IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 September 2024, accepted 4 October 2024, date of publication 8 October 2024, date of current version 21 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3476211

RESEARCH ARTICLE

SUNet: Coffee Leaf Disease Detection Using Hybrid Deep Learning Model

DEEPAK THAKUR[1], TANYA GERA[1], AMBIKA AGGARWAL[2], MADHUSHI VERMA[3], MANJIT KAUR[4], (Senior Member, IEEE), DILBAG SINGH[5], (Senior Member, IEEE), AND MOHAMMED AMOON[6]

[1]Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab 140401, India
[2]Department of Computer Science and Engineering, UPES, Dehradun 248007, India
[3]School of Computer Science Engineering and Technology, Bennett University, Greater Noida 201310, India
[4]School of Computer Science and Artificial Intelligence, SR University, Warangal, Telangana 506371, India
[5]Research and Development Cell, Lovely Professional University, Phagwara, Punjab 144411, India
[6]Department of Computer Science, Community College, King Saud University, Riyadh 11437, Saudi Arabia

Corresponding author: Manjit Kaur (manjit@ieee.org)

This work was supported by the Researchers Supporting Project number (RSPD2024R968), King Saud University Riyadh, Saudi Arabia.

ABSTRACT Leaf mining, rust, bacterial blight, and berry pathology are major diseases in coffee plants. These diseases not only reduce yield but also affect quality. Early detection and targeted treatment are crucial to mitigate their effects. This paper introduces an efficient hybrid deep learning model, SUNet, for prediction and classification of healthy and diseased coffee leaves. SUNet integrates U-Net with Segnet's encoding system, using VGG16 for robust feature extraction. A decoder with skip connections is used to preserve spatial details. Mask R-CNN is also employed for instance segmentation, accurately localizing disease spots. A pyramid pooling module captures multi-scale contextual information. The model is tested using two benchmark datasets, JMuBEN and JMuBEN2. These datasets contain a wide range of coffee leaves affected by phoma, cercospora, or rust, along with healthy samples. SUNet achieved significant performance improvement over other models in terms of accuracy, Intersection over Union (IoU), F1-score, precision, and recall by 1.22%, 1.21%, 1.17%, 1.19%, and 1.24%, respectively. These improvements demonstrate that SUNet can be used for the early detection and classification of coffee leaf diseases. Therefore, with precise and timely interventions, SUNet can help farmers minimize crop losses, enhance coffee production quality, and reduce reliance on harmful chemical treatments.

INDEX TERMS Agricultural applications, coffee leaf disease, deep learning, semantic segmentation, SUNet, SegNet, U-Net.

I. INTRODUCTION

Coffee is consumed on a large scale in the world, and its production contributes greatly to the economic development of many countries, along with supporting millions of small-scale farmers around the globe. But the coffee industry is under siege by pests and diseases, market instability, and climate change, all of which ultimately impact the quality and productivity of coffee crops. The diseases that seriously affect coffee production are phoma, cercospora, and rust, which cause significant reductions in the quality and yield of this crop [1]. Coffee leaf diseases must be detected early and kept under control to mitigate their deleterious effects. The detection of these diseases was carried out by the traditional manual inspection, which is time-consuming and labor-intensive, leading to human error. These methods are also labor-intensive and require a lot of training and knowledge, making it extremely difficult for smaller farmers or those in very remote locations. Additionally, the current diagnostics are challenged in dealing with the early stages of diseases or distinguishing between diseases that present similar symptoms and signs at their initial stages [2]. The breakthroughs in deep learning promise to address the problem of automatic detection and recognition of plant

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

diseases. One of the most important fields where deep learning models (particularly convolutional neural networks (CNNs)) are thought to be useful is image and object detection. Such models can be trained to detect data within images and classify them according to patterns evident in the underlying characteristics. Deep learning has been shown to be effective in detecting diseases of crops such as potatoes, tomatoes, and grapes [3]. Applying these methods to coffee leaf disease detection remains a relatively new area of research.

In this paper, a hybrid deep learning model (named SUNet) is proposed for the detection and classification of healthy leaves vs. diseased leaves in coffee plants. SUNet is designed as a combination of the strengths of U-Net and SegNet, which uses a VGG-16-based encoder for effective feature extraction with a lightweight decoder like in U-Net style, along with skip connections to keep fine-grained spatial details alive, yet narrows down its field-of-view towards small regions [4]. SUNet also leverages Mask R-CNN to perform instance segmentation, enabling accurate detection of disease regions. We have used both synthesized JMuBEN and JMuBEN2 high-quality datasets [5], with plenty of images for every coffee leaf case in terms of healthiness or the disease stage (Phoma, Cercospora, and Rust).

The key contributions of the paper are:

- The proposed SUNet offers an efficient deep learning methodology for predicting and classifying diseased and healthy coffee leaves.
- SUNet combines U-Net and SegNet segmentation methods to enhance the precision of segmenting disease-affected regions.
- The model is fine-tuned through modifications to its architecture's layers, filters, and activation functions for optimal performance in coffee leaf disease classification.
- Comparative analysis shows the superior performance of SUNet against other deep learning models, using benchmark datasets for disease detection and classification.

This paper is organized into several sections. Section II discusses work related to coffee leaf disease and machine learning-based image segmentation. Section III presents the proposed methodology. Section IV demonstrates the results and discussion. Section V concludes the paper, summarizes the limitations, and suggests directions for further research.

II. RELATED WORK

Recently, many deep learning-based techniques have been implemented by researchers to detect diseases in plants. For example, Thakur and Sheory utilized the VGG-ICNN model to identify plant diseases. The initial four layers of the VGG-ICNN model consist of VGG-16 convolutional layers, while the last convolutional layers are from Inception v7. VGG-ICNN efficiently detected diseases across 93 categories of disease classes and achieved 99.16% accuracy [6]. Chen and Pan detected apple leaf diseases using color, shape, and texture features through various extraction methods. They classified the diseases using algorithms like cycle-consistent generative adversarial network (CycleGAN) and ResNet. Their CNN-based approach achieved 97.78% accuracy in automatically identifying plant leaf diseases [7].

Jiang et al. assessed the performance of seven popular CNNs (ResNet-50, Inception-v3, VGG-16, DenseNet-121, ShuffleNet-v2, MobileNetV3, and EfficientNet-B6) for detecting three major wheat leaf diseases: stripe rust, leaf rust, and powdery mildew using field images. To complement the public 'PlantVillage' collection, which features individual leaves scanned under controlled conditions, they developed the field-based wheat diseases images (FWDI) dataset. All CNN models achieved maximum accuracy using a transfer learning technique that involved fine-tuning all parameters. Inception-v3 demonstrated the highest identification accuracy at 92.5% on the test dataset [8].

Khan and Narvekar developed a methodology to segment diseased lesions in leaf images. Their approach involved color balancing and superpixel segmentation. They applied thresholding with Histogram of Gradients (HOG) and color channels to isolate the leaf from the background. K-means clustering was used to identify diseased areas. They combined Pyramid of HOG (PHOG) with Gray Level Co-occurrence Matrix (GLCM) features for detailed representation. Random Forest (RF) was utilized for classification. The method achieved an accuracy of 93.12% [9]. Rangarajan et al. diagnosed leaf diseases in tomato crops at an early stage using AlexNet and VGG16 architectures. They analyzed the performance of both models based on learning rate, batch size, number of image datasets, weights, and bias. The accuracy achieved using 13,262 images was 97.49% for AlexNet and 97.29% for VGG16. AlexNet provided better accuracy compared to VGG16 in terms of computational load [10].

Karthik et al. proposed an inception-based global context attention network (IGCAN) to detect coffee leaf diseases. IGCAN leveraged inception modules, a global context block, and a multi-head attention module. These components enhanced feature extraction and achieved high classification accuracy. IGCAN outperformed existing models, achieving an accuracy of 98.57% [11]. Salamai et al. introduced a dual-path visual transformer network (DVTN) for automating coffee leaf disease diagnosis. It integrated lesion segmentation and classification paths for real-time disease diagnosis. DVTN featured a novel visual transformer block called COFFORMER. This block enhanced the efficiency of token mixing by replacing traditional attention mechanisms with a multiscale convolutional pooling operator [12].

Marin et al. developed a UAV-based framework that used vegetation indices and decision tree models, particularly Logistic Model Tree (LMT), to detect and assess the severity of coffee leaf rust (CLR). The method involved capturing multispectral images and extracting vegetation indices. These indices were then used with machine learning to classify CLR severity levels. This approach proved effective for early

detection and precision agriculture [13]. Tassis et al. proposed a deep learning framework for identifying diseases and pests in coffee leaves using in-field images. It used instance segmentation with Mask R-CNN, semantic segmentation with U-Net and PSPNet, and classification with ResNet. This combination enabled accurate detection of coffee leaves diseases [14]. Bordin Yamashita and Leite proposed a system for coffee disease classification using deep learning. The system was integrated into a low-cost microcontrolled board. Their approach aimed to provide in-loco disease diagnosis without needing an internet connection. The proposed architectures included both cascade and single-stage models. These models achieved accuracy values of approximately 98% and 96%, respectively [15].

Nawaz et al. introduced CoffeeNet, a deep learning model designed for recognizing coffee plant leaf diseases. CoffeeNet used a spatial-channel attention strategy based on ResNet-50. CoffeeNet achieved a classification accuracy of 98.54% and a mean Average Precision (mAP) of 0.97 [16]. Abuhayi and Mossa developed a deep learning method for coffee disease classification using CNNs. Gaussian filtering and data augmentation were applied to improve model robustness. CNN model combined GoogLeNet and RESNET architectures for feature extraction. Multi-Layer Perceptrons (MLPs) were used for classification. Their model achieved a testing accuracy of 99.08% [17].

From the literature, it is evident that recent deep learning techniques—ranging from combining different neural network architectures to integrating novel methods like UAV-based imaging and transformer networks—have demonstrated promising results in detecting coffee leaf diseases. However, challenges remain, including adapting models to diverse environmental conditions and addressing hardware limitations. There are also difficulties in deploying these technologies in real-world settings, where factors such as lighting variations and background clutter can impact performance.

III. METHODOLOGY

This section outlines the data preprocessing steps and the development of the proposed SUNet. The diagrammatic flow of the proposed model is illustrated in Figure 1. The process begins with the input of a coffee leaf image, followed by preprocessing and instance segmentation using Mask R-CNN. Subsequently, SegNet, U-Net, and PSPNet are applied for semantic segmentation. Two different classifiers, ResNet152V2 and VGG19, are used to classify the coffee leaf disease. Table 1 provides a simplified explanation of the proposed methodology.

A. DATASETS DESCRIPTION AND CLEANING

In this paper, we used the JMuBEN and JMuBEN2 datasets in our study (see [5] for more details). These datasets contain high-quality images of coffee leaves captured from various coffee plantations. The images contain both healthy leaves and those affected by common diseases such as Phoma, Cercospora, Rust, and Miner. Initially, data cleaning is performed as follows:

- Quality Check: Initially, we removed low-quality images, i.e., images which were blurry, underexposed, or overexposed. To ensure the model's ability to accurately recognize disease patterns, this step focuses on training the model with high-quality images. A total of 450 such images were excluded from the training dataset.
- Removal of Unwanted Pixels or Objects: We eliminated non-leaf elements such as stems, non-leaf parts of coffee plants, and background noise. This step ensures that only relevant features and patterns from the infected areas of coffee leaves are included. As a result, 400 images were removed to focus on the critical regions.
- Elimination of Duplicate Images: To prevent overfitting and biased learning, we removed repeated or duplicated images. This important step allows the model to learn from a diverse set of images, enhancing its generalization capabilities. We identified and removed 430 duplicate images.
- Annotation Consistency: We verified that all images in the dataset had accurate annotations, correctly indicating the disease on the coffee leaf. Any inconsistencies or errors were corrected to ensure the model was trained on reliable ground truth data. No additional images were removed during this process.

During the preprocessing phase, a total of 1,280 images were removed, leaving 57,275 images in the dataset. These 57,275 images were then used for model training and performance evaluation.

B. DATA AUGMENTATION

Data augmentation increases the number of images in the dataset by applying various operations to the original images. This step is crucial for enhancing the deep learning model's ability to recognize characteristics and patterns in the images. The augmentation techniques used include zooming, scaling, rotating, and translating the images, which effectively expand the dataset [18]. As a result, the model becomes more robust to minor modifications in input data, allowing it to learn more refined features from the augmented image set.

Here are the data augmentation strategies we utilize with our dataset:

- **Rotation:** We randomly rotate images by 90, 180, and 270 degrees, generating three new images for each original image.
- **Horizontal and Vertical Flipping:** We create flipped versions of the images by mirroring them horizontally and vertically, resulting in two additional images for each original image.
- **Scaling:** We randomly scale the images by factors of 0.8 and 1.2, producing two more images for each original image.

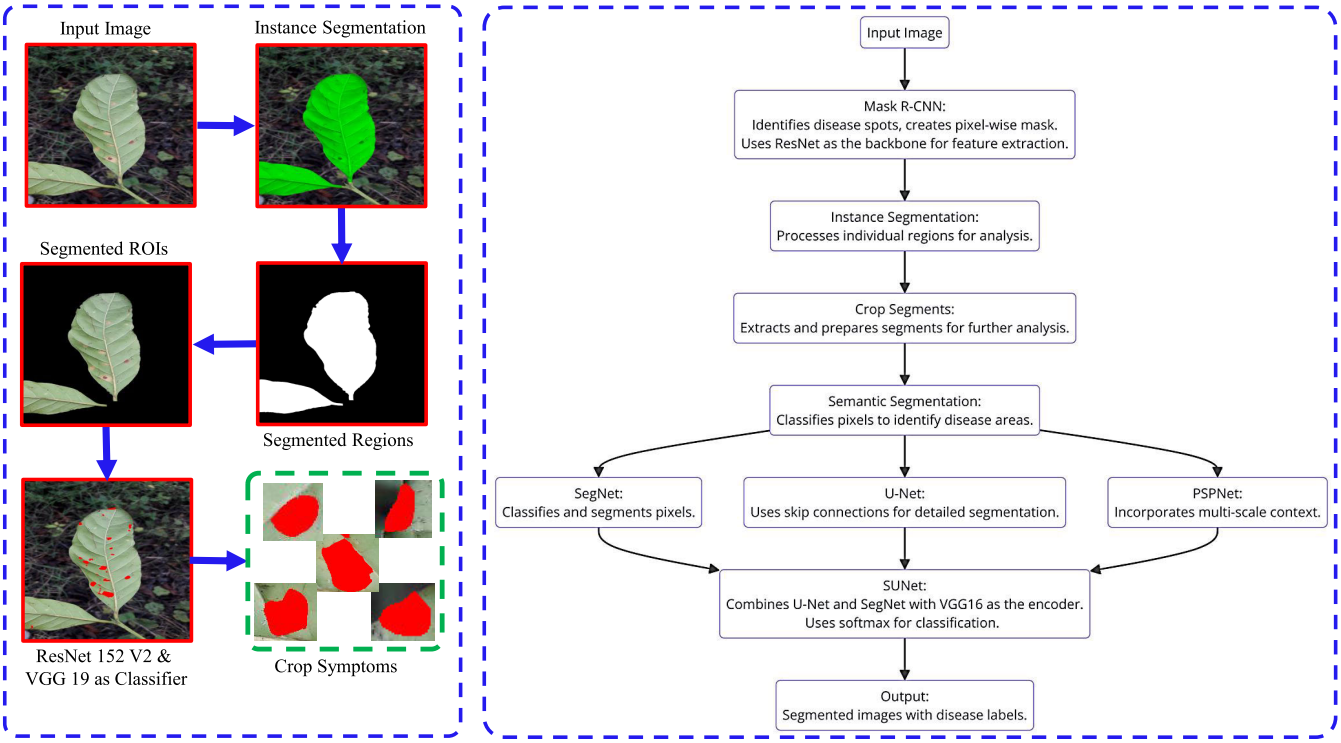


FIGURE 1. **Diagrammatic flow of the proposed model.**

- **Translation:** We translate the images by shifting them both by 10% horizontally and vertically, creating two new images for each original image.

Through data augmentation, nine new images are generated from each cleaned image, resulting in a total of 515,475 images from the pre-processing method (57275 × 9). These augmented images are combined with the original dataset, leading to a total of 572,750 images available for training and testing the model. Algorithm 1 outlines the steps for augmenting the coffee leaf images.

C. COFFEE LEAF DISEASE DETECTION USING MASK R-CNN-BASED INSTANCE SEGMENTATION

To perform segmentation, a Mask R-CNN is utilized. This deep learning model extends faster R-CNN by incorporating an additional parallel branch specifically for predicting segmentation masks. This enhancement allows Mask R-CNN to combine object detection and semantic segmentation tasks effectively [19]. Mask R-CNN operates in two stages. The first stage generates region proposals and the second stage classifies these regions using bounding boxes. A detailed explanation of these stages is provided in the following steps.

i. Region Proposal Stage: Initially, Mask R-CNN generates a set of potential bounding boxes for objects in the input image using a Region Proposal Network (RPN). The entire image is scanned with a sliding window method and RPN computes anchor boxes at each point. These anchor boxes vary in size and aspect ratio to accommodate different image dimensions. The objectness score is used to quantify the probability that an anchor box contains an object.

For training, the RPN assigns a low objectness score to anchor boxes with no overlap with ground truth objects and a high score to those that overlap with them. Mask R-CNN examines the bounding boxes generated using RPN. ResNet or VGG are then used to extract features from the image denoted by $F(I)$. Here, I represents the coffee leaf image. RPN processes the feature map to produce a set of candidate bounding boxes as $B = \{b_1, b_2, \ldots, b_n\}$, where n is the number of boxes which contains objects. For each bounding box b_i, the RPN calculates an objectness score s_i as:

$$s_i = \mathrm{P}\,(\mathrm{obj} \mid b_i) \tag{1}$$

where $P(obj|\ b_i)$ is the probability of the bounding box b_i containing a coffee leaf disease instance.

ii. Bounding Box Classification: In this stage, the model evaluates candidate bounding boxes obtained from the region proposal phase and classifies them into various object categories while refining their dimensions. Each candidate bounding box is processed to extract fixed-size feature maps using a Region-of-Interest (RoI) pooling layer. Thereafter, the classification head assigns each bounding box to a specific object class by generating class probabilities. Concurrently, the bounding box regression head adjusts the dimensions of the bounding boxes by regressing their coordinates towards the ground truth coordinates of the corresponding object.

The model classifies all potential bounding boxes b_i into one of $K + 1$ categories (where K represents the coffee leaf

TABLE 1. Simplified explanation of the methodology.

Step	Description
Data Collection and Preparation	**Datasets Used:** The study uses two datasets [5], JMuBEN and JMuBEN2, which contain images of both healthy coffee leaves and leaves affected by diseases such as Phoma, Cercospora, and Rust. **Image Quality Check:** The images are manually inspected to remove those of poor quality (e.g., blurry or poorly lit). **Removing Unwanted Parts:** Unwanted parts of the images, such as stems or backgrounds, are removed to focus only on the leaves. **Eliminating Duplicates:** Duplicate images are removed to prevent biased learning. **Annotation Consistency:** Annotations are checked and corrected to ensure they accurately represent the diseased areas.
Data Augmentation	**Purpose:** Data augmentation increases the number of images in the dataset by making various modifications to the original images. **Techniques Used:** The images are rotated, flipped, scaled, and translated to create new images. This helps the model learn better and become more resistant to minor changes in the data.
Image Segmentation Using Mask R-CNN	**What is Mask R-CNN?:** Mask R-CNN is a deep learning model that can detect and segment different objects in an image. **How it Works:** **Region Proposal:** The model scans the image to find potential areas (bounding boxes) where diseases might be present. **Bounding Box Classification:** The model then classifies these areas into different classes (types of diseases). **Mask Prediction:** Finally, it creates a detailed mask (segmentation) for each detected disease area.
Semantic Segmentation Using SegNet and U-Net	**SegNet:** **Encoder-Decoder Architecture:** SegNet uses an encoder to extract features from the image and a decoder to create the segmentation mask. **Purpose:** It helps in identifying and classifying pixels in the image to detect diseases. **U-Net:** **Similar to SegNet but with skip connections that help retain detailed information.** **Purpose:** It also segments the image to identify diseased areas.
Semantic Segmentation Using PSPNet	**What is PSPNet?:** PSPNet is another segmentation model that captures information at multiple scales using pyramid pooling. **Purpose:** It helps detect diseased areas even if they are of different sizes.
Proposed SUNet Hybrid Model	**Combining SegNet and U-Net:** SUNet integrates the strengths of SegNet and U-Net to create a more powerful model. **How it Works:** **Encoder:** Uses the VGG16 architecture from SegNet to extract features. **Decoder:** Uses a U-Net style decoder to create segmentation masks. **Skip Connections:** Connect the encoder and decoder layers to retain detailed information. **Output Layer:** Uses a Softmax function to classify each pixel into different disease categories.
Training and Evaluation	**Loss Function:** A cross-entropy loss function is used to train the model by comparing the predicted masks with the actual masks. **Performance Metrics:** The model's performance is evaluated using metrics like Intersection over Union (IoU), F1 Score, Precision, and Recall.
Hardware and Software Requirements	**Software:** Uses TensorFlow for building and training the model, OpenCV and NumPy for data preprocessing, and Scikit-learn for performance metrics. **Hardware:** Requires a high-performance GPU (NVIDIA Tesla V100), a powerful CPU (Intel Xeon), 128 GB RAM, and 2 TB SSD storage to handle the large dataset and complex computations.

disease classes and one background class). Let $C(b_i)$ denotes a $K+1$ dimensional vector representing the classification outcome for the bounding box b_i:

$$C(b_i) = [P_0, P_1, \ldots, P_K] \tag{2}$$

where P_j is the probability that bounding box b_i belongs to class j (0 for background, 1 to K for the coffee leaf disease classes). Additionally, the model refines the bounding box dimensions using a regression function $R(b_i) = [\Delta x, \Delta y, \Delta w, \Delta h]$. Here, Δx, Δy, Δw, and Δh are the offsets for the x and y coordinates, width, and height, respectively.

Mask R-CNN introduces a new parallel branch called the mask prediction branch, which generates binary masks for every object instance. Combined with the bounding box classification stage, this sub-stage takes as input the fixed-size feature maps retrieved by RoI pooling layer. To generate a binary mask for each item class, the mask prediction branch employs a pixel-wise sigmoid activation function after a series of convolutional layers (see [20] for more deatils). During training, the model learns to generate masks for every object detected in the segmentation process using ground truth as a reference. The model's loss function includes mask loss, optimizing bounding box, class, and mask predictions.

The model predicts a binary mask $M(b_i)$ of size $H \times W$ for each bounding box b_i identified as a coffee leaf disease (i.e., $C(b_i)$ corresponds to one of the K disease classes). Here, H and W represent the mask's height and width, respectively. The mask $M(b_i)$ contains pixels with values of 0 (background) or 1 (disease instance).

To enhance image recognition and classification, Mask R-CNN is effective, so we are combining three segmentation methods to achieve precise segmentation for each image. Hence, this combination will achieve better results even with complex backgrounds and overlapping objects in the RoI, and can distinguish objects with similar patterns. Instance segmentation with Mask R-CNN is detailed in Algorithm 2. Table 2 outlines the steps Mask R-CNN follows to recognize and classify diseased coffee leaves including inputs, outputs, hyperparameters, and individual

TABLE 2. Mask R-CNN model's stages and components for identifying and classifying coffee leaf diseases, including input, output, and parameters/hyperparameters for each stage.

Component / Stage	Description	Input	Output	Parameters / Hyperparameters
Input Image (I)	A coffee leaf image used as input for the model.	-	I	Image dimensions, color channels
Feature Map ($F(I)$)	The output of a CNN backbone (e.g., ResNet, VGG) applied to the input image, capturing high-level features.	I	$F(I)$	CNN architecture, weights
RPN	A network that processes the feature map to generate candidate bounding boxes (B) for potential objects in the image.	$F(I)$	B, s_i	RPN architecture, anchor scales/ratios
Objectness Score (s_i)	The probability of each bounding box (b_i) containing a coffee leaf disease instance.	B	s_i	Threshold for selecting bounding boxes
Bounding Box Classification ($C(b_i)$)	A $K+1$ dimensional vector representing the classification probabilities for each bounding box (b_i) across K disease classes and a background class.	B, s_i	C(b_i)	Number of disease classes (K)
Bounding Box Regression ($R(b_i)$)	Refines the dimensions of each bounding box (b_i) using offsets for the x and y coordinates, width, and height.	B, $C(b_i)$	$R(b_i)$	Regression loss function, weights
Mask Prediction ($M(b_i)$)	A binary mask of size $H \times W$ for each bounding box (b_i) classified as a coffee leaf disease, where the mask's height and width are H and W.	$R(b_i)$, $C(b_i)$	$M(b_i)$	Mask dimensions (H, W), mask loss function

TABLE 3. Configuration of Mask R-CNN.

Aspect	Description
Model Architecture	Mask R-CNN
Backbone Network	ResNet-50 or ResNet-101
RPN	3 anchor scales, 3 aspect ratios
Activation Functions	ReLU for the backbone, Softmax for classification, Sigmoid for mask prediction
Loss Function	Classification Loss + BBox Regression Loss + Mask Loss
Performance Metrics	mAP, IoU, F1 Score, Precision, Recall
Training Set	458,200 images (80%)
Validation Set	57,275 images (10%)
Test Set	57,275 images (10%)

components. The configuration of Mask R-CNN is shown in Table 3.

D. SEMANTIC SEGMENTATION OF COFFEE LEAF DISEASES USING SEGNET

In this section, we apply semantic segmentation to classify diseases on coffee leaves using SegNet. Semantic segmentation is a technique for identifying and classifying individual pixels in an image, allowing for the precise detection of multiple coffee leaf diseases. SegNet has an ability to effectively capture contextual data both local and global levels. It employs an encoder-decoder architecture in which the encoder extracts high-level feature representations from the input images to obtain segmentation masks. In this paper, VGG-16 is used to capture spatial information through max-pooling indices. These indices are then used by the decoder to obtain segmentation mask using a process called unpooling. This approach preserves the spatial information of the network while reducing computational complexity. The encoder function E transforms the input image I into a set of feature maps F_m as follows:

$$\mathrm{E(I)} = \mathrm{F_m} \tag{3}$$

The decoder can be expressed as a function D that maps the feature maps F and the max-pooling indices M to a segmentation mask S. It can be represented as:

$$\mathrm{D(F, M)} = \mathrm{S} \tag{4}$$

To implement SegNet for detection and categorization of coffee leaf disease, we follow the steps outlined below.

1) The coffee leaf images from the JMuBEN and JMuBEN2 datasets are resized and normalized for input into the SegNet. The preprocessing function $P(I)$, initially, resizes the original image I to $R(I)$ using the resizing factor r; $R(I) = I \times r$. The resized image is then normalized to produce $N(R(I))$, computed as $N(R(I)) = \frac{R(I)-\mu}{\sigma}$, where μ is the mean pixel value, and σ is the standard deviation of $R(I)$.
2) The dataset D is divided into three subsets as a training set T, a testing set Q, and a validation set V. The partition ratios for these subsets are denoted by α, β, and γ, respectively. These subsets are disjoint and can be represented as $D = T \cup V \cup Q$. Here, $T \cap V = \phi$, $T \cap Q = \phi$, and $V \cap Q = \phi$. Also, $|T| = \alpha \cdot |D|$, $|V| = \beta \cdot |D|$, and $|Q| = \gamma \cdot |D|$ denote the sizes of the training, validation, and testing sets, respectively. $|D|$ represents the total size of the dataset. Additionally, $\alpha + \beta + \gamma = 1$.
3) The SegNet architecture is modified for disease classification using coffee leaves by adjusting its structure. This adaptation is represented by a function M, which takes the original SegNet, the number of layers L, filters F, and output classes C as inputs. The modified SegNet denoted as SegNet' can be demonstrated as:

$$\text{SegNet'} = M(\text{SegNet}, L, F, C) \tag{5}$$

Algorithm 1 Data Augmentation for Coffee Leaf Disease Dataset

Input: Cleaned dataset $D = \{d_1, d_2, \ldots, d_n\}$ where $n = 57,275$
for each image d_i in D **do**
  Apply Rotation:
  for each angle $\theta \in \{90°, 180°, 270°\}$ **do**
    Rotate d_i by θ degrees clockwise and save the new image.
  end for
  $|\text{A_rot}| = n \times 3 = 57,275 \times 3 = 171,825$
  Apply Horizontal and Vertical Flipping:
  Flip d_i along the x-axis (horizontal flip) and save the new image.
  Flip d_i along the y-axis (vertical flip) and save the new image.
  $|\text{A_flip}| = n \times 2 = 57,275 \times 2 = 114,550$
  Apply Scaling:
  for each scale factor $s \in \{0.8, 1.2\}$ **do**
    Scale d_i by factor s and save the new image.
    New dimensions: width $\times s$, height $\times s$
  end for
  $|\text{A_scale}| = n \times 2 = 57,275 \times 2 = 114,550$
  Apply Translation:
  for each translation $(tx, ty) \in \{-0.1, 0.1\} \times \{-0.1, 0.1\}$ **do**
    Translate d_i and save the new image.
    New position: $(x + tx \times \text{width}, y + ty \times \text{height})$
  end for
  $|\text{A_trans}| = n \times 2 = 57,275 \times 2 = 114,550$
end for
Compute total number of augmented images for the entire dataset D:
$|\text{A}| = |\text{A_rot}| + |\text{A_flip}| + |\text{A_scale}| + |\text{A_trans}|$
$|\text{A}| = 171,825 + 114,550 + 114,550 + 114,550 = 515,475$
Combine the original dataset D with the augmented dataset A to create the final dataset D':
$|\text{D'}| = |D| + |A| = 57,275 + 515,475 = 572,750$

The function M uses the parameters L, F, and C to adjust the SegNet architecture accordingly.

4) The modified SegNet is then trained and tested using the training and validation datasets with optimal hyper-parameters. The training process is represented by the function Tr, which takes as input the modified SegNet, the training set T, validation set V, batch size (bs), number of epochs (ep), and learning rate (lr). It returns the trained model, denoted as SegNet_trained:

$$\text{SegNet_trained} = \text{Tr}(\text{SegNet'}, T, V, lr, bs, ep) \quad (6)$$

where SegNet_trained is the trained model.

Algorithm 3 presents the process for coffee leaf disease segmentation and classification using the modified SegNet.

Algorithm 2 Mask R-CNN Instance Segmentation

Input: Augmented dataset $D = \{I_i\}$ (572,750 images), Pre-trained weights W, Hyperparameters $H = \{\alpha, B, L\}$
Output: Trained Mask R-CNN model with updated weights W', Performance metric $P(W', T)$
Apply Preprocessing:
for each image I_i in D **do**
  $R \leftarrow$ resizing factor
  $N_i \leftarrow$ normalization *factor for image* I_i
  $I_i' \leftarrow (I_i \times R)/N_i$
end for
Apply Transfer Learning:
$W' \leftarrow W + \Delta W$
Apply Fine-Tuning:
Adjust hyperparameters $H = \{\alpha, B, L\}$
Start Training:
$X \leftarrow$ input *images from the training set*
$Y \leftarrow$ corresponding *labels*
Initialize loss function $L(W', X, Y) = L_c + L_r + L_m$
Minimize $L(W', X, Y)$ by updating the weights W'
Do Validation:
$V \leftarrow$ validation set
$P(W', V) \leftarrow$ *metric* (*e.g.*, *accuracy*, *F1score*)
Iteratively fine-tune H to maximize $P(W', V)$
Perfrom Testing:
$T \leftarrow$ test set
$P(W', T) \leftarrow$ final *performance metric*
Trained Mask R-CNN with updated W' & $P(W', T)$

Table 4 shows the configuration of the modified SegNet. It features five hidden layers and employs filters ranging from 64 to 1024 in the convolutional layers. It also includes a softmax activation function in the final layers with ReLU activation functions which are used in both the encoder and decoder layers. Performance is assessed using precision, recall, IoU, and cross-entropy Loss. Cross-Entropy Loss is also used as a loss function. The dataset consists of 572,750 images, with 80% images for training, 10% for testing, and 10% for validation.

TABLE 4. SegNet configuration for coffee leaf disease classification.

Aspect	Description
Model Architecture	SegNet
Number of Layers	5
Number of Filters	64, 128, 256, 512, 1024
Activation Functions	ReLU and Softmax
Loss Function	Cross-Entropy Loss
Performance Metrics	IoU, F1 Score, Precision, Recall
Training Set	80% - 458,200 images
Validation Set	10% - 57,275 images
Test Set	10% - 57,275 images

E. SEMANTIC SEGMENTATION OF COFFEE LEAF DISEASES USING U-NET

The next step in the proposed methodology is to apply semantic segmentation to detect and classify diseases on coffee leaves. It assigns a class to each pixel in an image

Algorithm 3 Coffee Leaf Disease Segmentation and Classification Using Modified SegNet

Input: JMuBEN and JMuBEN2 datasets (D)
Output: Trained SegNet (SegNet_trained)
Apply preprocessing on the datasets (D):
Initialize an empty list (preprocessed_images) to store the preprocessed images.
for each image I in D **do**
    Resize the image using the resizing factor r: $R(I) = I \times r$
    Calculate μ and σ of $R(I)$.
    Normalize the resized image: $N(R(I)) = \frac{R(I)-\mu}{\sigma}$
    Add the preprocessed image I' to the preprocessed_images list.
end for
Split the preprocessed_images into T, V, and Q sets:
Use partition ratios α, β, and γ such that $\alpha+\beta+\gamma = 1$.
Compute the sizes of T, V, and Q: $|T| = \alpha \times |D|$, $|V| = \beta \times |D|$, $|Q| = \gamma \times |D|$
Randomly assign the preprocessed images to T, V, and Q.
Modify the SegNet architecture:
Define a function M(SegNet, L, F, C)
Use M to create SegNet': SegNet' = M(SegNet, L, F, C) (see Eq. (5))
Train SegNet':
SegNet_trained = Tr(SegNet', T, V, lr, bs, ep)

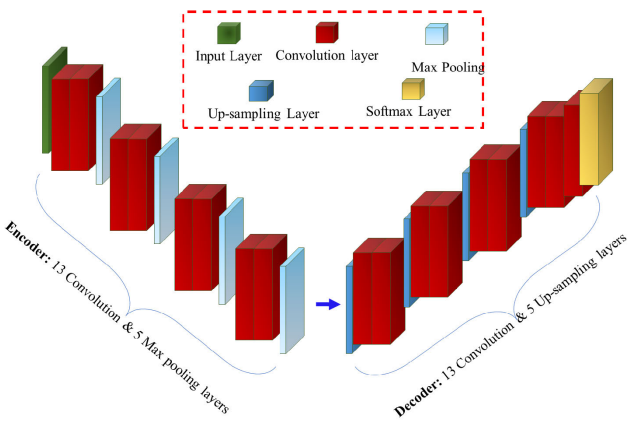


FIGURE 2. U-Net for leaf spot identification.

to enable reliable detection of diseased areas on the leaves. To achieve this, the U-Net is used (see Figure 2). The U-Net has an ability to effectively captures and stores image contextual data and features (refer [21]). To prevent gradient vanishing problem, U-Net also uses skip connections which preserve spatial information [22], [23].

Input images are initially pre-processed by resizing and normalizing. Thereafter, the U-Net is fine-tuned to identify disease areas by adjusting the activation functions, filter configurations, and the number of layers. By assigning class labels to each pixel and capturing context and features, the U-Net efficiently segments diseases on coffee leaves. This precision enables accurate diagnosis and classification even when diseases have similar visual characteristics. The subsequent section outlines each step briefly.

1) Ensure that the input images, derived from the combination of augmented and preprocessed dataset images, are in the correct size and format for the U-Net. For each image I, let N_{ui} represent normalization and R_u represent resizing. The resized and normalized images are denoted as I'_{ui}, where
$$I'_{ui} = \frac{I_{ui} * R_u}{N_{ui}} \tag{7}$$
for each image i, with I_{ui} being the original image.
2) Customize the U-Net for disease categorization in coffee leaves. Let L_u represents the number of layers, F_u denote the number of filters, and A_u refer to the activation functions used in U-Net. U-Net configuration can be expressed as U(L_u, F_u, A_u).
3) Train U-Net using the training set from the augmented coffee leaf disease dataset. Let X_u represent the input images and Y_u denote the corresponding labels. The loss function $L_u(W_u, X_u, Y_u)$ aims to minimize the difference between the ground truth masks and the predicted segmentation masks. The objective is to update the weights W_u of the U-Net to reduce the loss $L_u(W_u, X_u, Y_u)$.
4) Evaluate the performance of trained U-Net on the validation set. Let V_u denotes the validation set and $P_u(W'_u, V_u)$ represents the performance metric, such as IoU or F1 score. Iteratively adjust the U-Net configuration (L_u, F_u, A_u) to maximize $P_u(W'_u, V_u)$, where W'_u represents the updated weights after training.
5) After the U-Net achieves satisfactory performance on the validation set, evaluate its performance on the unseen test set T_u. The final performance metric can be expressed as $P_u(W'_u, T_u)$.

The algorithm for training U-Net to detect and classify coffee leaf diseases are presented in Algorithm 4. Table 5 presents the U-Net configuration.

Table 5 outlines the key components of the customized U-Net. It features five L_u, with an increasing F_u that start at 64 and double with each layer up to 1024 filters. The output layer employs a softmax activation function, while both the encoder and decoder layers use ReLU activation functions. Performance is evaluated using metrics such as IoU, F1 Score, precision, and recall, with the loss function being cross-entropy loss. The dataset is divided as 80% (458,200 images) for training, 10% (57,275 images) for validation, and 10% (57,275 images) for testing.

F. SEMANTIC SEGMENTATION OF COFFEE LEAF DISEASES USING PSPNET

The next step involves using the Pyramid Scene Parsing Network (PSPNet) for semantic segmentation to detect and classify whether coffee leaves are affected by disease.

Algorithm 4 U-Net-Based Coffee Leaf Disease Segmentation and Classification

Input: Augmented dataset of coffee leaf images (I_u)
Output: Segmented and classified coffee leaf diseases
Do preprocessing:
for each image i in I_u **do**
  $I'_{ui} = \frac{(I_{ui} \times R_u)}{N_{ui}}$
end for
Configure U-Net architecture:
Define the U-Net configuration as $U(L_u, F_u, A_u)$
Do training:
Optimize the U-Net's W_u to minimize $L_u(W_u, X_u, Y_u)$
Do validation:
Calculate the performance metric $P_u(W'_u, V_u)$ and iteratively fine-tune L_u, F_u, and A_u to maximize it.
Do testing:
Evaluate the final performance metric $P_u(W'_u, T_u)$ on the unseen T_u.

Semantic segmentation assigns a categorical label to each pixel in an image. PSPNet excels in capturing global contextual information at various scales, which is crucial for precise segmentation [24].

PSPNet utilizes a ResNet as its backbone to generate feature maps from input images [25]. It incorporates a pyramid pooling module that divides the feature map into progressively smaller regions, pools average features from each region, and then integrates them. This approach enhances the network's ability to detect objects of different sizes, leveraging its multi-scale capabilities. The final segmentation mask is produced through a convolutional layer (refer [26]).

The steps to apply PSPNet for disease detection and classification in coffee leaves are as follows:

- The input images are initially resized and normalized for PSPNet. Let R_p denotes the resizing factor and N_{pi} represents the normalization factor for each image i. The resized and normalized images can be computed as:

$$I'_{pi} = \frac{I_{pi} \cdot R_p}{N_{pi}} \tag{8}$$

  where I_{pi} is the original image.
- PSPNet is then customized for coffee leaf disease classification. Let B_p represents the backbone network, M_p denotes the pyramid pooling module, and F_p signifies the output layer. The PSPNet configuration can be expressed as:

$$\text{PSPNet}(B_p, M_p, F_p) \tag{9}$$

- The augmented coffee leaf disease dataset's training set is used to train PSPNet. The input images are represented by X_p, while the corresponding labels are represented by Y_p. To minimize the difference between the predicted segmentation masks S_p and the ground truth masks G_p, the loss function $L_p(W_p, X_p, Y_p)$ is utilized. The loss function can be mathematically expressed as

$$L_p(W_p, X_p, Y_p) = \sum_i D(S_{p_i}, G_{p_i}), \tag{10}$$

  where D is a distance metric like cross-entropy loss, and W_p denotes the PSPNet's weights.
- Evaluate the performance of trained PSPNet using the validation set. Let V_p represents the validation set, and $P_p(W'_p, V_p)$ denotes the performance metric such as IoU or F1 score, where W'_p are the model's weights. Adjust the PSPNet configuration (B_p, M_p, F_p) iteratively to maximize $P_p(W'_p, V_p)$ with W'_p representing the updated weights after training.
- Once the PSPNet achieves satisfactory performance on the validation set, it will be evaluated on unseen test set T_p. The final performance metric is determined by $P_p(W'_p, T_p)$, where W'_p denotes the model's weights.

Efficient segmentation and detection of diseased regions on coffee leaves are achieved using PSPNet, despite the complex patterns and varying sizes of the diseases. Figure 3 and Table 6 illustrate sample results of the segmentation of disease-affected regions. The procedure of PSPNet is outlined in Algorithm 5.

Algorithm 5 Detection and Classification of Coffee Leaf Diseases Using PSPNet

Input: Augmented dataset of coffee leaf images (I_p) and corresponding labels (Y_p)
Output: Predicted segmentation masks (S_p) for test set (T_p) with performance metric $P_p(W'_p, T_p)$
Do preprocessing:
For each image i, compute I'_p using Eq. (8)
Configure PSPNet architecture:
Define customized PSPNet(B_p, M_p, F_p) (refer Eq. (9))
Do training:
To minimize the loss function $L_p(W_p, X_p, Y_p)$ (refer Eq. (10)) on X_p and Y_p, update the weights W_p.
Do validation:
Assess the trained PSPNet with updated weights W'_p on V_p. Fine-tune the architecture (B_p, M_p, F_p) to maximize $P_p(W'_p, V_p)$.
Do testing:
Evaluate the PSPNet with updated weights W'_p on T_p, and compute the final performance metric $P_p(W'_p, T_p)$.

Table 6 outlines the key features of PSPNet cutomized for coffee leaf disease classification. It employs a ResNet50 backbone for feature extraction and pyramid pooling with various bin sizes (1×1, 2×2, 3×3, and 6×6) to capture diverse spatial information. The cross-entropy loss function is used. The output layer consists of a convolution layer followed by a Softmax activation function. Performance is evaluated using metrics such as IoU, F1 Score, Precision, and Recall. The dataset is divided into 80% (458,200 images)

TABLE 5. Configuration of U-Net.

Aspect	Description
Model Architecture	U-Net
L_u	5
F_u	64, 128, 256, 512, 1024
A_u	ReLU for encoder and decoder, Softmax for the output layer
Loss Function	Cross-Entropy Loss
Performance Metrics	IoU, F1 Score, Precision, Recall
Training Set	80% - 458,200 images
Validation Set	10% - 57,275 images
Test Set	10% - 57,275 images

TABLE 6. Configuration of PSPNet.

Aspect	Description
Model Architecture	PSPNet
Backbone	ResNet50
Pyramid Pooling	1x1, 2x2, 3x3, 6x6 bins
Output Layer	Conv + Softmax
Loss Function	Cross-Entropy Loss
Performance Metrics	IoU, F1 Score, Precision, Recall
Training Set	458,200 images (80%)
Validation Set	57,275 images (10%)
Test Set	57,275 images (10%)

for training, 10% (57,275 images) for validation, and 10% (57,275 images) for testing. This setup exemplifies the customized PSPNet designed specifically for segmenting and classifying coffee leaf diseases.

G. PROPOSED MODEL ARCHITECTURE: SUNET FOR COFFEE LEAF DISEASE CLASSIFICATION

To classify diseases on coffee leaves, we propose a hybrid network combining SegNet and U-Net. The algorithm for this hybrid model is outlined in Algorithm 6. The architecture aims to provide an accurate segmentation and classification performance. Table 7 provides the architectural details of proposed SUNet. Table 8 outlines the detailed steps for customization. The explanation of the steps of SUNet is provided in the following subsections.

- The encoder in SUNet utilizes the VGG16 backbone from SegNet. VGG16 includes sixteen hidden layers for feature extraction. From these layers, the last three layers utilize the ReLU activation function, while the remaining thirteen are convolutional layers.
- For each class of coffee leaf diseases, the model can provide accurate segmentation masks because the decoder generates the segmented output as same as the resolution of original image. Up sampling, convolutional, and activation layers forms the five-layer architecture of the U-Net based decoder. 1024, 512, 256, 128, and 64 filters, respectively, are used in the decoder levels. To prevent gradient degrdation, the skip connections between encoder and decoder layers are used as $S_i = E_i + D_i$. Here, E_i denotes the output from encoder layer. D_i denotes the decoder layer.
- SUNet is customized to classify affected regions on coffee leaves. This customization involves adjusting the number of layers, filters, and activation functions. For each pixel in the segmented mask, the Softmax activation function is used to predict the classification probability. The probability of class i, denoted by $P(c_i)$ can be calculated as:

$$P(c_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{11}$$

where z_i is the input value for class i, and the summation is performed over all classes.
- During training, the cross-entropy loss function is used to compare the ground truth masks G_p and the predicted segmentation masks S_p. The objective is to minimize the difference between S_p and G_p. The cross-entropy loss function is mathematically defined as:

$$L(W, X, Y) = -\sum_i G_{p_i} \cdot \log(S_{p_i}) \tag{12}$$

where W represents the weights of SUNet. X denotes the input images, Y refers to the corresponding labels, and i iterates over all pixels in the masks.
- Performance of the SUNet is evaluated using the F1 score, IoU, precision, and recall metrics. IoU can be calculated as:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{13}$$

where TP denotes the number of true positive pixels, FP represents the number of false positive pixels, and FN indicates the number of false negative pixels.

Table 9 presents the detailed configuration of the SUNet. Table 10 outlines the modifications made to SUNet. Table 11 presents the strengths and weaknesses of each individual model.

SUNet employs the SegNet VGG16 encoder which benefits from pre-trained weights, unlike U-Net that does not utilize such weights. This advantage allows the model to achieve faster training and improved performance. While pre-trained encoders are advantageous for PSPNet, the VGG16 backbone utilized in SUNet is remarkable for its compact size and efficient performance [27].

SUNet enhances U-Net's architecture by using an advanced decoder design that matches the resolution of the original images. This improves segmentation performance compared to SegNet's simpler decoder. Unlike PSPNet,

TABLE 7. Design of the proposed hybrid network (SUNet).

Integration Aspect	Advantages	Improvements	Configurations used in the Paper
Encoder (SegNet with VGG16 Backbone)	Utilizes the robust feature extraction capability of VGG16, which is pre-trained on a large dataset, leading to faster convergence and better initial feature representation	By using a pre-trained encoder, SUNet can extract hierarchical features more effectively	Backbone: VGG16, Pre-trained on ImageNet
Decoder (U-Net Style with Skip Connections)	Retains detailed spatial information through skip connections, which is crucial for precise segmentation of disease areas	Ensures that the segmentation masks are reconstructed with high fidelity to the input images, maintaining fine-grained details	Skip Connections: Between corresponding encoder and decoder layers, Activation: ReLU
Mask R-CNN for Instance Segmentation	Provides accurate instance segmentation, which helps in identifying and delineating individual disease spots on the leaves	Adds a layer of precision to the initial segmentation, ensuring that the model can handle complex and overlapping disease regions	Region Proposal Network (RPN), Backbone: ResNet-50
PSPNet for Contextual Information	Captures global context, which helps in understanding the broader distribution of disease on the leaf	By integrating multi-scale context, SUNet can effectively segment disease areas that vary in size and shape	Pyramid Pooling: 1x1, 2x2, 3x3, 6x6 bins, Backbone: ResNet-50

TABLE 8. Detailed steps of customization.

Step	Description
Fine-Tuning the Encoder	The pre-trained VGG16 model is fine-tuned on the coffee leaf dataset by unfreezing the top layers and training them on the new dataset.
Configuring the Decoder	The U-Net style decoder is customized with five layers, each with specific filters (1024, 512, 256, 128, 64). ReLU activations and skip connections are used.
Implementing Mask R-CNN	The RPN within Mask R-CNN generates candidate regions for potential disease spots, which are then classified and segmented.
Integrating PSPNet	The pyramid pooling module of PSPNet pools feature maps at different scales and concatenates them for multi-scale contextual information.
Designing the Output Layer	The final layer uses a Softmax activation function to classify each pixel into one of the predefined categories.
Augmentation and Preprocessing	The dataset is augmented with various transformations, and images are resized and normalized for uniform input dimensions.

TABLE 9. Configuration of SUNet.

Component	Description	Configuration
Encoder	VGG16 backbone from SegNet; 16 weight layers (13 convolutional, 3 fully connected) with ReLU activations	Pre-trained weights, 13 Conv layers, 3 FC layers
Decoder	U-Net style; 5 layers with up-sampling, convolutional, and activation layers	Filters: {1024, 512, 256, 128, 64}; ReLU activations
Skip connections	Fine-grained information is retained by linking in the encoder and decoder, there are matching layers	$S_i = E_i + D_i$
Output Layer	Customized for coffee leaf disease classification; Softmax activation function	Softmax activation
Loss Function	Loss function for cross-entropy; utilized to decrease the disparity between the predicted and ground truth masks	L(W, X, Y) = -$\Sigma_i G_{p_i} * log(S_{pi})$

which uses pyramid pooling to gather contextual information, SUNet benefits from skip connections between its encoder and decoder. These connections help preserve fine-grained details more efficiently. These skip connections allow SUNet to maintain detailed information across layers. Customized for coffee leaf disease classification, SUNet adjusts its layers, filters, and activation functions. It also utilizes the Softmax function to generate precise class probabilities for each pixel.

Figure 3 shows the infected leaf image sourced from [28]. This image is utilized as the input for evaluating the proposed SUNet model on an unseen dataset. The impact of instance segmentation on the input image is also shown. It demonstrates how instance segmentation effectively identifies and highlights the infected regions within the leaf image.

Figure 4 shows the computed mask obtained from the proposed model. This mask is used to indicate the region of interest (ROI) within the input image. Thus, this figure enhances the understanding of SUNet's focus during the plant leaf diagnostic process.

Figure 5 presents the proposed SUNet-based detection of infected regions marked in red. This demonstrates that SUNet can effectively detect and classify the infected areas for coffee leaf disease diagnosis.

H. SEGMENTATION AND DISTINGUISHING DISEASE AREAS

- This model incorporates an RPN that generates candidate regions likely to contain a disease based on feature extraction. Subsequently, these regions are classified

TABLE 10. Targeted modifications for SUNet.

Modification	Description	Details
Customized Encoder (VGG16 Backbone from SegNet)	- Pre-trained Weights: Uses VGG16 architecture pre-trained on ImageNet for robust feature extraction. - Adaptation: Fine-tuned on coffee leaf datasets (JMuBEN and JMuBEN2) to adapt features to coffee leaf imagery.	$f(x) = VGG16(x)$
Enhanced Decoder (U-Net Style with Skip Connections)	- Skip Connections: Retains fine-grained details. - Layer Customization: Adjusted layers and filters (1024, 512, 256, 128, 64) to balance detail preservation and efficiency.	$D_i = ReLU(E_i + D_{i+1})$
Integration of Instance Segmentation (Mask R-CNN)	- Instance Segmentation: Handles segmentation of individual disease spots. - Region Proposal Network (RPN): Generates candidate regions likely to contain disease spots.	$P = RPN(F)$
Multi-Scale Contextual Information (PSPNet)	- Pyramid Pooling Module: Captures contextual information at multiple scales. - Contextual Adaptation: Pools features at different scales (1x1, 2x2, 3x3, 6x6 bins) for understanding local and global patterns.	$F_{pyramid} = concat(pool(F, 1 \times 1), pool(F, 2 \times 2), pool(F, 3 \times 3), pool(F, 6 \times 6))$
Custom Output Layer for Disease Classification	- Softmax Activation: Classifies each pixel into one of the disease categories (healthy, Phoma, Cercospora, Rust, Miner). - Class-Specific Training: Trained with annotated data for precise disease classification.	$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$
Data Augmentation and Preprocessing	- Augmentation Techniques: Applies rotation, flipping, scaling, and translation to increase training data variability. - Preprocessing: Resizes and normalizes images for consistency and improved training stability.	$Normalization(x) = \frac{x - mean}{std}$

(a) Infected leaf image and (b) impact of instance segmentation

FIGURE 3. Impact of instance segmentation on the Infected leaf image.

according to the type of disease, and a pixel-wise mask is created for precise segmentation.

- U-Net and SegNet: These models utilize an encoder-decoder architecture. The encoder first extracts features at various levels of abstraction, which are then passed to the decoder. The decoder uses this information to reconstruct the pixel-wise mask. U-Net enhances this process with skip connections, which help preserve fine details. A pixel-wise mask is advantageous as it classifies each pixel based on the features from its specific field of view (FOV).

(a) Infected leaf image, (b) computed binary mask and (c) ROI within the input image.

FIGURE 4. Region of interest (ROI) computed using proposed SUNet.

- PSPNet: This network model includes a Pyramid pooling module that integrates contextual information at multiple scales. This allows the capture of both local and global features for accurate segmentation of diseases with varying sizes.
- SUNet: This network combines the strengths of both U-Net and SegNet to optimize feature utilization and improve feature extraction. VGG16 backbone from SegNet enhances this network by delivering robust feature extraction capabilities.

IV. RESULT AND DISCUSSION

This research uses a range of performance metrics to assess the model's segmentation and classification capabilities. Key indicators such as F1 Score, IoU, Recall, and Precision are employed to evaluate the effectiveness of the SUNet. Table 12 outlines the software and hardware specifications used for

Algorithm 6 SUNet Algorithm for Classifying Coffee Leaf Disease

1: **Input:** Set of preprocessed images of coffee leaves $\{I_1, I_2, \ldots, I_n\}$ and labels $\{L_1, L_2, \ldots, L_n\}$
2: **Output:** Predicted segmentation masks $\{S'_1, S'_2, \ldots, S'_n\}$ and their corresponding predicted class labels $\{C'_1, C'_2, \ldots, C'_n\}$
3: **Initialize the SUNet:**
   a. Configure the VGG16 encoder from SegNet with pre-trained weights:
$$E = \text{VGG16(pretrained_weights)}$$
   b. Configure the U-Net style decoder with 5 layers, filters {1024, 512, 256, 128, 64}, and ReLU activation functions:
$$D = \text{U-NetDecoder(filters} = \{1024, 512, 256, 128, 64\})$$
   c. Create a Softmax activation function-specific output layer for identifying coffee leaf disease:
$$O = \text{SoftmaxLayer()}$$
4: **Train the SUNet:**
   a. Apply a cross-entropy loss function to the training set to train the model:
$$L(W, X, Y) = -\sum_i G_i * \log(S_{p_i})$$
   b. Update the weights (W) of the SUNet during training.
5: **Validate the SUNet:**
   a. Evaluate the performance of trained SUNet on validation set using various metrics.
   b. Iteratively fine-tune the SUNet configuration to maximize the performance $P_p(W', V_p)$.
6: **Test the SUNet:**
7: Evaluate the performance of SUNet on unseen data.
8: **Generate predictions:**
   a. Use the trained SUNet to generate predicted segmentation masks $\{S'_1, S'_2, \ldots, S'_n\}$:
$$S'_i = E(I_i) + D(I_i)$$
   b. Class probabilities for each pixel in the segmentation mask are generated by applying the Softmax activation function:
$$P(c_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

developing, training, and evaluating the SUNet for detecting diseases on coffee leaves.

Figure 6 illustrates the training and validation loss curves for U-Net, SegNet, PSPNet, and the proposed SUNet.

(a) Infected leaf image and (b) Computed infected regions

FIGURE 5. Proposed SUNet-Based Detection of Infected Regions Marked in Red.

In Figures 6(a) to 6(c), it is observed that although the loss decreases as training progresses, a significant gap exists between the training and validation losses which indicate that these models exhibit some degree of overfitting. Figure 6 (d) reveals that SUNet achieves better performance with the better convergence as shown by the closely aligned training and validation loss curves. Thus, SUNet has lesser impact of overfitting. Additionally, SUNet achieves significantly lower loss values compared to the other models. This indicates that it not only converges faster but also generalizes more effectively.

To evaluate the effectiveness of the proposed SUNet, confusion matrix heatmaps are computed for all models, namely SegNet, U-Net, PSPNet, and SUNet. These heatmaps are shown in Figures 7, 8, 9, and 10, respectively. Figure 7 shows SegNet's confusion matrix, which reveals a relatively higher number of misclassifications, particularly in distinguishing between disease types and healthy leaves. Figure 8 presents U-Net's confusion matrix, which indicates improved performance over SegNet but still shows some misclassifications. Figure 9 depicts PSPNet's confusion matrix, which shows further improvements but still has limitations in accurately identifying all disease-affected areas. Figure 10 illustrates SUNet's confusion matrix, which clearly demonstrates SUNet's superior performance, with fewer misclassifications and higher accuracy in identifying and classifying disease-affected areas.

Figure 11 shows the Matthews Correlation Coefficient (MCC) and Dice coefficient for four semantic segmentation models such as U-Net, PSPNet, SegNet, and SUNet. It is found that U-Net achieves an MCC of 0.8218 and a Dice coefficient of 0.8457. This indicates a strong correlation between the predicted and actual classifications, with a relatively high

TABLE 11. Comparison of different segmentation models.

Model	Advantages	Disadvantages	Special Features
Mask R-CNN	- Effective instance segmentation of individual objects (disease spots). - High precision in detecting and delineating disease regions. - Flexible in handling complex backgrounds and overlapping objects.	- Computationally intensive. - Complex architecture involving multiple stages (region proposal, classification, mask prediction).	Instance Segmentation: Precisely identifies and segments individual disease spots.
U-Net	- Retains fine-grained details through effective skip connections. - High accuracy, especially in biomedical images. - Efficient end-to-end training with moderate computational resources.	- Limited context capturing due to small receptive field. - High parameter count depending on network depth and width.	Biomedical Specialization: Particularly effective in medical imaging scenarios.
Seg-Net	- Efficient feature extraction and image reconstruction using encoder-decoder architecture. - More computationally efficient compared to other models. - Can leverage pre-trained VGG16 weights for faster training.	- May lose fine details in the decoding process. - Less flexible due to a fixed structure.	Feature Extraction: Utilizes VGG16 backbone for robust initial feature extraction.
PSP-Net	- Captures multi-scale context using pyramid pooling. - Robust in segmenting images with varying object sizes. - Improved accuracy by considering global context.	- Complex architecture due to pyramid pooling module. - High computational resource demand.	Contextual Awareness: Excellent at integrating local and global contextual information.
SUNet (Hybrid Model)	- High accuracy combining multiple models. - Detailed segmentation retaining fine details. - Robust contextual understanding. - Efficient training leveraging pre-trained weights.	- Significant computational complexity for training and inference. - Increased implementation and tuning complexity.	Hybrid Design: Integrates U-Net and SegNet strengths, enhanced by specific features from Mask R-CNN and PSPNet for superior performance.

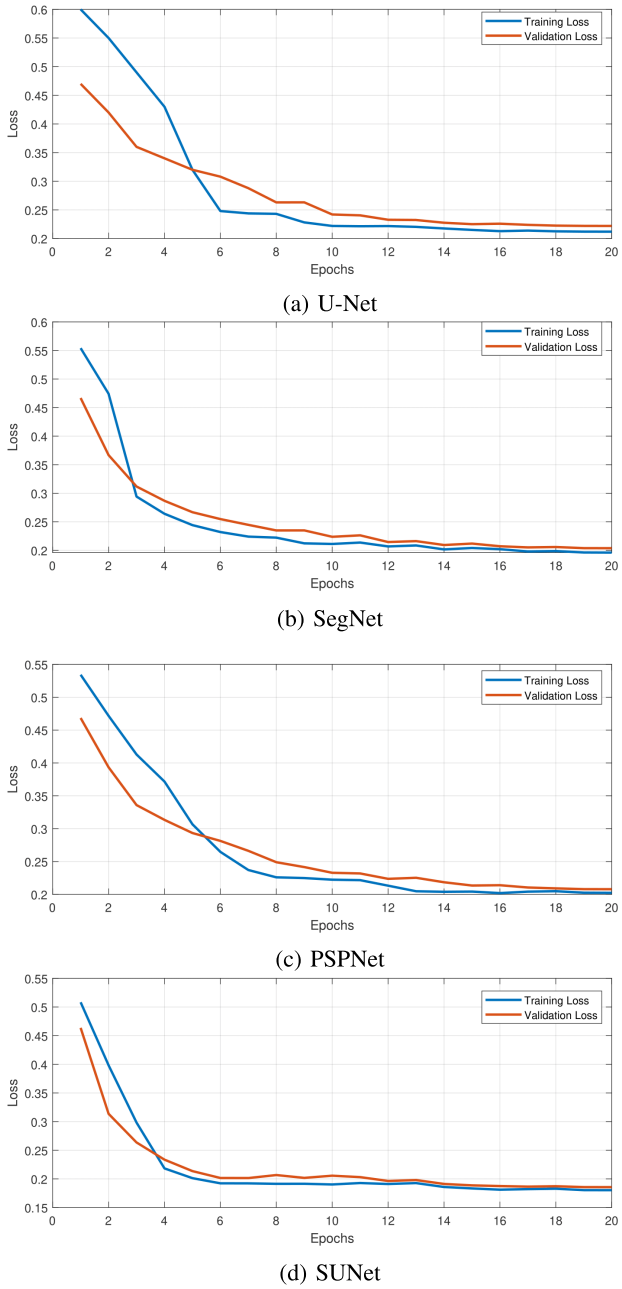
TABLE 12. Software and hardware requirements for SUNet implementation and evaluation.

Requirement Type	Component	Specification	Purpose
Software	Deep learning	TensorFlow	Building and training the SUNet
	Data preprocessing	OpenCV, NumPy	Data augmentation and preprocessing for consistent and efficient image manipulation
	Model evaluation	Scikit-learn	Calculating performance metrics (IoU, F1 Score, Precision, Recall)
Hardware	GPU	NVIDIA Tesla V100	Efficient training and inference of the model, leveraging high-performance capabilities
	CPU	Intel Xeon (multiple cores)	Parallel processing during data preprocessing and augmentation
	RAM	128 GB	Handling the memory requirements of the dataset and model
	Storage	2 TB SSD	Storing the large dataset, model weights, and intermediate results for fast data access during training and evaluation

Dice coefficient reflecting good overlap between predicted and true regions. PSPNet achieves an MCC of 0.8157 and a Dice coefficient of 0.8437. While similar to U-Net, PSPNet shows slightly lower MCC and Dice coefficients, indicating comparable performance but with marginally less accuracy in classifying the disease-affected areas. SegNet achieves an MCC of 0.8122 and a Dice coefficient of 0.8411. SegNet has the lowest values among the models shown, suggesting it has a lower correlation and overlap accuracy compared to U-Net, PSPNet, and SUNet. Finally, SUNet achieves an MCC of 0.9705 and a Dice coefficient of 0.9840. SUNet achieves the highest MCC and Dice coefficients, demonstrating its superior performance in correlating and overlapping with the true disease-affected areas. This indicates that SUNet provides the most accurate and reliable classification results among the models evaluated.

Figure 12 illustrates the class-wise error rates for each model which shows that the proposed SUNet achieves significantly lower error rates for all classes. These classes are Cercospora, Rust, Phoma, Miner, and Healthy. Specifically, the error rates for SUNet are 0.017 for Cercospora, 0.0172 for Rust, 0.0179 for Phoma, 0.0181 for Miner, and 0.0189 for Healthy. These results highlight SUNet's superior performance in minimizing errors across all categories. Thus, the proposed SUNet demonstrates its effectiveness in accurately classifying various disease types and healthy leaves compared to other models.

Figures 13 to 16 show the misclassification heatmaps for SegNet, U-Net, PSPNet, and SUNet. The heatmap for SegNet reveals a higher number of misclassified areas, particularly where the model struggles to differentiate between different disease types and healthy leaves (see Figure 13). This indicates that SegNet has a higher rate of errors and confusion in its predictions. The heatmap for U-Net shows improvements over SegNet, with fewer misclassifications (see Figure 14). However, it still demonstrates some residual



(a) U-Net

(b) SegNet

(c) PSPNet

(d) SUNet

FIGURE 6. Training and validation loss analysis.

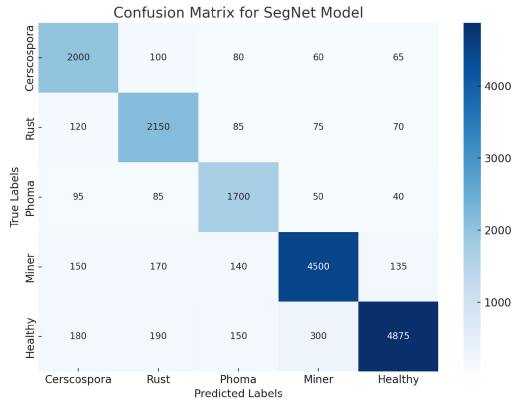


FIGURE 7. Confusion Matrix Heatmap for SegNet.

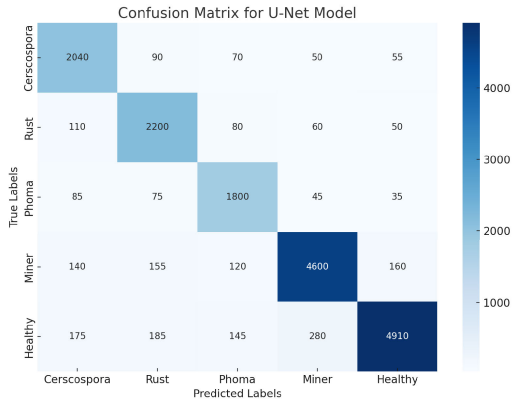


FIGURE 8. Confusion Matrix Heatmap for U-Net.

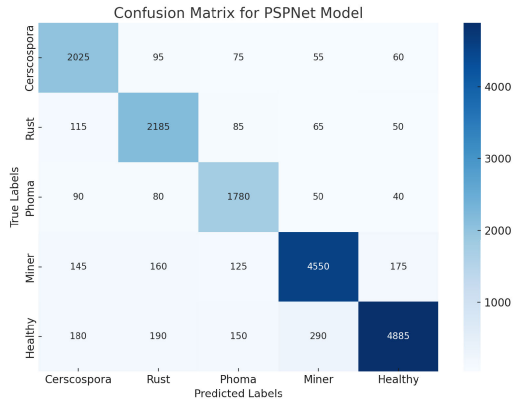


FIGURE 9. Confusion Matrix Heatmap for PSPNet.

errors in distinguishing disease types and healthy leaves, indicating that while U-Net performs better, it is not without limitations. The heatmap for PSPNet shows further reductions in misclassification compared to SegNet and U-Net (see Figure 15). Although PSPNet achieves better performance, there are still areas where the model struggles to accurately classify all disease-affected regions. The heatmap for SUNet clearly illustrates its superior performance (see Figure 16). It shows significantly fewer misclassifications, indicating that SUNet accurately identifies and classifies disease-affected areas with high precision. This highlights SUNet's effectiveness in reducing errors and providing reliable classification results. Overall, the misclassification heatmaps reveal that the proposed SUNet achieves significantly fewer misclassification errors compared to SegNet, U-Net, and PSPNet.

Table 13 presents a comparative analysis of semantic segmentation models, namely SUNet, SegNet, U-Net, and PSPNet. The proposed SUNet demonstrates significantly better performance across all metrics. It achieves an accuracy of 98.45%, an IoU score of 97.8%, and an F1 score of 98.1%. Additionally, SUNet's high precision (98.3%) and recall (98.6%) make it the most reliable choice for coffee leaf disease classification compared to SegNet, U-Net, and PSPNet.

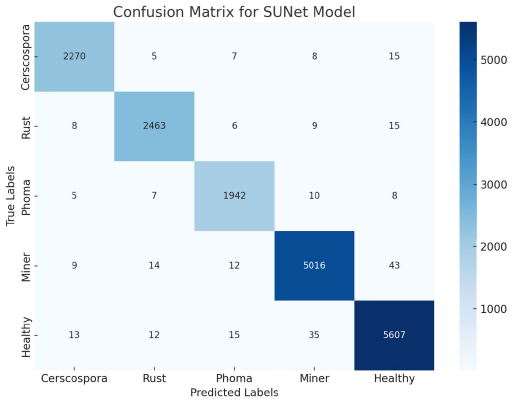


FIGURE 10. Confusion Matrix Heatmap for SUNet.

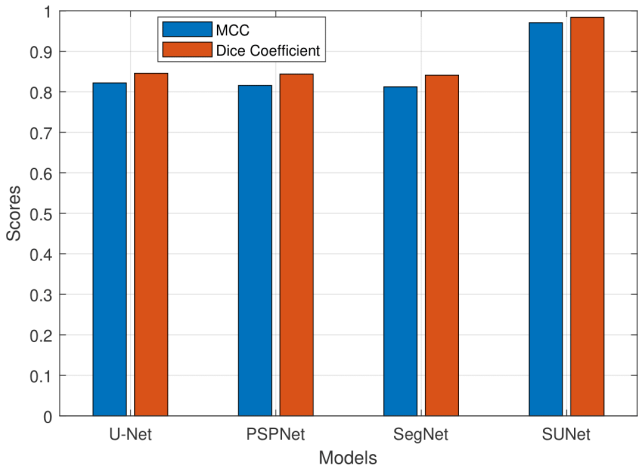


FIGURE 11. MCC and Dice Coefficients for Each Model.

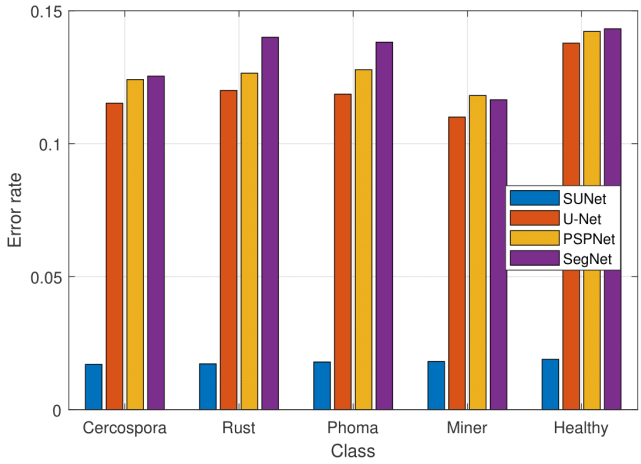


FIGURE 12. Error Rates for Each Model by Class.

TABLE 13. Performance Comparison of SUNet, SegNet, U-Net, and PSPNet on Coffee Leaf Disease Classification.

Metric	SUNet	SegNet	U-Net	PSPNet
Accuracy (%)	98.45	86.70	88.50	87.90
IoU (%)	97.80	78.60	81.40	80.10
F1 Score (%)	98.10	85.34	86.67	85.98
Precision (%)	98.30	84.34	85.64	84.97
Recall (%)	98.60	86.61	87.97	87.27

Table 14 presents a comparison between ResNet152V2 and VGG-19 for coffee leaf disease detection. It evaluates both bounding box detection and semantic segmentation. ResNet152V2 achieves higher accuracy and better IoU scores at thresholds of 0.5, 0.75, and 1.0 compared to VGG-19.

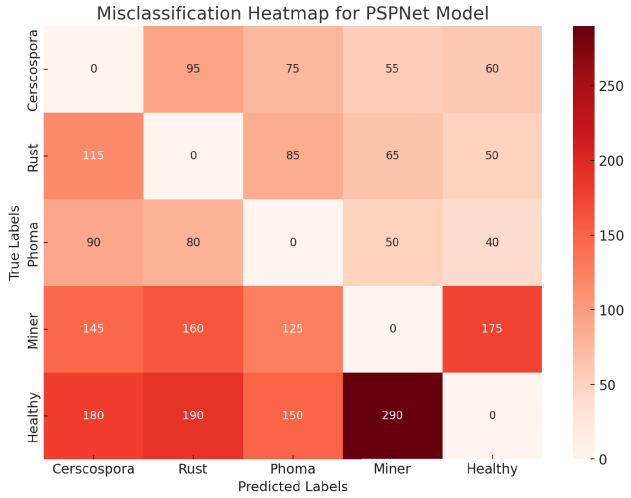


FIGURE 13. Misclassification Heatmap for PSPNet.

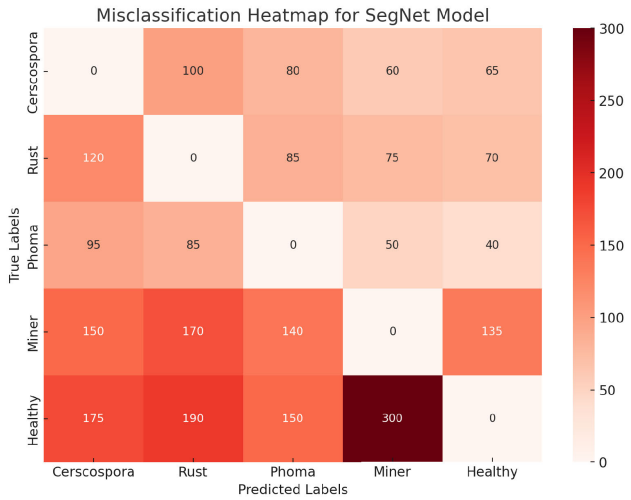


FIGURE 14. Misclassification Heatmap for SegNet.

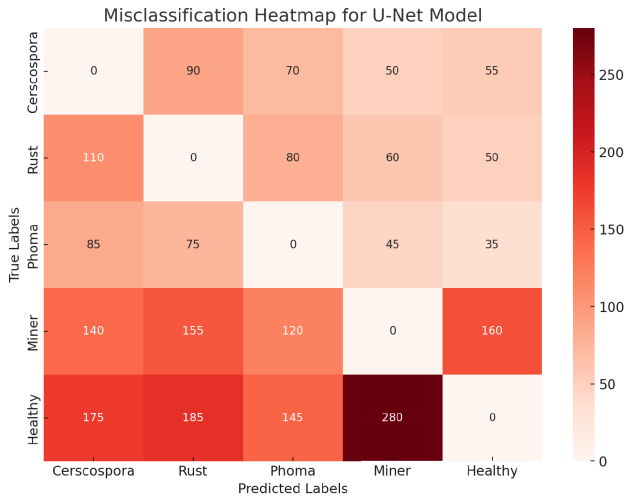


FIGURE 15. Misclassification Heatmap for U-Net.

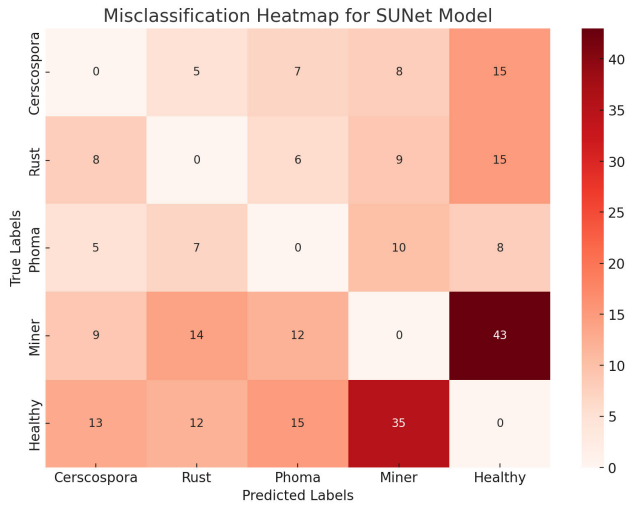


FIGURE 16. Misclassification Heatmap for SUNet.

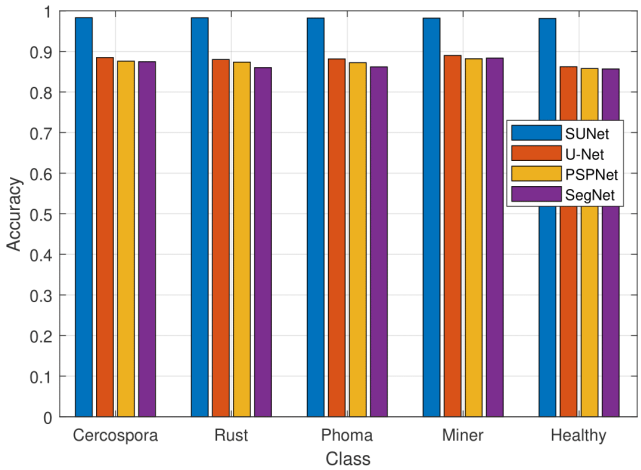


FIGURE 17. Class-wise Accuracy for Each Model.

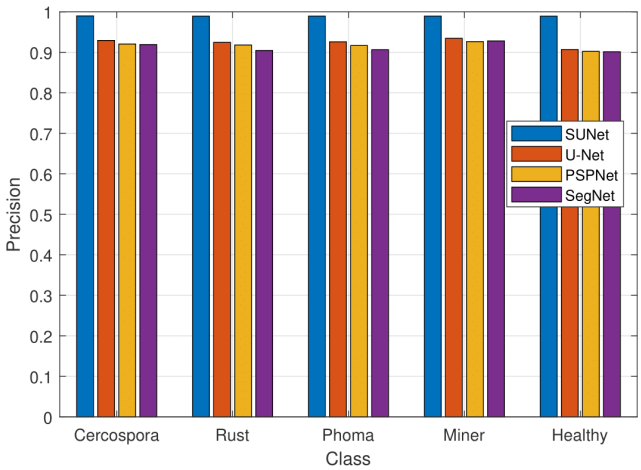


FIGURE 18. Class-wise Precision for Each Model.

However, ResNet152V2 requires more training time and has a longer inference time. It also has increased model complexity, including more parameters and higher memory usage. In contrast, VGG-19 is faster and less resource-intensive. Although it offers lower accuracy, it uses the

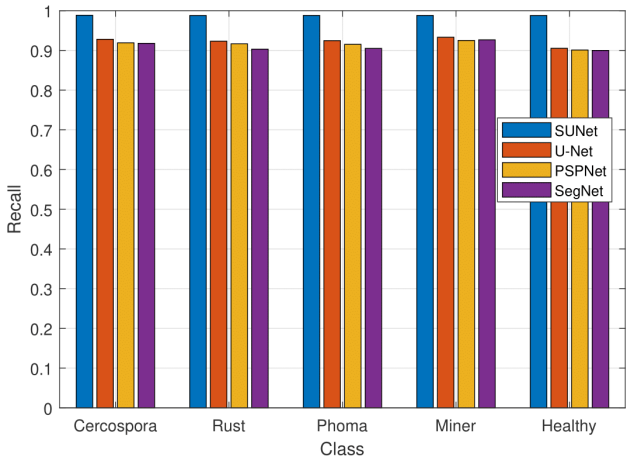


FIGURE 19. Class-wise Recall for Each Model.

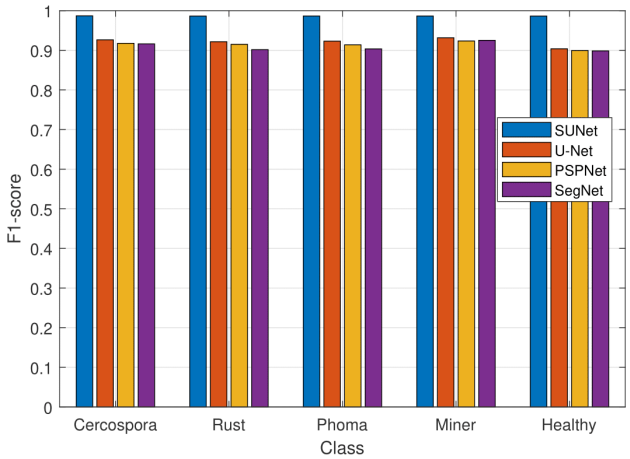


FIGURE 20. Class-wise F1-score for Each Model.

simpler SGD optimizer, while ResNet152V2 employs Adam. This comparison highlights the trade-off between accuracy and efficiency when selecting a model for coffee leaf disease detection.

Table 15 provides a comparison of features across different models, including the proposed SUNet. It highlights that SUNet encompasses all the features listed, while other models such as SegNet, U-Net, and PSPNet lack certain features. Consequently, SUNet is shown to be more comprehensive and is therefore preferred for coffee leaf disease detection due to its ability to cover all relevant features.

Figure 17 presents a class-wise accuracy analysis among U-Net, PSPNet, SegNet, and SUNet. The proposed SUNet outperforms the other models, achieving the highest accuracy for each class as Cercospora with 0.9830, Rust with 0.9828, Phoma with 0.9821, Miner with 0.9819, and Healthy with 0.9811. This demonstrates SUNet's superior performance in accurately classifying each disease type and healthy leaves compared to U-Net, PSPNet, and SegNet.

Figure 18 presents a class-wise precision analysis for U-Net, PSPNet, SegNet, and SUNet. The proposed SUNet outperforms the other models, achieving the highest precision for each class: 'Cercospora' (0.9898), 'Rust' (0.9892), 'Phoma' (0.9893), 'Miner' (0.9893), and 'Healthy' (0.9892). These

TABLE 14. Performance comparison of ResNet152V2 and VGG-19 for coffee leaf disease detection: Metrics, Time, and model complexity.

Model	Task	IoU=0.5 (%)	IoU=0.75 (%)	IoU=100 (%)	Training Time (hrs)	Inference Time (ms)	Parameters (M)	Memory Usage (GB)	Layers	Best Optimizer
ResNet 152V2	Segmentation	89.36	78.32	97.25	12	60	60.2	11	152	Adam
ResNet 152V2	Bounding Box	85.36	75.89	96.25	12	60	60.2	11	152	Adam
VGG-19	Segmentation	77.21	74.76	96.33	10	45	20	8	19	SGD
VGG-19	Bounding Box	85.62	76.29	95.25	10	45	20	8	19	SGD

TABLE 15. Comparing the different models' features with the proposed SUNet.

Feature	Mask R-CNN	Seg-Net	U-Net	PSP-Net	Proposed SUNet
Region Proposal	✓	✗	✗	✗	✓
Bounding Box Classification	✓	✗	✗	✗	✓
Mask Prediction	✓	✗	✗	✗	✓
Encoder-Decoder Architecture	✗	✓	✓	✗	✓ (Customized Encoder with VGG16 Backbone from SegNet and Enhanced Decoder with U-Net Style)
Skip Connections	✗	✗	✓	✗	✓ (Retains fine-grained details with layer customization)
Pyramid Pooling	✗	✗	✗	✓	✓ (Multi-Scale Contextual Information with Pyramid Pooling Module)
Combines SegNet and U-Net Strengths	✗	✗	✗	✗	✓
Uses VGG16 Architecture	✗	✓	✗	✗	✓ (Pre-trained on ImageNet, fine-tuned on coffee leaf datasets)
Softmax Function for Pixel Classification	✓	✓	✓	✓	✓
Handles Different Disease Sizes	✗	✗	✗	✓	✓
High Performance on Coffee Leaf Disease Data	✗	✗	✗	✗	✓
Instance Segmentation (Mask R-CNN)	✓	✗	✗	✗	✓ (Integration of Mask R-CNN for individual disease spots)
Region Proposal Network (RPN)	✓	✗	✗	✗	✓ (Generates candidate regions likely to contain disease spots)
Multi-Scale Contextual Information (PSPNet)	✗	✗	✗	✓	✓ (Pyramid Pooling Module captures contextual information at multiple scales)
Custom Output Layer for Disease Classification	✓	✓	✓	✓	✓ (Softmax Activation for precise disease classification)
Data Augmentation Techniques	✗	✗	✗	✗	✓ (Applies rotation, flipping, scaling, and translation to increase training data variability)
Fine-Tuning the Encoder	✗	✓	✗	✗	✓ (Fine-tuning VGG16 on the coffee leaf dataset)
Configuring the Decoder	✗	✗	✓	✗	✓ (U-Net style decoder with customized layers and filters)
Implementing Mask R-CNN	✓	✗	✗	✗	✓ (RPN within Mask R-CNN generates candidate regions for potential disease spots)
Integrating PSPNet	✗	✗	✗	✓	✓ (Pyramid pooling module pools feature maps at different scales and concatenates them for multi-scale contextual information)
Designing the Output Layer	✓	✓	✓	✓	✓ (Softmax activation function to classify each pixel into one of the predefined categories)
Augmentation and Preprocessing	✗	✗	✗	✗	✓ (Dataset augmented with various transformations; images resized and normalized for uniform input dimensions)

results demonstrate SUNet's superior ability to accurately identify each class, highlighting its effectiveness in precision compared to the other models.

Figure 19 presents a class-wise recall analysis for U-Net, PSPNet, SegNet, and SUNet. The proposed SUNet outperforms the other models by achieving the highest recall rates for each class: 'Cercospora' (0.9884), 'Rust' (0.9878), 'Phoma' (0.9879), 'Miner' (0.9879), and 'Healthy' (0.9878). These results demonstrate SUNet's superior ability to correctly identify and classify each type of coffee leaf disease.

Figure 20 presents a class-wise F1-score analysis for U-Net, PSPNet, SegNet, and the proposed SUNet. The analysis reveals that SUNet outperforms the other models, achieving superior F1-scores across all classes: 'Cercospora' with 0.9869, 'Rust' with 0.9863, 'Phoma' with 0.9864, 'Miner' with 0.9864, and 'Healthy' with 0.9863. These results highlight SUNet's exceptional ability to balance precision and recall for each class, demonstrating its effectiveness in both accurately identifying disease-affected areas and minimizing false positives and false negatives.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed an efficient hybrid deep learning model, SUNet, for the segmentation and classification of coffee leaf diseases. SUNet incorporated U-Net, SegNet, Mask R-CNN, and PSPNet to handle the complexity and variety of disease manifestations. The SegNet architecture, with a VGG16 backbone, ensured accurate feature extraction. U-Net's skip connections provided detailed image descriptions of diseased areas. Mask R-CNN enabled precise instance segmentation, while PSPNet enhanced multi-scale contextual understanding. SUNet demonstrated significant performance improvements compared to other models. It achieved improvements in accuracy, IoU, F1-score, precision, and recall by 1.22%, 1.21%, 1.17%, 1.19%, and 1.24%, respectively. These results show that SUNet efficiently distinguished healthy from diseased regions and accurately classified various types of coffee leaf diseases.

Future work could focus on the optimization of SUNet's performance using various optimizers such as Bayesian optimization and metaheuristics. SUNet could also be implemented on Internet of Things (IoT) platforms for real-time disease detection and management. Further research could explore the SUNet's robustness to noisy and images with complex backgrounds by designing and embedding customized layers into it. Additionally, to achieve more generalized results, SUNet can be used as a transfer learning model for training on other datasets. Finally, SUNet could be applied in a federated learning environment for broader applications.

CONFLICT OF INTEREST

The authors declare no conflict of interest regarding the publication of this article.

REFERENCES

[1] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, p. 939, Jul. 2019.

[2] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosyst. Eng.*, vol. 144, pp. 52–60, Apr. 2016.

[3] J. Boulent, S. Foucher, J. Théau, and P.-L. St-Charles, "Convolutional neural networks for the automatic identification of plant diseases," *Frontiers Plant Sci.*, vol. 10, p. 941, Jul. 2019.

[4] J. G. Arnal Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Eng.*, vol. 180, pp. 96–107, Apr. 2019.

[5] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "Arabica coffee leaf images dataset for coffee leaf disease detection and classification," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107142.

[6] P. S. Thakur, T. Sheorey, and A. Ojha, "VGG-ICNN: A lightweight CNN model for crop disease identification," *Multimedia Tools Appl.*, vol. 82, no. 1, pp. 497–520, Jan. 2023.

[7] Y. Chen, J. Pan, and Q. Wu, "Apple leaf disease identification via improved CycleGAN and convolutional neural network," *Soft Comput.*, vol. 27, no. 14, pp. 9773–9786, 2023.

[8] J. Jiang, H. Liu, C. Zhao, C. He, J. Ma, T. Cheng, Y. Zhu, W. Cao, and X. Yao, "Evaluation of diverse convolutional neural networks and training strategies for wheat leaf disease identification with field-acquired photographs," *Remote Sens.*, vol. 14, no. 14, p. 3446, Jul. 2022.

[9] S. Khan and M. Narvekar, "Novel fusion of color balancing and superpixel based approach for detection of tomato plant diseases in natural complex environment," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3506–3516, Jun. 2022.

[10] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Proc. Comput. Sci.*, vol. 133, pp. 1040–1047, Jan. 2018.

[11] R. Karthik, J. J. Alfred, and J. J. Kennedy, "Inception-based global context attention network for the classification of coffee leaf diseases," *Ecological Informat.*, vol. 77, Nov. 2023, Art. no. 102213. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157495412300242X

[12] A. A. Salamai, "Towards automated, efficient, and interpretable diagnosis coffee leaf disease: A dual-path visual transformer network," *Exp. Syst. Appl.*, vol. 255, Dec. 2024, Art. no. 124490. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424013575

[13] D. B. Marin, G. A. E. S. Ferraz, L. S. Santana, B. D. S. Barbosa, R. A. P. Barata, L. P. Osco, A. P. M. Ramos, and P. H. S. Guimarães, "Detecting coffee leaf rust with UAV-based vegetation indices and decision tree machine learning models," *Comput. Electron. Agricult.*, vol. 190, Nov. 2021, Art. no. 106476. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921004932

[14] L. M. Tassis, J. E. T. de Souza, and R. A. Krohling, "A deep learning approach combining instance and semantic segmentation to identify diseases and pests of coffee leaves from in-field images," *Comput. Electron. Agricult.*, vol. 186, Jul. 2021, Art. no. 106191. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921002088

[15] J. V. Y. B. Yamashita and J. P. R. R. Leite, "Coffee disease classification at the edge using deep learning," *Smart Agricult. Technol.*, vol. 4, Aug. 2023, Art. no. 100183. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375523000138

[16] M. Nawaz, T. Nazir, A. Javed, S. T. Amin, F. Jeribi, and A. Tahir, "CoffeeNet: A deep learning approach for coffee plant leaves diseases recognition," *Exp. Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121481. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423019838

[17] B. M. Abuhayi and A. A. Mossa, "Coffee disease classification using convolutional neural network based on feature concatenation," *Informat. Med. Unlocked*, vol. 39, Jan. 2023, Art. no. 101245. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914823000874

[18] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, Oct. 1959.

[19] H. Inoue, "Data augmentation by pairing samples for images classification," 2018, *arXiv:1801.02929*.

[20] A. Johannes, A. Picon, A. Alvarez-Gila, J. Echazarra, S. Rodriguez-Vaamonde, A. D. Navajas, and A. Ortiz-Barredo, "Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case," *Comput. Electron. Agricult.*, vol. 138, pp. 200–209, Jun. 2017.

[21] R. Krohling, J. G. Esgario, and J. A. Ventura, "BRACOL—A Brazilian arabica coffee leaf images dataset to identification and quantification of coffee diseases and pests," Tech. Rep., 2019, doi: 10.17632/yy2k5y8mxg.1.

[22] R. Krohling, J. E. T. de Souza, and L. M. Tassis, "BRACOT—A Brazilian arabica coffee tree images dataset for instance segmentation of coffee leaves. Mendeley data," Tech. Rep., 2021, doi: 10.17632/pmkbyjpf6k.1.

[23] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *Handbook of Brain Theory and Neural Networks*, vol. 3361, 1995.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[25] P. Wen, Y. Guan, J. Li, X. Xu, H. Peng, J. Yang, Y. Jia, X. Xie, M. Li, X. Wang, Y. Xin, and Y. He, "A-PSPNet: A novel segmentation method of renal ultrasound image," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 40–45.

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3431–3440.

[28] J. Parraga-Alava, K. Cusme, A. Loor, and E. Santander, "RoCoLe: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition," *Data Brief*, vol. 25, Aug. 2019, Art. no. 104414. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340919307693