journal of Computational and Applied Mathematics, Volume 451, 2024, 116083, ISSN 0377-0427, https://doi.org/10.1016/j.cam.2024.116083.

[20] S. Al-Abri, T. X. Lin, M. Tao and F. Zhang, "A Derivative-Free Optimization Method With Application to Functions With Exploding and Vanishing Gradients," in IEEE Control Systems Letters, vol. 5, no. 2, pp. 587-592, April 2021, doi: 10.1109/LCSYS.2020.3004747.

[21] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[22] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[23] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[24] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[25] Wang, Wenhai, et al. "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[26] Yao, Ting, et al. "Wave-vit: Unifying wavelet and transformers for visual representation learning." European conference on computer vision. Cham: Springer Nature Switzerland, 2022.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[28] Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks." arXiv preprint arXiv:1909.09586 (2019).

[29] F. X. Sikounmo, C. Deffo and C. T. Djamegni, "Social and Environmental Impact of a Plant Disease Analysis Method Based on Object Extraction," in IEEE Technology and Society Magazine, vol. 43, no. 3, pp. 65-71, Sept. 2024, doi: 10.1109/MTS.2024.3455110.

[30] Elsayed Said Mohamed, AA. Belal, Sameh Kotb Abd-Elmabod, Mohammed A El-Shirbeny, A. Gad, Mohamed B Zahran, Smart farming for improving agricultural management, The Egyptian Journal of Remote Sensing and Space Science, Volume 24, Issue 3, Part 2, 2021, Pages 971-981, ISSN 1110-9823, https://doi.org/10.1016/j.ejrs.2021.08.007.

**DELHI TECHNOLOGICAL UNIVERSITY**
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bhawana Road, Delhi-42

# PLAGIARISM VERIFICATION

Title of the Thesis **Precise Identification of Diseased Leaf Regions Using Deep Learning-Based Semantic Segmentation Techniques**

Total Pages **32** Name of Scholar **Anurag Nandaghale**

Supervisor **Rahul Katarya**

Department **Computer Science and Engineering**

This is to report that the above thesis was scanned for similarity detection. Process and outcome are given below:

Software used: Turnitin Similarity Index: **3%** Total word count: **9,518**

Date:

**Candidate's Signature** **Signature of Supervisor**

# Anurag Thesis.pdf

Delhi Technological University

## Document Details

Submission ID

trn:oid:::27535:98773853

Submission Date

Jun 1, 2025, 12:38 PM GMT+5:30

Download Date

Jun 1, 2025, 12:41 PM GMT+5:30

File Name

Anurag Thesis.pdf

File Size

1.1 MB

32 Pages

9,518 Words

54,650 Characters

# 3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)

## Match Groups

**21** Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks

**0** Missing Quotations 0%
Matches that are still very similar to source material

**0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

## Top Sources

| | |
|---|---|
| 2% | Internet sources |
| 1% | Publications |
| 2% | Submitted works (Student Papers) |

## Integrity Flags

**0 Integrity Flags for Review**

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

**21** Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks

**0** Missing Quotations 0%
Matches that are still very similar to source material

**0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

## Top Sources

| | |
|---|---|
| 2% | Internet sources |
| 1% | Publications |
| 2% | Submitted works (Student Papers) |

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | | |
|---|---|---|---|
| 1 | Internet | etheses.whiterose.ac.uk | <1% |
| 2 | Internet | ouci.dntb.gov.ua | <1% |
| 3 | Internet | dblp.dagstuhl.de | <1% |
| 4 | Publication | Iliass Zine-dine, Anass Fahfouh, Jamal Riffi, Khalid El Fazazy, Ismail El Batteoui, M... | <1% |
| 5 | Internet | koreascience.or.kr | <1% |
| 6 | Internet | metall-mater-eng.com | <1% |
| 7 | Submitted works | Liverpool John Moores University on 2023-12-17 | <1% |
| 8 | Internet | genesys-lab.org | <1% |
| 9 | Internet | www.davidinouye.com | <1% |
| 10 | Internet | www.pulsion.co.uk | <1% |

| | | | |
|---|---|---|---|
| 11 | Submitted works | Anna University on 2025-01-08 | <1% |
| 12 | Submitted works | Liverpool John Moores University on 2025-05-04 | <1% |
| 13 | Publication | Olfa Mzoughi, Itheri Yahiaoui. "Deep learning-based segmentation for disease id... | <1% |
| 14 | Submitted works | Liverpool John Moores University on 2023-10-07 | <1% |
| 15 | Publication | Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023 | <1% |
| 16 | Submitted works | University of Moratuwa on 2023-06-11 | <1% |
| 17 | Submitted works | University of Sydney on 2024-05-24 | <1% |
| 18 | Internet | assets.researchsquare.com | <1% |
| 19 | Internet | www.csauthors.net | <1% |
| 20 | Internet | www.nature.com | <1% |