IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 August 2023, accepted 3 September 2023, date of publication 7 September 2023,
date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312718

RESEARCH ARTICLE

Cascading Autoencoder With Attention Residual U-Net for Multi-Class Plant Leaf Disease Segmentation and Classification

S. ABINAYA, KANDAGATLA UTTEJ KUMAR, AND A. SHERLY ALPHONSE
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India

Corresponding author: S. Abinaya (s.abinaya@vit.ac.in)

This work was supported by the Vellore Institute of Technology, Chennai, for the article processing charge (APC).

ABSTRACT Plant leaf diseases pose a significant threat to global food security and cause substantial economic losses. The objective of this study is to develop an effective approach for early detection and accurate identification of plant leaf diseases using computer vision techniques. The proposed method, Cascading Autoencoder with Attention Residual U-Net (CAAR-UNet), leverages deep learning to achieve precise segmentation and classification of plant leaf diseases. By cascading Symmetric Autoencoders with Attention Residual U-Net model and training on a custom dataset, it surpassed existing methods in identifying four disease classes. The model achieves remarkable accuracy, with a mean pixel accuracy of 95.26% and a weighted mean intersection over union of 0.7451, accurately capturing individual pixels and delineating disease class boundaries. This approach holds great potential in facilitating early plant disease detection and improving crop management practices. Its adoption can significantly impact food security worldwide, addressing a critical gap in the agricultural sector. The results highlight the effectiveness of the proposed strategy in plant disease management and open the door for further research in this field.

INDEX TERMS Plant disease, semantic segmentation, classification, symmetric autoencoder, attention residual U-Net.

I. INTRODUCTION

Accurate segmentation and classification of plant-borne illnesses is vital for smart farming, which aims to detect and diagnose diseases early and improve crop yield while reducing losses. This motivation stems from the significant impact that plant diseases have on global food security and economic stability. Early detection and accurate identification of these diseases can enable timely interventions, reducing crop damage and ensuring efficient management of agricultural resources.

Deep learning models have demonstrated great potential in the field of plant leaf disease segmentation and classification. However, there are challenges that need to be addressed to ensure their effectiveness and practical applicability. One such challenge is the need to improve the accuracy and generalizability of these models to handle variations in plant leaves and diseases, as well as noisy and low-quality images commonly encountered in real-world scenarios.

To address these challenges, previous researchers have explored different models for plant-borne illness segmentation and categorization. For instance, DenseNet, a CNN-based model, achieved remarkable accuracy in cassava disease recognition [1]. Attention U-Net, a medical image segmentation method [2], successfully identified the pancreas in low-contrast CT scans. However, these studies have limitations in terms of the scope of plant diseases studied and the robustness of the models. There is a need for more advanced models capable of handling variations in plant leaves and diseases, as well as noisy and low-quality images.

These challenges have been overcome in models utilizing deep learning techniques to identify and diagnose diseases present in the leaves of plants by introducing a novel approach that uses a custom dataset containing four distinct disease

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

classes. The method involves a hybrid architecture called the Cascading Autoencoder with Attention Residual U-Net (CAAR-UNet), which integrates attention mechanisms and residual connections in the segmentation module. By doing so, the model can capture more informative features and mitigate the effects of noise and variations in input images. Moreover, the symmetric autoencoder architecture in the segmentation module allows the model to acquire hierarchical visual features obtained from images used as input, resulting in more robust performance in detecting plant leaves and diseases.

The effectiveness of the CAAR-UNet model was evaluated on a custom dataset comprising five classes, including background class. The outcomes of the experiment demonstrate that the model suggested performs better than previous cutting-edge models, achieving an impressive average mean pixel accuracy (Am) of 95.26% and a weighted mean intersection over union (wmIoU) of 0.7451. Notably, the model also exhibits robust performance on noisy and low-quality images, demonstrating its accuracy and generalizability. The results from Table 5 of Section IV demonstrates the effectiveness of the proposed CAAR-UNet model for class-wise plant leaf disease segmentation, particularly for the detection of northern corn leaf blight and coffee leaf miner, which are two of the most economically important plant diseases. The model also shows promise for the detection of strawberry leaf scorch and grape black measles, although further improvements may be possible with more diverse datasets and additional optimization of the model architecture. Overall, the proposed CAAR-UNet model has the potential to enhance the effectiveness and precision of plant disease detection systems, thereby promoting sustainable agricultural management. Furthermore, the CAAR-UNet model's capability to manage imbalanced class distributions and differing numbers of samples in each class is a considerable advantage. This allows the model to generalize better to new and previously unseen data. The class-weighted ratio loss function helped to mitigate the issue of class imbalance, resulting in improved performance in the segmentation of minority classes. This feature is particularly useful in real-world scenarios where acquiring balanced datasets may be challenging.

The main contributions of this research are as follows:

1) Symmetric Autoencoder Preprocessing: Extracting latent features and generating an initial mask image with pixel-level classification for defined classes using a symmetric autoencoder model.
2) Attention Residual U-Net Refinement: Fine-tuning the mask image using an attention residual U-Net model to capture disease patterns and boundaries with attention mechanisms and residual connections more effectively.
3) Patch-based Analysis for Hidden Insights: Utilizing a patch-based analysis strategy to analyze images at a local level, uncovering finer details and contextual information.
4) Model Training and Evaluation: Training and evaluating the CAAR-Unet model on a custom dataset, using various performance metrics and comparing against state-of-the-art methods.

These contributions highlight the innovative approaches used in preprocessing, refinement, analysis, and evaluation stages of the proposed model, showcasing advancements in the field of plant leaf disease detection and segmentation. This research fills an important need in the agriculture industry by giving a more accurate and efficient way to segment and classify plant leaf diseases. This can be used in precision agriculture and for managing plant leaf diseases. The proposed model has demonstrated its robustness and effectiveness in handling complex scenarios with multiple disease symptoms and diverse image qualities. The remainder of this paper is organized as follows. Section II describes the related works. Section III explains the proposed methodology, overall workflow, data preparation, data preprocessing, proposed model in detail, including its architectures. Section IV provides evaluation metrics, and experimental results which provide evidence of the model's effectiveness and usefulness. Finally, conclusion and future works are presented in Section V.

II. RELATED WORKS

Over the past few years, the effectiveness of deep learning approaches in accurately detecting and categorizing plant leaf diseases has been increasingly evident. A multitude of studies have been conducted to explore the potential of different deep learning architectures, such as MobineNet, VGG16 and Unet variants, for plant leaf illness identification and categorization. This section examines some of the relevant literature in this field and discusses the benefits and limitations of various methods.

Several studies have employed various forms of the Unet architectures to identify and categorize plant leaf diseases. For example, researchers in [3] proposed a novel method for detecting leaf diseases in soybeans. Their framework utilised a DIM UNet to extract features and an LSTM to classify. Their proposed method achieved high accuracy in soybean leaf disease detection, with an F1-score of 0.96, when tested on a public dataset of soybean leaf images. Similarly, in [4], the authors presented a hybrid deep learning model dubbed "RA-UNet" that employs attention gates to identify liver and tumour regions from CT scans. Combining the UNet architecture with a residual network, the RA-UNet model accomplished state-of-the-art outcomes in liver and tumour detection task. For LiTS dataset, model accomplished a dice coefficient of 0.947 and 0.731 for liver and tumour segmentation.

The authors of [5] developed an automated system to detect and assess the extent of damage caused by grape black measles. For detection, the system utilised Faster R-CNN and DeepLabV3+ models, and for severity analysis, a fuzzy logic system. Their method achieved a disease detection accuracy of 93.5% and a mean absolute error of 0.076 for severity analysis. Likewise, [6] proposed an automated method utilizing MobileNet for detecting and categorizing diseases

present in plant leaves. The authors fine-tuned a pre-trained MobileNet model for disease and pest classification using transfer learning. The overall accuracy of their method for identifying diseases and pests was an impressive 97.84%. The authors of [32] presented SDDNet, a deep learning method for segmenting concrete cracks in images. SDDNet achieved real-time performance and effectively handles complex backgrounds and crack-like features. The model consists of standard convolutions, DenSep modules, a modified ASPP module, and a decoder module. It is trained on a manually created crack dataset and achieves an mIoU of 0.846 on the test set. SDDNet outperforms recent models with significantly fewer parameters and processes images in real-time (36 FPS) at a resolution of 1025 $\times$ 512 pixels, making it a promising approach for practical crack segmentation applications.

Numerous complex architectures have been investigated by researchers in the domain of identifying and categorizing diseases found in plants in an effort to increase comprehension. The authors of [7] reviewed recent research on deep learning-based approaches intended for detecting and categorizing diseases in plants. They analysed various architectures and methodologies used in the field, identified challenges, and proposed potential future research directions. Their analysis highlighted the need for larger and more diverse datasets, robust and scalable models, and the incorporation of intelligent agriculture systems. Overall, this paper provided useful information about the current well-known methods in employing deep learning methods to detect and categorize plant diseases.

Several studies have investigated the use of backbones in models for segmentation and classification of plant leaf diseases. In [10], the authors proposed an automated architecture for banana leaf diseases based on segmentation and classification using deep learning techniques such as CNNs and transfer learning. Their method achieved a high degree of disease detection precision, achieving an overall accuracy of 96%. Similarly, in [11], the authors introduced a DCNN backbone for the recognition of rice plant diseases and pests by video detection and deep convolutional neural networks. Transfer learning was used to refine a pre-trained VGG16 model for disease and pest classification, resulting in an overall accuracy of 93.9 percent. Both papers evaluated their proposed methods on their respective datasets of banana leaf and rice plant videos, demonstrating promising disease and pest recognition outcomes. Overall, employing backbones in deep learning models for detecting and categorizing plant diseases has potential to significantly enhance precision and performance.

In [8], the authors introduced an edge-based coffee disease classification method based on deep learning. Transfer learning was utilised to refine the parameters of a pre-existing MobileNet model for coffee disease classification on edge devices, achieving an impressive 98.7% accuracy. The authors of [9] proposed a two-stage cascade model for segmenting MRI brain tumours. The model was tested on the BraTS dataset and combined variational autoencoders and attention gates to achieve high accuracy and outperformed several other well-known segmentation approaches. The hybrid architecture, comprising both a cascaded autoencoder and a CNN, has been shown to be advantageous for classification and segmentation.

Numerous research has studied the use of multi-stage and hybrid architectures for segmentation of plant-borne diseases. For instance, the authors of [12] presented a CRUN-based architecture for segmentation and identification of leaf disease stages. Their model acquired a high F1-score of 0.95 for segmenting leaf diseases and successfully identified the disease stage using morphological characteristics. To address the difficulty of detecting various illnesses on distinct plant parts, the authors of [13] suggested a CNN-CRF hybrid model for plant disease recognition. Using a publicly available dataset, the model attained an average accuracy of 95.5%, indicating its efficacy in plant disease recognition. The model was implemented on a PC with a Tesla P100 GPU and 27 GB of RAM, and the processing time required for inference could range from several seconds to a few minutes.

The authors of [14] proposed a two-stage segmentation technique based on deep learning and corn field data for crop disease quantification. In the first step of the model, modified Unet was utilised to segment leaves, followed by a DeepLabV3+ model for segmenting disease lesions. Their method achieved best accuracy in segmentation and classification, with an F1-score of 0.9 for segmentation and a classification accuracy of 93.8%. They suggested a deep learning-based method for image segmentation and classification to recognize tomato plant diseases in [15]. They utilised a modified Unet with InceptionNet architecture to accurately divide and classify leaf pictures into illness groups. With an average F1 score of 0.96 across three distinct tomato plant diseases, their model demonstrated a high degree of accuracy. The success of these types of hybrid model served as inspiration to incorporate similar design elements into our work.

Some researchers have tried to employ autoencoder-based models for segmenting and classifying plant leaf diseases. For instance, in [16], the authors developed an autoencoder-based model for diagnosing agricultural diseases and assisting treatment recommendations. The model first uses a cascaded autoencoder to extract features from images of a plant's leaf and then employs a support vector machine (SVM) classifier to categorise the images as either healthy or unhealthy. The model's accuracy in detecting rice and tomato leaf diseases was 96.3% and 95.4%, respectively. This research has limitations as it was only tried out on a small subset of crop diseases. A system for automatic detection and classification of defects on metallic surfaces was proposed using the CASAE model and compact CNNs [17]. In this two-stage process, defects are identified, and then those identified are placed into predetermined categories. In order to pinpoint problem areas on the metal's surface, a cascaded autoencoder network is used

during the defect detection phase. To determine the nature of the flaws within the ROIs, a compact convolutional neural network is employed during the defect classification phase. When applied to a set of images of metallic surfaces, they managed a 93.2% flaw detection rate and a 91.3% defect recognition rate.

The authors of [30] proposed promising deep learning-based approach to enhance colorectal polyp detection and segmentation called PSNet. By combining various deep learning modules, including PS encoder, transformer encoder, PS decoder, enhanced dilated transformer decoder, partial decoder, and merge module, they successfully addressed the challenges of model overfitting, poor boundary pixel definition, and capturing diverse polyp characteristics. Through extensive comparative studies on five existing polyp datasets, PSNet outperforms state-of-the-art results with mDice and mIoU scores of 0.863 and 0.797, respectively. The authors introduced a new modified polyp dataset and achieved significantly improved performance with an mDice of 0.941 and mIoU of 0.897. The authors of [31] introduced STRNet, a novel semantic transformer representation network, for real-time crack segmentation in complex scenes. It addresses deficiencies in previous studies related to ground truth data preparation, complex scene handling, object-specific networks, and evaluation methods. STRNet incorporates attention-based encoder and decoder, coarse upsampling, focal-Tversky loss, and learnable swish activation functions to achieve both speed and accuracy. The network is trained on 1203 images with extensive augmentation and evaluated on 545 images, outperforming advanced networks with the highest evaluation metrics and processing speed (49.2 frames per second). STRNet offers a significant advancement in crack segmentation, providing practical applications in real-time scenarios.

As we explore the possibilities of deep learning approaches for detecting and categorizing plant leaf diseases, it's important to also consider the challenges and limitations that come with these methods. One significant hurdle is the need for large, diverse datasets to effectively train deep learning models. Obtaining representative images of plant leaves with various diseases and environmental conditions can be a real challenge, but it's crucial for building accurate models. Additionally, the computational complexity of deep learning models requires significant computing resources for training, including powerful GPUs and vast amounts of memory. This can make inference on resource-limited devices, such as edge devices in agricultural fields, a challenging task.

In the field of plant leaf disease detection, deep learning models have shown promising results in accurately categorizing and detecting plant diseases. Their application is often limited by the need for large, diverse datasets and the computational resources required for training, there is a need to develop methods to improve model generalization and overcome the challenges of training with small datasets. To address these challenges, a novel architecture, the Cascading Autoencoder with Attention Residual U-Net (CAAR-UNet), has been developed. This architecture employs a two-stage learning process, incorporating a symmetric autoencoder model for preprocessing input images, followed by fine-tuning using the Attention Residual U-Net architecture. To further improve model generalization and robustness, patch-based approach and data augmentation techniques were utilized. By dividing the images into smaller patches and applying data augmentation, the model can learn from a larger and more diverse set of examples, improving its ability to accurately classify and detect plant diseases. The proposed CAAR-UNet model was demonstrated on a custom dataset of four commonly occurring plant leaf disease classes and has shown its applicability in multi-class plant leaf disease segmentation. Through continued exploration of alternative methods, such as ensemble methods, transfer learning, image processing, and hyperspectral imaging, deep learning-based plant leaf disease detection can become even more precise and effective in agricultural settings.

III. PROPOSED METHODOLOGY

This section contains an overview of the proposed work; the overall workflow, data preparation, data preprocessing, and the architecture of the CAAR-UNet model are discussed.

A. OVERALL WORKFLOW

Fig. 1, illustrates the overall process of the method for detecting and classifying diseased regions in plant leaf images, including northern corn leaf blight, grape black measles, strawberry leaf scorch, and coffee leaf miner. To accomplish this task, publicly available Plant Village dataset [27] is utilized, which contains a diverse array of labelled images. Prior to being input into the proposed model, the images underwent enhancement using a variety of techniques to ensure highly accurate detection of diseased regions. Data preprocessing task includes the use of the Patchify library and various data augmentation methods to bring out more finer details and contextual informative features from the scalable image data. Comprising two components, the proposed model first employs a symmetric autoencoder to enhance the predicted image, without any irrelevant information (no leaf backgrounds). The enhanced predicted image (consists only background class and pixel wise predicted classes) is then fed as input into the attention residual U-Net, which outputs classified and segmented regions of disease on plant leaves.

B. DATASET DESCRIPTION

In this study, publicly available datasets, namely the PlantVillage Dataset [27] and the Coffee Leaf Dataset [28], were utilized to create a custom dataset for training and evaluation. The PlantVillage Dataset contains a collection of 54,303 healthy and unhealthy leaf images divided into 38 groups, representing different plant species and diseases. From this dataset, we selected 100 images per class from the classes Northern Corn Leaf Blight, Grape Black Measles, and Strawberry Leaf Scorch to represent distinct diseases across multiple species.

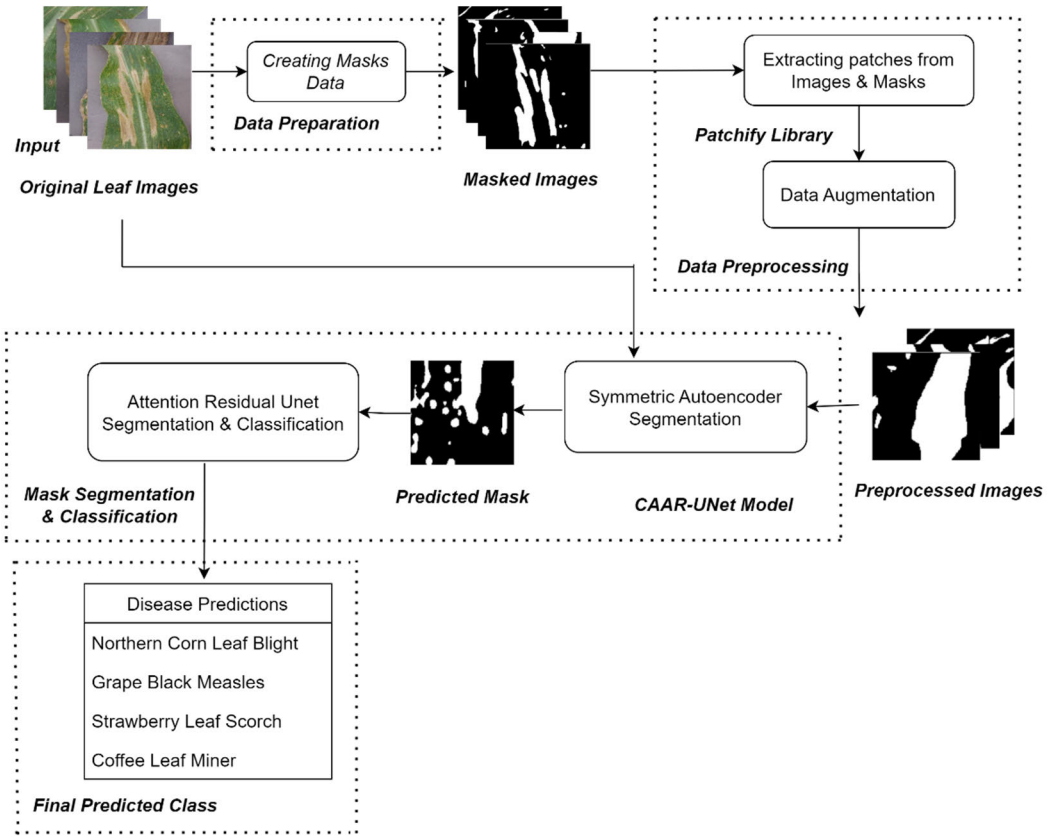

FIGURE 1. Overall proposed workflow.

TABLE 1. Custom data statistics.

Dataset	Plant Leaf Class	Image Count
PlantVillage Dataset	Northern corn leaf blight	100
	Grape black measles	100
	Strawberry leaf scorch	100
Coffee Leaf Dataset	Coffee leaf miner	100
	Total	400

Additionally, we incorporated the Coffee Leaf Dataset [28], which consists of coffee leaf images with two classes: rust and miner. From this dataset, we selected 100 images specifically belonging to the coffee leaf miner class.

Overall, our custom dataset comprises a total of 400 RGB images, each with a resolution of 256 × 256 pixels. By combining images from different classes and datasets, we aimed to create a diverse and representative dataset that encompasses various plant species and disease types, enabling comprehensive training and evaluation of our proposed model. Table 1 showcases the distribution of images across different disease classes, providing a comprehensive overview of the composition of the dataset.

C. DATA PREPARATION

The Apeer platform [29] is utilised in order to annotate diseased regions that were visible in the plant leaf images. Via its user-friendly interface, the Apeer platform provides access to the most cutting-edge deep learning algorithms currently available for image segmentation. Great caution was employed to mark and highlight the diseased regions of the plant leaf images using this platform, and then retrieved the corresponding masked images for each category. Fig. 2, displays some examples of plant leaf images alongside their respective leaf masks that were obtained from apeer platform.

The model is trained using the Apeer platform's produced labels as the ground truth. As the process of marking and highlighting exact shape and size of unhealthy parts in each image needs a great deal of time and attention, it was not possible to generate these annotated images in huge quantities. The techniques described in sub-section D will be utilized to improve the model's generalization and robustness. These methods allow the model to learn from a more extensive and diverse set of examples, resulting in increased accuracy in detecting and classifying plant diseases.

D. DATA PREPROCESSING

In the preprocessing phase, patch-based approach with patchify library and data augmentation techniques are implemented to facilitate the learning process of the model. By dividing the input images into smaller patches and

TABLE 2. Data augmentation techniques applied.

Technique	Description	Value
Horizontal and Vertical Flips	Flipping the visual input horizontally or vertically to make the model robust to diverse leaf orientations and directions.	Applied to all images
Random Rotation	Applying random rotations to assist the model in understanding leaf segmentation in various orientations and angles.	Random rotation angles: -45° to 45°
Random Cropping	Randomly cropping and resizing images to help the model learn leaf segmentation for different sizes and shapes.	Crop size range: 70-90% of original size
Grid Distortion	Distorting images by adjusting a grid placed over them, generating distorted images to enhance model performance.	Random shift of grid control points up to 10 pixels

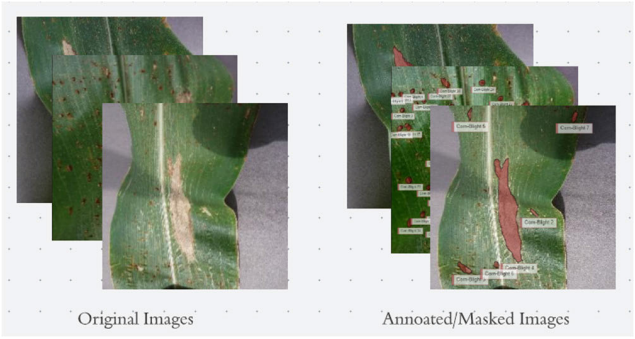

FIGURE 2. Creating masks by highlighting and labelling the diseased regions using apeer platform.

applying data augmentation, the model can learn from a more diverse and extensive dataset, which can bring out more latent features within the images and improve model's ability to generalize and make accurate predictions on new, unseen data. Furthermore, patch-based approach and data augmentation reduces the computational burden and makes it possible to handle datasets more efficiently.

Scalable data processing techniques were employed to effectively fine-tune the dataset consisting of 400 images. Utilized the patchify library to extract multiple smaller, non-overlapping patches from each original image, resulting in 4 non-overlapping patchified images of size (128, 128) from each original image of size (256, 256) as presented in subsection I and Fig. 3. Applying data augmentation techniques to the patches extracted from the images resulted in an increased dataset diversity and complexity, as outlined in Table 2. For each patchified image of size (128, 128), five additional images of the same size were generated, resulting in a total of 9,600 images. This augmentation approach greatly expanded the dataset, providing a more diverse and extensive set of examples for training. To ensure a rigorous evaluation, the test set data is separated from the entire dataset at the outset of the experiment. The remaining data is then utilized for the 10-fold cross-validation process. In each fold of the cross-validation, the training and validation data are altered while maintaining a constant size. Specifically, the dataset is divided into ten subsets or folds, with each fold serving as the validation set once while the other nine folds are used for training.

This process is repeated ten times, ensuring that every data point is included in the validation set exactly once and is part of the training set in nine out of the ten folds. By consistently maintaining the size of the training and validation sets across all folds, we aim to comprehensively assess the model's performance, reduce the risk of overfitting, and yield meaningful and reliable results for our proposed method. Table 3 complements this information by presenting the distribution of data across the training, validation, and testing sets for each category class, allowing for a comprehensive understanding of the dataset composition and partitioning.

1) PATCHIFICATION BASED ANALYSIS

The patchify library was utilized to partition the input images into smaller, non-overlapping patches of size (128 × 128) from each original image of size (256 × 256). This approach enabled the model to uncover finer details and contextual information within the images and detect subtle variations that may not be visible in the original image. The extraction of multiple patches from each image resulted in a more diverse dataset and reduced computational burden, improving the model's learning ability and allowing the model to handle datasets more efficiently. Patchified images are depicted in Fig. 3 and patchify library is applied as mentioned in (1).

$$patchify\left(image_{to_{patch}}, patch_{shape}, step = no.of \;\; steps\right) \quad (1)$$

where the original images from the dataset, denoted as $image_{to_{patch}}$, and these were divided into patches of size (128, 128, 3) for leaf images and (128, 128) for mask images using the $patch_{shape}$ parameter, with no overlapping between patches as defined by the *step* parameter set to 128.

TABLE 3. Data statistics after patch extraction and data augmentation.

Plant Leaf Image	Total Samples	Train Data Size	Test Data Size	Validation Data Size
Northern corn leaf blight	2400	1920	240	240
Grape black measles	2400	1920	240	240
Strawberry leaf scorch	2400	1920	240	240
Coffee leaf miner	2400	1920	240	240
Total	9600	7680	960	960

FIGURE 3. Images after applying patchify library to the dataset.

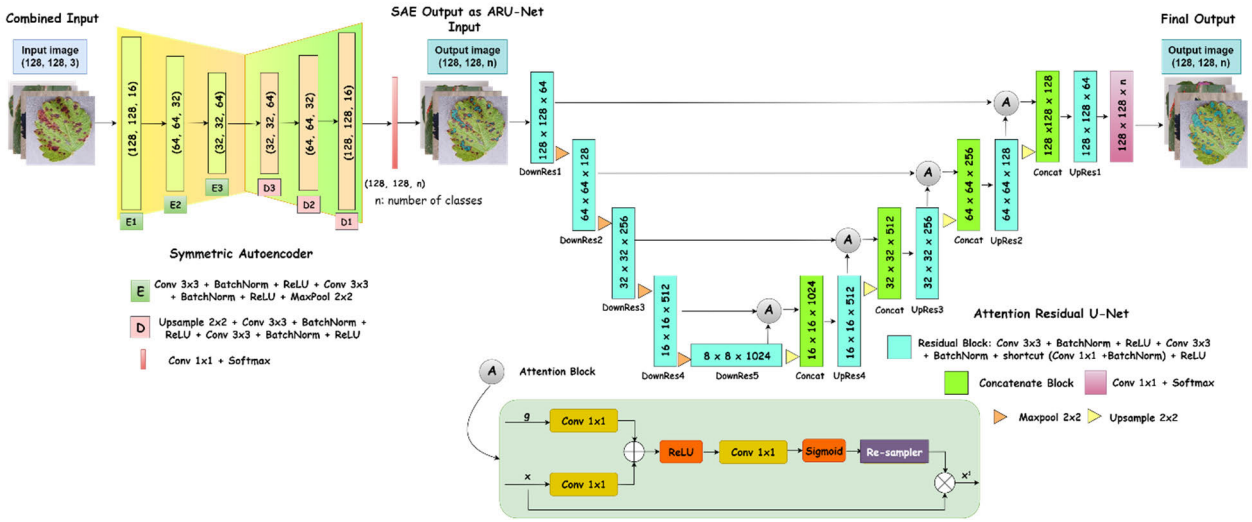

FIGURE 4. Architecture of the proposed CAAR-UNet model.

E. CASCADING AUTOENCODER WITH ATTENTION RESIDUAL U-NET (CAAR-UNET)

The Cascading Autoencoder and Attention Residual U-Net (CAAR-UNet) model is a novel deep learning model proposed for plant leaf disease segmentation and classification. The proposed model architecture was chosen based on the results of studying numerous deep learning-based model architectures [2], [4], [9], [16], [20], [24], which led to the adoption of the Attention Residual U-Net model architecture and Autoencoders. The model architecture provides a promising method for image segmentation problems, with enhanced efficiency, less overfitting, and greater generalisation capabilities. The resulting model takes an image of a plant leaf as input and produces a segmented image with diseased areas highlighted and classified.

The proposed CAAR-UNet model is formed by cascading two separate models, the Symmetric Autoencoder (SAE) [9], [17] and Attention Residual U-Net (ARU-Net) [4], [24] as presented in Fig. 4, initially they were defined separately for compilation and training purpose. Then, a new input layer is defined for the combined hybrid model outlined in Fig. 4 to match the input shape of the separate models from Fig. 7, 10. The SAE model takes preprocessed RGB leaf images of shape (128, 128, 3) as input and outputs a pixel-wise predicted mask of shape (128, 128, n), where n is the number of classes to be predicted, in our case it is five classes including the background class. This output is then fed into the ARU-Net model, which further refines the predicted mask and produces a final result of the same shape (128, 128, n). The training process involves training the SAE and ARU-Net models separately using established training techniques, and then combining them to form the CAAR-UNet model. Pseudocode of the proposed CAAR-UNet model is given in the Appendix for more information.

The combined model is then trained on custom dataset using established training techniques from Table 4. After training phase, the predict method is used to apply the model to new samples of unseen data to test generalisability our model. Overall, the proposed CAAR-UNet model leverages the strengths of each component to achieve improved segmentation accuracy.

A more powerful model that can tackle a broader set of segmentation problems is achieved by combining their architectures. For instance, the Symmetric autoencoder excels at images with multiple objects and backgrounds, but it may have trouble with objects of wildly varying sizes and shapes. While ARU-Net performs well when presented with objects of wildly varying sizes and shapes,

it may struggle when presented with complex backgrounds and overlapping objects. A model can be constructed that is more versatile and effective across a wider range of image segmentation problems by fusing the two architectures. In conclusion, the proposed model, Symmetric Autoencoder cascading with Attention Residual U-Net architecture, is a robust and versatile technique for plant-borne disease detection and categorization, with the ability to learn useful features from the input image and focus on important regions for the segmentation task.

1) SYMMETRIC AUTOENCODER (SAE)

The Symmetric autoencoder's structure is established in the first phase of the proposed approach. This is a neural network that takes an input image and reduces it to a lower-dimensional format before reconstructing it. This method enables the network to learn valuable image characteristics that can be used for subsequent segmentation. The symmetric autoencoder is made up of numerous layers of an encoder network that transform the input image into a low-dimensional feature vector and a decoder network that transform the feature vector into an output image.

The Symmetric autoencoder network consists of three encoder blocks and three corresponding decoder blocks. The architecture design of the SAE model is presented in Fig. 7. The mathematical representation of the symmetric autoencoder architecture can be expressed as follows.

a: ENCODER BLOCKS

Each encoder block in this setup is tasked with maximising the number of feature maps while decreasing the spatial dimensions of the input data. In order to reduce the dimensionality of the input data while maintaining important features, each encoder block uses two layers of convolution with varying filter sizes, a batch normalization layer, a max pooling layer with ReLU activation function in between. Each block's forward flow is shown in (2).

$$h_k = f_k\left(W_k * h_{k-1} + b_k\right), \quad k = 1, 2, 3 \tag{2}$$

where h_{k-1} is the result of the preceding encoder block or the input x if k=*1*, W_k and b_k are the weights and biases of the encoder block indexed k, and f_k is the activation function of the encoder block indexed k. The architecture design of encoder block in SAE model is presented in Fig. 5.

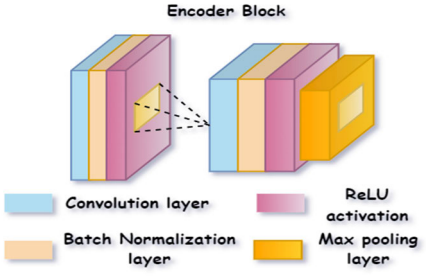

FIGURE 5. Architecture design of encoder block in SAE.

b: DECODER BLOCKS

Each decoder block, on the other hand, performs the reverse operation of the corresponding encoder block. It takes the lower-dimensional representation produced by the encoder block and transforms it back into the original spatial dimensions while decreasing the number of feature maps.

This is achieved by applying each decoder block with upsampling layers to expand the spatial size of the feature maps, followed by two convolutional layers with the same number of filters as in the corresponding encoder block. The last decoder block is subsequently a 1×1 convolutional layer with a softmax activation function that produces a segmentation mask with a number of class channels. The forward flow of each block is represented in (3).

$$g_k = f_k\left(W_k * g_{k-1} + b_k\right), \quad k = 1, 2, 3 \tag{3}$$

where g_{k-1} is the outcome of the preceding decoder layer or the result of the k-th encoder layer if k=*3*, W_k and b_k are the weights and biases of the indexed k decoder layer, f_k is the activation function of the k-th decoder layer. The architecture design of decoder block in SAE model is presented in Fig. 6.

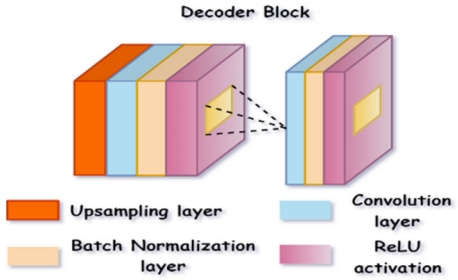

FIGURE 6. Architecture design of decoder block in SAE.

c: FINAL LAYER

The output layer can be represented as shown in (4).

$$y = softmax\left(W_{out} * g_1 + b_{out}\right) \tag{4}$$

where W_{out} and b_{out} are the output layer's weights and biases, and y is the predicted output of the symmetric autoencoder.

d: LOSS FUNCTION

The Symmetric autoencoder strives to minimize the dissimilarity between the input data x and its anticipated output y, as gauged by the reconstruction loss expressed in (5).

$$L(x, y) = -sum_i\{x_i * log(y_i)\} \tag{5}$$

where x_i and y_i are the *i-th* elements of x and y, respectively, and sum_i denotes the sum over all elements of x and y.

By minimizing the reconstruction loss across all building blocks, the symmetric autoencoder aims to achieve optimal performance, as reflected in (6), which involves summing up the reconstruction loss for each block.

$$J(x, y) = sum_k\{L(h_{k-1}, g_k)\} \tag{6}$$

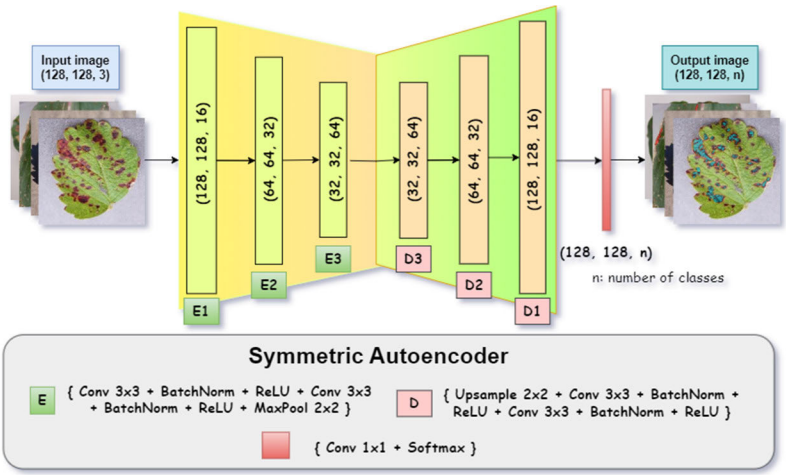

FIGURE 7. Architecture of symmetric autoencoder in CAAR-UNet.

where h_{k-1} is the outcome of the *(k-1)-th* encoder block, g_k is outcome of *k-th* decoder layer, and sum_k refers to the sum over all building blocks (i.e., *k=1,2,3* in this case).

Overall, encoder blocks aid in the extraction of significant features from input data, while decoder blocks aim to reconstruct the original input data using the extracted features. The cascading of these block allows for deeper feature extraction and reconstruction, which can result in more expressive and accurate models.

During training, the SAE learns to reconstruct the image by mapping it to a lower-dimensional representation and then decoding it back to its original dimensions. In this process, the SAE effectively suppresses noise and irrelevant variations (e.g., lighting variations or leaf background), emphasizing the relevant features necessary for disease segmentation resulting in a cleaner image containing only the predicted class or category of the pixel, such as different types of diseases or background. These pixel values indicate the classification and segmentation information generated by the SAE for further processing and refinement in subsequent stages of the model.

2) ATTENTION RESIDUAL U-NET (ARU-NET)

The second phase of the proposed approach specifies architecture of the Attention Residual U-Net (ARU-Net) model, which is intended for image segmentation. This model's architecture is intended for refinement of sub-section I segmentation task, which includes dividing an image into regions and labelling each region based on its class (e.g., various disease types). The architecture design of the Attention Residual U-Net model is presented in Fig. 10. This model is comprised of convolutional layers that downsample the image, followed by convolutional layers that upsample the image to its original size. In addition, the model includes skip connections to preserve spatial information during downsampling and upsampling operations. In addition, the model includes an attention mechanism that assists the network in focusing on the most relevant and important image regions for segmentation, as well as residual connections in the network that are used to address the vanishing gradient problem that can arise during the process of training networks.

The model's attention gates are defined by the attention block function, which accepts input feature maps and computes attention coefficients using two distinct convolutional layers. The attention coefficients are then passed through a softmax activation function to obtain the attention map, which is then element-wise multiplied using the input feature maps to generate attended feature maps. To achieve the final result, the attended process of concatenating feature maps is performed in conjunction with output of the corresponding decoder block. The Attention Residual U-Net architecture can be represented as follows:

a: DOWNSAMPLING LAYERS

The downsampling block is an important component that aims to reduce the spatial dimensionality of feature maps while increasing the number of channels, leading to the extraction of higher-level features from the input image. It consists of convolutional layers, ReLU activation functions, and max pooling layers, and uses residual blocks to learn more complex representations and extract hierarchical features. The formula for the forward pass of the downsampling block with residual connections is shown in (7).

$$h_k = f\left(W_k * h_{k-1} + b_k + (W_{k-1} * h_{k-1} + b_{k-1})\right), \quad k = 1, 2, \ldots, N \tag{7}$$

The equation (7) represents the computation for the k-th layer of the downsampling block, where h_0 represents the input image, h_{k-1} is the output of the (k-1)-th layer, which is then multiplied by the weights W_k and added to the biases b_k. The result is then passed through an activation function f, which is typically ReLU. In addition to this, there is a residual connection that is added to this output, which is computed by taking the output of the (k-1)-th layer, multiplying it with the weights W_{k-1} and adding the biases b_{k-1}, and then adding

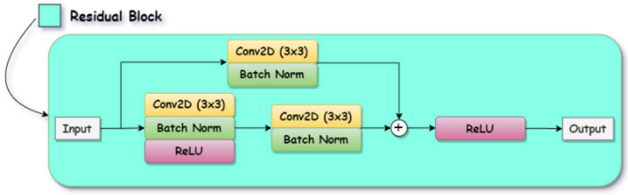

FIGURE 8. Architecture design of residual block in ARU-Net.

this to the output of the k-th layer. This residual connection helps in improving the flow of gradients through the network and enables the network to learn better representations. The architecture design of residual block in ARU-Net network is presented in Fig. 8.

b: ATTENTION MECHANISM

The attention gates are used to selectively focus on the most informative regions of the feature maps while suppressing irrelevant or redundant information. The attention gate consists of two parallel convolutional layers, one for the encoder feature map, downsampling layer and another for the decoder feature map, upsampling layer. The formula for the forward pass of the attention block is shown in (8).

$$\begin{aligned} g &= f\left(AVGPOOL\left(W_g * h_N + b_g\right)\right. \\ &\quad \left. + MAXPOOL\left(W'_g * h_N + b'_g\right)\right) \\ z &= SIGMOID\left(W_z * g + b_z\right) \\ s &= z * h_N \end{aligned} \tag{8}$$

The attention gate g from (8) consists of a global average pooling and a global max pooling operation, followed by two separate convolutional layers (W_g and W'_g) as shown in Fig. 9 with batch normalization and ReLU activation, and a sigmoid activation layer (W_z and b_z) that outputs a scalar value between 0 and 1. The output of the attention gate s is obtained by element-wise multiplication of the sigmoid output z and the output of the last convolutional layer of the corresponding downsampling block h_N.

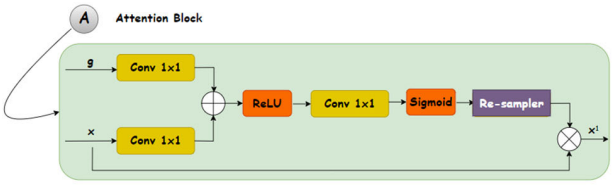

FIGURE 9. Architecture design of attention block in ARU-Net.

c: UPSAMPLING LAYERS

Similar to the downsampling block, the upsampling block also uses a series of residual connections to increase the spatial resolution of the feature maps while recovering the spatial details of the input image. The upsampling block usually consists of an upsampling layer, followed by a concatenation with the corresponding encoder feature map (from the same resolution). The formula for the forward pass of the upsampling is shown in (9).

$$\begin{aligned} h'_k &= f\left(h'_{k-1} * W'_k + b'_k + \left(h_N * W'_{k-1}\right.\right. \\ &\quad \left.\left. + b'_{k-1}\right) + s\right), \quad k = 1, 2, \ldots, M \end{aligned} \tag{9}$$

The equation (9) represents the computation for the k-th layer of the upsampling block, where h'_{k-1} is the output of the (k-1)-th layer, which is then multiplied by the weights W'_k and added to the biases b'_k. The result is then passed through an activation function f, which is ReLU. In addition to this, there is a double residual connection (i.e. the residual connection between the current layer and the corresponding layer in the encoder, as well as the residual connection between the current layer and the previous layer in the decoder) that is added to this output. The first part of this connection is computed by taking the output of the (k-1)-th layer, multiplying it with the weights W'_{k-1} and adding the biases b'_{k-1}, and then adding this to the output of the corresponding downsampling layer (h_N) that feeds into the current upsampling layer. The second part of this connection is the attention map s, which is added to the output.

d: FINAL LAYER

The final output uses a convolutional layer with an activation function (e.g., *softmax*) to produce a probability map of each pixel belonging to each class. The final layer formula is shown in (10).

$$y = f(h_M * W_M + b_M) \tag{10}$$

where W_M and b_M are the weights and biases of the final convolutional layer of upsampling block, and f is a activation function, which is *softmax*. The output of the final layer y will be a probability distribution over the different classes for each pixel. The class with the highest probability will be selected as the predicted class for that pixel during inference.

e: LOSS FUNCTION

For the purpose of classification tasks, a widely used loss function is cross-entropy loss, which has been employed in this particular study. It quantifies the discrepancy between the predicted probabilities and the true labels. The formula for cross-entropy loss, shown in (11), calculates the negative sum of the logarithm of the predicted probabilities for each pixel.

$$L = -\left(\frac{1}{N}\right) * \sum_{i=1}^{h} \sum_{j=1}^{\omega} \sum_{k}^{c} y_{i,j,k} * \log\left(p_{i,j,k}\right) \tag{11}$$

here, N refers to overall count of pixels contained within the image, $y_{i,j,k}$ represents the ground-truth label assigned to the pixel located at position (i,j) for class k, and $p_{i,j,k}$ denotes the probability assigned to each individual pixel (i,j) for class k. During the training phase, the cross-entropy loss

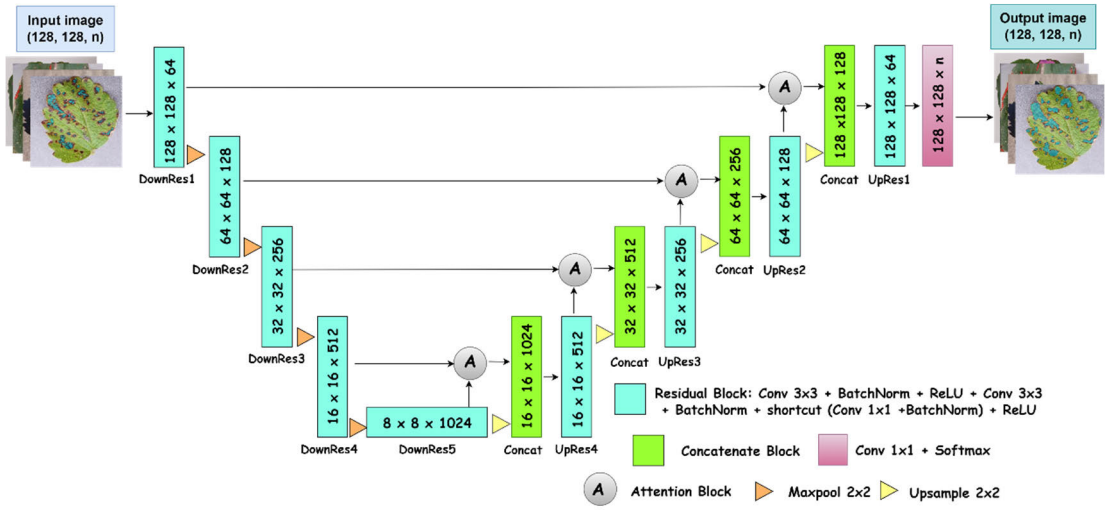

FIGURE 10. Architecture design of attention residual U-Net in CAAR-UNet.

function is utilized to update the model parameters, thereby minimizing the dissimilarity between the predicted and actual labels.

Overall, the Attention Residual U-Net with attention mechanism repeats the downsampling and upsampling layers multiple times, with attention gates inserted between corresponding encoder and decoder blocks. The output of the final upsampling layer is a segmentation map with the same spatial resolution as the input image, where each pixel corresponds to a predicted class label. The use of residual connections allows the gradient to flow more efficiently through the network during training while attention mechanism allows the network to focus on the most informative regions of the input image, which can improve the accuracy of the segmentation results.

The CAAR-UNet model presented in sub-section E is a novel hybrid neural network architecture designed to improve the accuracy of plant disease segmentation. By combining the strengths of a Symmetric autoencoder, an Attention Residual Unet and incorporating patch-based analysis, this model offers a unique approach that builds upon existing concepts. The Symmetric autoencoder preprocesses the input data to extract latent features improving the quality of the input data and lead to better segmentation results, while the patch-based approach and data augmentation techniques help to uncover more finer details and contextual information within the images. The attention mechanism selectively focuses on informative regions, and the use of residual connections improves the flow of gradients during training. Although each of these components has been used in other contexts, their specific combination in the CAAR-Unet model for plant disease segmentation is novel. Overall, this approach has the potential to improve the accuracy of multi-class segmentation tasks and represents an important contribution to the field of plant disease diagnosis and treatment.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The Windows 10 operating system with specifications of NVIDIA Tesla P100 GPU and 16GB RAM was used with the Spyder platform to implement the CAAR-UNet architecture model proposed for plant-borne disease detection. The Keras framework was used alongside a Tensorfow backend. The experimental results showed the best split between training, validation, and testing to be 80/10/10.

The training of the proposed CAAR-UNet model followed a supervised learning approach. The dataset was loaded and preprocessed using the patchify technique, extracting patches of size (128, 128, 3) from the input images, ensuring the images and their corresponding masks are appropriately formatted. These patches served as the input for the model. As the segmentation task involves multiple classes, we have modified the loss function to incorporate the class weight ratio. This modification helps address class imbalance issues and ensures that each class receives appropriate emphasis during training. Next, hyperparameter tuning is performed using 10-fold cross-validation. This allowed to optimize key hyperparameters such as batch size, number of epochs, and learning rate. The inclusion of cross-validation enabled robust evaluation of the model's performance across different subsets of the training data, enhancing its ability to generalize.

The model architecture is then constructed, it is designed to effectively capture spatial features and extract meaningful representations from the input images. Then the model is compiled, specifying the modified loss function with the class weight ratio, along with an appropriate optimizer and evaluation metrics. This ensures that the training process focuses on minimizing the weighted loss, considering the class distribution. During training, the model iteratively processes batches of training images and corresponding masks. Backpropagation and parameter updates occur to minimize the weighted loss and improve the model's performance.

The training process continues for a specified number of epochs, allowing the model to learn complex patterns and refine its segmentation predictions. After training, the model is evaluated on a separate validation dataset. Evaluation metrics mentioned in sub-section A, are calculated to assess the model's performance in accurately delineating the different classes in the validation images.

The inclusion of patch-based preprocessing aimed to enable the model to capture localized patterns within the dataset, enhancing its learning capabilities. By incorporating modified loss function accounting for class weights and using 10-fold cross-validation, our proposed model is trained to effectively segment multiple classes in images. This comprehensive training approach aims to enhance the model's ability to capture class-specific details and generalize well to unseen data. Table 4 displays the hyperparameters that were adjusted throughout the training procedures.

TABLE 4. Hyperparameters utilized in training process.

Hyperparameter	Value
Optimizers	SGD, Adam, RMSProp
Learning rate	0.05
Early stopping	Yes
Epochs	100
Batch size	64
Loss function	Categorical cross entropy
Momentum	0.9

During the training stage, we tested three different optimizers: Adam, SGD, and RMSprop. Among them, the CAAR-UNet model trained with the Adam optimizer achieved the highest segmentation accuracy. Therefore, all the comparisons and results presented in the tables are based on the CAAR-UNet model trained specifically with the Adam optimizer. We utilized a batch size of 64 and trained the model for over 100 epochs. The evaluation of the model's performance was conducted on the validation dataset, yielding promising results. This confirms the effectiveness of our approach for image segmentation tasks using the CAAR-UNet model and the Adam optimization algorithm.

A. EVALUATION METRICS

The performance of the CAAR-UNet model for multi-plant leaf disease segmentation and classification can be evaluated using various metrics. The basic concepts of some of these performance metrics along with the mathematical details are presented in the following subsections.

1) MEAN PIXEL ACCURACY

The mean pixel accuracy for multi-class segmentation is the average pixel accuracy over all images and classes. It measures the proportion of correctly classified pixels across all the classes and provides a measure of how well the model can perform multi-class segmentation. For image i in the data, (12) presents the pixel accuracy for each class j.

$$A_{ij} = \frac{(TP_{ij} + TN_{ij})}{(TP_{ij} + TN_{ij} + FP_{ij} + FN_{ij})} \tag{12}$$

In the aforementioned context, the notation TP refers to the count of true positives, TN denotes true negatives, FP represents false positives, and FN indicates false negatives.

The mean pixel accuracy (A_m) is the average of overall images and classes, and it is represented in (13).

$$A_m = \frac{1}{n} * \sum_{i=1}^{n} \sum_{j=1}^{C} A_{ij} \tag{13}$$

In this context, the variable n corresponds to count of overall images contained within the dataset, and c is the count of overall classes.

2) WEIGHTED MEAN INTERSECTION OVER UION

Weighted mean Intersection over Union (IoU) is a common evaluation metric for image segmentation that considers both the pixel-wise accuracy and the class distribution in the ground truth. The weighted mean IoU formula can be expressed as follows:

For each class j (j = 1, 2, …, c) and each image i (i = 1, 2, …, n), the IoU score is presented as (14).

$$IoU_{ij} = \frac{TP_{ij}}{(TP_{ij} + FP_{ij} + FN_{ij})} \tag{14}$$

The per-class IoU by averaging the IoU scores over all images for each class is presented in (15)

$$IoU_j = \frac{1}{n} * \sum_{i=1}^{n} IoU_{ij} \tag{15}$$

The class weights w_j expressed as a percentage of total ground truth pixel count for class j is presented in (16).

$$w_j = \frac{P_j}{\left(\sum_{j=1}^{c} P_j\right)} \tag{16}$$

where P_j is the total count of pixels for class j in the ground truth segmentation masks.

The weighted mean IoU as the weighted mean of the per-class IoU scores using the class weights is represented in (17).

$$wmIoU = \sum_{j=1}^{c} w_j * IoU_j \tag{17}$$

The weighted mean IoU is obtained by computing the average IoU score across all classes while considering the proportional representation of pixels for each class in the ground truth. This accounts for class imbalance in the dataset and gives more weight to the classes with more pixels in the ground truth.

3) MEAN BOUNDARY F1-SCORE

Mean boundary F1-score is a measure of the accuracy of the detected boundaries, calculated as the F1 score of the boundary pixels. Mathematically, precision, recall and F1 score for class j, w_j is the count of boundary pixels per class j, is represented in (18).

$$\begin{aligned} precision_j &= \frac{TP_j}{(TP_j + FP_j)} \\ recall_{jj} &= \frac{TP_j}{(TP_j + FN_j)} \\ F1_j &= \frac{2 * precision_j * recall_j}{(precision_j + recall_j)} \end{aligned} \tag{18}$$

The weighted mean BF Score across all classes and images is represented as in (19).

$$wmBF1 = \frac{1}{n} * \sum_{j=1}^{C} w_j * F1_j \tag{19}$$

The Weighted Mean Boundary F1 score ($wmBF1$) provides a more balanced evaluation of the boundary detection algorithm's performance across different classes, by considering the count of boundary pixels per class.

4) DICE COEFFICIENT

The Dice coefficient is used to evaluate how close an algorithm's prediction is to the true segmentation. Mathematically dice coefficient for class j is represented as in (20).

$$D_j = \frac{2 * TP_j}{(2 * TP_j + FP_j + FN_j)} \tag{20}$$

The weighted mean Dice coefficient across all classes and images is represented as in (21).

$$wmDC = \frac{1}{n} * \sum_{j=1}^{C} w_j * D_j \tag{21}$$

The $wmDC$ ranges from 0 to 1, with 1 indicating perfect correlation between what was expected and what really happened. A higher Dice coefficient indicates better segmentation performance.

The evaluation of the CAAR-UNet model's efficacy can be conducted using a combination of these metrics to assess its ability to accurately segment and classify multi-plant leaf diseases.

B. PERFORMANCE ASSESSMENT: EXISTING ARCHITECTURES VS PROPOSED MODEL

The purpose of this section was to evaluate distinct network architectures for the task of segmenting plant leaf diseases. Images of plant leaves were used to test the effectiveness of the architectures such as Mask R-CNN, Symmetric Autoencoder, U-Net, Attention U-Net, SegNet, SDDNet, Attention Residual U-Net, STRNet, PSNet and Cascading Autoencoder with Attention Residual U-Net. The loss function was updated by including a class weighted ratio to enhance the models' performance. This method assisted in resolving the problem of class imbalance within the dataset, where some classes had significantly fewer samples than others. The effectiveness of the trained and tested models was evaluated by analyzing the segmentation results, as presented in Table 5. Various metrics, such as average pixel accuracy, weighted mean Intersection over Union (IoU), weighted mean dice coefficient, and weighted mean boundary F1-score, were used to measure the performance of each model.

TABLE 5. Semantic segmentation of multi class plant leaf diseases.

Network	A_m	$wmIoU$	$wmDC$	$wmBF1$
Mask R-CNN	0.9068	0.5851	0.5503	0.5278
SAE	0.8952	0.5637	0.5358	0.5102
U-Net	0.9074	0.6285	0.5425	0.5170
Attention U-Net	0.9148	0.6819	0.5734	0.5224
SegNet	0.9102	0.6725	0.5607	0.5301
SDDNet	0.9185	0.7028	0.5793	0.5308
ARU-Net	0.9287	0.7179	0.5831	0.5431
STRNet	0.9246	0.7082	0.5803	0.5478
PSNet	0.9341	0.7274	0.6027	0.5386
CAAR-UNet	**0.9526**	**0.7451**	**0.6176**	**0.5554**

The outcomes demonstrated in Table 5 establish the supremacy of the proposed CAAR-UNet model over the individual SAE and UNet models and also existing models in the task of plant-borne disease segmentation and categorization. The proposed model achieved highest $wmIoU$ of 74.51 percent and $wmDC$ of 61.76 percent, indicating that in most of the cases its predictions are more accurate and closely aligned with the ground truth. This improvement in performance can be attributed to the use of Symmetric autoencoder architecture as a preprocessing step to extract high-level features from the input images, reducing irrelevant information and improving the relevance of visual features. The attention residual unet model may have been able to learn more effective feature representations from the output of the symmetric autoencoder due to the presence of residual connections and attention mechanisms in its architecture. These mechanisms allow the model to focus on relevant parts of the input features and facilitate the flow of information through the network, leading to more effective feature representations and further improvement in the model's segmentation accuracy, leading to more accurate segmentation.

While the individual SAE, UNet, and Mask R-CNN models may have struggled to effectively capture high-level contextual information and learn effective feature representations, the attention unet and attention residual unet models may have faced limitations in capturing complex disease interactions and mitigating the vanishing gradient problem. The proposed hybrid CAAR-UNet model overcomes these limitations and achieves improved segmentation accuracy by leveraging the strengths of both the symmetric autoencoder and attention residual unet architectures.

TABLE 6. Class-wise performance analysis of experimented segmentation models.

Network	Class	$A_{m,class}$	IoU	DC_{class}	$mBF1_{class}$
Mask R-CNN	nclb	0.9046	0.6823	0.6105	0.5417
	gbm	0.6453	0.4845	0.5301	0.4352
	sls	0.7480	0.5312	0.5284	0.4553
	clm	0.8671	0.5847	0.5982	0.5268
	background	0.9527	0.8156	0.5806	0.4925
SAE	nclb	0.8957	0.6716	0.5923	0.5624
	gbm	0.6235	0.4732	0.5238	0.4352
	sls	0.7351	0.5118	0.5535	0.4649
	clm	0.8483	0.5744	0.5891	0.5435
	background	0.9397	0.8145	0.5714	0.4873
U-Net	nclb	0.9154	0.6861	0.6013	0.5527
	gbm	0.6417	0.4811	0.5221	0.4676
	sls	0.7598	0.5176	0.5467	0.4750
	clm	0.8730	0.5879	0.5961	0.5122
	background	0.9549	0.8366	0.5762	0.5265
Attention U-Net	nclb	0.9181	0.7063	0.6114	0.5385
	gbm	0.6714	0.4972	0.5334	0.4528
	sls	0.7336	0.5729	0.5623	0.5052
	clm	0.8895	0.6058	0.6028	0.5047
	background	0.9621	0.8491	0.5849	0.5106
SegNet	nclb	0.9174	0.7124	0.6147	0.5397
	gbm	0.6613	0.5158	0.5378	0.4627
	sls	0.7465	0.5450	0.5561	0.4814
	clm	0.9052	0.6192	0.6085	0.5120
	background	0.9605	0.8437	0.5838	0.5282
SDDNet	nclb	0.9197	0.7127	0.6218	0.5514
	gbm	0.6724	0.5103	0.5413	0.4701
	sls	0.7625	0.5480	0.5734	0.4837
	clm	0.9077	0.6207	0.6151	0.5124
	background	0.9613	0.8548	0.5870	0.5295
ARU-Net	nclb	0.9250	0.7246	0.6317	0.5576
	gbm	0.6854	0.5070	0.5461	0.4721
	sls	0.7648	0.5934	0.5774	0.4981
	clm	0.9131	0.6234	0.6229	0.5158
	background	0.9625	0.8563	0.6068	0.5344
STRNet	nclb	0.9274	0.7246	0.6348	0.5671
	gbm	0.7004	0.5122	0.5524	0.4748
	sls	0.7712	0.6137	0.5789	0.5091
	clm	0.9170	0.6483	0.6276	0.5353
	background	0.9642	0.8521	0.6082	0.5379
PSNet	nclb	0.9362	0.7366	0.6383	0.5690
	gbm	0.7046	0.5224	0.5840	0.4801
	sls	0.7928	0.6140	0.5854	0.5107
	clm	0.9192	0.6508	0.6291	0.5448
	background	0.9708	0.8612	0.6154	0.5436
CAAR-UNet	nclb	0.9431	0.7542	0.6556	0.5746
	gbm	0.7053	0.5313	0.5693	0.4927
	sls	0.7960	0.6206	0.5947	0.5183
	clm	0.9254	0.7140	0.6315	0.5584
	background	0.9712	0.8627	0.6176	0.5472

Overall, the proposed hybrid CAAR-UNet model have outperformed other models due to its ability to effectively capture complex features and interactions between different disease classes, incorporate both local and global contextual information, and mitigate training issues such as the vanishing gradient problem.

To further assess the performance of the models, a class-wise analysis was conducted for each disease class. The results presented in Table 6 demonstrate that the proposed CAAR-UNet architecture achieved higher pixel classification accuracy ($A_{m,class}$), IoU, Dice coefficient (DC_{class}), and mean boundary F1-score ($mBF1_{class}$) for all four disease classes compared to the other models. These findings suggest that the proposed model has the potential to accurately detect and classify a wide range of plant leaf diseases.

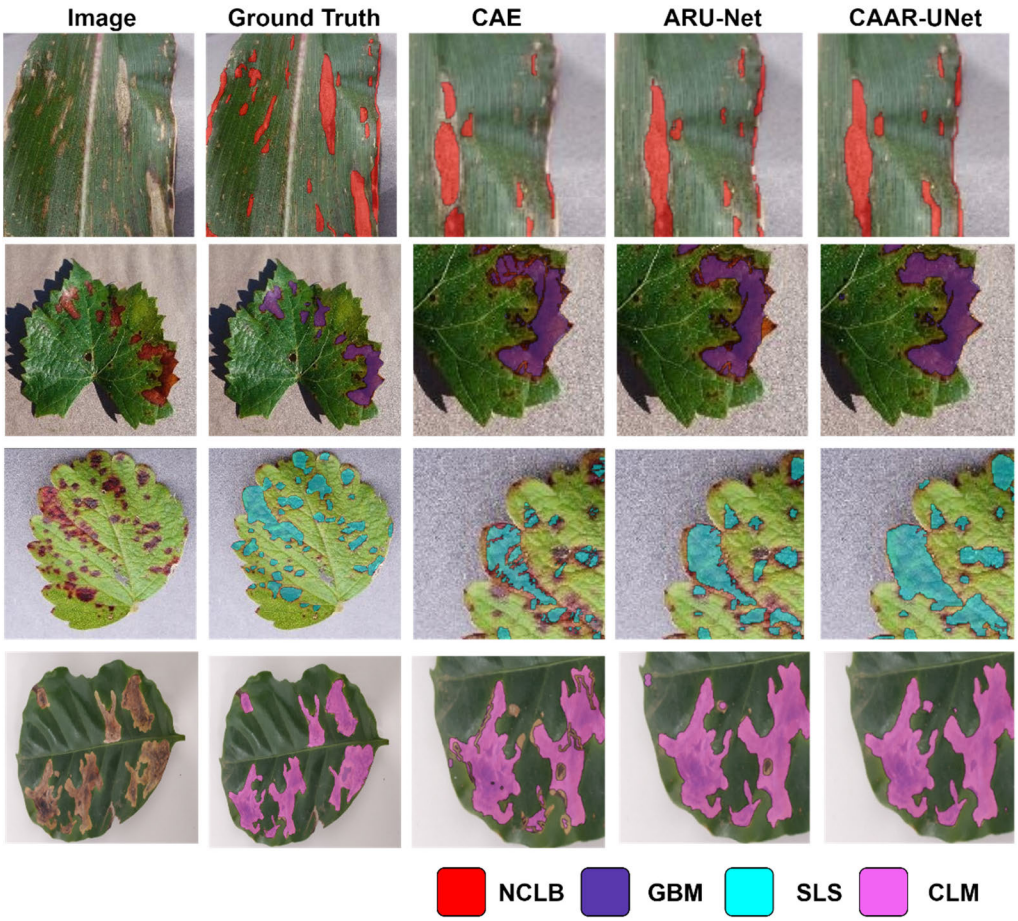

FIGURE 11. A Closer Look at the image, ground truth masks, and segmentation outputs of SAE, ARU-Net, and CAAR-UNet models.

To provide a visual representation of the effectiveness of the proposed model in segmenting different disease classes, final predicted mask was generated for each class and presented in Fig. 11. These results highlight the model's ability to accurately identify and segment different plant leaf diseases, where *nclb*, *gbm*, *sls* and *clm* are the class names of northern corn leaf blight, grape black measles (commonly known as esca), strawberry leaf scorch and coffee leaf miner respectively. Overall, the experimental outcomes highlight the potential of the proposed CAAR-UNet architecture in effectively segmenting and classifying plant leaf diseases.

The analysis from Table 6 revealed that among the all the classes, northern corn leaf blight (*nclb*) and coffee leaf miner (*clm*) have highest IoU scores across all models, while the grape black measles (*gbm*) has the lowest IoU score, and strawberry leaf scorch (*sls*) has an average performance across all models.

The hybrid Cascading Autoencoder with Attention Residual U-Net model achieved highiest IoU score of 0.75 for the nclb class and 0.71 for the clm class, while the IoU score for the gbm class was only 0.5. The PSNet, STRNet, Attention Residual U-Net, Attention U-Net, and SegNet models also achieved high IoU scores for the nclb and clm classes, with scores ranging from 0.62 to 0.72, but struggled with the gbm class, achieving scores around 0.5 to 0.6. The traditional U-Net model and Mask R-CNN achieved an IoU score of 0.7 for the nclb, but struggled with the other classes, achieving scores ranging from 0.5 to 0.6. The SAE model had moderate IoU scores for all classes, with scores ranging from 0.5 to 0.7.

The performance of each model is likely due to the combination of the specific architecture of the model and the characteristics of the images in each class. The distinct shape and size of the lesions in the nclb and clm classes such as large oval-shaped brown or greyish lesions and irregular shaped brown tunnel spots respectively may have contributed to the models' high performance in segmenting these classes. The small irregular shaped dark spots in the gbm class may have made it particularly challenging for all models to accurately segment this class. Finally, multiple small circular or irregular-shaped reddish lesions all over the strawberry leaf may have made it moderate performed for most models, to segment the sls class accurately.

The CAAR-UNet model demonstrated impressive performance not only on the custom dataset but also on the additional images from the PlantVillage dataset and the Coffee Leaf dataset, which were unseen during the training phase. This was done to assess the generalizability of the model and to test its ability to accurately segment plant leaf diseases in different scenarios. These findings suggest that the proposed CAAR-UNet model can be effective in accurately segmenting plant leaf diseases across different datasets and scenarios and may have potential for wider use in agricultural research and crop management.

The visual representation depicted in Fig. 11 offers an elaborate analysis of the image segmentation results obtained from implemented models; SAE, ARU-Net, and CAAR-Unet. By comparing the outputs of each model, it is evident that the CAAR-Unet model exhibited exceptional accuracy and precision, while ARU-Net and SAE also yielded good segmentation results overall. The quarter portions of the original image sizes were used in the Fig. 11 to provide a close-up view of the segmentation outputs, allowing for a more detailed analysis of how each model performs. By visualizing the segmentation results in this way, a better understanding of the strengths and weaknesses of each model in capturing fine-grained details of the input image can be achieved. The figures provide a comparative view of the performance of each model and helps in identifying the model that yields the most accurate segmentation outputs.

V. CONCLUSION AND FUTURE WORKS

Our research introduces the Cascading Autoencoder with Attention Residual U-Net (CAAR-UNet) model, a novel deep learning architecture for multi-plant leaf disease segmentation and classification tasks. The study demonstrates outstanding performance, surpassing other models with a remarkable wmIoU of 0.7451 and mean pixel accuracy of 95.26%. These results hold promise for practical applications in precision agriculture and disease monitoring.

However, it is important to acknowledge the limitations of our research. The computational complexity and training time of the CAAR-UNet model are relatively high, requiring significant computational resources. While our experiments yield promising results on a specific dataset, further research is needed to evaluate the model's performance on unseen datasets and different plant species.

Addressing these limitations and conducting further research can enhance the practicality and effectiveness of the CAAR-UNet model in plant leaf disease image analysis. Future studies should aim to explore the model's generalization capabilities, reduce its computational complexity, and assess its performance on diverse datasets and plant species. By overcoming these limitations, the CAAR-UNet model can unlock its full potential for advancing precision agriculture and disease monitoring. field.

APPENDIX

Pseudocode of the proposed CAAR-UNet model:

```
# Patchify the original images into smaller patches
FUNCTION patchify_images(original_images, patch_size):
    patchified_images = []
    FOR image IN original_images:
        patches = patchify.patchify(image, patch_size)
        patchified_images.extend(patches)
    RETURN patchified_images
# Preprocess the patchified images using Symmetric Autoencoder model
FUNCTION SymmetricAutoencoder(input):
    FOR patch IN patchified_images:
        # Encoder Path
        encoder_outputs = []
        current_layer = patch
        FOR i IN range(num_encoder_blocks):
            current_layer=EncoderBlock(current_layer, filters[i])
            encoder_outputs.append(current_layer)
    # Decoder Path
    FOR i IN range(num_decoder_blocks):
    current_layer = DecoderBlock(current_layer, filters[-(i+1)])
    # Reconstructed Output
    preprocessed_images = Convolution(current_layer, num_channels)
    RETURN preprocessed_images
# Refine the preprocessed images using Attention Residual U-Net model
FUNCTION AttentionResidualUNet(input):
    # Encoder Path
    encoder_outputs = []
    skip_connections = []
    current_layer = input
    FOR i IN range(num_encoder_blocks):
     current_layer, skip = EncoderBlock(current_layer, filters[i])
     encoder_outputs.append(current_layer)
     skip_connections.append(skip)
     # Decoder Path
     FOR i IN range(num_decoder_blocks):
        current_layer = DecoderBlock(current_layer, skip_connections[-(i+1)], filters[-(i+1)])
     # Attention Mechanism
     attention_map = AttentionBlock(encoder_outputs[-1], current_layer)
     attended_features = ApplyAttention(attention_map, current_layer)
     # Residual Connections
     residual_connections = []
     FOR i IN range(num_residual_blocks):
        current_layer = ResidualBlock(current_layer, filters[-(i+1)])
```

```
        residual_connections.append(current_layer)
    # Final Prediction
    output = FinalPrediction(residual_connections[-1], filters[0])
    RETURN output
  # Main Cascading Autoencoder with Attention Residual U-Net (CAAR-UNet) model
  FUNCTION CAAR_Unet(input):
    refined_images = []
    # Symmetric Autoencoder preprocessing
    preprocessed_images = SymmetricAutoencoder(patchified_images)
    # Attention Residual U-Net refinement
    output = AttentionResidualUNet(preprocessed_images)
    refined_images.append(output)
    RETURN refined_images
  # Unpatchify the refined images to reconstruct the output images
  FUNCTION unpatchify_images(refined_output, original_image_size):
    output_images = []
    FOR refined_image IN refined_images:
      output_image = patchify.unpatchify(refined_image, original_image_size)
      output_images.append(output_image)
    return output_images
```

REFERENCES

[1] Y. Zhong, B. Huang, and C. Tang, "Classification of cassava leaf disease based on a non-balanced dataset using transformer-embedded ResNet," *Agriculture*, vol. 12, no. 9, p. 1360, Sep. 2022.

[2] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.

[3] A. Srilakshmi and K. Geetha, "A novel framework for soybean leaves disease detection using DIM-U-net and LSTM," *Multimedia Tools Appl.*, vol. 82, pp. 28323–28343, Feb. 2023.

[4] Q. Jin, Z. Meng, C. Sun, H. Cui, and R. Su, "RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans," *Frontiers Bioengineering Biotechnol.*, vol. 8, p. 1471, Dec. 2020.

[5] M. Ji and Z. Wu, "Automatic detection and severity analysis of grape black measles disease based on deep learning and fuzzy logic," *Comput. Electron. Agricult.*, vol. 193, Feb. 2022, Art. no. 106718.

[6] S. Ashwinkumar, S. Rajagopal, V. Manimaran, and B. Jegajothi, "Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks," *Mater. Today, Proc.*, vol. 51, pp. 480–487, Jun. 2022.

[7] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—A review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021.

[8] 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.

[9] C. Lyu and H. Shu, "A two-stage cascade model with variational autoencoders and attention gates for MRI brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Lima, Peru: Springer, 2020, pp. 435–447.

[10] V. G. Krishnan, J. Deepa, P. V. Rao, V. Divya, and S. Kaviarasan, "An automated segmentation and classification model for banana leaf disease detection," *J. Appl. Biol. Biotechnol.*, vol. 10, no. 1, pp. 213–220, Oct. 2022.

[11] D. Li, R. Wang, C. Xie, L. Liu, J. Zhang, R. Li, F. Wang, M. Zhou, and W. Liu, "A recognition method for Rice plant diseases and pests video detection based on deep convolutional neural network," *Sensors*, vol. 20, no. 3, p. 578, Jan. 2020.

[12] J. Sujithra and M. F. Ukrit, "CRUN-based leaf disease segmentation and morphological-based stage identification," *Math. Problems Eng.*, vol. 2022, pp. 1–13, Jun. 2022.

[13] N. G. Rezk, A.-F. Attia, M. A. El-Rashidy, A. El-Sayed, and E. E.-D. Hemdan, "An efficient plant disease recognition system using hybrid convolutional neural networks (CNNs) and conditional random fields (CRFs) for smart IoT applications in agriculture," *Int. J. Comput. Intell. Syst.*, vol. 15, no. 1, p. 65, Aug. 2022.

[14] L. G. Divyanth, A. Ahmad, and D. Saraswat, "A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery," *Smart Agricult. Technol.*, vol. 3, Feb. 2023, Art. no. 100108.

[15] M. Shoaib, T. Hussain, B. Shah, I. Ullah, S. M. Shah, F. Ali, and S. H. Park, "Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease," *Frontiers Plant Sci.*, vol. 13, Oct. 2022, Art. no. 1031748.

[16] S. Abinaya and M. K. Devi, "Enhancing crop productivity through autoencoder-based disease detection and context-aware remedy recommendation system," in *Application of Machine Learning in Agriculture*. New York, NY, USA: Academic Press, 202, pp. 239–262.

[17] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.*, vol. 8, no. 9, p. 1575, Sep. 2018.

[18] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, and S. Singh, "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges," *Arch. Comput. Methods Eng.*, vol. 29, no. 1, pp. 641–677, Jan. 2022.

[19] A. Pandey and K. Jain, "A robust deep attention dense convolutional neural network for plant leaf disease identification and classification from smart phone captured real world images," *Ecological Informat.*, vol. 70, Sep. 2022, Art. no. 101725.

[20] A. A. Albishri, S. J. H. Shah, and Y. Lee, "CU-net: Cascaded U-net model for automated liver and lesion segmentation and summarization," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2019, pp. 1416–1423.

[21] S. Di, Y. Zhao, M. Liao, Z. Yang, and Y. Zeng, "Automatic liver tumor segmentation from CT images using hierarchical iterative superpixels and local statistical features," *Exp. Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117347.

[22] L. C. Ngugi, M. Abdelwahab, and M. Abo-Zahhad, "Tomato leaf segmentation algorithms for mobile phone applications using deep learning," *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105788.

[23] S. J. J. Jebadurai, I. J. Jebadurai, and G. J. L. Paulraj, "Early detection and classification of plant diseases/abiotic disorders using deep learning—A review," in *Proc. 4th Int. Conf. Inventive Res. Comput. Appl. (ICIRCA)*, Sep. 2022, pp. 646–655.

[24] L. Chen, X. Zhou, M. Wang, J. Qiu, S. Liu, and K. Mao, "ARU-net: Research and application for wrist reference bone segmentation," in *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2019, pp. 1–5.

[25] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Trends in vision-based machine learning techniques for plant disease identification: A systematic review," *Exp. Syst. Appl.*, vol. 208, Dec. 2022, Art. no. 118117.

[26] R. Kumar, D. Singh, A. Chug, and A. P. Singh, "Evaluation of deep learning based Resnet-50 for plant disease classification with stability analysis," in *Proc. 6th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2022, pp. 1280–1287.

[27] J. A. Pandian and G. Geetharamani. (2019). *Data for: Identification of Plant Leaf Diseases Using a 9-Layer Deep Convolutional Neural Network*. Mendeley Data, V1. [Online]. Available: http://dx.doi.org/10.17632/tywbtsjrjv.1

[28] L. B. Silva, Á. L. C. Carneiro, M. S. A. R. Faulin. (2020). *Rust (Hemileia Vastatrix) and Leaf Miner (Leucoptera Coffeella) in Coffee Crop (Coffea Arabica)*. Mendeley Data, V5. [Online]. Available: http://dx.doi.org/10.17632/vfxf4trtcg.5

[29] Apeer. (2023). *Automated Image Analysis*. [Online]. Available: https://www.apeer.com

[30] J. Lewis, Y.-J. Cha, and J. Kim, "Dual encoder–decoder-based deep polyp segmentation network for colonoscopy images," *Sci. Rep.*, vol. 13, no. 1, p. 1183, Jan. 2023.
[31] D. H. Kang and Y.-J. Cha, "Efficient attention-based deep encoder and decoder for automatic crack segmentation," *Struct. Health Monit.*, vol. 21, no. 5, pp. 2190–2205, 2022.
[32] W. Choi and Y.-J. Cha, "SDDNet: Real-time crack segmentation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8016–8025, Sep. 2020.

S. ABINAYA received the B.E., M.E., and Ph.D. degrees in artificial intelligence from Anna University, in 2022.

She is currently an Assistant Professor in Senior Grade 1 with the Vellore Institute of Technology, Chennai. She has proposed various novel information filtering algorithms for recommender systems and pattern recognition algorithms in image processing. She has published various articles in recognized international journals. She has published various book chapters on application of machine learning in agriculture and the IoT. Her research interests include artificial intelligence, machine learning, deep learning, recommender systems, and fuzzy logic and optimization technique. She is a Lifetime Member of The Institution of Engineers (India) (IEI).

KANDAGATLA UTTEJ KUMAR was born in Warangal, Telangana, India, in 2001. He is currently pursuing the B.Tech. degree in computer science and engineering with specialization in artificial intelligence and machine learning with the Vellore Institute of Technology, Chennai, Tamil Nadu, India. His research interests include machine learning, deep learning, and computer vision.

A. SHERLY ALPHONSE received the B.E. degree from Manonmaniam Sundaranar University and the M.E. and Ph.D. degrees in affective computing from Anna University, in 2018.

She is currently an Assistant Professor in Senior Grade 2 with the Vellore Institute of Technology, Chennai. She has proposed various novel pattern recognition algorithms for facial expression recognition and has published various articles in recognized international journals. She has published various book chapters on the IoT and blockchain. She has various patents published on image processing in the fields, such as detecting abnormalities in the liver which is a major contribution in the field of the health sector. She has also published patents like "A Robotic Device for Killing Bedsheet Bacteria" and "Mathematical Application Technology for IoT Data Analysis and Optimization." Her research interests include image processing, machine learning, and facial expression analysis. She is a Lifetime Member of IAENG.