

Deep Learning-Driven Citrus Disease Detection: A Novel Approach with DeepOverlay L-UNet and VGG-RefineNet

Deep Learning-Driven Citrus Disease Detection

P Dinesh, Ramanathan Lakshmanan*

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

Abstract—Agriculture is essential to the world's desire to produce food, generate income, and maintain livelihoods. Citrus fruits are produced worldwide and have a significant impact on food production, nutrition, and agriculture. During production, farmers face difficulties due to diseases that affect plant growth. Black spot, canker, and greening are some citrus leaf diseases that risk citrus production, resulting in economic losses as well as reduced supply stability. Early detection of these diseases through recent technologies like deep learning will help farmers with better yields and quality. The current methods fall short in marking the area affected by the disease with accuracy and more performance. This work has a novel method proposed for the segmentation and classification of citrus leaf diseases. The method consists of three phases. In the first phase, DeepOverlay L-UNet is used to segment the affected regions. In the second phase, disease detection is carried out using VGG-RefineNet, and in the third phase, the affected region is highlighted in the original image with a severity level. On the other hand, the DeepOverlay L-UNet model proves to be effective in detecting affected areas, thereby enabling clear visualization of the spread of the disease. The result affirms that the proposed method outperforms with a better training IOU of 0.9864 and a validation IOU of 0.9334.

Keywords—Citrus disease detection; highlighting affected region; Deep learning; semantic segmentation; DeepOverlay L-UNet; VGG-RefineNet

I. INTRODUCTION

Agriculture plays a major influence in the world economy since it is essential to supporting livelihoods, promoting economic expansion, and raising national GDPs [1]. Concerns regarding food shortages and rising demand arise as the global population is projected to surpass ten billion by 2060, underscoring the significance of agriculture in addressing these issues [2]. However, threats to crop production include diseases, pests, and long climate change, which have an impact on production yield and quality worldwide [3].

Citrus is one of the species of plants that are produced globally, with output reaching 157.98 million tons. It is an essential part of global agriculture and is utilized in various sectors, particularly the food and nutrition industries. Examples of these plants are lemons and oranges. Diseases that affect output and quality pose serious concerns for citrus crops. Citrus

production is at risk from diseases including blackspot, canker, and greening, which can result in financial losses and a less stable supply. Monitoring disease conditions through plant observation and the direct use of pesticides in agriculture in every adverse situation are two alternate approaches to disease protection. This approach is known globally and is simple for producers to use. This method's drawback is that certain producers unknowingly utilize chemical pesticides. The most frequent issue with medication usage that occurs unknowingly is the incorrect medicine used due to incorrect disease detection in plants. The welfare of people affected by unknowing drug usage. Citrus black spot, citrus cancer, and greening are the most prevalent illnesses in the citrus production field. These diseases are quite common in commercial citrus production [4]. These are the reasons why the disease wants to be detected immediately, and appropriate action should be taken. If not, it results in a loss of product quality and quantity.

Citrus greening is a highly destructive citrus disease worldwide. These diseases can affect any commercial citrus variety. Asian Citrus Psellid (ACP) (*Diaphorina citri*) is one of the diseases responsible for this illness. To stop greening from spreading further, trees impacted by the disease must be destroyed [5]. Worldwide, many commercial citrus cultivars, particularly grapefruit, sweet oranges, and lemons, are afflicted with citrus bacterial cancer. Humid-wet areas with extreme temperatures, precipitation, and wind are more conducive to the spread of this illness. This disease is characterized by early fruit and leaf loss, dark blotches on the leaves, and bubble-like diseases in different tree sections [6]. On the fruit and leaves of citrus trees, the citrus black spot typically takes the form of freckle marks. It can also be observed as lesions on the crop's branches. It is a condition that is more frequent in warm climates, like citrus cancer. *Phyllosticta citricarpa* is the fungus disease that causes Citrus black spot. These must degrade the yield and quality [7]. These diseases affect crop quality and yields. Sensible practices in agriculture, including spraying and new technologies, are being used to avoid plant diseases. Deep learning approaches are utilized to segment and categorize plant diseases, as evidenced by the literature study.

In agriculture, statistics on image data are essential for disease detection, image segmentation, and crop assessment on yields. Statistical data is used in agriculture to assess disease levels according to pigment factors such as hue, saturation, and

brightness [8]. Furthermore, statistical data on images is essential for recognizing how plant diseases affect crop yields, particularly in nations like India, where agriculture plays a major economic role [9]. For measuring and diagnosing diseases, image segmentation methods like genetic algorithms use statistical analysis to divide pictures into discrete sections [10]. Disease severity and disease grades in crops are estimated using statistical indicators such as Region of Interest (ROI) and percentage of Occurrence of Infection (POI) [11]. Farming operations may be enhanced with the use of this data, resulting in higher yields and more environmentally friendly farming methods.

Many methods were recently investigated in studies to identify plant diseases. Standard approaches include importing leaf images, segmenting the damaged region by pre-processing for noise reduction, utilizing algorithms for disease identification, and extracting features using methods like LBP and HoG [12]. To diagnose diseases based on observable symptoms on leaves, recent advances have focused on deep learning models, namely CNNs, that perform away with the need for manual feature description [2]. According to the category of disease and severity of damage, potential treatments are then suggested by applying deep learning techniques such as DenseNet for disease categorization and segmentation [9]. Furthermore, techniques like CAAR-UNet models use preprocessing, data preparation, and architectural improvements to identify and categorize sick areas in plant leaf images [13].

Segmentation in identifying diseases helps locate and characterize the boundaries of diseased regions that lie within images [14] [15]. It distinguishes between healthy and diseased areas, which assists in determining the severness and position of the diseases on the plant's leaf. By segmenting images, it is easy to identify particular disease signs, such as blackspots and canker, based on texture, colour, and shape-defining features. Segmentation techniques such as semantic segmentation and instance segmentation are used to properly identify and categorize various diseases in plants, hence improving disease detection accuracy. Overall, segmentation is a key phase in disease detection, allowing for focused examination and categorization of plant diseases.

Semantic segmentation is important for identifying plant diseases. Accurate diseased region segmentation is possible with refined deep learning models such as DenseNet and Hybrid-DSCNN. These models can identify damage to plant leaves at the pixel level, enabling accurate disease identification [16]. Semantic segmentation not only helps to categorize diseases and determine the stage of disease, but it also gives useful information for recommending appropriate treatments. Additionally, the use of weakly supervised learning approaches improves classifier performance by showing disease symptoms and infected regions, allowing for a better understanding of plant disease. Overall, semantic segmentation is a strong method for properly detecting and managing plant diseases.

Disease classification requires the use of advanced methods like Support vector Machines, Convolutional Neural Networks, and DenseNet for accurate categorization [17] [18]. These

methods use image analysis and the extraction of features to recognize and categorize different diseases in plants [19]. ShuffleNetV2, for example, is used to classify plant leaf diseases while maximizing the model's accuracy through parameter setups and feature selection. Furthermore, ML and DL algorithms are utilized to diagnose diseases, demonstrating the importance of these classification methods. By using these tools, researchers can improve agricultural disease detection and treatment processes.

This research effort uses deep learning technology to recognize and categorize citrus diseases on leaves in their early stages. A discussion on segmentation and classification has been concluded after a literature review. The segmentation phase receives most of the attention. The whole focus is concentrated on the diseased region. There is also a need for developing and utilizing a broader plant segmentation approach that may be applied in both regulated and natural circumstances. Using pre-trained deep learning models, some authors have analysed the classification of citrus diseases. Following the initial step in image processing, which involved preprocessing the data set, the authors have used classification models to identify disease areas. Till now, disease borders and classes have not been accurately highlighted by the segmentation and classification models. This work proposed a novel DeepOverlay L-UNet approach for highlighting the affected region with precise boundaries based on severity and improved intersection over union. The disease classifications are recognized by VGG-RefineNet. The proposed method, DeepOverlay L-UNet, uses semantic segmentation to divide the diseased area from the leaf, and the flow is shown in Fig. 1.

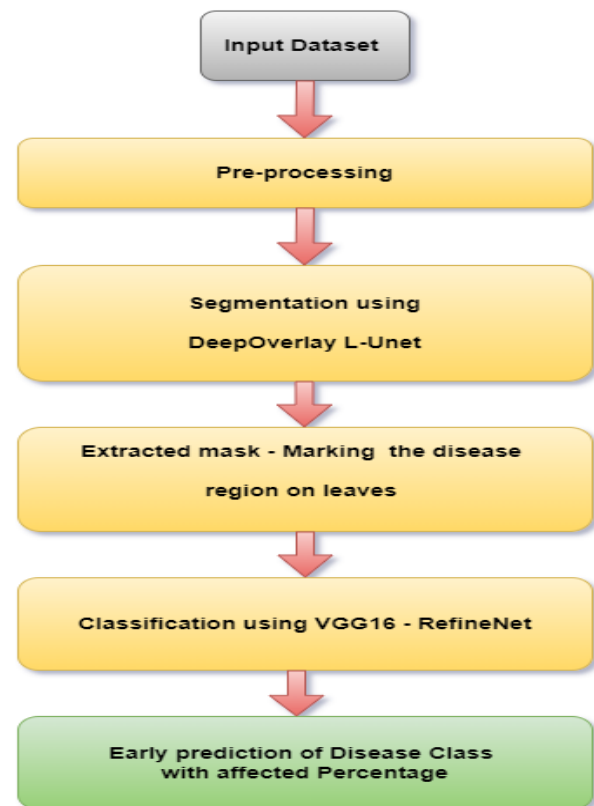


Fig. 1. Block diagram.

The paper's layout is organized in this way: Section II contains the literature survey. Section III discusses the description of the dataset. Section IV provides a detailed explanation of the entire methodology's working process. Section V, outlines the assessment criteria for the proposed method and discusses the usefulness of this method in performing real-time testing, highlighting, and detecting. Finally, the general summary, constraints, and prospects are covered in Section VI.

II. LITERATURE SURVEY

This section offers an extensive examination of various techniques for detecting plant diseases in order to understand how they work and recognize any possible limitations. As the global population surges, agriculture becomes increasingly vital for the energy needs of nations. Yet plant diseases reduce crop yields and quality, creating obstacles to agricultural progress. Accurate diagnosis of disease is essential to successful prevention and control. Disease identification has always been a manual procedure carried out by professionals that takes time. To overcome these inefficiencies, an automated system for identifying plant diseases was developed and implemented. Moreover, recent technological advancements have been employed to examine plant diseases and pests within the agricultural sector. The core of contemporary research in this domain is artificial intelligence, particularly its subset, machine learning. Furthermore, deep learning (DL) techniques have proven effective in various image processing applications, such as detecting, segmenting, recognizing, and categorizing diseases.

With the use of pre-processing and hybrid optimisation approaches, [14] creates an optimized framework using YOLOv7. YR2S (YOLO-Enhanced Rat Swarm Optimizer) incorporates Red Fox Optimization alongside ShuffleNetv2. Using ShuffleNet with ERSO for classification, the framework creates feature maps for leaf detection, and FCN-RFO is used to segment regions that are prone to illness. When applied to a tailored dataset, the model performs better than existing methods.

For diagnosing diseases and detecting damage to plant leaves, an automated method is suggested. [9] With 100% classification accuracy, the first step utilizes DenseNet to diagnose illnesses based on leaf pictures. In step two, a 1D Convolutional Neural Network (CNN) with 97% accuracy is used to identify leaf damages through semantic segmentation using deep learning. Depending on the type of disease and the extent of damage, the third stage recommends treatment. [20] A MULTINET approach was created to address the problem of 3D plant leaf disease detection and severity predictions by integrating multi-agent DRL and EfficientNet. Four processes are used in the framework: segmentation, species detection with classification by using a block divider model, Enhanced Deep Q-Network, EMMARO-based data augmentation, and numerous agents utilizing Deep Reinforcement Learning (DRL).

The method for automatically recognizing and identifying multi-biotic tomato leaf lesions is presented in study [16] and utilizes multiple CNNs. This system utilizes Hybrid-DSCNN for semantic segmentation, Mask R-CNN for segmentation,

and a CNN for classification. The Hybrid-DSCNN two-layer Layer-Convolution achieved segmentation and classification accuracy of 98.25%, along with a precision of 95.7%. [13] The CAAR-UNet, an autoencoder with attention and residual connections, utilizes a cascading structure in the computer vision method created to precisely detect and diagnose diseases in plant leaves early on. Achieving an average pixel precision of 95.26%, the deep learning approach achieves good precision.

In study [21], presents the dataset of Wheat Rust Disease at NUST (NWRD), which classifies wheat rust disease (WRD) into several kinds and categories using multi-leaf pictures from wheat fields. The UNet semantic segmentation model paired with the adaptive patching with feedback approach, showed encouraging outcomes. The research in [22] work explores the segmentation of disease using the U-Net architecture. The research utilized VGG16, MobileNet-v2, AlexNet, and DenseNet201 deep learning methods on a set of 60 images containing angular leaf spot and bean rust diseases. This work found that segmented pictures had greater classification accuracy than the original ones.

The authors in study [23] suggested technique for precisely identifying and classifying agricultural diseases, such as early and late blight, to assess disease damage is the Detection Transformer for Disease Segmentation (DS-DETR). To increase convergence speed, the model utilizes the Plant Disease Classification Dataset for unsupervised pre-training. To improve model accuracy, the query box is given Gaussian-like spatial weights using Spatially Modulated Co-Attention (SMCA). Evaluating this model on the Tomato Leaf Disease Segmentation Dataset resulted in a disease grading accuracy of 0.9640.

For integrated fusarium head blight (FHB) severity identification, [8] they developed a system that fuses multiple models based on deep learning. High-throughput wheat spike photos showed 97.6% segmentation accuracy, whereas fine and complicated FHB spots showed 99.8% accuracy. The approach also improved the classification of wheat FHB grading, moving from stages of disease management to the breeding process.

Utilizing Felzenszwalb's graph-based segmentation technique with annotated citrus fruits [24], a model of the deep neural network is developed to detect the severity of the condition. The prognostic model attains a 99% accuracy rate for minor severity levels, 98% accuracy for major severity levels, 96% accuracy for good conditions, and 97% accuracy for moderate severity levels. There are four severity categories for citrus fruit illnesses, and this method is effective and valid for identifying them.

The authors in study [1] aims to differentiate and categorize canker, greening, and blackspot diseases in citrus crops by utilizing image processing and machine learning algorithms. Preparation and segmentation tasks are performed on various images from the dataset Citrus Leaves Prepared. A new CNN structure is designed to consist of four blocks and brief directions. The model can effectively differentiate between citrus black spot, bacterial canker, and huanglongbing (greening), as they are predominantly categorized by it.

The study utilizes segmented images to focus on and feeds them into deep neural networks in order to create an ensemble stacked deep learning model for automatically detecting mango-leaf diseases [25]. Combining the output of the deep neural network with an ML model is utilized for detecting leaf illness. With an accuracy rate of 98.57%, the model performs better than existing models.

With the purpose of identifying and classifying biotic stress in coffee leaves early on, this work presents the extracted feature ensemble (EFE) approach. [26] The method enhances classification performance by utilizing custom-designed features and convolutional neural networks (CNNs) based on transfer learning. The effect of dimensionality on the performance of the model is evaluated, and three approaches are suggested to analyse extracted feature sets. This work indicates that the process of feature concatenation improves the accuracy and discriminative power of classification models.

However, existing approaches have limitations such as inadequate background conditions, cost complexity, misclassifications, and overfitting [27]. The AgriDet system, composed of a fusion of Kohonen-based deep learning networks and the traditional INC-VGGN, is introduced to address this issue. By incorporating a dropout layer, a Kohonen learning layer, and a pre-trained INC-VGGN model, the system is able to effectively identify and categorize illnesses.

Research indicates that using attention-based dilated CNN logistic regression is an effective approach for identifying tomato leaf disease. In study [19] images are preprocessed with bilateral filtering and Otsu segmentation; a synthetic image is generated using the Conditional Generative Adversarial Network model, features are normalized, and a logistic regression classifier is used to categorize the images. The results of this study show that the accuracy in training, testing, and validation for identifying multiclass tomato leaf diseases is 100%, 100%, and 96.6%, respectively.

Detecting diseases in sugarcane plants using current methods is inaccurate. The study in [28] identify and classify sugarcane leaf disease with high accuracy, The deep transfer learning model presented in this work is based on quantum-behaved particle swarm optimization (QBPSO-DTL). SqueezeNet, a deep-stacked autoencoder, and optimum region-expanding segmentation are all used in the modelling process.

In [29], study is centred on developing a fusion model for detecting and classifying diseases in rice plants using Efficient Deep Learning techniques (EDLFM-RPD). This method utilises preprocessing techniques such as median filtering, K-means segmentation, a manually created Gray Level Co-occurrence Matrix (GLCM), deep features from Inception, and Swarm Optimization with a Fuzzy Support Vector Machine (FSVM) model. Tests demonstrate improved efficiency, with a top accuracy of 96.170%.

TABLE I. COMPARISON OF VARIOUS SEGMENTATION TECHNIQUES WITH THE DEEPOVERLAY L-UNET

Application	Dataset	Number of images	Methods	Performance	Author
Disease classification	A new plant disease dataset	18345	Optimized ShuffleNet v2	Average Accuracy: 99.69%	[14]
Damage detection on leaves	Plant village dataset (four diseases)	8,875	1D-CNN	Average accuracy of 97%	[9]
Segmentation and detection of plant disease	The Tomato Leaf Disease Dataset (TLDD)	1004	Hybrid-DSCNN	Accuracy: 98.24%, IoU: 92.91%, Precision: 92.83%, Recall: 94.36%	[15]
Citrus disease detection and classification	citrus dataset	598	Modified CNN	Average Accuracy: 95%.	[1]
Segmentation and detection	The Tomato Leaf Disease Dataset (TLDD)	1680	Hybrid-DSCNN (2Layer-USN)	IoU: 92.8%, mIoU: 94.24%, accuracy: 98.25%.	[16]
Earlier disease detection.	Mango leaf diseases	2000	Ensemble Stack neural network.	Accuracy 98.57%	[25]
Precise detection of diseases in plant leaves.	The Plant Village Dataset and the Coffee Leaf Dataset.	400	Cascade Autoencoder incorporating Attention Residual U-Net	Pixel accuracy mean: 95.26%, IoU: 0.7451.	[13]
Early disease detection	NUST Wheat Rust Disease Dataset	100	Octave-UNet	IoU of 0.316, F1 score of 0.529.	[21]
Disease Segmentation and Classification	Dry bean leaves	120	U-Net and DenseNet201	IoU: 0.7725, F1-score: 0.9459%.	[22]
Plant disease detection and categorization.	coffee leaves	4000	modified VGG16	test accuracy: 97.9%.	[30]
Diagnosing the disease Severity.	Fusarium Head Blight on Wheat.	3875	Mobilev3 and Deeplabv3+	MIoU: 83.61, accuracy: 98.54%. overall accuracy rate of 86.9%.	[8]
Disease severity classification	Plantdoc and Plant Village datasets	2,598	INC-VGGN	Training accuracy: 98.9%, and validation accuracy: 96.00%.	[27]
Detection and severity analysis of disease	Grape dataset	500	DeepLabV3+ is based on ResNet50.	overall accuracy: 97.75%	[11]
Leaf disease detection.	Tomato leaf from the Plant Village dataset	18,161	Modified UNet	Test accuracy: 98.66%, IoU of 98.5%, dice: 98.73%. (Separate leaf and background)	[31]
Highlighting the affected disease region with a severity percentage.	Citrus plant dataset	±5000	DeepOverlay L-UNet, and VGG-RefineNet	Train IOU of 0.9864 , validation IOU of 0.9334 , overall classification accuracy of 98% .	Proposed

Various methods are proposed for citrus leaf segmentation and classification in the literature. Table I summarizes the efforts made to identify diseases in several plant species using these methods. Even though the methods proposed in the literature perform well in segmentation and classification, they fail in severity-based classification and highlighting the region of interest, which plays a major role in early and accurate disease detection. The proposed method is chosen since the model balances global context encoding and local detail refinement for accurate segmentation boundaries, which is the major contribution of this work.

III. DATASET DESCRIPTION

This study utilizes Citrus Leaves Setup, a citrus dataset available for open access. There are four categories in the collection: citrus black spot, citrus canker, and greening, which are the most common citrus diseases, along with pictures of healthy leaves. The dataset from study [32] contains images with a resolution of 256x256. The dataset has a smaller number of images as well as reduced quality, with an unbalanced distribution of data classes across different data.

This study focuses on enhancing disease detection accuracy in plant pathology by improving citrus leaf image quality and stability through preprocessing and augmentation techniques. The initial stage of preprocessing is essential, as it transforms simple images into a format that is easier to analyse. This is achieved by making several improvements: increasing contrast by 10% to highlight features, boosting brightness by 60% to counteract possible underexposure, and tripling sharpness to bring out intricate details [25]. These enhancements are essential to ensuring that upcoming machine learning models can recognize and absorb the fundamental aspects of the data.

Augmentation techniques artificially expand the dataset by approximately five thousand images, leading to a wider and more inclusive collection of images for training the model. This study involves examining both axes, rotating at angles of 0 and 90 degrees, and utilizing scaling factors of 0.5 and 1. These changes mimic the various positions, inclinations, and dimensions that leaves can exhibit in their natural surroundings, enabling the models to handle the diversity found in real-world situations. Additionally, employing the HSV colour scheme for image masking could help separate unhealthy areas. [10], [33] Establishing specific thresholds enables various masks to effectively separate colour data and reveal distinct areas of interest, which are then extracted and evaluated for disease identification.

The last step in preprocessing and augmentation involves placing masks on top of the original images. The use of a partially transparent layer allows for a direct visual comparison between healthy and diseased regions, leading to a better understanding of the impact of the disease, and this overlaid data is highly effective in segmentation. This overlay serves as both a helpful aid for visual examination and a useful tool for displaying data in a user-friendly way. Fig. 2, Fig. 3, and Fig. 4 included in this study show the detailed process of preparing citrus leaf images for effective disease identification, highlighting the crucial role of thorough preprocessing in image analysis for detecting plant diseases. A detailed breakdown of the number of images in a given class within this dataset, as well as its numbers after augmentation, is presented in Table II. The data specifically prepared for segmentation is displayed in Table III.

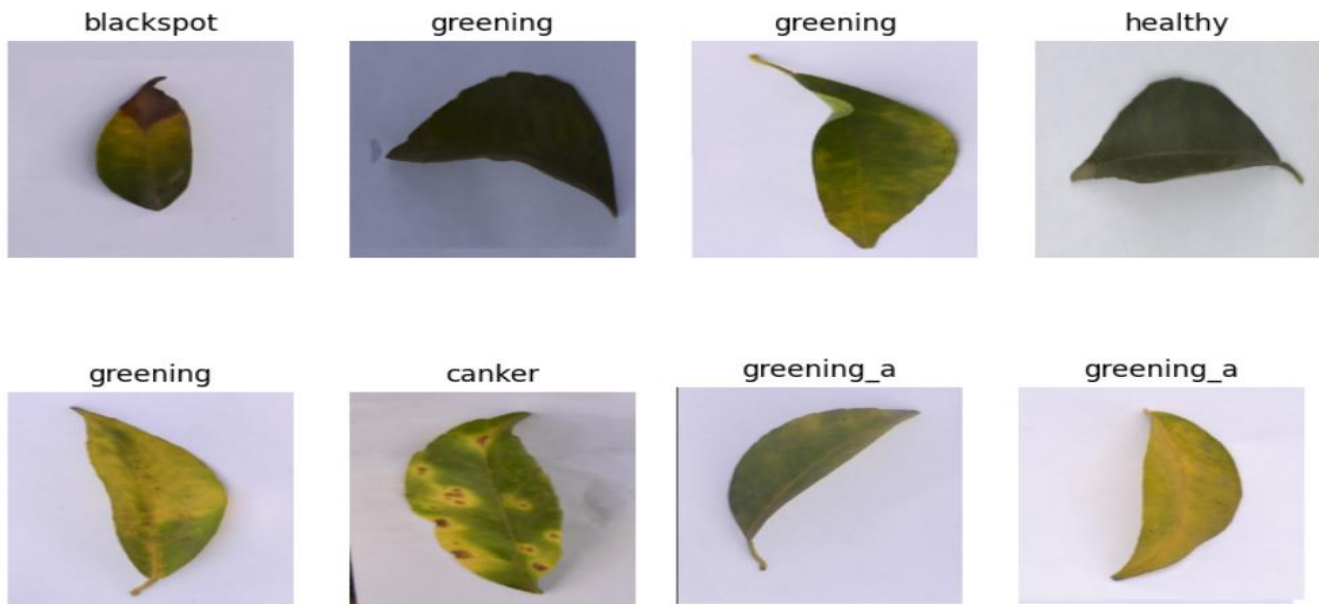


Fig. 2. Representative images extracted from the dataset.



Fig. 3. Representative images from dataset after preprocessing and augmentation for classification.

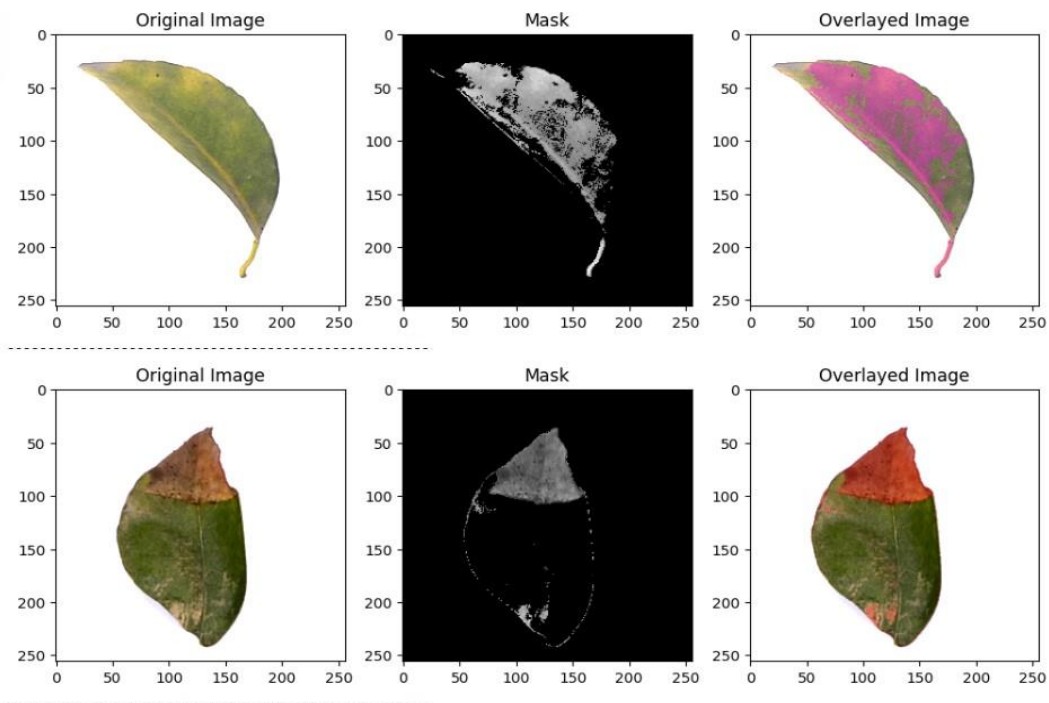


Fig. 4. Sample input of original image, masked images, and overlaid images for segmentation.

TABLE II. TRAIN, TEST AND A VALIDATION SET OF ORIGINAL AND AUGMENTED DATASETS FOR CLASSIFICATION

	Training		Testing		Validation		Total sample	
	original	augmented	original	augmented	original	augmented	original	augmented
Blackspot	122	852	15	106	15	106	152	1064
Canker	121	845	15	106	15	106	151	1057
Greening	171	1192	21	149	21	149	213	1491
Healthy	46	326	6	40	6	40	58	406
Total	460	3215	57	401	57	401	574	4018

TABLE III. TRAIN SET, TEST SET OF OVERLAYED AND MASKED IMAGES FOR SEGMENTATION

	Overlaid image	Masked image	Total sample
Train set	460	460	920
Test set	114	114	228
Total	574	574	1148

IV. METHODOLOGY

In this work, a DeepOverlay L-UNet architecture was used in the proposed method. The output of the method was then fed into the VGG-RefineNet deep learning architecture to detect and categorize plant diseases based on the severity percentage of the citrus-diseased plants. Initially, preprocessing involves resizing, enhancing, improving contrast, and augmenting images to avoid inconsistencies in the dataset. The HSV threshold-based colour scheme is used to separate diseased areas in image masking to address complex, multiple background challenges. This research provides distinct information about colour related to specific diseases. Following that, place the masks on top of the original image. In the first phase, masked images and overlaid images are fed as input for semantic segmentation. The semantic segmentation method DeepOverlay Leaky Relu-based deep learning is used to extract and learn features from diseased areas of the citrus leaf region. The affected area of the diseased citrus leaf is predicted using a combination of DeepOverlay L-UNet characteristics. In the second phase, disease classes are identified through classification using VGG-RefineNet. The model surpasses current methods in segmentation, detection, and classification, showing improved validation and accuracy. Finally, in the third phase, the severity of the disease on the citrus leaf is assessed by the Highlighting Disease Area with the Affected Percentage (HDAP) method, which calculates the percentage of the affected area by identifying the disease boundary and using the DeepOverlay L-UNet to segment and overlay the affected area on the leaf. Researchers calculate the total affected percentage on the citrus leaf by measuring both the entire leaf area and the area covered by the disease.

The detection and classification of diseases are crucial in the field of agriculture. This detection method using images helps make it easier to protect crops from disease compared to sensor-based solutions. Proposing an improved application is crucial to helping farmers. Citrus farmers have the ability to recognize suspicious images of their crops and gather essential data, such as the severity percentage of the disease. Here researcher emphasizes the use of a neural network approach, utilizing DeepOverlay L - UNet and VGG - RefineNet deep neural networks to accurately identify the affected region on the leaves. This allows for the trustworthy recognition and detection of citrus plant diseases. This technique involves three separate steps: segmentation, highlighting the disease-affected area, and categorizing with a deep neural network. Fig. 10 provides an overall overview of the proposed work. The disease identification method consists of three stages: a proposed (DeepOverlay L-UNet) enhanced base network for segmentation, an improved (VGG-RefineNet) network for classification, and the introduction of the Highlighting Disease Area with Affected Percentage (HDAP) method. The following paragraphs will outline each of these phases:

A. DeepOverlay L-UNet for Segmentation

Once the masked image and overlay images have been preprocessed, they are used as input for the training process. The DeepOverlay L-UNet is an advanced neural network created specifically for image segmentation. The process starts with selecting an input image. The network's design is built on

the U-Net model, featuring both an encoder to capture image context and a decoder for accurate localization. Within the encoder part, convolutional layers are used along with batch normalization and Leaky ReLU activation functions with an alpha value of 0.1. This particular activation function is selected for its capacity to enable low gradients when the unit is not active, thereby addressing the vanishing gradient issue often associated with conventional ReLU functions. The decoder segment uses transposed convolutions to enlarge the feature maps, which are then joined with the encoder feature maps that correspond. This stage is essential for the network to accurately pinpoint and outline the edges of the objects in the image. The output layer contains a convolutional process combined with a sigmoid activation function, producing a probability map that displays the segmented sections of the image.

Custom metrics like IoU, mean IoU, weighted mean IoU, pixel accuracy, mean pixel accuracy, mean boundary F1 score, and dice coefficient are utilized to assess the network's performance. These measurements offer a thorough evaluation of the quality of segmentation. The training procedure includes building the model with an Adam optimizer and a binary cross-entropy loss function. The model is trained for 150 epochs with a batch size of 2, showing a thorough optimization process to enhance segmentation skills.

Finally, the network includes a visualization component where the predicted segmentation masks can be overlaid on the original images. This visual inspection is essential for verifying the model's predictions and ensuring the segmentation's accuracy. In essence, the DeepOverlay L-UNet with its Leaky ReLU-enhanced encoder and comprehensive metrics, offers a robust solution for image segmentation tasks, ensuring detailed and accurate delineation of image features. The model architecture is explained below:

The DeepOverlay L-UNet architecture is designed for semantic segmentation tasks, where the goal is to categorize every pixel in an image into various classes, such as separating foreground from background.

1) *Encoder*: The model's encoder part includes convolutional layers and pooling layers. It transforms the input image into a compact, meaningful representation known as the latent space. Convolutional layers derive hierarchical features from the input data, while pooling layers decrease the spatial dimensions of the feature maps. The model includes two encoder blocks, each composed of a convolutional layer, batch normalization, and max pooling.

- The first encoder block takes an input tensor with the shape (128 x 128 x 3) and uses 32 filters, producing output tensors x1 and p1.
- The second encoder block takes the output tensor p1 from the first block as its input and uses 64 filters, resulting in output tensors x2 and p2. The number of filters increases with each subsequent encoder block to capture more complex features.

a) *Convolutional layer*: The fundamental component of CNNs is the convolutional layer. It applies kernels (filters) to

the input data and extracts local features. This layer uses filters to capture specific characteristics from the input data by moving the filter across the input and conducting element-wise multiplications. Convolutional layers learn spatial hierarchies of features, capturing patterns like edges, textures, and shapes.

b) Batch normalization: Batch normalization (BN) stabilizes training by normalizing the activations of each layer, addressing issues like vanishing or exploding gradients. BN normalizes the mean and variance of activations within a mini-batch. It introduces learnable parameters (gamma and beta) to scale and shift the normalized values. It improves convergence speed, generalization, and robustness.

c) Activation function: An activation function, such as ReLU, brings non-linear elements to the model, enabling it to learn complex patterns.

d) Max pooling: Pooling layers decrease spatial dimensions while maintaining essential features. Max pooling selects the maximum value within a region (e.g., 2x2) of the feature map. Pooling layers downsample the feature maps, reducing the computation and preventing overfitting. It summarizes features, enhancing the model's robustness to changes in position.

2) Leaky ReLU activation: Leaky ReLU activation is applied to both x_1 and x_2 . Negative values in x_1 and x_2 are scaled by the alpha value (usually set to a small positive number). Positive values remain unchanged. Leaky ReLU is applied after each encoder block to allow a small gradient when the input is negative, preventing vanishing gradients and helping in learning complex features.

3) Bottleneck block: A bottleneck block with 128 filters follows the encoder blocks, capturing high-level features and compressing them into a compact representation.

4) Decoder: The decoder reconstructs the original input from the latent representation using transposed convolutional layers to increase spatial dimensions. The decoder's output aims to match the original input (e.g., a reconstructed image). Two decoder blocks are specified, each consisting of a transposed convolutional layer for the upsample, concatenated with the relevant encoder output, and an additional convolutional layer.

- The first decoder block takes an input tensor b (bottleneck output), concatenated with the corresponding encoder output x_2 and uses 64 filters.
- The second decoder block takes the output from the first decoder block as its input, resulting in output tensors x_1 and 32 filters. Each decoder block includes a transposed convolutional layer, concatenation, and another convolutional layer. The decoder blocks progressively refine the features and recover spatial information lost during downsampling in the encoder.

a) Transposed convolutional layer (Deconvolution): Transposed convolutional layers (also known as deconvolutional layers) perform upsampling. It increases spatial dimensions, allowing the network to generate higher-

resolution outputs. Transposed convolutions apply filters in reverse; they project a smaller feature map onto a larger one. These are commonly used in image generation tasks (e.g., GANs) and semantic segmentation.

b) Concatenation with skip connection: The decoder block concatenates the upsampled feature maps with the corresponding feature maps from the encoder (skip connection). Utilize skip connections (also known as residual connections) for transmitting information between layers. This aids in spreading intricate details. By integrating both low-level and high-level characteristics, the model is able to improve the localization of objects and boundaries. In U-Net architectures, the encoder's feature maps are combined with the decoder's feature maps to enhance segmentation outcomes.

An additional convolutional layer: The combined feature maps undergo further enhancement with an additional convolutional layer.

5) Output layer: The final output, showing the likelihood of each pixel being part of the foreground, is generated by a 1x1 convolutional layer with a sigmoid activation function, with an output shape of 128 x 128 x 1.

6) Model training and evaluation: The model is built using binary cross-entropy loss and Adam optimizers with multiple evaluation metrics. During training, the model learns to minimize the loss function by adjusting its weights. The evaluation metrics (IoU, accuracy, etc.) assess the model's performance on validation data.

Prediction and Visualization: Predictions are made on training and validation data. Overlay images are created by combining the original image and the predicted mask. The overlay helps visualize the segmentation results, which helps highlight the affected area. The segmented output of different citrus diseases is shown in Fig. 5, and the architecture diagram of DeepOverlay L-UNet is shown in Fig. 6.

In this method leaky Relu is introduced to gain more features in the citrus image and overlay the segmented output to original image, this will visualize affected area. These differentiates the proposed method from the U-Net. By using the DeepOverlay L-UNet Severity region exactly extracted compared to other methods.

Algorithm: Disease Region Detection using DeepOverlay L-UNet

Input: Training overlaid images (X_{train}), corresponding ground truth masks (Y_{train}), and validation images (X_{val}).

Output: Predicted masks for leaf disease regions.

Step 1: Load the training images and ground truth masks.

Step 2: Define the L-UNet model architecture:

- Input layer to accept images of shape (128, 128, 3).
 - Encoder blocks to capture features at different scales. After each convolutional block within the encoder, apply LeakyReLU activation.
 - Bottleneck block with convolutions but no pooling. Apply LeakyReLU activation after the bottleneck to maintain gradient flow.
 - Decoder blocks to upsample and restore the original image
-

dimensions. After each convolutional block within the decoder, apply LeakyReLU activation.

- Output layer utilizes the sigmoid activation function for prediction of the segmentation mask.

Step 3: Compile the model with Adam optimizer, binary_crossentropy loss, and custom metrics (e.g., IoU, F1-score).

Step 4: Train the model on the training data with a validation split for monitoring performance.

Step 5: Predict masks on the validation set (X_val) using the trained model.

Step 6: Apply a threshold to convert the model's predictions to binary masks.

Step 7: Overlay the predicted masks on the original validation images to visualize the results.

Step 8: Calculate the affected area (number of pixels) based on the overlay mask.

Step 9: Repeat steps 5 - 8 for a random validation sample.

Step 10: Display the original image, masked ground truth, predicted mask, and overlay.

End Algorithm

B. VGG - RefineNet Network for Classification

Following preprocessing, the dataset images are given as input to the training stage. The base model starts by using transfer learning to understand the characteristics. Transfer learning begins by utilizing a pre-existing base model. The training of these models was done using extensive datasets, such as ImageNet, and have acquired the ability to identify basic characteristics such as edges, textures, and shapes. The convolutional layers of the basic model function as feature extractors. They acquire the ability to identify basic characteristics in images. Utilizing a pre-trained base model allows you to leverage its acquired features without starting the training process from the beginning. The model's custom layer merges the feature extraction abilities of the base model with custom layers that understand specific patterns related to the task. The VGG RefineNet model was modified by incorporating custom layers (flatten, dense, batch normalization, dropout) to suit researcher dataset. With the VGG-RefineNet model, the characteristics of citrus plant diseases are being learned more precisely. The final classification probabilities are provided by the output layer.

The disease samples are subsequently processed by the VGG – RefineNet following modification of the current model. This recently created neural network is utilized for classifying the citrus diseases. The newly created neural network consists of the input layer, convolution layer, pooling layer, flatten layer, dense layer, batch normalization layer, dropout layer, and output layer. In this neural network, a deeper understanding is gained on the disease detection process to capture all essential features related to each disease category. Better precision is achieved in learning the texture, shape, size, colour, and other characteristics which avoids misclassification. The roles of the various layers are outlined below: Every layer in the neural network carries out a distinctive role.

1) *The base layer (VGG16)*: The VGG16 base model is made up of 13 convolutional layers which are responsible for

extracting features from the input image. Filters are applied in each convolutional layer to learn specific patterns in the local area. VGG-16 employs compact convolutional filters (3x3 in size). A ReLU activation function follows every convolutional layer. Proceeded by five layers of max pooling. Max pooling decreases spatial dimensions by downsampling the feature maps. It aids in preserving critical characteristics while decreasing the amount of computation needed. In implementation, VGG-16 utilizes max pooling with a stride-2 and a window size of 2x2.

a) *Layers that are being frozen*: Freezing the last four layers stops their weights from being modified in the training process. This is crucial for transfer learning, as it enables the model to keep the knowledge gained from ImageNet while adjusting the top layers for your particular task.

2) *Flatten Layer*: The Flatten layer converts the results from the convolutional layers into a single-dimensional array. Following the extraction of features by the base model, the flattened representation is used as input for the following fully connected layers (dense layers). When the convolutional layers output shape is (batch_size, h

3) eight, width, channels), the shape of the flattened output becomes (batch_size, height x width x channels).

4) *Dense layers*: After the flattened representation, researchers added two dense layers: The initial dense layer consisted of 1024 units and used ReLU activation. It grasps features at a high level. The following layer, with n_classes units (4 in this case), is the second dense layer and has softmax activation. It generates probabilities for different classes.

First dense layer:

$$Output = ReLU (Input \cdot Weights + Bias) \quad (1)$$

Second dense layer (output layer):

$$Output = Softmax (Input \cdot Weights + Bias) \quad (2)$$

5) *Batch normalization layer*: Batch normalization normalizes the outputs of the preceding layer, reducing internal covariate shifts during training. It scales and shifts the normalized activations using learned parameters to stabilize training, improve convergence, and accelerate learning.

$$Output = \frac{Input - \mu}{\sigma} \cdot \gamma + \beta \quad (3)$$

μ – mean.

σ – standard deviation.

γ – scaling factor.

β – shift factor.

6) *Dropout layer*: Dropout prevents overfitting by randomly turning off neurons while training. During each training iteration, a fraction of neurons (specified by the dropout rate 0.5) is randomly dropped out. The mask randomly sets some values to zero.

$$Output = Input \cdot Mask \quad (4)$$

The general structure involves extracting features from the primary model and incorporating specialized layers designed for the plant disease categorization assignment.

C. Highlighting Disease Area with Affected Percentage (HDAP)

HDAP is the process that carried out throughout the work because it needs to get the original leaf area and predicted

mask area it means disease disease-affected area to highlight the affected area with the severity percentage. It is helpful to avoid the cause of disease affected by the minimal usage of pesticides, fertilizers, and climate change. In this work researcher used DeepOverlay L-UNet and VGG-RefineNet to extract the details of the disease region, disease class to highlight the disease region, class on the original image to visualize for the farmer.

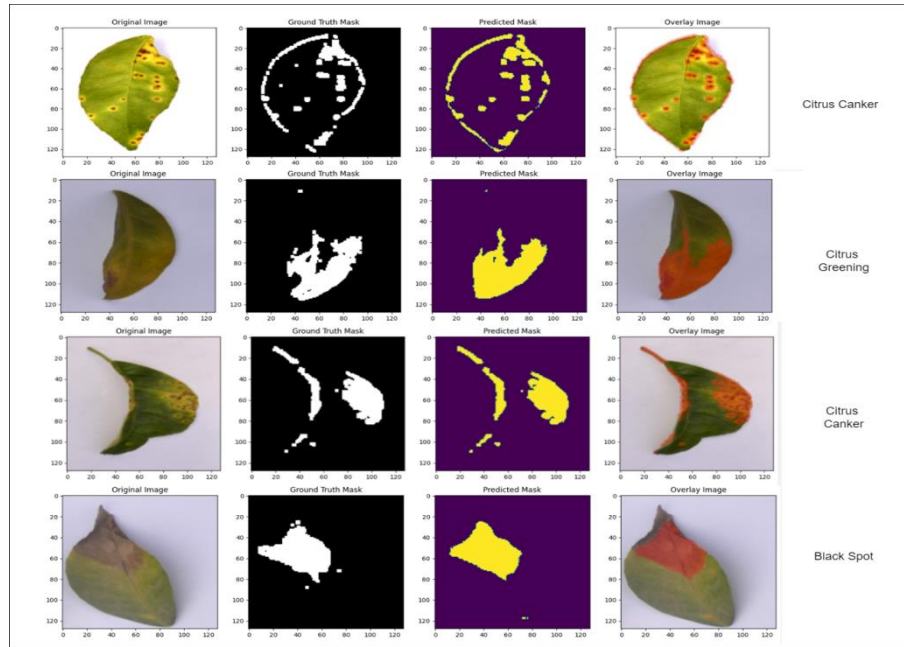


Fig. 5. Segmented output of different citrus diseases.

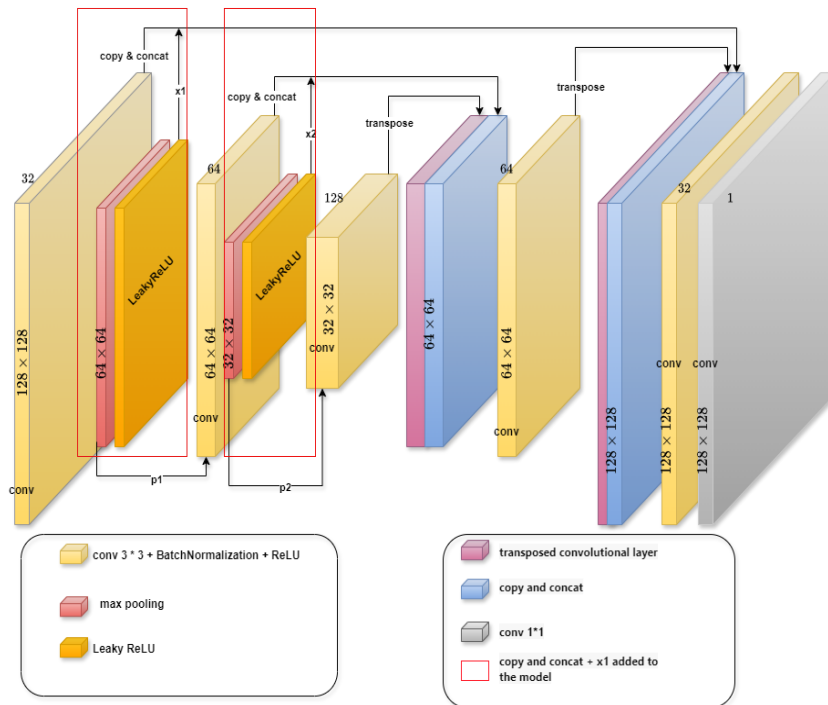


Fig. 6. Architecture diagram of DeepOverlay L-UNet.

The Highlighting Disease Area with Affected Percentage (HDAP) is a methodical process used to identify and quantify disease-affected areas in images, such as leaves in citrus plant disease detection. Before highlighting the disease area in citrus leaves researcher identified the area of the entire leaf by using contour function and it acquire coordinates of the leaf boundary pixels, then draw contour iterates over the list of contour points and draws lines between these points on the original image till it formed a loop. The region enclosed by the contour, which gives the area of the polygon defined by the contour points is used to get the area of the disease part in leaf. Using the trained segmentation model (Deepoverlay L-UNet) to predict the disease mask then Overlay the predicted mask on the original image. After overlaying the mask, identify and highlighting the contours of the disease-affected regions using contour function. Calculating the pixel area of the overlay image to get the area of the diseased part in citrus leaf. The HDAP is process computed the ratio of the area that is affected to the total area of the leaf, expressed as a percentage. This HDAP method provides important information about the existence and severity of disease within a leaf. These are explained in below steps:

1) *Original image and contour extraction:*

- Begin with an original image, such as a citrus leaf.
- Convert the original RGB image to grayscale. Grayscale simplifies edge detection, which is essential for contour extraction.
- Apply thresholding using Otsu's method to create a binary image. This step separates the foreground (citrus leaf regions) from the background. Pixels above the threshold become white, while others remain black.
- Detect contours within the binary image using `cv2.findContours()`. These contours represent the boundaries of the disease-affected areas.
- Draw these contours on the original image to visualize the leaf areas. The `cv2.drawContours()` function overlays the contours, providing a clear outline of the affected regions.

2) *Leaf area calculation:*

- Use `cv2.contourArea()` to calculate the area of each contour. This function applies Green's theorem to compute the area enclosed by the contour points.
- Sum up the areas of all detected contours to determine the total leaf area affected by the disease (in pixels).

3) *Predicted mask, overlay and classify:*

- Employ a trained segmentation model to predict the disease mask (segmentation mask) for the given image.
- Threshold the predicted mask to obtain a binary image.
- Overlay the predicted mask on the original image using a contrasting color (e.g., red) to highlight the affected regions.

4) *Diseased area calculation:*

- Calculate the pixel area of the overlay image by count number of non-zero pixels (white pixels) in the binary mask using NumPy functions. This provides the pixel area of the affected regions.

5) *HDAP calculation:*

- Calculating the HDAP by expressing the affected area as a percentage of the total leaf area.
- The HDAP ratio helps quantify the disease's extent, aiding in diagnosis, treatment planning, and monitoring.

The HDAP process is versatile and applicable beyond leaf disease detection, serving as a valuable tool in various fields requiring detailed image analysis and segmentation. Whether in agriculture imaging, HDAP provides insights into the presence and severity of disease, facilitating informed decisions and interventions.

D. *Evaluation Metrics*

1) *Precision:* It calculates the proportion of accurately predicted positive observations out of all predicted positives. A low false positive rate is associated with high precision.

$$\text{Precision} = \frac{\text{True Positives}}{\text{Predicted Positives} + \epsilon} \quad (5)$$

2) *Recall:* It is also referred to as sensitivity and calculates the ratio of accurately predicted positive observations to all true positives. A low false negative rate is associated with high recall.

$$\text{Recall} = \frac{\text{True Positives}}{\text{Possible Positives} + \epsilon} \quad (6)$$

3) *F1 score:* It is determined by averaging precision and recall, calculated using the harmonic mean. Balancing precision and recall are helpful when needed.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall} + \epsilon} \quad (7)$$

4) *Specificity:* Measures the percentage of correct predictions for negative outcomes compared to the total number of actual negative outcomes. It represents the frequency of accurate negatives.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{Possible Negative} + \epsilon} \quad (8)$$

5) *Accuracy:* It is a popular metric for assessing a model's performance on a specific dataset, it is determined by the ratio of accurately classified samples to the overall sample size.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (9)$$

6) *Intersection over union (IoU):* IoU is a commonly used measurement in segmentation challenges for evaluating the overlap between predicted and actual segmentation. It measures the proportion of the overlapping region compared to the total area.

$$IoU = \frac{\text{Intersection}}{\text{Union} + \epsilon} \quad (10)$$

7) *Mean IoU*: The average IoU score across all samples. It provides an overall performance measure for segmentation tasks.

$$\text{Mean IoU} = \frac{1}{N} \sum_{i=1}^N \frac{\text{Intersection}_i}{\text{Union}_i + \epsilon} \quad (11)$$

8) *Weighted mean IoU*: Similar to average Intersection over Union, this metric assigns more weights to classes with larger pixel counts. This is useful when class imbalance is present.

$$\text{Weighted Mean IoU} = \frac{\sum_{i=1}^N IoU_i \times \text{Weights}_i}{\sum_{i=1}^N \text{Weights}_i} \quad (12)$$

9) *Pixel accuracy*: Calculates the proportion of pixels classified correctly. It is an easy method to evaluate the overall accuracy of a segmentation model.

$$\text{Pixel Accuracy} = \frac{\text{Correct Pixels}}{\text{Total Pixels}} \quad (13)$$

10) *Dice coefficient*: Like IoU, calculates the intersection between two samples. The total number of pixels in both images divided by two is equal to the area of overlap. The Dice Coefficient incorporates a smoothing parameter to avoid dividing by zero. It evaluates the intersection of two samples and is especially beneficial for tasks involving binary segmentation.

$$\text{Dice Coefficient} = \frac{2 \times \text{Intersection} + \text{smooth}}{\text{Sum of True} + \text{Sum of Pred} + \text{smooth}} \quad (14)$$

Intersection is the overlap between the actual labels and the predicted labels. Sum of True is the overall count of true labels. Pred sum is the overall count of predicted labels. A tiny constant, smooth, is added to avoid division by zero and to enhance the metric for improved stability and performance.

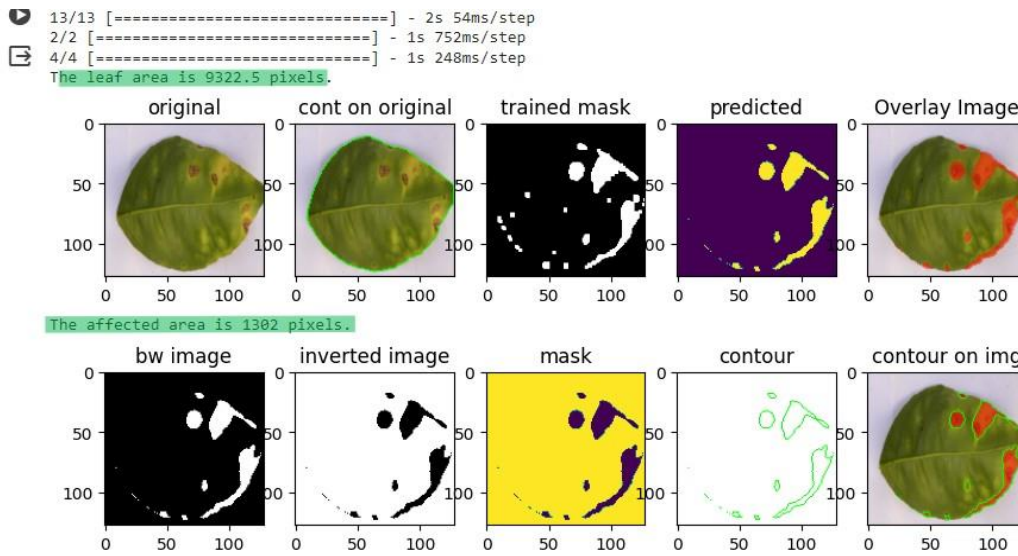


Fig. 7. Citrus Canker leaf sample on HDAP process.

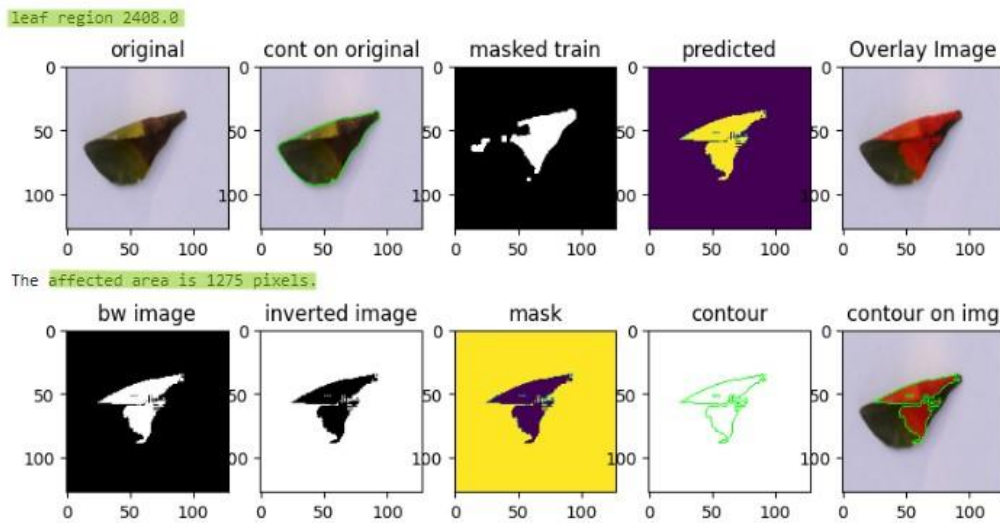


Fig. 8. Citrus Blackspot leaf sample on HDAP process.

```
1/1 [=====] - 0s 18ms/step
Predicted class: canker_a with confidence: 100.0%
1/1 [=====] - 0s 17ms/step
The total area is 23957.0 pixels.
```

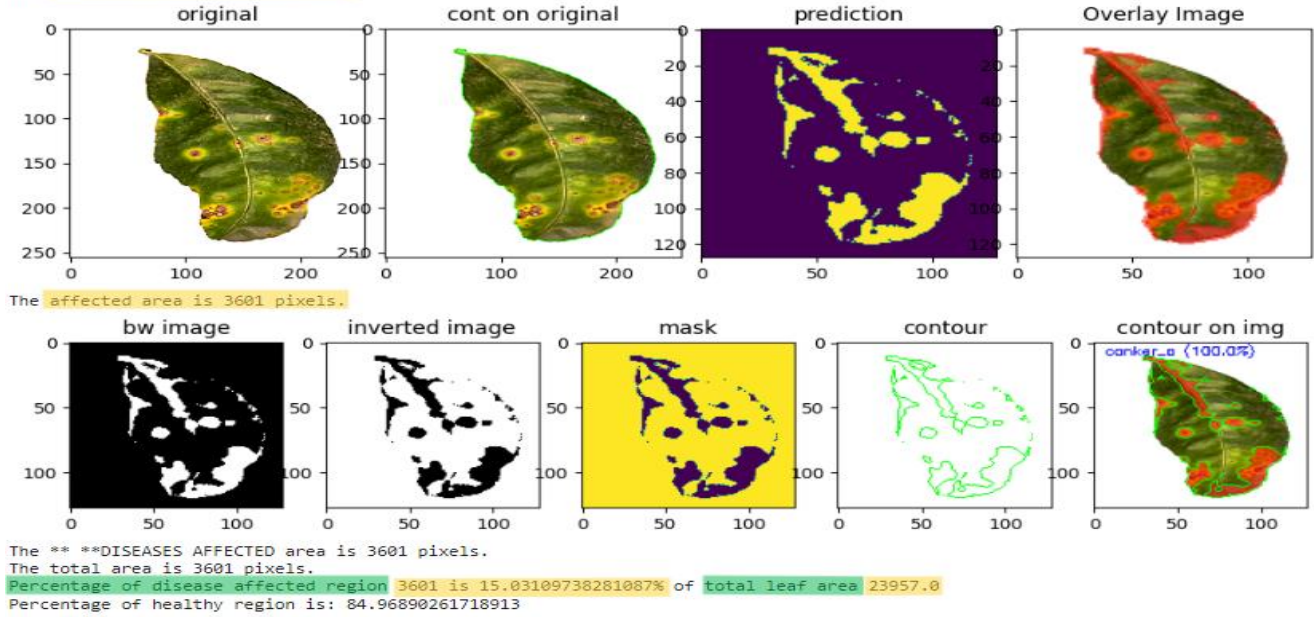


Fig. 9. Sample on highlighting the disease affected area with affected.

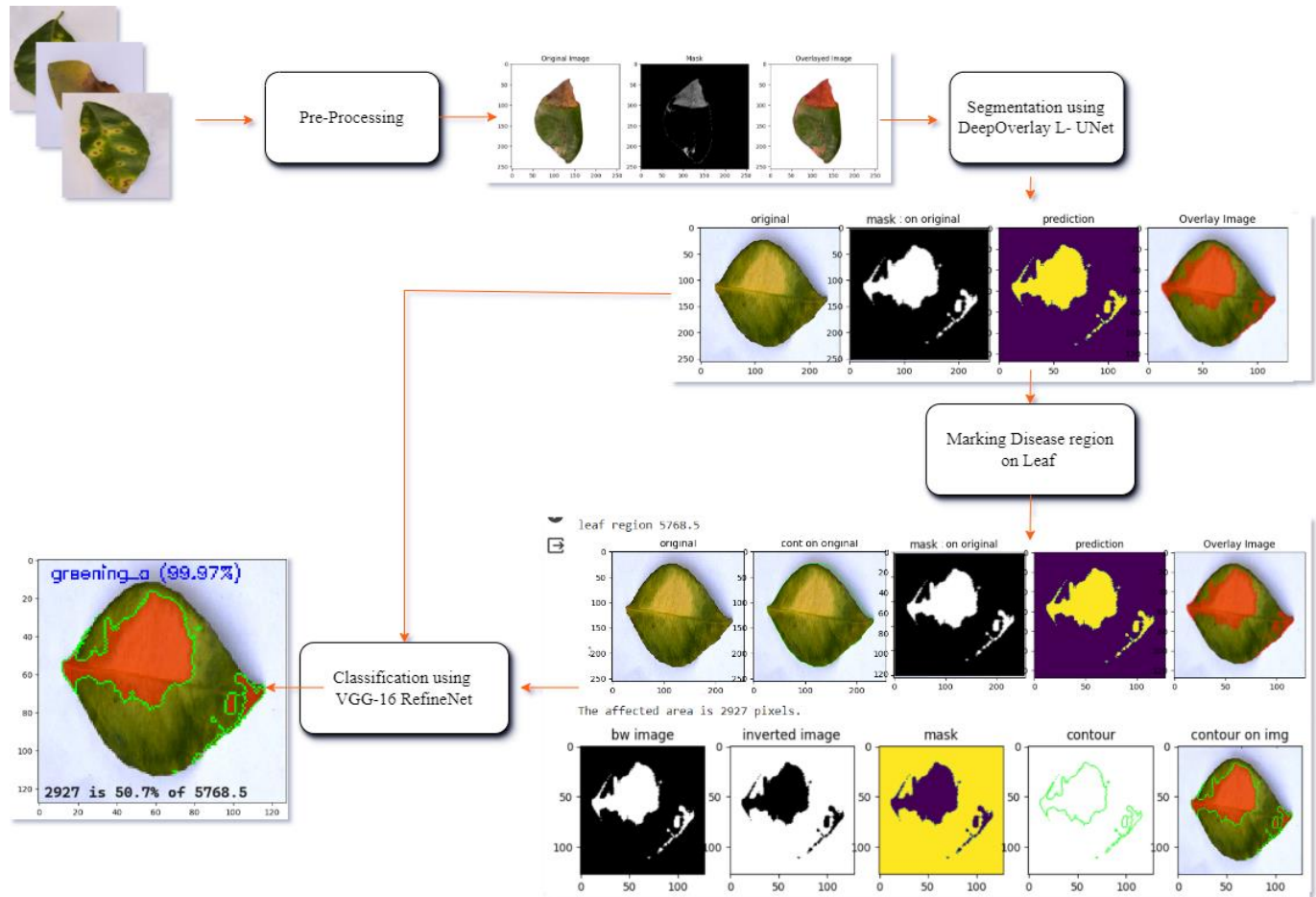

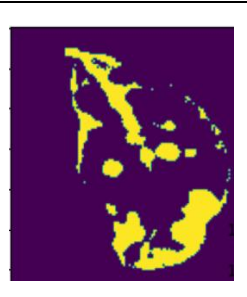












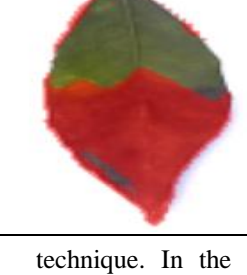


Fig. 10. Percentage. Overall Process of Citrus Disease Segmentation, Classification and HDAP.

TABLE IV. CITRUS DISEASE PREDICTED, SEGMENT OVERLAYED, CLASSIFIED AND HIGHLIGHTED THE DISEASE AFFECTED AREA WITH PERCENTAGE.

Input Image	Disease area Predicted by the proposed model	Overlay predicted into the input image	Proposed Classification and HDAP (by pixels)
			canker_a (100.0%) 3601 is 15.03% of 23957
			greening_a (99.97%) 1916.56 is 91.05% of 21049.5
			greening_a (99.97%) 2927 is 50.7% of 5768.5
			canker_a (99.95%) 2788.17 is 11% of 26347
			blackspot_a (99.98%) 1917.23 is 61% of 3143

V. RESULTS AND DISCUSSION

The research consists of three sections. The initial phase involved extracting disease from masked images of the afflicted, which were segmented using DeepOverlay L - UNet. This deep learning approach helped in identifying the disease. In the latter section, the segmented layered images were categorized using the VGG - RefineNet Deep learning

technique. In the third section, the disease area that was extracted was shown independently and the affected area of the disease was measured compared to the total leaf area using the HDAP method. After that, the segmentation and classification performance metrics were acquired and assessed. Fig. 10 illustrates the complete process of segmenting, classifying, and performing HDAP on citrus diseases in the study.

DeepOverlay L-UNet was utilized to segment the disease affected areas in the segmentation section. In this model proposal, the data consists of pre-processed data from the original images, masked images, and overlaid masks on original images. Following this step, the data is divided into sets for training and testing purposes. Next, the DeepOverlay L-UNet is put together and acquires a deeper understanding through the use of training data.

After finishing the training, the original images are displayed by overlaying the real values, predictions, and projections on top of them. Ultimately, the trained model's performance was assessed. The following are the listed steps for the proposed method of the DeepOverlay L-UNet architecture. 1) Uploaded the pre-processed data, 2) generated plots showing the images alongside the mask and overlay images, 3) divided data into training and validation sets, 4) Compiled and trained the model, then displayed the original image, ground truth, predicted image, and overlay of predicted

on original image, 5) Assessed the model using precision, recall, accuracy, IoU, Dice-Coefficient, and other metrics.

By utilizing Overlaid masked images as inputs and applying Leaky ReLU activation after each encoder block, this model gains finer details to improve performance in IoU, Pixel Accuracy, and Dice-Coefficient metrics. 80% of the dataset is allocated for training, while 20% is reserved for testing and segmented using DeepOverlay L-UNet. To explore the effectiveness of the DeepOverlay L-UNet architecture, the model's performance was assessed with 25, 50, 75, 100, 125 and 150 epochs to determine its impact on enhancing performance. Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15 and Table V display the precision, recall, specificity, f1 score, accuracy, loss, IoU, Mean IoU, Pixel Accuracy, and Dice-coefficient graphs, values obtained from the experimental results. Examples of predicted diseased images using the proposed model can be seen in images Table IV.

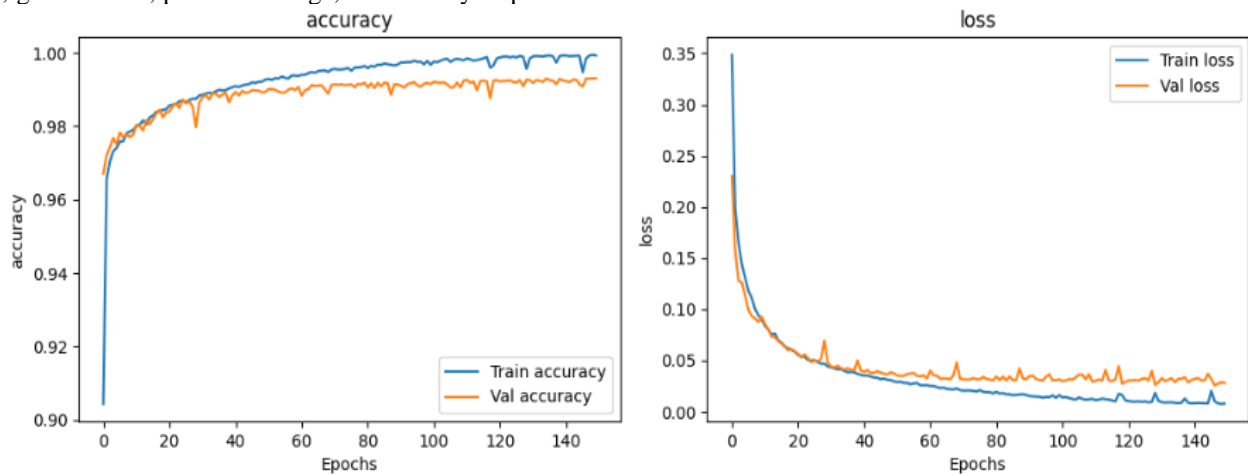


Fig. 11. Graphs showing the accuracy and loss during 150 epochs of training and validation.

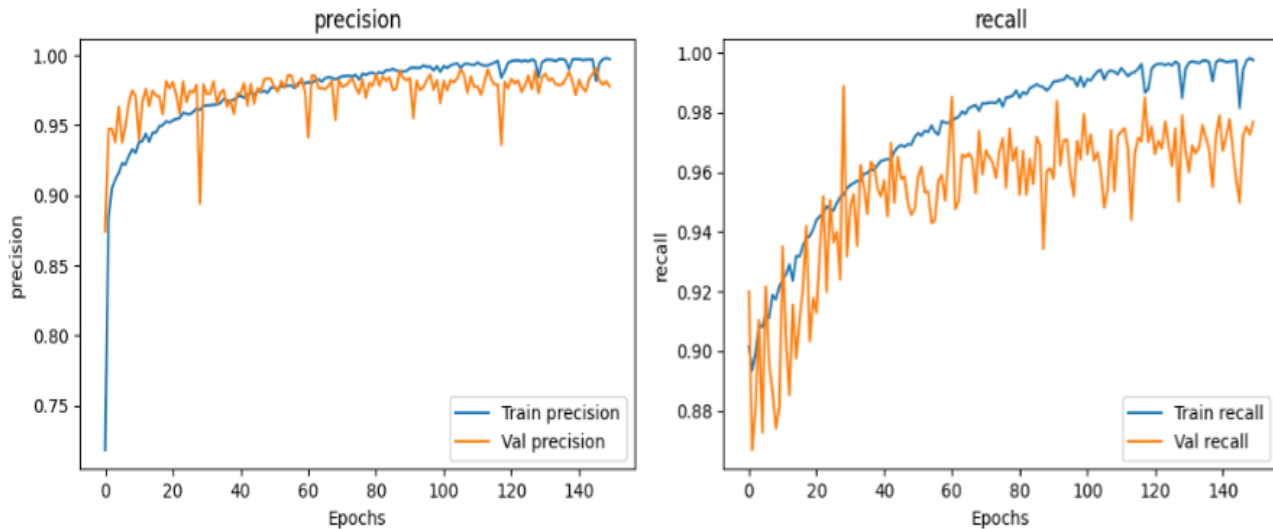


Fig. 12. Graph showing the precision and recall during 150 epochs of training and validation.

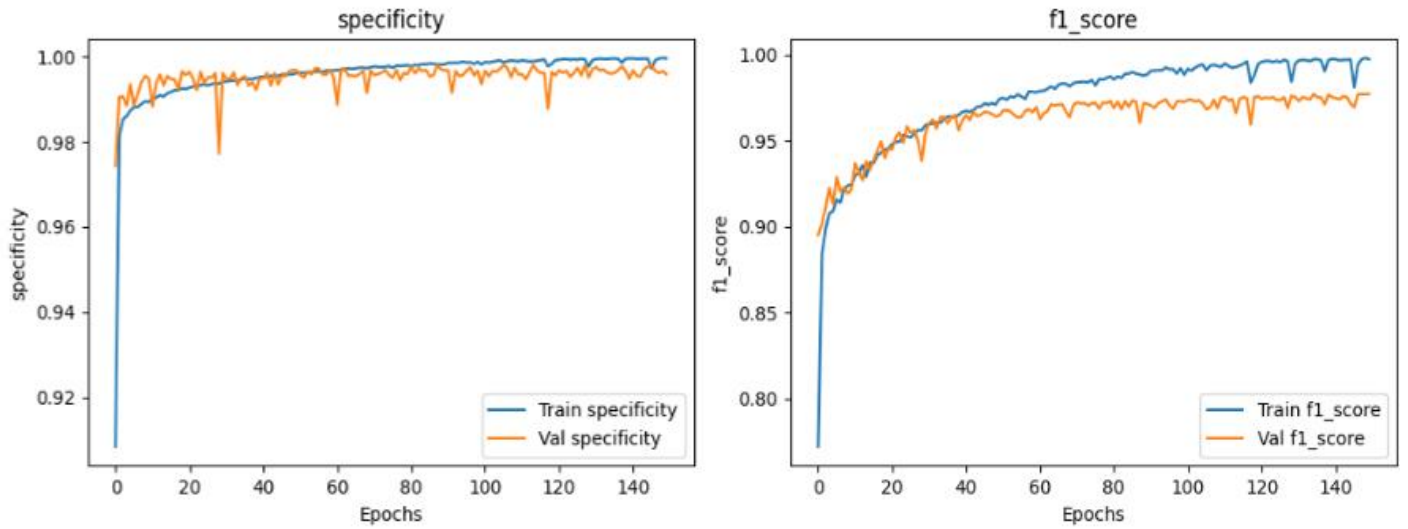


Fig. 13. Graph showing the specificity and f1_Score during 150 epochs of training and validation.

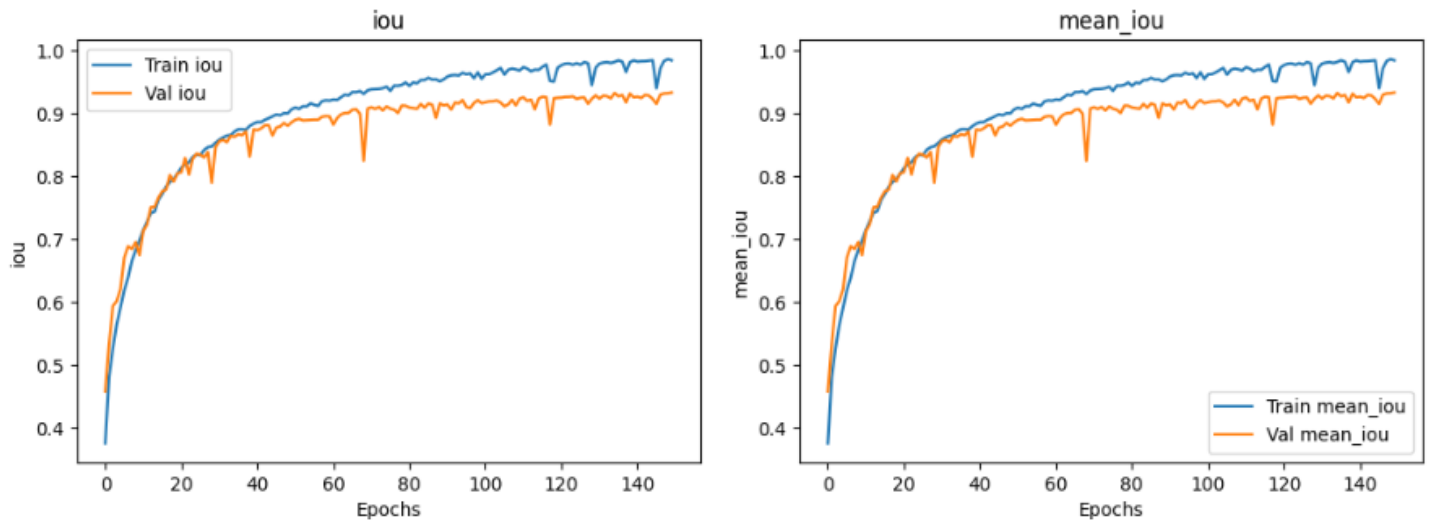


Fig. 14. Graph showing the IoU and Mean IoU during 150 epochs of training and validation.

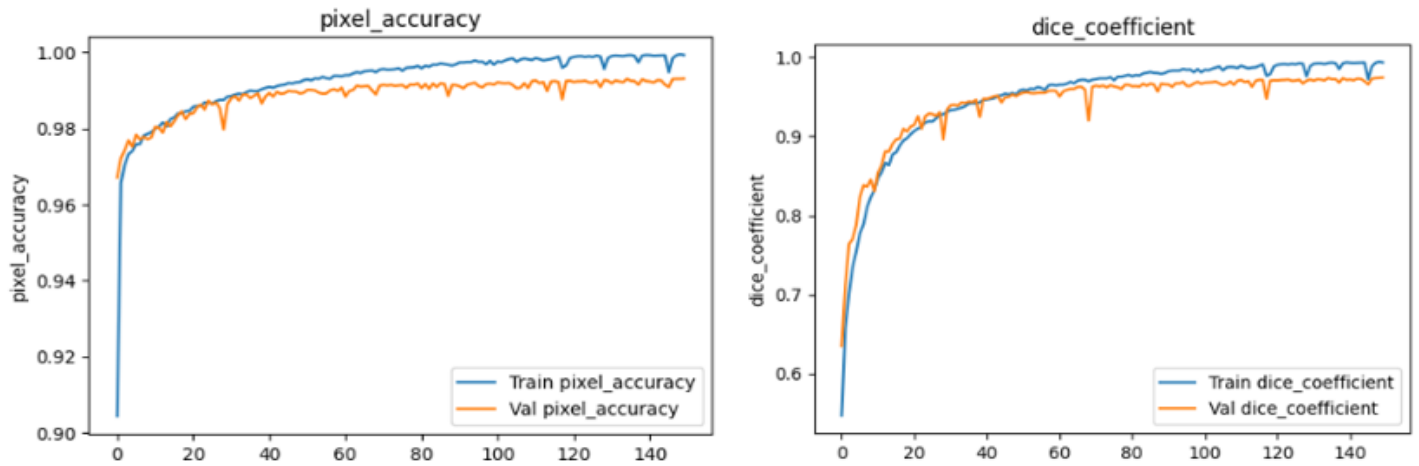


Fig. 15. Graph showing the pixel accuracy and dice coefficient during 150 epochs of training and validation.

TABLE V. EVALUATION METRICS OBTAINED ON DEEPOVERLAY L-UNET

Training Model	Evaluation Metrics									
Epochs	Accu	Val- Accu	IOU	Val-IOU	Pixel Accu	Val Pixel Accu	Dice	Val-Dice	Loss	Val- loss
25	0.9869	0.9873	0.8347	0.8361	0.9869	0.9873	0.9190	0.9288	0.0500	0.0484
50	0.9927	0.9894	0.9066	0.8861	0.9927	0.9894	0.9579	0.9537	0.0296	0.0374
75	0.9958	0.9913	0.9427	0.9113	0.9958	0.9913	0.9749	0.9647	0.0197	0.0319
100	0.9968	0.9915	0.9552	0.9165	0.9968	0.9915	0.9810	0.9667	0.0160	0.0315
125	0.9990	0.9921	0.9801	0.9234	0.9990	0.9921	0.9918	0.9703	0.0096	0.0321
150	0.9993	0.9931	0.9847	0.9334	0.9993	0.9931	0.9936	0.9744	0.0079	0.0282

During the classification phase, the segmented overlaid image is categorized into a particular disease class. In our modern society, technology plays a crucial role in our daily lives. In this research, pre-trained models were selected based on their top classification accuracy. Moreover, these models were chosen to examine the impact of varying depths and parameter counts on the accuracy of classifying segmented images. Once these images have been pre-processed, they are then transferred to the training phase. After the images are pre-processed, they are then fed as an input to the training phase. Utilizing a pre-existing base model in VGG-RefineNet allows for leveraging the advantages of learned features without the need to start training from the beginning. The custom layers in the VGG-RefineNet model, such as flatten, dense, batch normalization, and dropout, enhance the base model's feature extraction abilities to learn specific patterns for tasks like disease classification. Fig. 16 provides detailed values for recall, precision and F1-score. It is evident that the proposed classification method gives better precision for black spot and greening, while the recall is better for canker and healthy. Additionally, the F1-score improves across all diseases. Furthermore, Fig. 17 illustrates the confusion matrix for the proposed CNN model. It is observed that there are four misclassifications on black spot and five misclassifications in greening, while there is no misclassification on canker and healthy.

The percentage of severity on the citrus leaf is determined by highlighting and measuring the disease affected area. This HDAP method starts with the original leaf image, applies Otsu's method to separate the leaf region from the background,

then obtains the contour and overlays it onto the leaf region. Determine the leaf region's area by measuring the contour's area. Next, utilize the segmentation model to forecast the disease mask and superimpose it onto the initial image with a different colour. Subsequently, identify the contour, sketch the outline for the affected part, and determine the area of the affected part by analysing the overlaid area, which includes the count of non-zero pixels representing the diseased portion. Following that, determine the percentage of citrus disease affected by comparing the disease area to the total area of citrus leaves. In conclusion, the output image displayed the classified result and the percentage of citrus disease affected, as seen in Table IV. Next, the citrus leaf image samples processed using the HDAP algorithm are displayed in Fig. 7 and Fig. 8, while the entire process is illustrated in Fig. 9.

By utilizing the segmentation model as proposed, the training model achieved a success rate with an IOU of 0.9847, mean IOU of 0.9846, weighted mean IOU of 0.9874, pixel accuracy of 0.9993, mean pixel accuracy of 0.9993, dice coefficient of 0.9936, mean boundary f1 score of 0.9557, precision of 0.9974, recall of 0.9975, f1 score of 0.9974, specificity of 0.9996, loss of 0.0079, and accuracy of 0.9993. The Validation model achieved success with the following metrics: intersection over union (IOU) of 0.9334, mean IOU of 0.9334, weighted mean IOU of 0.9517, pixel accuracy of 0.993, mean pixel accuracy of 0.9931, dice coefficient of 0.9744, mean boundary F1 score of 0.9504, precision of 0.9778, recall of 0.9769, F1 score of 0.9773, specificity of 0.9959, loss of 0.0282, and accuracy of 0.9931.

	precision	recall	f1-score	support
blackspot_a	1.00	0.97	0.98	118
canker_a	0.96	1.00	0.98	112
greening_a	0.99	0.97	0.98	148
healthy_a	0.95	1.00	0.97	38
accuracy			0.98	416
macro avg	0.97	0.98	0.98	416
weighted avg	0.98	0.98	0.98	416

Fig. 16. Recall, precision, F1-Score and accuracy.

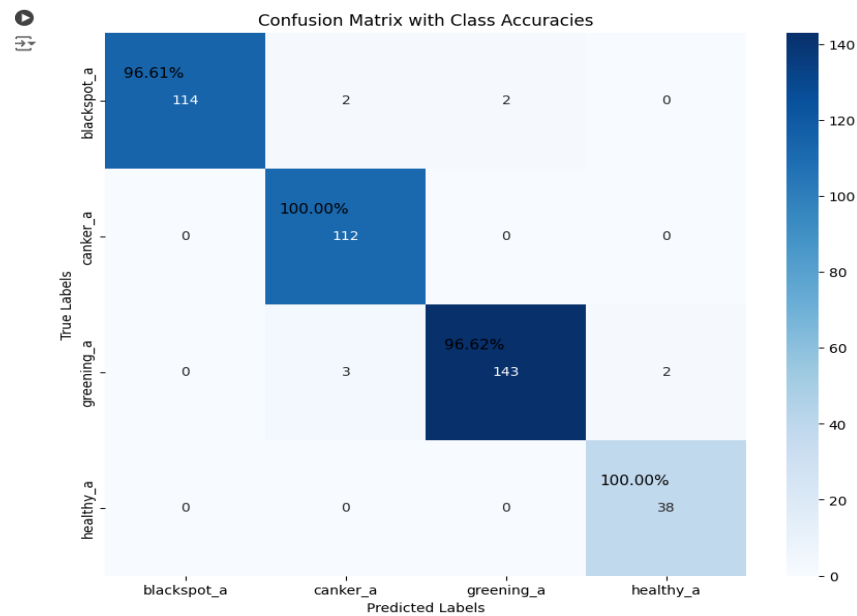


Fig. 17. Confusion matrix of the proposed VGG-RefineNet.

VI. CONCLUSION

This research introduces a new deep learning model using convolution, leaky relu, and transfer learning to detect, segment, highlight, and classify citrus canker, blackspot, and citrus greening diseases found on citrus leaves, a topic not covered in existing literature. The method being suggested involves four different stages. Initially, the unhealthy region is determined using HSV preprocessing, and the mask is applied to the original image. Following this, standard image preprocessing techniques are performed to improve contrast, brightness, scale, and rotation. In the next phase, DeepOverlay L-UNet is used to extract additional features from the citrus image by enabling slight gradients in the negative inputs with the introduction of Leaky Relu on the perimeter of each convolution block in the encoder block. This aids in acquiring complex features for a more precise and accurate segmentation of disease areas. During the third phase, feature extraction and classification are carried out using transfer learning techniques. The proposed VGG-RefineNet design shortens training time by leveraging a pre-trained base model to extract features without starting training from scratch. During stage four, the HDAP method calculates the affected disease area by obtaining the leaf area through contour functions and determining the area of the affected region by counting the number of pixels in the overlay image that is not zero. Next, once these values are identified, the next step is to calculate the ratio of the impacted area to the entire leaf area, representing it as a percentage. The limitation of the proposed work is after the segmentation process the larger area of the region of interest is considered for classification purpose, further it can be extended for the smaller regions.

Based on the IOU measures, the proposed method outperforms the existing segmentation algorithm to extract the affected regions. Further, the classification model achieved a maximum success rate for blackspot, citrus canker, citrus

greening, and healthy class in the citrus leaves dataset, with a 98% success rate for overall citrus plant disease.

REFERENCES

- [1] H. Çetiner, "Citrus disease detection and classification using based on convolution deep neural network," *Microprocess Microsyst*, vol. 95, Nov. 2022, doi: 10.1016/j.micpro.2022.104687.
- [2] R. Satya Rajendra Singh and R. K. Sanodiya, "Zero-Shot Transfer Learning Framework for Plant Leaf Disease Classification," *IEEE Access*, vol. 11, pp. 143861–143880, 2023, doi: 10.1109/ACCESS.2023.3343759.
- [3] N. G. Rezk, E. E. D. Hemdan, A. F. Attia, A. El-Sayed, and M. A. El-Rashidy, "An efficient IoT based framework for detecting rice disease in smart farming system," *Multimed Tools Appl*, vol. 82, no. 29, pp. 45259–45292, Dec. 2023, doi: 10.1007/s11042-023-15470-2.
- [4] S. F. Syed-Ab-Rahman, M. H. Hesamian, and M. Prasad, "Citrus disease detection and classification using end-to-end anchor-based deep learning model," *Applied Intelligence*, vol. 52, no. 1, pp. 927–938, Jan. 2022, doi: 10.1007/s10489-021-02452-w.
- [5] Blake Bextine and George G. Kennedy, *A Review of the Citrus Greening Research and Development Efforts Supported by the Citrus Research and Development Foundation*. Washington, D.C.: National Academies Press, 2018. doi: 10.17226/25026.
- [6] S. A. De Carvalho et al., "Comparison of resistance to asiatic citrus canker among different genotypes of citrus in a long-term canker-resistance field screening experiment in Brazil," *Plant Dis*, vol. 99, no. 2, pp. 207–218, 2015, doi: 10.1094/PDIS-04-14-0384-RE.
- [7] V. Guarnaccia et al., "Phyllosticta citricarpa and sister species of global importance to Citrus," *Mol Plant Pathol*, vol. 20, no. 12, pp. 1619–1635, Dec. 2019, doi: 10.1111/mpp.12861.
- [8] Y. H. Wang, J. J. Li, and W. H. Su, "An Integrated Multi-Model Fusion System for Automatically Diagnosing the Severity of Wheat Fusarium Head Blight," *Agriculture (Switzerland)*, vol. 13, no. 7, Jul. 2023, doi: 10.3390/agriculture13071381.
- [9] B. Sai Reddy and S. Neeraja, "Plant leaf disease classification and damage detection system using deep learning models," *Multimed Tools Appl*, vol. 81, no. 17, pp. 24021–24040, Jul. 2022, doi: 10.1007/s11042-022-12147-0.
- [10] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Information Processing in Agriculture*, vol. 4, no. 1, pp. 41–49, Mar. 2017, doi: 10.1016/j.inpa.2016.10.005.

- [11] M. Ji and Z. Wu, "Automatic detection and severity analysis of grape black measles disease based on deep learning and fuzzy logic," *Comput Electron Agric*, vol. 193, Feb. 2022, doi: 10.1016/j.compag.2022.106718.
- [12] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Information Processing in Agriculture*, vol. 7, no. 4, pp. 566–574, Dec. 2020, doi: 10.1016/j.inpa.2019.11.001.
- [13] S. Abinaya, K. U. Kumar, and A. S. Alphonse, "Cascading Autoencoder With Attention Residual U-Net for Multi-Class Plant Leaf Disease Segmentation and Classification," *IEEE Access*, vol. 11, pp. 98153–98170, 2023, doi: 10.1109/ACCESS.2023.3312718.
- [14] C. Madhurya and E. A. Jubilson, "YR2S: Efficient Deep Learning Technique for Detecting and Classifying Plant Leaf Diseases," *IEEE Access*, vol. 12, pp. 3790–3804, 2024, doi: 10.1109/ACCESS.2023.3343450.
- [15] P. Kaur, S. Harnal, V. Gautam, M. P. Singh, and S. P. Singh, "Performance analysis of segmentation models to detect leaf diseases in tomato plant," *Multimed Tools Appl*, vol. 83, no. 6, pp. 16019–16043, Feb. 2024, doi: 10.1007/s11042-023-16238-4.
- [16] P. Kaur, S. Harnal, V. Gautam, M. P. Singh, and S. P. Singh, "Hybrid deep learning model for multi biotic lesions detection in solanum lycopersicum leaves," *Multimed Tools Appl*, vol. 83, no. 3, pp. 7847–7871, Jan. 2024, doi: 10.1007/s11042-023-15940-7.
- [17] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," *Environ Monit Assess*, vol. 195, no. 1, Jan. 2023, doi: 10.1007/s10661-022-10656-x.
- [18] J. Deng et al., "Applying convolutional neural networks for detecting wheat stripe rust transmission centers under complex field conditions using RGB-based high spatial resolution images from UAVs," *Comput Electron Agric*, vol. 200, Sep. 2022, doi: 10.1016/j.compag.2022.107211.
- [19] M. S. Islam et al., "Multimodal Hybrid Deep Learning Approach to Detect Tomato Leaf Disease Using Attention Based Dilated Convolution Feature Extractor with Logistic Regression Classification," *Sensors*, vol. 22, no. 16, Aug. 2022, doi: 10.3390/s22166079.
- [20] S. Allaoua Chelloug, R. Alkanhel, M. S. A. Muthanna, A. Aziz, and A. Muthanna, "MULTINET: A Multi-Agent DRL and EfficientNet Assisted Framework for 3D Plant Leaf Disease Identification and Severity Quantification," *IEEE Access*, vol. 11, pp. 86770–86789, 2023, doi: 10.1109/ACCESS.2023.3303868.
- [21] H. Anwar et al., "The NWRD Dataset: An Open-Source Annotated Segmentation Dataset of Diseased Wheat Crop," *Sensors*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156942.
- [22] R. Kursun, K. K. Bastas, and M. Koklu, "Segmentation of dry bean (*Phaseolus vulgaris* L.) leaf disease images with U-Net and classification using deep learning algorithms," *European Food Research and Technology*, vol. 249, no. 10, pp. 2543–2558, Oct. 2023, doi: 10.1007/s00217-023-04319-5.
- [23] J. Wu et al., "DS-DETR: A Model for Tomato Leaf Disease Segmentation and Damage Evaluation," *Agronomy*, vol. 12, no. 9, Sep. 2022, doi: 10.3390/agronomy12092023.
- [24] P. Dhiman et al., "A Novel Deep Learning Model for Detection of Severity Level of the Disease in Citrus Fruits," *Electronics (Switzerland)*, vol. 11, no. 3, Feb. 2022, doi: 10.3390/electronics11030495.
- [25] V. Gautam, R. K. Ranjan, P. Dahiya, and A. Kumar, "ESDNN: A novel ensemble stack deep neural network for mango leaf disease classification and detection," *Multimed Tools Appl*, vol. 83, no. 4, pp. 10989–11015, Jan. 2024, doi: 10.1007/s11042-023-16012-6.
- [26] M. A. Latif et al., "Enhanced Classification of Coffee Leaf Biotic Stress by Synergizing Feature Concatenation and Dimensionality Reduction," *IEEE Access*, vol. 11, pp. 100887–100906, 2023, doi: 10.1109/ACCESS.2023.3314590.
- [27] A. Pal and V. Kumar, "AgriDet: Plant Leaf Disease severity classification using agriculture detection framework," *Eng Appl Artif Intell*, vol. 119, Mar. 2023, doi: 10.1016/j.engappai.2022.105754.
- [28] T. Tamilvizhi, R. Surendran, K. Anbazhagan, and K. Rajkumar, "Quantum Behaved Particle Swarm Optimization-Based Deep Transfer Learning Model for Sugarcane Leaf Disease Detection and Classification," *Math Probl Eng*, vol. 2022, 2022, doi: 10.1155/2022/3452413.
- [29] A. S. Almasoud et al., "Artificial Intelligence-Based Fusion Model for Paddy Leaf Disease Detection and Classification," *Computers, Materials and Continua*, vol. 72, no. 1, pp. 1391–1407, 2022, doi: 10.32604/cmc.2022.024618.
- [30] E. B. Milke, M. T. Gebiremariam, and A. O. Salau, "Development of a coffee wilt disease identification model using deep learning," *Inform Med Unlocked*, vol. 42, Jan. 2023, doi: 10.1016/j.imu.2023.101344.
- [31] M. E. H. Chowdhury et al., "Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques," *AgriEngineering*, vol. 3, no. 2, pp. 294–312, Jun. 2021, doi: 10.3390/agriengineering3020020.
- [32] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Brief*, vol. 26, Oct. 2019, doi: 10.1016/j.dib.2019.104340.
- [33] T. H. Nguyen, T. N. Nguyen, and B. V. Ngo, "A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, Dec. 2022, doi: 10.3390/agriengineering4040056.