IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 May 2023, accepted 15 June 2023, date of publication 26 June 2023, date of current version 10 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289750

RESEARCH ARTICLE

Lite-SRGAN and Lite-UNet: Toward Fast and Accurate Image Super-Resolution, Segmentation, and Localization for Plant Leaf Diseases

HOSAM S. EL-ASSIOUTI, HADEER EL-SAADAWY, MARYAM N. AL-BERRY, AND MOHAMED F. TOLBA, (Senior Member, IEEE)
Department of Scientific Computing, Ain Shams University, Cairo 11566, Egypt
Corresponding author: Hosam S. El-Assiouti (hossamsherif@cis.asu.edu.eg)

ABSTRACT Complex deep convolutional networks are typically designed to achieve state-of-the-art results. Such networks require powerful computing resources and cannot work efficiently on resource-constrained devices particularly for real-time use. To address these challenges, this study introduces resource-efficient lightweight approaches for segmentation, localization, super-resolution, and classification tasks. On this basis, we propose two novel lightweight architectures named: Lite-UNet and Lite-SRGAN. We validated the effectiveness of our proposed networks using the large publicly available Plant Village dataset. Lite-UNet network is used for performing segmentation and localization tasks, while Lite-SRGAN network is used for performing the super-resolution task. The proposed Lite-UNet outperforms U-Net with slight gains of 0.06% and 0.12% for dice coefficient and Intersection over Union (IoU) respectively while achieving significant reductions of 15.9x, 25x, and 6.6x in terms of parameters, floating-point operations per second (FLOPs), and inference time respectively. In addition, the proposed Lite-SRGAN achieves comparable qualitative and quantitative results compared to SRGAN with significant reductions of 7.5x, 7.8x, and 2.7x in terms of parameters, FLOPs, and inference time respectively when upsampling the low-resolution images from 64×64 to 256×256 (4x upscaling). Similarly, it achieves a reduction of 7.1x, 11.2x, and 1.9x when upsampling from 128×128 to 256×256 (2x upscaling). For classification purposes, a two-stage classification approach is introduced, in which the crop species and their leaf diseases are recognized respectively. Different models are utilized in both stages including MobileNetV3, DenseNet121, and ConvNeXt. The best accuracy obtained on the testing set is 99.76% when using the proposed methods together, which outperforms several other related studies. Source code is available at https://github.com/hosamsherif/LiteSRGAN-and-LiteUNet

INDEX TERMS Lightweight networks, super resolution, generative adversarial networks (GANs), object localization, segmentation, convolutional neural networks.

I. INTRODUCTION

Plant and crop diseases are considered one of the major diseases that can threaten people's life, as the crops is one of the main foods that people rely on in their daily lifestyle [1], [2], [3]. Automatic detection of plant diseases in its early stages can lessen its harmful effects [4], [5].

Significantly so far, deep convolutional networks have been designed to be complex, comprising an increased number of learnable parameters to achieve a higher accuracy. However, these powerful deep convolutional networks require higher computational resources and come at high latency and cost. Thus, these complex networks cannot work efficiently on mobile and embedded devices particularly for real-time use. This study focuses on building robust lightweight approaches for performing segmentation, localization, super-resolution, and enhancing classification. To achieve this purpose, two novel efficient lightweight architectures, named Lite-UNet and Lite-SRGAN are introduced. Both architectures are designed to work efficiently

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

in real-time environments with minimal possible latency and low complexity especially for resource-constrained devices while achieving promising results.

The proposed Lite-UNet network is used to perform the segmentation task. Moreover, it is utilized to localize the plant leaves in the images with the help of the contour detection algorithm [6]. The resulting masks from the segmentation step are used to crop the plant leaves from the input images to neglect the effect of the surrounding background that does not contain any relevant features. Cropping the region of interest helps the classification model to focus more on the discriminative features and reduce the convergence time needed for training. However, training a Convolutional Neural Network (CNN) classification model typically requires input images with a fixed resolution. Consequently, the cropped leaves from the input images are mapped to a unified resolution before being fed into the classification network. Since the use of basic interpolation techniques (e.g., bicubic, bilinear, and nearest neighbor) for enlarging images leads to unsatisfactory recovery of textures and high-frequency details in the resulting interpolated image, especially when the cropped image resolution is relatively small. Thus, a novel accurate, lightweight super-resolution architecture (Lite-SRGAN) is introduced to efficiently map the low-resolution cropped image to the required resolution while preserving the texture and high-frequency details. The segmentation, super-resolution, and classification experiments are conducted and evaluated on the large publicly available Plant Village dataset [4], [7].

To analyze the efficacy of the Lite-UNet architecture, it is compared to the U-Net network [8] in terms of performance (i.e., dice coefficient, IOU, accuracy), model complexity (i.e., parameters, FLOPs), and latency (i.e., inference time). Lite-UNet generates accurate segmentation masks with slight increases of 0.06% and 0.12% for the dice coefficient and IOU respectively compared to U-Net. However, Lite-UNet achieves a significant improvement in terms of complexity and latency with 15.9x fewer parameters than U-Net as well as 25x fewer FLOPs. In addition, our model shows faster inference compared to U-Net with an inference speed-up of 6.6x and 1.6x on CPU and GPU respectively.

Lite-SRGAN is also compared to one of the state-of-the-art perception-based super-resolution models which is SRGAN [9]. The proposed novel Lite-SRGAN generates high quality super-resolved images with a noticeable reduction in terms of parameters, FLOPs, and inference time, while achieving comparable qualitative and quantitative results with SRGAN. In this study, two versions of the proposed Lite-SRGAN are introduced. The first version upscales the low-resolution image (LR) by a factor of 4 (i.e., 64×64 to 256×256), whereas the second version upscales the LR image by a factor of 2 (i.e., 128×128 to 256×256). The first version of Lite-SRGAN achieves a 7.5x, 7.8x, 2.7x, and 1.2x reduction in terms of parameters, FLOPs, CPU inference time, and GPU inference time compared to the corresponding SRGAN version. On the other hand, the second version of Lite-SRGAN achieves a 7.1x, 11.2x, 1.9x, and 1.5x reduction in terms of parameters, FLOPs, CPU inference time and GPU inference time compared to the corresponding SRGAN version. It is noteworthy to mention that the proposed Lite-UNet and Lite-SRGAN models can be applied to any other task or application where real-time segmentation and super-resolution are crucial.

Finally, a two-stage hierarchical classification approach is introduced, where the first stage classifies the given input image into one of the 9 plant leaf species, and the second stage classifies the input image into one of the classes that belongs to the determined category from the first stage. Different state-of-the-art pre-trained CNN networks are used in both stages including MobileNetV3 [10], DenseNet121 [11], and ConvNeXt [12]. These models achieved high comparable results, however, MobileNetV3 is considered the most suitable one for the proposed full lightweight approach due to its low computational cost and fast inference.

The main contributions in this paper can be summarized as follows:

1. A novel lightweight architecture for segmentation named Lite-UNet is proposed, which take advantage of achieving superior results with significantly fewer parameters, low complexity, and high-speed inference.
2. A novel lightweight architecture for super-resolution named Lite-SRGAN is proposed. It efficiently maps a given low-resolution image to a high-resolution one with a significant reduction in terms of the parameters, FLOPs, and latency. We also demonstrate the effect of combining different loss functions and how they contribute to the proper reconstruction of the super-resolved images.
3. Localizing the plant leaves in the given input image by utilizing the segmentation masks obtained by the Lite-UNet network along with the contour detection algorithm.
4. A two-stage hierarchal classification approach is proposed, where the crop species and their diseases are classified respectively.
5. Extensive experiments were conducted to demonstrate the efficiency of each proposed method on its own and how these methods can be combined to form a superior methodology.

The rest of this paper is organized as follows: Related work is discussed in Section II. The proposed methods and the overall approach are discussed in detail in Section III. Experimental results and extensive analysis for each method are presented in Section IV. Finally, Section V provides the conclusions and future work.

II. RELATED WORK

Automatic classification and detection of different plant diseases have gained significant interest from many researchers over the past few years. Advances in image processing, machine learning and deep learning techniques have played a vital role for developing different approaches and innovative solutions for classifying and detecting different plant diseases

in the early stages. This section discusses some recent state-of-the-art studies related to our proposed work.

Harakannanavar et al. [13] introduced an approach for leaf disease identification based on machine learning and image processing algorithms. This study considered 6 different tomato leaf disorders extracted from the Plant Village dataset.

Histogram equalization and k-means clustering were used for the preprocessing step. Different feature extractions algorithms are utilized for extracting features from the images, including Discrete Wavelet Transform (DWT), Gray-level Co-occurrence Matrix (GLCM), and Principal Component Analysis (PCA). Finally, different models including K-Nearest Neighbor (KNN), Support Vector Machines (SVM), and CNN, are used to classify the extracted features. Their proposed method achieved an accuracy of 99.09% by utilizing DWT+PCA+GLCM as feature extraction techniques and custom CNN as their classifier.

Abbas et al. [14] proposed a deep learning method for identifying different tomato diseases. Conditional generative adversarial network (cGAN) was utilized as an augmentation technique for extending the training set with new synthetic samples. A pre-trained DenseNet121 model is trained on tomato leaf images extracted from the Plant Village dataset and the synthetic samples generated by cGAN. This method was carried out on 3 different number of classes including 5 classes, 7 classes, and 10 classes, achieving accuracies of 99.51%, 98.65%, and 97.11% respectively.

In [15], Bedi and Gole proposed a hybrid model based on convolutional neural network (CNN) and convolutional autoencoder (CAE). Their proposed method considered detecting bacterial diseases in peach leaf images extracted from plant village dataset. The extracted peach dataset consisted of 2 classes including healthy and bacterial spot disease with 4457 total number of images. The convolutional auto encoder network reduces the dimensionality of the input images by compressing the domain representations of the images while maintaining the most important features. Thereafter, the compressed representations of peach leaf images were fed into a CNN architecture to classify whether the leaf is healthy or has a bacterial spot disease. This method achieved an accuracy of 99.35% on the training set and 98.38% accuracy on the testing set with only 9,914 trainable parameters. In [16], Alatawi et al. utilized a VGG-16 model pre-trained on the ImageNet dataset with 3 custom dense layers to classify different plant diseases. 19 different classes including apple, corn, tomato, grape diseases as well as healthy classes are considered. This method achieved an accuracy of 95.2% using the Plant Village dataset.

Hassan and Maji [17] introduced a robust lightweight CNN approach based on inception building block and residual connections for plant disease classification. Standard convolution layers in the inception block are replaced with depth-wise separable convolution, and thus reducing the total number of parameters by a margin of 70%. This model was trained and evaluated on 3 different plant disease datasets. The testing accuracy obtained on the rice disease dataset was 99.66%, and on the Plant Village dataset when using 17 different classes for corn, potato, and tomato diseases was 99.36%. Finally, the testing accuracy achieved on the cassava dataset was 76.59%.

A novel approach for automatic and reliable leaf disease detection based on the Modified U-Net and EfficientNet was proposed by Chowdhury et al. [18]. Their experiments were conducted on different tomato leaf diseases from the Plant Village dataset. The introduced Modified U-Net segments the leaf region from the given image, whereas EfficientNet classifies the segmented images obtained from the Modified U-Net. Different preprocessing techniques were applied to the images including data rescaling and normalization. In addition, different augmentation techniques have been applied to balance the dataset including image translation, rotation, and scaling. The modified U-Net achieved a Dice coefficient of 98.73%. Moreover, EfficientNet-B7 achieved accuracies of 99.95% and 99.12% for classifying 2 classes and 6 classes respectively. Finally, EfficientNet-B4 achieved 99.89% accuracy for classifying 10 different classes.

Tuncer [19] proposed a novel approach for plant leaf disease detection using a hybrid cost-optimized CNN. The proposed hybrid model is based on inception network and depth-wise separable convolution, resulting in a significant reduction in the model parameters. This hybrid model was trained and tested on 30 different classes extracted from the Plant Village dataset, achieving an accuracy of 99.27% with a reduction of 75% in the total number of parameters.

In [20], Zhao et al. introduced a novel two-stage Generative Adversarial Network called Double-GAN. It consists of two stages. The first stage utilizes the Wasserstein generative adversarial network (WGAN) to generate unhealthy leaves with a resolution of 64×64, while in the second stage a super-resolution generative adversarial network (SRGAN) was used to obtain high-quality images with a resolution of 224×224 while preserving the image quality as much as possible. The DoubleGAN-based method was utilized to balance the dataset classes by increasing the number of instances for the minor classes. They also introduced a two-stage classification approach. where the plant type is classified in the first stage into one of the 5 different classes extracted from the Plant Village dataset (apple, corn, grape, potato, and tomato) using three different classifiers (VGG16, ResNet50, and DenseNet121). Subsequently, the second stage classifies the given image into one of the 10 tomato leaf categories using the same three classifiers used in the first stage. This method achieved accuracies of 99.74% and 99.53% for the first and the second stages respectively.

III. METHODOLOGY & PROPOSED WORK

The proposed methods for segmentation, localization, super-resolution, and multi-stage classification are discussed in this section. The following subsections provide a detailed description of each method, as well as the overall proposed methodology.

A. SEGMENTATION

MobileNets is one of the most famous CNNs introduced by a team of Google researchers [21]. MobileNet architecture mainly focuses on reducing the number of operations and trainable parameters to match the limited design requirements for mobile vision applications while retaining high classification accuracy compared to the state-of-the-art architectures. This is achieved by replacing the high computation regular convolution operations with depth-wise separable convolution operations. Different versions of MobileNets were recently released [10], [21], [22], we used MobileNetV2 [22] because of its high performance, fast computation and because it is well-suited for our whole segmentation proposed architecture.

MobileNetV2 architecture introduces an inverted residual structure consisting of two types of inverted bottleneck blocks: one with stride=1 and the other with stride=2 for downsampling the feature map resolution with a factor of 2. The block with stride=1 has a residual connection, whereas the block with stride=2 does not have a residual connection as shown in Figure 1. Both blocks consist of three consecutive layers. The first layer consists of three consecutive operations: 1 × 1 convolution (i.e., expansion convolution), batch normalization, and ReLU6 activation function, whereas the second layer consists of 3×3 depth-wise convolution followed by batch normalization and ReLU6 activation. Finally, the third layer consists a of 1 × 1 convolution (i.e., projection convolution) followed by batch normalization without a non-linear activation function.

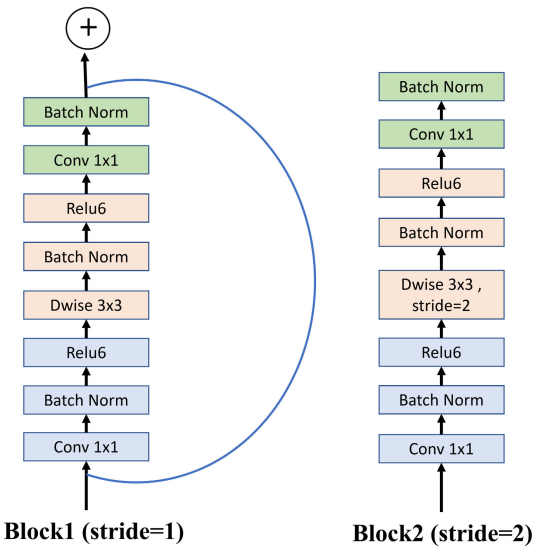

FIGURE 1. MobileNetV2 blocks.

U-Net architecture [8] is a well-known CNN that was released in 2015. Since then, it has shown promising results in many different tasks such as biomedical image segmentation, satellite image segmentation, image generation, and image inpainting [23], [24], [25]. The main structure of U-Net consists of an encoder and a decoder and skip connections connecting them together to form a U-shaped architecture.

The proposed Lite-UNet model main building blocks is inspired from both the U-Net and MobileNetV2 architectures. Similar to U-Net, the Lite-UNet model consists of two paths: a contracting path (encoder network) followed by an expansive path (decoder network). The encoder part is mainly based on the MobileNetV2 network. Moreover, the MobileNetV2 encoder network is pre-trained on the ImageNet dataset [26]. The proposed architecture took advantage of using a pre-trained encoder to help the segmentation model converge faster and learn better feature representations. The encoder part encodes high-level semantic features from the given input image through a sequence of encoder blocks of the MobileNetV2 architecture, shown in Figure 1. To build a smaller and faster encoder with a reduced number of parameters, the width multiplier of MobileNetV2 is set to 0.35. The decoder part takes the high-level semantic features obtained from the encoder and generates a segmentation mask that corresponds to the given input image. Skip connections are used to concatenate the feature maps in the encoder part with their corresponding feature maps in the decoder part, to transfer the information from the earlier layers in the encoder path to the later layers in the decoder path, and thus allowing the recovery of the spatial information lost during downsampling, and enabling the segmentation model to produce more accurate masks.

As shown in Figure 2, the input image, and layers (4,10,19,40) are extracted from the encoder part and concatenated with their corresponding feature maps in the decoder part via skip connections to refine the details in the decoding stage. The feature maps that are extracted from the encoder part to be concatenated with the decoder part are those produced from the 1 × 1 expansion convolution layers (i.e., after the ReLU activation function) as they are the deeper feature maps in the MobileNetV2 encoder. Thus, we can preserve as much information as possible from the earlier layers in the encoder part. Each step in the decoding (expansive) path includes an upsampling layer which doubles the resolution of the previous feature map, a concatenation with the corresponding feature map in the encoder path, followed by 2 convolution layers, where each convolutional layer uses a 3 × 3 kernel size followed by a batch normalization layer and ReLU activation function. Finally, a 1 × 1 convolution layer is applied with only one kernel, followed by a sigmoid activation function to produce the desired segmentation mask with the same resolution of the given input image.

The Dice loss function is used to optimize the proposed Lite-UNet architecture. The Dice coefficient has been considered in many research papers as a main metric for evaluating the performance of different segmentation networks. Thereafter, it was adapted by Milletari et al. [27] to be used as a loss function for optimizing segmentation networks. The relationship between the Dice coefficient and Dice loss is inversely proportional. Thus, when the Dice

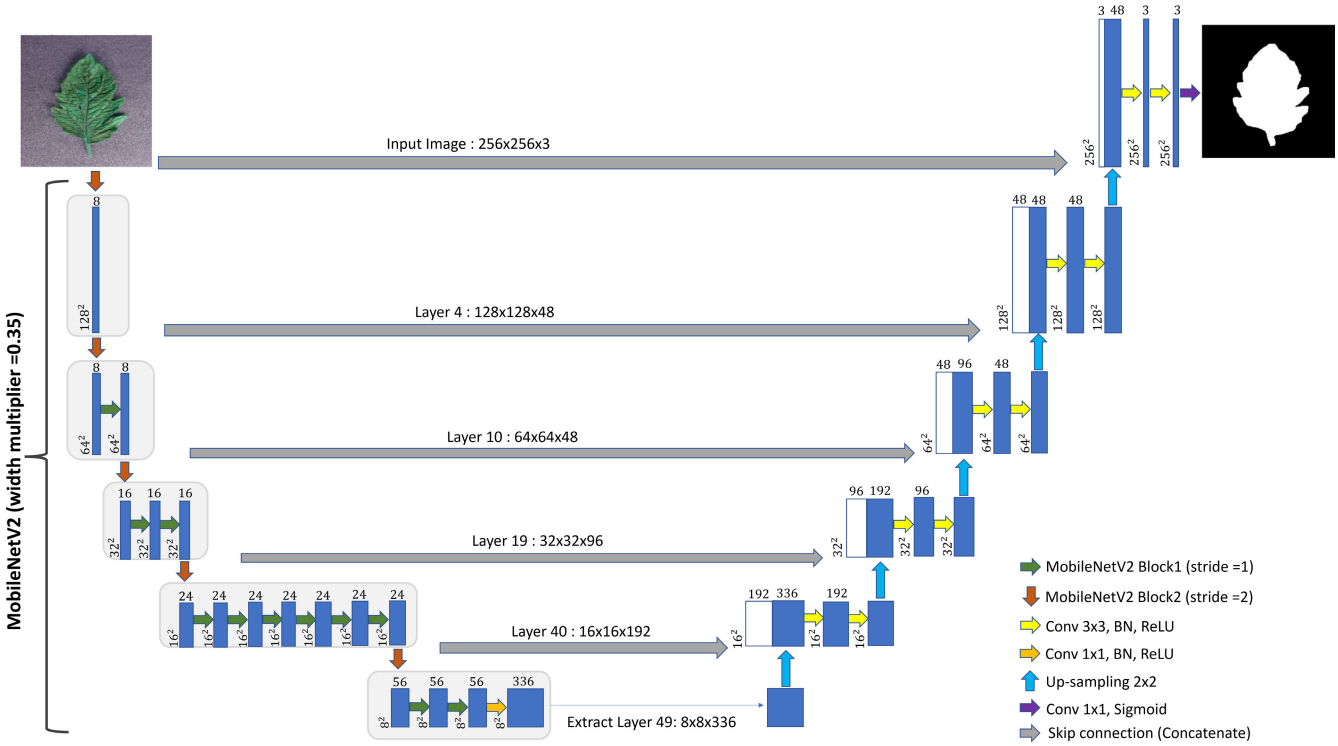

FIGURE 2. The proposed Lite-UNet architecture.

coefficient value increases, the more intersection occurs between the predicted mask and the ground truth mask, and the Dice loss value approaches zero. On the other hand, when the Dice coefficient value decreases, the Dice loss value approaches one. The Dice loss function is given by equation (1) and (2):

$$l_{dice} = 1 - dice\_coefficent \tag{1}$$

$$l_{dice} = 1 - \frac{2y\hat{y} + \epsilon}{y + \hat{y} + \epsilon} \tag{2}$$

where y refers to the ground truth mask, $\hat{y}$ refers to the segmentation mask generated by the network and ϵ term is a very small value added to the equation to avoid division by zero issue when $y = \hat{y} = 0$.

B. SUPER RESOLUTION

Super-resolution has attracted significant interest over the last decades. Super-resolution is widely used in various applications such as improving the resolution of medical, ariel and satellite images, as well as video streaming [28], [29], [30]. Many deep-learning-based approaches have been released over the past few years and have attracted much popularity due to their efficiency in mapping a low-resolution image into a higher one. Some perceptual-oriented approaches focused on improving the quality of super-resolved images [9], [31], [32], [33], while other studies focused on building a lightweight super-resolution model for real time image enhancement [34], [35], [36], [37].

One of the aims of this study is to build a robust, lightweight, and resource-efficient super-resolution network with a significantly fewer number of parameters and faster inference time. Thus, enabling our model to work smoothly in a real-time environment for either enhancing images or videos. On the other hand, we focus on improving the perceptual quality of the super-resolved images. In this section, the proposed Lite-SRGAN architecture is discussed in detail as well as the loss functions used to optimize the generator and discriminator networks.

1) LITE-SRGAN ARCHITECTURE

The main purpose of the introduced Lite-SRGAN network is to obtain a super-resolved image (SR image) from a given low-resolution image (LR image) in a very short inference time. This enables its deployment on mobile devices for real-time applications. In addition, the proposed Lite-SRGAN also focuses on estimating a good-looking super-resolved image where the clarity of the obtained super-resolved image details surpasses many state-of-the-art recent studies that address the same problem but with much less inference time and fewer trainable parameters. The generator training process is done by feeding a low-resolution image as an input to the generator network. Subsequently, the generator predicts a super-resolved image. The total generator loss is then

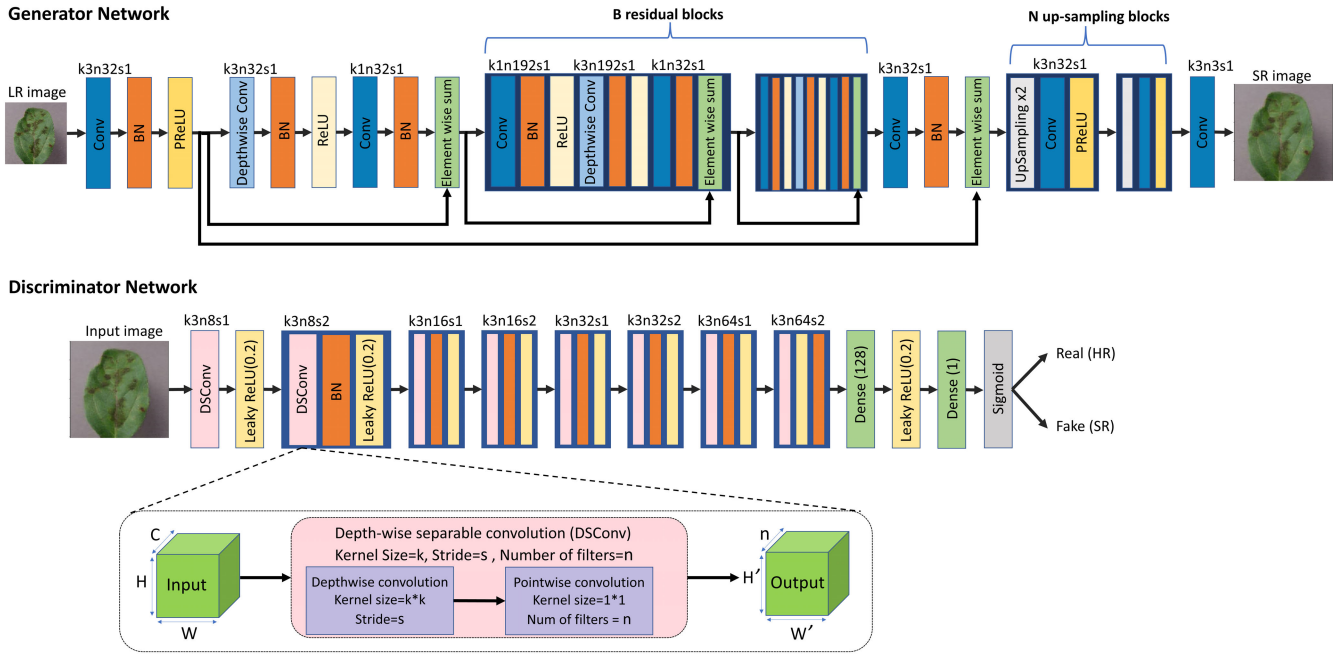

FIGURE 3. The proposed Lite-SRGAN architecture.

calculated based on the original high-resolution (HR) image and the super-resolved (SR) image, where the generator network is optimized to generate a SR image with realistic textures and minimal visual distortion so it can fool the discriminator.

The generator architecture blocks are inspired from the inverted residual blocks in the MobileNetV2 architecture, [22] which was previously depicted in Figure 1. These blocks focus on substituting high computation standard convolution with separable depth-wise convolution, and thus the total number of parameters of the generator network is reduced. As shown in Figure 3, the generator network consists of B identical blocks (B = 12), where each block consists of a 1×1 convolution layer (i.e., expansion convolution) followed by a batch normalization (BN) layer and a ReLU activation function, then a 3×3 depth-wise convolution layer takes place followed by a BN layer and a ReLU activation function. Finally, another 1×1 convolution layer (i.e., projection convolution) takes place followed by a BN layer and an element-wise sum layer to sum the output of the previous block (i-1) with the output of the current block (i). The role of these B blocks is to obtain high level semantic features. Following the B blocks there exist upsampling blocks that are responsible for increasing the resolution of the LR image, where each upsampling block consists of an upsampling layer followed by 3×3 convolution layer and a PReLU activation function [38]. At the end, we have a convolution layer with 3 kernels each of size 3×3, and a tanh activation function to generate a 3-channel super-resolved image that has a range of values from -1 to 1 due to the tanh activation function.

Two different versions of the generator are used in this study depending on the dimensionality of the input LR image: one uses only one upsampling block to increase the resolution of the given LR image by a factor of 2, while the other version uses two upsampling blocks to increase the resolution of the given LR image by a factor of 4.

The discriminator network is simply a classifier that tries to distinguish between the high-resolution images coming from the real data and the super-resolved images obtained by the generator network. Both generator and discriminator networks are competing against each other during training which is the general idea of any GANs based architecture [39]. The discriminator network is optimized during training to get better each time at distinguishing between real samples and generated super-resolved samples, whereas the generator is optimized based on how well the generated samples deceive the discriminator, so both are improving depending on each other. The discriminator architecture is shown in Figure 3. It consists of 8 blocks, those used blocks are inspired from the blocks used in the SRGAN discriminator, but the main modification is that each convolution layer is replaced with a depth-wise separable convolution layer, and hence leading to much fewer trainable parameters and less training time. The depth-wise separable convolution layer consists of a 3×3 depth-wise convolution followed by a 1×1 convolution (pointwise convolution).

Each discriminator block consists of a depth-wise separable convolution layer followed by a BN layer and a leaky ReLU activation function ($\alpha = 0.2$), except for the first block, it does not contain a BN layer. Four of these eight blocks

use a strided convolution to downsample the resolution of the feature map by a factor of 2. The number of filters is doubled every two blocks to increase the depth of the resulting feature map. The 64 features obtained from the last layer are then fed to a dense layer followed by a leaky ReLU activation function ($\alpha = 0.2$) and a sigmoid activation function is used in the last layer for classification purposes. The discriminator network is optimized using the binary cross-entropy (BCE) loss function.

2) LOSS FUNCTIONS

a: PERCEPTUAL LOSS

Perceptual loss [40], [41] is mainly used to increase the perceptual similarity between the original image and the generated one by optimizing the network in the content (feature) space to focus on retaining the content of the original image while generating a new one, rather than optimizing the network in the image (pixel) space. Perceptual loss is simply defined by first finding the content features (high-level features) for the original HR image and the generated SR image, and then calculating the mean square error between these extracted features. The i^{th} convolutional layer in the pre-trained VGG19 network [42] is utilized to extract the content features. Since CNNs' early convolution layers extract low-level features, while later convolutional layers capture more information regarding the content of the image. Therefore, the last convolutional layer (before ReLU activation) is used to define our perceptual loss function. The perceptual loss function is defined as follows:

$$l_{perceptual} = \frac{1}{W_i H_i C_i} \sum_{x,y,c} (\phi_i \left(I^{SR}\right)_{x,y,c} - \phi_i \left(I^{HR}\right)_{x,y,c})^2 \tag{3}$$

where ϕ_i refers to the feature map extracted from the i^{th} convolutional layer within the network; W_i, H_i, and C_i refers to the feature map dimensions, I^{HR} refers to the original HR image; and I^{SR} refers to the reconstructed SR image obtained by the generator network.

b: ADVERSARIAL LOSS

The adversarial loss role is to fool the discriminator, so the generator is optimized to deceive the discriminator. The adversarial loss function contributes to optimizing the generator learning process by penalizing the generator if the discriminator figures out that the generated image is not realistic with some probability. Adversarial loss was introduced in the original GAN paper by Goodfellow et al. [39] and then updated by the researchers [9] for improving the gradient behavior to be defined by the following equation:

$$l_{adv} = -\log\left(D_{\theta_D}\left(I^{SR}\right)\right) \tag{4}$$

where θ_D is the parameters of the discriminator network, and $D_{\theta_D}\left(I^{SR}\right)$ is the probability of how the generated SR image looks like a realistic HR image from the discriminator perspective.

c: PER-PIXEL LOSS

The Per-Pixel loss simply finds the mean absolute difference (L1 loss) between each pixel in the original image and the corresponding pixel in the generated image, it gives a better visual appearance for the generated image by slightly smoothing it towards high frequency noise, but on the other hand it sometimes lessens the proper reconstruction of high frequency content details. It works well beside the perceptual loss, as the perceptual loss focuses mainly on retaining the content of the original image, and the per-pixel loss slightly smooths the reconstructed image; thus, its contribution to the total generator loss should be carefully weighted. The Per-Pixel loss function is defined as follows:

$$l_{MSE} = \frac{1}{WHC} \sum_{x,y,c} \left| {I^{SR}}_{x,y,c} - {I^{HR}}_{x,y,c} \right| \tag{5}$$

where W, H, and C refer to dimensions of the image; I^{HR} refers to the original HR image; and I^{SR} refers to the reconstructed SR image obtained by the generator network.

d: STYLE LOSS

Style loss is similar to the perceptual loss, but it measures the square of the differences between the gram matrix of high-resolution image features and the gram matrix of super-resolved image features, [40], [43] where the gram matrix represents the amount of correlation between feature maps resulting from a given convolutional layer in the network. The feature maps obtained by a given convolutional layer l in a CNN network should have dimensions of $N_l \times H_l \times W_l$, where N_l refers to the number of feature maps obtained at layer l and H_l, W_l refer to the dimensions of each feature map at layer l. This can be reshaped into a matrix with dimensions of N_l rows and $H_l W_l$ columns, so the final features matrix is given by shape $N_l \times M_l$ where M_l is a vectorized representation of $H_l \times W_l$. Therefore, the feature matrix at layer l is denoted as $F^l \in R^{N_l \times M_l}$, where $F^l_{i,k}(I)$ refers to the activation map at layer l of the i^{th} feature at position k when image (I) is passed as an input to the network.

The correlations between feature maps at layer l are given by Gram matrix $G^l \in R^{N_l \times N_l}$, where $G^l_{i,j}$ is computed as the inner product between the vectorized feature maps i and j in layer l normalized by the spatial dimension of the feature matrix at layer l

$$G^l_{i,j}(I) = \frac{1}{M^l N^l} \sum_k F^l_{i,k}(I)\, F^l_{j,k}(I) \tag{6}$$

The style loss at layer l is defined as the mean square of the differences between the gram matrices of the high-resolution image features and the super-resolved image features, those features are obtained by passing the high-resolution and super-resolved images to a pre-trained VGG16 network. [42]

$$E_l = \frac{1}{N^l N^l} \sum_{i,j} (G^l_{i,j}\left(I^{HR}\right) - G^l_{i,j}(I^{SR}))^2 \tag{7}$$

where $G^l\left(I^{HR}\right)$ refers to the gram matrix at layer l given a high-resolution image, while $G^l\left(I^{SR}\right)$ refers to the gram

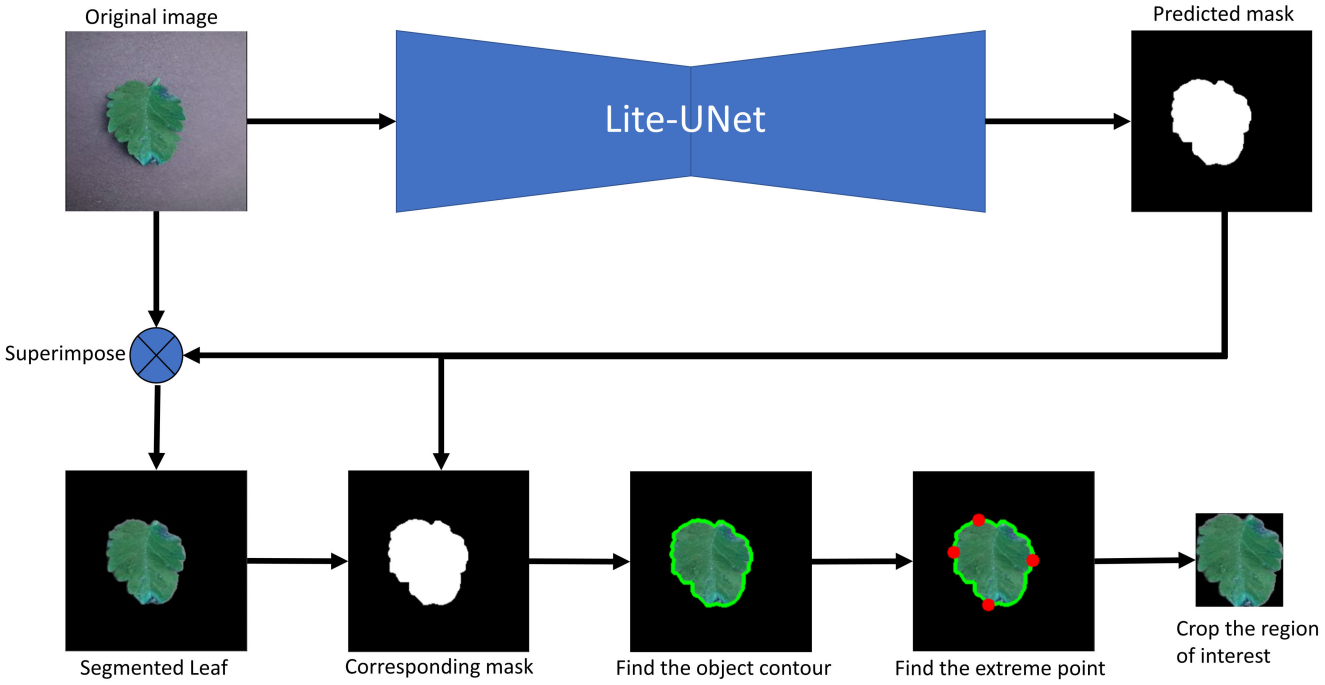

FIGURE 4. The process of localization and cropping the region of interest object.

matrix at layer l given a super-resolved image obtained by the generator. The network is optimized by minimizing the style loss across multiple layers, so the final equation of the style loss is defined by:

$$l_{style} = \sum_{l=0}^{L} w_l E_l \tag{8}$$

where w_l are the weighting factors that control the contribution of each layer to the total style loss function.

e: TOTAL LOSS

The final loss function used to optimize our generator network is a weighted combination of the 4 sub-loss functions and is given by equation (9). Where $\lambda's$ are the weighting parameters used to balance the contribution of each loss term.

$$l_{total} = \lambda_1 l_{perceptual} + \lambda_2 l_{adv} + \lambda_3 l_{MSE} + \lambda_4 l_{style} \tag{9}$$

C. PREPROCESSING AND LOCALIZATION

Figure 4 shows the steps for localizing and cropping the region of interest (i.e., plant leaf area). The input leaf image is fed into the proposed Lite-UNet architecture to obtain the corresponding segmentation mask. Thereafter, we superimpose the original image with the predicted segmentation mask to obtain the segmented leaf. Since most of the images contain a large background area, which may reduce the performance of the classifier and slow its convergence process [44]. Thus, the undesired surrounding background is neglected by cropping only the region of interest. Moreover, the segmentation step is also very important in the proposed methodology, as we localize the leaf beside classifying it. The cropping process is done by first obtaining the segmentation mask by applying the proposed Lite-UNet on the input images. Then a set of erosion operations followed by dilation operations (morphological opening) are performed on the predicted segmentation mask to remove any existing noise that may result due to the segmentation step. Afterwards, each object contour is grabbed from the segmentation mask using the contour detection algorithm, [6] and the extreme points of each object are calculated accordingly. Finally, the calculated extreme points are used to crop the object. The coordinates of these calculated extreme points are also projected onto the original image and connected to perform the localization task prior doing the classification task.

After obtaining the cropped leaf images from the given input images as shown in Figure 4, we need to get a unified resolution for all the training images to train a classifier. Each cropped image is upsampled based on its resolution using three different ways as depicted in Figure 5; If both the width and height of the cropped image are less than 64, then resize with padding technique is applied on the cropped image to obtain an image of resolution 64 × 64, then the Lite-SRGAN generator with 2 upsampling blocks is used to increase the image resolution by a factor of 4 to obtain a super-resolved image with a resolution of 256 × 256. However, if the given cropped image width and height are less than 128, then resize with padding technique is applied to obtain an image of resolution 128 × 128, then the Lite-SRGAN generator with 1 upsampling block is used to increase the image resolution by a factor of 2 to obtain a super-resolved image with a resolution of 256 × 256. Otherwise, if either the width or

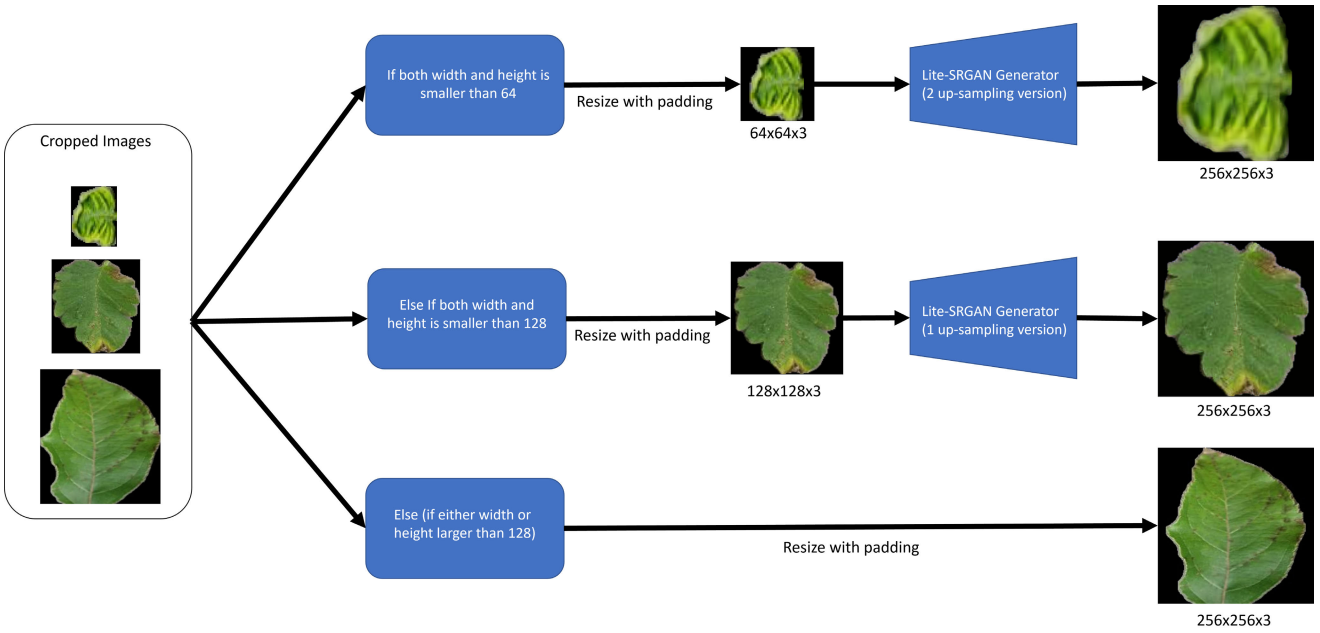

FIGURE 5. The process of obtaining a super-resolved image based on the given cropped image resolution.

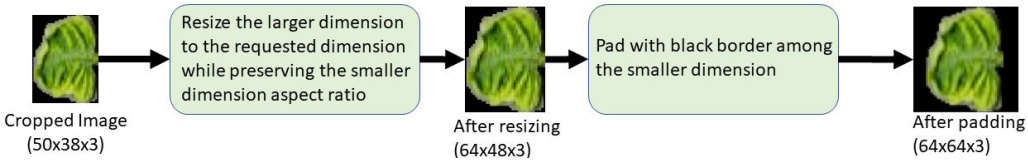

FIGURE 6. An example illustrating how resize with padding technique is applied to the cropped region of interest.

height is greater than 128, then resize with padding technique is applied to obtain a 256 × 256 upsampled image.

The idea behind utilizing the resize with padding technique instead of resizing the cropped image to a fixed size directly without padding, is to preserve the aspect ratio of the cropped object, as changing the original object shape may lead to undesired results in classification beside undesired visual distortion of the appearance of the object. Resize with padding technique idea is to simply resize an image to a fixed resolution without changing the aspect ratio of the object shape. As shown in the example in Figure 6, resize with padding technique works by first resizing the given image while preserving the aspect ratio, followed by padding the smaller dimension with zeros to match the requested dimensions.

D. TWO-STAGE CLASSIFICATION HIERARCHICAL APPROACH

In this work, a two-stage hierarchical classification approach is proposed, as depicted in Figure 7. The first stage classifies the input image into one of the nine different plant types, whereas the second stage identifies the diagnosis of the determined plant type whether, it is healthy or diseased and the type of disease if exists.

E. FULL METHODOLOGY

The full pipeline of the whole proposed method is shown in Figure 8. First, the input image is fed into the proposed Lite-UNet network to obtain the corresponding segmentation mask that highlights the exact location of the plant leaf. Afterwards, the obtained mask is superimposed with the original input image to obtain the segmented leaf only. Thereafter, we focus on cropping the region of interest, which is the plant leaf in our case, to neglect the effect of the background and to make it easier for the classifier network to learn the relevant features only related to the foreground object of interest, thereby enhancing the performance of the classifier network, and reducing its convergence time. On the other hand, segmentation is a crucial step in the proposed work, as it enables us to localize the plant leaf by finding the object contour with the help of the contour detection algorithm. The extreme points of the obtained contours are calculated accordingly. These extreme points represent the coordinates of the object boundary, which are subsequently projected onto the original input image to perform the localization task. In addition, these extreme points are used to crop the object of interest out of the entire image. Resize with padding technique is then applied to the cropped image based on the conditions defined in Figure 5. If the dimension of the

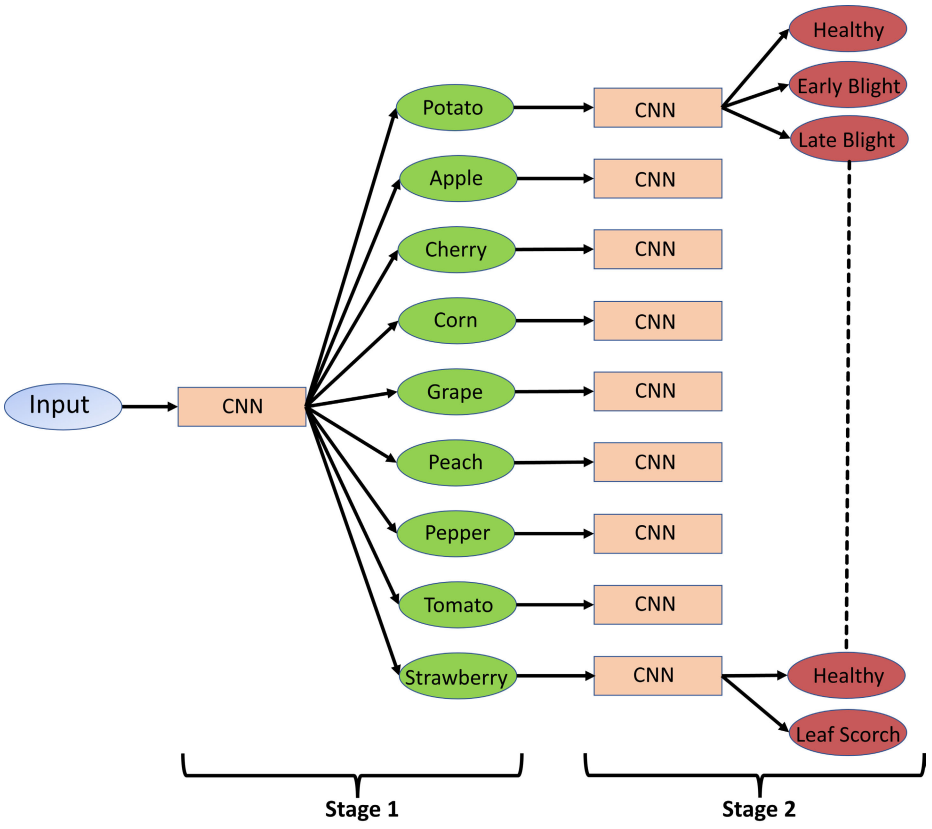

FIGURE 7. The proposed two stage classification hierarchical approach.

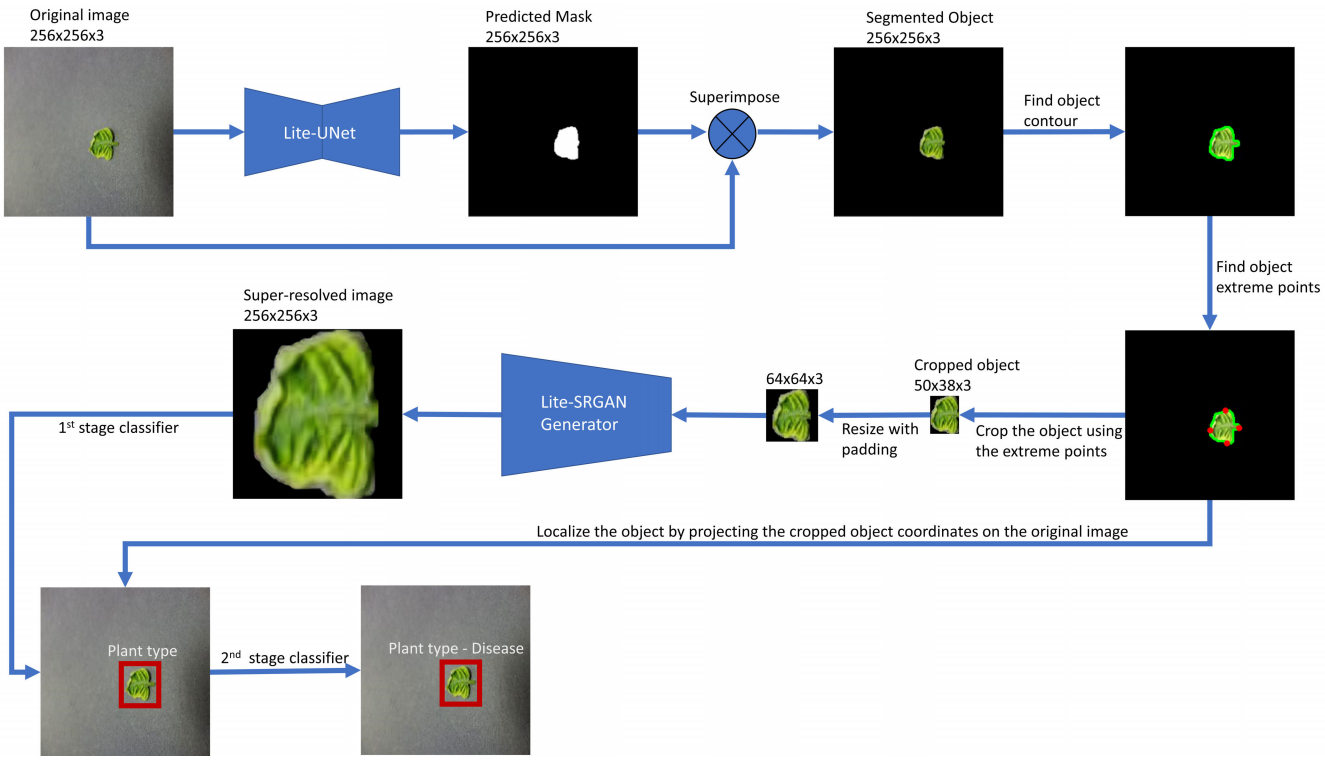

FIGURE 8. The pipeline for the whole proposed methodology.

resulting image after using the resize with padding technique is either 64 × 64 or 128 × 128, then it is fed into the trained generator of Lite-SRGAN to obtain a high-quality super-resolved image instead of using other traditional upscaling techniques such as bicubic and bilinear. as these techniques will lead to image distortion when enlarging the image, especially if the dimension of the image is relatively small. Finally, the super-resolved image is fed into the first stage classifier to identify the plant leaf type. After that, the second stage classifier identifies the diagnosis of the given plant leaf image. Moreover, as depicted in Figure 8, we project the coordinates of the leaf boundaries on the original input image to perform the localization task beside classification.

IV. EXPERIMENTAL RESULTS

All experiments in this work were conducted on Nvidia Tesla T4 GPU. This section provides a detailed description for the 1) dataset 2) evaluation metrics 3) segmentation results 4) super-resolution results & analysis and 5) results of the two-stage hierarchical classification approach.

A. DATASET

The plant village dataset [4] is the largest and most popular open-source dataset for different plant leaf diseases utilized by many researchers in the last few years for training and testing purposes. It is mainly used to classify different leaf crop diseases. The dataset consists of 54303 healthy and diseased leaf images divided into 38 categories from different 14 plant crop species with a fixed resolution (i.e., 256 × 256). In this research 9 different crop species are selected, belonging to 33 different categories. These categories are extracted from the plant village dataset and their corresponding masks are obtained from an open-source leaf mask dataset [7] for segmentation purposes.

The 9 different plant species are selected so we can make a multi-stage classification network. The first stage is responsible for identifying the plant leaf type, whereas the second stage is responsible for identifying the leaf diagnosis. Each type of plant leaf from the 9 types considered contains one healthy class and one or more diseased classes. Table 1 lists the categories belonging to each plant crop type and the number of training and testing images used for each category. The number of training and testing images varies from one category to another depending on the number of images in each category; the percentage of the training set used is 80%, while the remaining 20% of the data are used for testing purposes.

B. EVALUATION METRICS

The evaluation metrics used for assessing the performance of the considered classification models in this study are defined in equations (10) – (13).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Sensitivity(Recall) = \frac{TP}{TP + FN} \tag{12}$$

$$Specificity = \frac{TN}{TN + FP} \tag{13}$$

The evaluation metrics used to assess the performance of the proposed segmentation model in addition to accuracy are defined in equations (14) and (15):

$$IoU\ (Jaccard\ index) = \frac{TP}{TP + FP + FN} \tag{14}$$

$$Dice\ Coefficient = \frac{2xTP}{(TP + FP) + (TP + FN)} \tag{15}$$

where true positive (TP) refers to the number of samples from a given class that are correctly classified, false positive (FP) is the number of samples that are incorrectly classified to a given class while belonging to other classes, false negative (FN) is the number of samples that are misclassified to other classes, and true negative (TN) is the number of samples that are correctly classified to other classes.

C. SEGMENTATION RESULTS

This section presents in detail the training configurations and results obtained using the proposed Lite-UNet model. The model is trained and evaluated on the plant village dataset and its corresponding leaf mask dataset. The proposed segmentation method is mainly designed to work on mobile and low computing devices by using a lightweight encoder, and thus reducing the network latency and enabling it to work smoothly in a real time environment.

To illustrate the efficiency of the proposed Lite-UNet, we compared its obtained results with U-Net network. We trained the U-Net network on the same training samples used to train our proposed Lite-UNet network, and fairly evaluated both networks on the same test set based on the evaluation metrics stated in equations (14) and (15). Moreover, both models were compared in terms of complexity (parameters and FLOPs) and latency (inference time), as shown in Table 3. It is demonstrated from Table 3 that the proposed Lite-UNet surpasses the U-Net network in all evaluation aspects, which reveals the efficiency of the proposed Lite-UNet model. The configurations used to train both models are listed in Table 2. Figure 9 shows some random test samples for the plant leaf images and their corresponding ground truth masks, as well as the predicted mask generated by the proposed Lite-UNet network, and the segmented region of interest obtained by superimposing the original image with the obtained predicted mask.

Where LR reduce patience refers to the number of training epochs with no improvement, after which the learning rate will be decreased by the given reduce factor.

D. SUPER-RESOLUTION RESULTS & ANALYSIS

1) TRAINING DETAILS FOR LITE-SRGAN

All the experiments regarding super-resolution were performed on a 10,000 random sample from the plant village dataset. Two different versions of the proposed Lite-SRGAN

TABLE 1. Distribution of the training and testing samples for each category.

Plant type	Category	Number of training samples	Number of testing samples
Apple	Healthy	1316	329
	Apple scab	504	126
	Black rot	497	124
	Cedar Apple rust	220	55
Cherry	Healthy	683	171
	Powdery mildew	842	210
Corn	Healthy	930	232
	Cercospora leaf spot	410	103
	Common rust	954	238
	Nothern leaf blight	788	197
Grape	Healthy	338	85
	Black rot	944	236
	Esca (Black Measles)	1106	277
	Leaf blight	861	215
Peach	Healthy	288	72
	Bacterial spot	1838	459
Pepper	Healthy	1182	296
	Bacterial spot	798	199
Potato	Healthy	122	30
	Early blight	800	200
	Late blight	800	200
Strawberry	Healthy	365	91
	Leaf Scorch	887	222
Tomato	Healthy	1273	318
	Bacterial spot	1702	425
	Early blight	800	200
	Late blight	1527	382
	Leaf mold	762	190
	Septoria leaf spot	1417	354
	Spider mites	1341	335
	Target spot	1123	281
	Mosaic virus	298	75
	Yellow leaf curl virus	4286	1071

TABLE 2. Training parameters configurations of the proposed Lite-UNet model.

Hyperparameter	Value setting
Epochs	40
Learning rate	0.0001
Batch size	32
LR reduce patience and reduce factor	5, 0.2
Optimizer	Adam
Loss function	Dice loss

are utilized, the first version uses two upsampling layers, while the second version uses only one upsampling layer. For data preparation purposes, the LR images (64×64) are obtained by downsampling the HR images (256×256) with a scaling factor of 4 using the bicubic interpolation function in the first version. However, in the second version the LR images (128×128) are obtained by downsampling the HR images with a scaling factor of 2.

The training process is divided into two steps. Similar to the PSNR oriented models, in the first step, the generator is pre-trained slightly with L1 loss for 10^5 update iterations to avoid getting stuck in the local minima. Moreover, pretraining the generator enables the network to produce a well-looking super-resolved image, and thus allowing the discriminator to receive good super-resolved images in the initial iterations instead of extremely fake ones, which makes its task harder by focusing more on discriminating texture details. After that, the pre-trained model weights act as an initialization for the proposed Lite-SRGAN generator network. Secondly, we train both the generator and discriminator networks simultaneously. The loss function used to train the generator network is given by equation (9) where $\lambda_1 = 4x10^{-4}$,

TABLE 3. Comparison of model performance (accuracy, Dice score, IoU), model complexity (Params, FLOPs) and latency (averaged inference time) for U-Net and the proposed Lite-UNet.

Model	Accuracy ↑	Dice ↑	IoU ↑	#Params ↓	#FLOPs ↓	Inference time (ms) ↓	
						CPU	NVIDIA Tesla T4 GPU
U-Net	98.77	98.72	97.48	31M	1.09G	573	98
Lite-UNet (Ours)	98.83	98.77	97.58	1.95M	0.043G	87	60

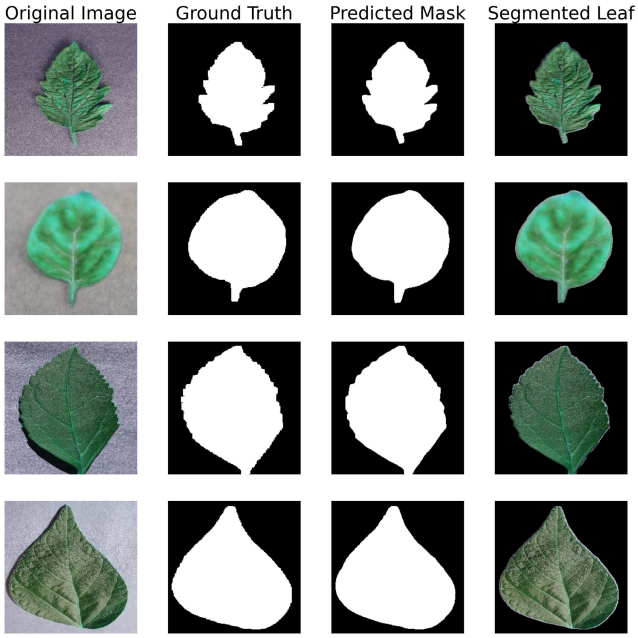

FIGURE 9. Random plant leaf samples (1st column), corresponding ground truth masks (2nd column), predicted masks generated by the proposed network (3rd column), corresponding segmented leaf (4th column).

$\lambda_2 = 1x10^{-3}, \lambda_3 = 0.2, \lambda_4 = 2x10^{-7}$. For optimizing the proposed Lite-SRGAN, we used the Adam optimizer [45] for both the generator and discriminator networks, where both the generator and discriminator learning rates are initialized with $1x10^{-4}$ and decayed by a factor of 2 every 10^5 mini-batch updates. The mini-batch size is set to 8.

2) QUANTITATIVE AND QUALITATIVE RESULTS

To fairly evaluate the effectiveness of the proposed Lite-SRGAN, it is compared to one of the state-of-the-art perception-driven super-resolution models SRGAN. Both the proposed model and SRGAN model are trained and evaluated using the same training and testing sets from plant village dataset. Since there is no unified and effective metric for evaluating the performance of super-resolution models. Thus, some visual qualitative results are shown in Figure 10.

In addition, some quantitative metrics are also considered for evaluation i.e., Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [46] (evaluated on the Y channel of the YCbCr color space), and Perceptual Index (PI). where higher value for PSNR and SSIM are better, while lower perceptual index indicates better perceptual quality and is defined as a combination of two non-reference quantitative measures: NIQE [47] and Ma's score [48] where PI=(NIQE+(10-Ma))/2. PSNR and SSIM tend to have better (higher) values with PSNR-oriented models as they produce smoothed results with less focus on texture and details which is somehow inconsistent with the human evaluation perspective, unlike perception-driven models, which focus more on reconstructing high-frequency details and textures in the super-resolved image. PI is considered a better choice when evaluating perception-driven approaches including SRGAN and the proposed Lite-SRGAN. It is also worth mentioning that the PI metric cannot be entirely relied upon as a superior metric for evaluating super-resolution models. As shown in Tables 4 and 5, it is illogical that the average PI for some methods is even better than the ground truth HR images.

Table 4 and Table 5 show the average of the three discussed quantitative metrics for each method on 500 random samples from the plant village dataset when using two upsampling layer (4x upscaling) and when using one upsampling layer (2x upscaling). Since our focus is also to build a lightweight network that can work smoothly in a real-time environment with a minimal desired latency, we also compared both models in terms of complexity (parameters and FLOPs) and latency (inference time) as shown in Table 6. Moreover, Figure 10 depicts some qualitative results from different cat-

TABLE 4. Quantitative results on 500 random samples from plant village dataset when upsampling from (64 to 256).

Metrics	PI ↓	PSNR ↑	SSIM ↑
HR	4.342	∞	1
Bicubic	7.004	28.576	0.536
SRGAN	4.316	27.064	0.474
Lite-SRGAN	4.396	27.802	0.490

TABLE 5. Quantitative results on 500 random samples from plant village dataset when upsampling from (128 to 256).

Metrics	PI ↓	PSNR ↑	SSIM ↑
HR	4.342	∞	1
Bicubic	4.898	31.630	0.714
SRGAN	4.248	29.237	0.594
Lite-SRGAN	4.203	30.330	0.659

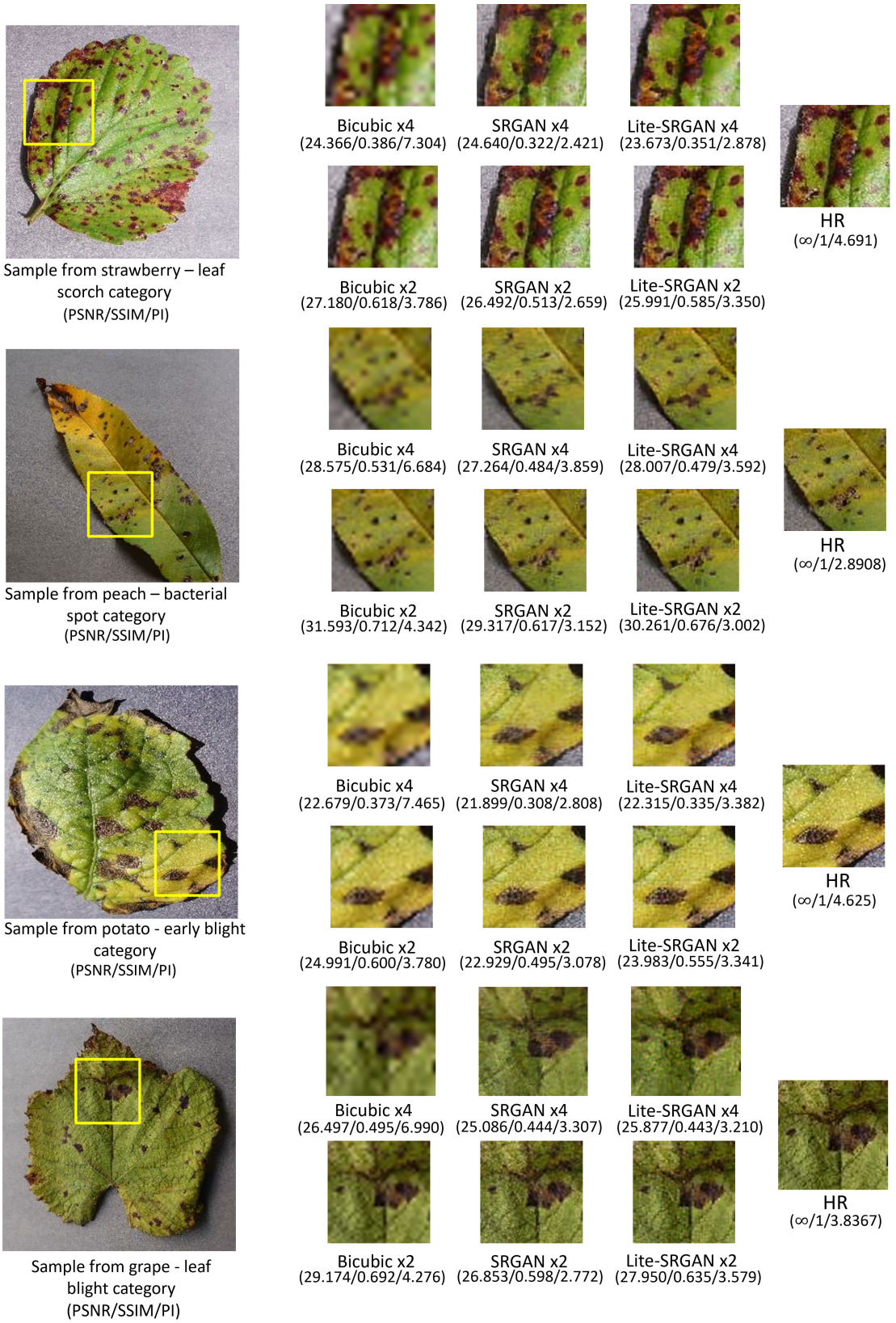

FIGURE 10. **Qualitative results of different methods (Zoom in for the best view), x4 (indicates upsampling the image from 64 × 64 to 256 × 256). However, x2 (indicates upsampling the image from 128 × 128 to 256 × 256).**

TABLE 6. Comparison of generator trainable parameters, FLOPs, and averaged inference time between SRGAN and the proposed Lite-SRGAN.

Upsampling resolution	Method	Generator #Params ↓	Generator #FLOPs ↓	Inference time (ms) ↓	
				CPU	NVIDIA Tesla T4 GPU
64p to 256p	SRGAN	1,554,883	181.7M	240	72
	Lite-SRGAN (Ours)	208,547	23.4M	88	61
128p to 256p	SRGAN	1,407,107	472.4M	427	103
	Lite-SRGAN (Ours)	199,267	42M	225	68

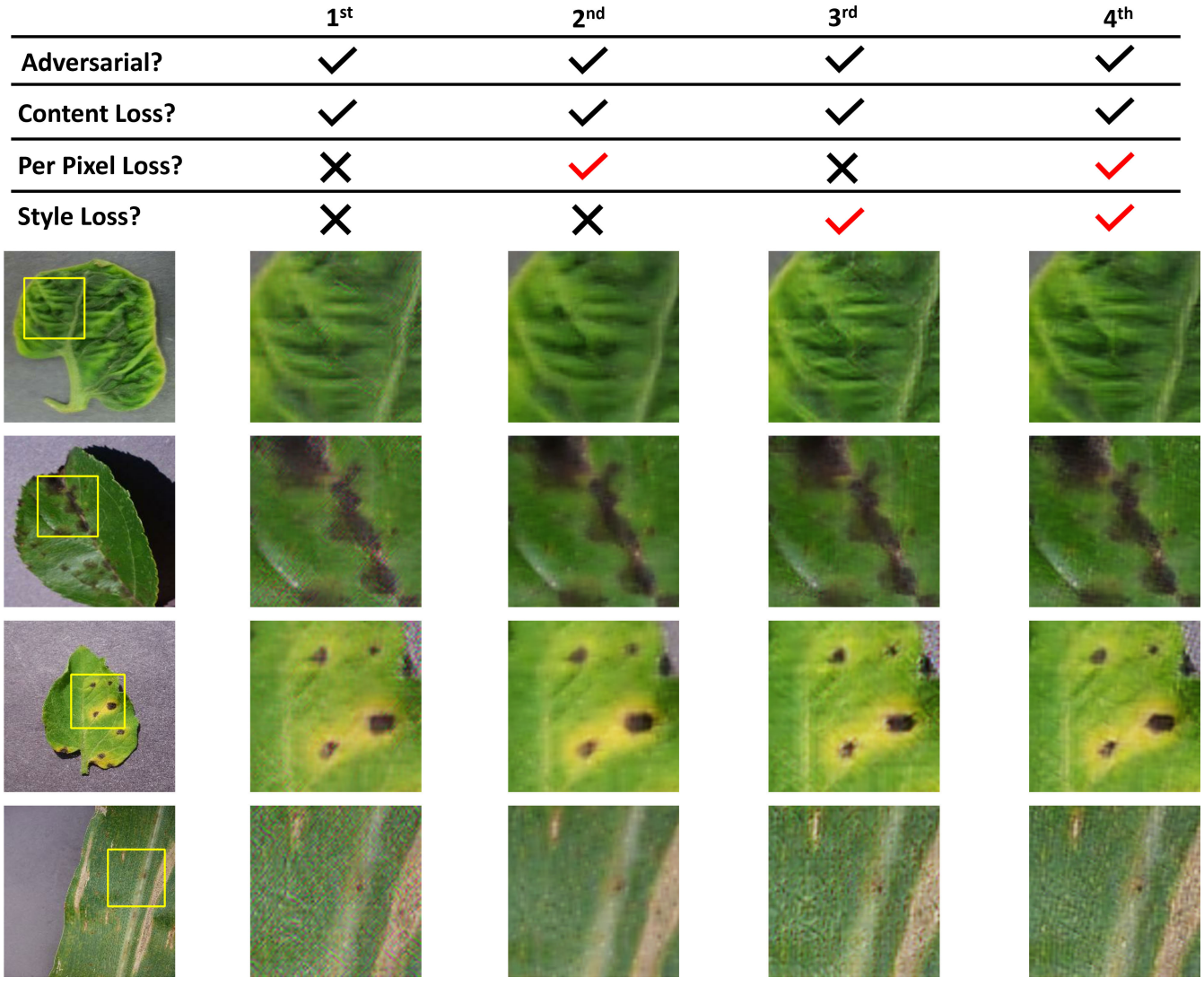

FIGURE 11. Visual comparison of combining different loss functions together (Zoom in for the best view), Each column represents a different experiment. The red sign indicates the added loss function in each experiment.

egories and states their PSNR, SSIM, and PI. Although, these metrics are calculated based on the whole super-resolved images, but we only show a batch from the image for better clarification of the visual impact when using each method.

From Table 4 and Table 5, it can be concluded that bicubic method has the highest PSNR and SSIM values due to the over-smoothed results obtained when using the bicubic interpolation technique. However, the perceptual index is low and comparable for both the SRGAN and Lite-SRGAN models and high for the Bicubic method, as it has less focus on reconstructing image textures and details. Furthermore, the qualitative results shown in Figure 10 demonstrate that the proposed Lite-SRGAN can reconstruct a superior SR image like SRGAN while being capable of producing sharper and enhanced textures in some images. In addition, the proposed model is optimized for real-time use due to its obvious

TABLE 7. Stage 1 results using the three pre-trained CNN models on the test set.

Model	Accuracy	Precision	Sensitivity	Specificity
DenseNet121	99.91%	99.91%	99.91%	99.99%
MobileNetV3	99.96%	99.96%	99.95%	100%
ConvNeXt	99.94%	99.94%	99.94%	99.99%

TABLE 8. Stage 2 results using the three pre-trained CNN models on the test set.

Model	Accuracy	Precision	Sensitivity	Specificity	Type
DenseNet121	100%	100%	100%	100%	Apple
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	100%	100%	100%	100%	Cherry
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	98.31%	98.31%	98.31%	99.46%	Corn
MobileNetV3	98.05%	98.05%	98.05%	99.37%	
ConvNeXt	98.57%	98.57%	98.57%	99.54%	
DenseNet121	100%	100%	100%	100%	Grape
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	100%	100%	100%	100%	Peach
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	100%	100%	100%	100%	Pepper
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	100%	100%	100%	100%	Potato
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	100%	100%	100%	100%	Strawberry
MobileNetV3	100%	100%	100%	100%	
ConvNeXt	100%	100%	100%	100%	
DenseNet121	99.83%	99.83%	99.83%	99.98%	Tomato
MobileNetV3	99.72%	99.78%	99.72%	99.98%	
ConvNeXt	99.81%	99.81%	99.81%	99.98%	

superiority in terms of complexity and latency compared to SRGAN, as depicted in Table 6.

3) EFFECT OF LOSS FUNCTIONS

To study the effect and contribution of each loss function used in the proposed Lite-SRGAN, we conducted different experiments when combining the used loss functions together as depicted visually in Figure 11. The first column represents the 1st experiment when using the adversarial loss along with the perceptual loss, which are the loss functions commonly used to optimize most of the recent perceptual driven super-resolution approaches, while the 2nd experiment shows the effect of adding per-pixel loss (L1 loss), and the 3rd experiment shows the effect of adding style loss. Finally, the 4th experiment shows the effect of using all loss functions together.

TABLE 9. Average accuracy, precision, sensitivity, and specificity for each model in stage 2.

Model	Average accuracy	Average precision	Average sensitivity	Average specificity
DenseNet121	99.79%	99.79%	99.79%	99.94%
MobileNetV3	99.75%	99.76%	99.75%	99.93%
ConvNeXt	99.82%	99.82%	99.82%	99.95%

TABLE 10. Comparison of plant diseases classification results with related works on plant village dataset.

Study	Method	Number of classes	Overall accuracy
[19]	Hybrid CNN	30	99.27%
[14]	C-GAN +DenseNet121	5	99.51%
		7	98.65%
		10	97.11%
[15]	CAE+CNN	2	98.38%
[18]	Modified U-Net + EfficientNet-B7	2	99.95%
	Modified U-Net + EfficientNet-B7	6	99.12%
	Modified U-Net + EfficientNet-B4	10	99.89%
[13]	DWT + PCA + GLCM + CNN	6	99.09%
[16]	VGG16	19	95.2%
[17]	CNN	17	99.36%
[20]	DoubleGAN + DenseNet121	10	99.53%
Our study	**Proposed method + MobileNetV3**	**33**	**99.68%**
	Proposed method + DenseNet121	**33**	**99.74%**
	Proposed method + ConvNeXt	**33**	**99.76%**

In this work, it is concluded from the conducted experiments that the main contribution of per-pixel loss is that it helps to reduce the undesired high-frequency noise and visual artifacts that may result in the super-resolved image, while the main contribution of style loss is that it produces more realistic and informative textures in the resulting image. However, the drawback of per-pixel loss is that it somehow smooths the high-frequency details, as shown in the 2nd experiment so its contribution to the total loss function should be carefully weighted. Based on our experiments, combining the four loss functions with appropriate weights is the best choice as it provides a better visual appearance for super-resolved images with less artifacts and good textures.

E. RESULTS OF THE PROPOSED TWO-STAGE HIERARCHICAL APPROACH

This section discusses the results obtained using the multi-stage classification approach according to the proposed pre-processing techniques. Since the proposed models used for preprocessing follow a lightweight design to gain the ability to work smoothly in a real-time environment. Thus, different pre-trained lightweight models are applied for classification, including: MobileNetV3 [10] and DenseNet121 [11]. In addition, the recent state-of-the-art ConvNeXt model [12] is experimented which is a pure convolutional neural network inspired by the design of vision transformers. These models are pre-trained on the ImageNet dataset [26]. For each model, we replaced the flatten layer with a global average pooling layer [49] to reduce the number of trainable parameters as well as overfitting. Following the global average pooling layer, two fully connected layers (dense layers) are added with 256 and 128 neurons respectively. Dropout regularization [50] is also added after each fully connected layer with a keep probability of 0.5. Finally, a softmax layer is used as an output layer with (n) neurons where n varies according to the number of classes.

For training purposes, Adam optimizer [45] is used to optimize the MobileNetV3 and DenseNet121 networks, while the ConvNeXt network is optimized using AdamW optimizer [51] as proposed by its authors. The training and testing sets are randomly selected according to the data splits listed in Table 1. The considered performance metrics used for evaluating the used models are accuracy, precision, sensitivity, and specificity which are given in equations (10) – (13). Tables 7 and 8 show the detailed results for both stages 1 and 2 respectively on the testing set.

It can be deduced from Table 7 that MobileNetV3 achieves the best performance for stage 1 with an overall accuracy of

99.96% on the test set, whereas ConvNeXt achieves the best performance for stage 2 with an average accuracy of 99.82%, as listed in Table 9. The three models achieved very high and comparable results in both stages.

It is also worth mentioning that MobileNetV3 is considered the lighter model among them due to its few trainable parameters. Although, the ConvNeXt performance is slightly better than MobileNetV3 and DenseNet121. However, ConvNeXt has much higher trainable parameters than both of them. Thus, MobileNetV3 is the best choice for the entire proposed lightweight approach due to its low complexity and its high achieved performance.

F. COMPARISON OF THE PROPOSED WORK CLASSIFICATION RESULTS WITH RELATED WORKS

The comparison between the proposed work and other studies is given in Table 10. The overall accuracy of the proposed two-stage classification approach is calculated based on the total misclassification in both stages. It is observed from Table 10 that our proposed work outperforms other studies while taking advantage of introducing a fully lightweight approach that is adequate for real-time use.

V. CONCLUSION

To facilitate the employment of deep convolutional networks on mobile, embedded, and resource-constrained devices particularly for real-time use. In this study, we focus on introducing a fully lightweight approach for performing different tasks, including segmentation, localization, super-resolution, and classification with low computational cost and minimal possible latency. To achieve this purpose, this paper introduces two novel lightweight-based architectures named Lite-UNet and Lite-SRGAN. All experiments in this study are conducted using the large publicly Plant Village dataset.

The full proposed approach can be described in the following manner. First, the Lite-UNet network is utilized to obtain the segmentation masks for the input plant leaf images. In addition, the obtained segmentation masks are used to crop the plant leaves from the given images to neglect the effect of the background area that does not contain any relevant features and thus, enhancing the classification results and enabling the classification model to converge faster. Thereafter, the cropped images are upsampled to a unified resolution. The proposed Lite-SRGAN is used to reconstruct a high-quality HR image from the given cropped LR image when the cropped image resolution is relatively small. In such case, it is used instead of basic interpolation techniques to preserve the texture and details of the enlarged image and avoid over-smoothed results. Finally, a two-stage classification approach is introduced, where the crop category is identified in the first stage, and its corresponding leaf disease is recognized in the second stage.

The proposed Lite-UNet allows memory-efficient inference with a significant reduction in parameters and FLOPs, while achieving promising results compared to U-Net. Moreover, the proposed Lite-SRGAN achieves comparable visual results with SRGAN, while outperforming it by a significant margin in terms of parameters, FLOPs, and inference time.

For classification purposes, three different pre-trained models are used in both stages including MobileNetV3, DenseNet121, and ConvNeXt. The extensive experiments conducted in this study demonstrates the efficiency of the proposed techniques for segmentation, localization, super-resolution, and classification. Considering the classification results obtained, it is concluded that the proposed work surpasses other related studies, which reveals the superiority of the overall proposed methodology.

In the future, we will consider experimenting the proposed Lite-UNet and Lite-SRGAN on other popular datasets.

REFERENCES

[1] J. B. Ristaino, P. K. Anderson, D. P. Bebber, K. A. Brauman, N. J. Cunniffe, N. V. Fedoroff, C. Finegold, K. A. Garrett, C. A. Gilligan, C. M. Jones, M. D. Martin, G. K. MacDonald, P. Neenan, A. Records, D. G. Schmale, L. Tateosian, and Q. Wei, "The persistent threat of emerging plant disease pandemics to global food security," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 23, Jun. 2021, doi: 10.1073/pnas.2022239118.

[2] R. N. Strange and P. R. Scott, "Plant disease: A threat to global food security," *Annu. Rev. Phytopathology*, vol. 43, no. 1, pp. 83–116, Sep. 2005, doi: 10.1146/annurev.phyto.43.113004.133839.

[3] D. M. Rizzo, M. Lichtveld, J. A. K. Mazet, E. Togami, and S. A. Miller, "Plant health and its effects on food safety and security in a one health framework: Four case studies," *One Health Outlook*, vol. 3, no. 1, Dec. 2021, doi: 10.1186/s42522-021-00038-7.

[4] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[5] 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, doi: 10.1109/ACCESS.2021.3069646.

[6] S. Suzuki and K. Be, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph., Image Process.*, vol. 30, no. 1, pp. 32–46, Apr. 1985, doi: 10.1016/0734-189X(85)90016-7.

[7] *spMohanty/PlantVillage-Dataset: Dataset of Diseased Plant Leaf Images and Corresponding Labels*. Accessed: Mar. 14, 2023. [Online]. Available: https://github.com/spMohanty/PlantVillage-Dataset

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[9] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114, doi: 10.1109/CVPR.2017.19.

[10] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[12] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976, doi: 10.1109/CVPR52688.2022.01167.

[13] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transitions Proc.*, vol. 3, no. 1, pp. 305–310, Jun. 2022, doi: 10.1016/j.gltp.2022.03.016.

[14] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106279, doi: 10.1016/j.compag.2021.106279.

[15] P. Bedi and P. Gole, "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network," *Artif. Intell. Agricult.*, vol. 5, pp. 90–101, 2021, doi: 10.1016/j.aiia.2021.05.002.

[16] A. A. Alatawi, S. M. Alomani, N. I. Alhawiti, and M. Ayaz, "Plant disease detection using AI based VGG-16 model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, p. 2022, 2022, doi: 10.14569/IJACSA.2022.0130484.

[17] S. M. Hassan and A. K. Maji, "Plant disease identification using a novel convolutional neural network," *IEEE Access*, vol. 10, pp. 5390–5401, 2022, doi: 10.1109/ACCESS.2022.3141371.

[18] M. E. H. Chowdhury, T. Rahman, A. Khandakar, M. A. Ayari, A. U. Khan, M. S. Khan, N. Al-Emadi, M. B. I. Reaz, M. T. Islam, and S. H. M. Ali, "Automatic and reliable leaf disease detection using deep learning techniques," *AgriEngineering*, vol. 3, no. 2, pp. 294–312, May 2021, doi: 10.3390/agriengineering3020020.

[19] A. Tuncer, "Cost-optimized hybrid convolutional neural networks for detection of plant leaf diseases," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 8, pp. 8625–8636, Aug. 2021, doi: 10.1007/s12652-021-03289-4.

[20] Y. Zhao, Z. Chen, X. Gao, W. Song, Q. Xiong, J. Hu, and Z. Zhang, "Plant disease detection using generated leaves based on DoubleGAN," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 3, pp. 1817–1826, May 2022, doi: 10.1109/TCBB.2021.3056683.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[23] A. Rakhlin, A. Davydow, and S. Nikolenko, "Land cover classification from satellite imagery with U-Net and Lovász-softmax loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 257–2574, doi: 10.1109/CVPRW.2018.00048.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632.

[25] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 89–105, doi: 10.1007/978-3-030-01252-6_6.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[27] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. on 3D Vis. (DV)*, Oct. 2016, pp. 565–571, doi: 10.1109/3DV.2016.79.

[28] Y. Chen, F. Shi, A. G. Christodoulou, Y. Xie, Z. Zhou, and D. Li, "Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2018, pp. 91–99, doi: 10.1007/978-3-030-00928-1_11.

[29] H. M. Keshk and X. Yin, "Satellite super-resolution images depending on deep learning methods: A comparative study," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Oct. 2017, pp. 1–7, doi: 10.1109/ICSPCC.2017.8242625.

[30] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3224–3232, doi: 10.1109/CVPR.2018.00340.

[31] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. Change Loy, Y. Qiao, and X. Tang, "ESRGAN: Enhanced super-resolution generative adversarial networks," 2018, *arXiv:1809.00219*.

[32] Y. Zhang, Z. Zheng, and R. Hu, "Super resolution using segmentation-prior self-attention generative adversarial network," 2020, *arXiv:2003.03489*.

[33] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021, *arXiv:2112.10752*.

[34] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5835–5843, doi: 10.1109/CVPR.2017.618.

[35] K. S. Krishnan and K. S. Krishnan, "SwiftSRGAN–rethinking super-resolution for efficient and real-time inference," in *Proc. Int. Conf. Intell. Cybern. Technol. Appl. (ICICyTA)*, Dec. 2021, pp. 46–51, doi: 10.1109/ICICyTA53712.2021.9689188.

[36] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Towards real-time image enhancement GANs," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2019, pp. 183–195, doi: 10.1007/978-3-030-29888-3_15.

[37] M. Ayazoglu, "Extremely lightweight quantization robust real-time single-image super resolution for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2472–2479, doi: 10.1109/CVPRW53098.2021.00280.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

[39] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[40] L. Gatys, A. Ecker, and M. Bethge, "A neural algorithm of artistic style," *J. Vis.*, vol. 16, no. 12, p. 326, Sep. 2016, doi: 10.1167/16.12.326.

[41] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711, doi: 10.1007/978-3-319-46475-6_43.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[43] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423, doi: 10.1109/CVPR.2016.265.

[44] K. Kc, Z. Yin, D. Li, and Z. Wu, "Impacts of background removal on convolutional neural networks for plant disease classification in-situ," *Agriculture*, vol. 11, no. 9, p. 827, Aug. 2021, doi: 10.3390/agriculture11090827.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[47] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013, doi: 10.1109/LSP.2012.2227726.

[48] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Comput. Vis. Image Understand.*, vol. 158, pp. 1–16, May 2017, doi: 10.1016/j.cviu.2016.12.009.

[49] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.

[50] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1–30, Jun. 2014.

[51] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

HOSAM S. EL-ASSIOUTI was born in New York, USA, in 1998. He received the B.Sc. degree (Hons.) in computer science from Ain Shams University, Egypt, in 2020, where he is currently pursuing the M.Sc. degree in scientific computing with the Faculty of Computer and Information Sciences. He is a Teaching Assistant with the Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University. His research interests include deep learning, machine learning, computer vision, and image processing.

HADEER EL-SAADAWY was born in Cairo, Egypt. She received the B.Sc. (Hons.), M.Sc., and Ph.D. degrees in scientific computing from Ain Shams University, in 2013, 2018, and 2021, respectively. She is currently a Lecturer with the Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University. Her current research interests include bioinformatics, signal processing, image processing, machine learning, and deep learning. She has published ten publications in these areas.

MARYAM N. AL-BERRY was born in Cairo, Egypt. She received the B.S., M.Sc., and Ph.D. degrees in scientific computing from Ain Shams University, in 2001, 2007, and 2015, respectively. From 2020 to 2021, she was the coordinator of three credit hour programs, namely digital multimedia, artificial intelligence, and cyber security. From 2021 to 2022, she was the Coordinator of the Digital Multimedia Program. She is currently the Coordinator of the Bioinformatics Program. She is also an Assistant Professor with the Scientific Computing Department, Faculty of Computers and Information Sciences, Ain Shams University. Her research interests include image processing and analysis, action and activity recognition, satellite image segmentation, machine learning, and deep learning.

MOHAMED F. TOLBA (Senior Member, IEEE) has been a Professor of scientific computing with Ain Shams University, since 1984, where he was the Vice President, from 2002 to 2006, and the Dean of the Faculty of Computers and Information Sciences, from 1996 to 2002. He has supervised more than 90 M.Sc. and 50 Ph.D. students at Ain Shams University and other Egyptian universities. He is currently a consultant to different local and international organizations for IT. He has more than 220 publications in the fields of AI, image processing, pattern recognition, OCR, scientific computing, and simulation and modeling. He was a member of the International Association for Science and Technology for Development (IASTED), Canada, from 1995 to 2007; the International Society for Computers and their Applications (ISCA), USA, from 1998 to 2007; the Advisory Committee of Strengthening Science and Technology Researchers Project—STRP Ministry of Scientific Research, from 2006 to 2009; and the Committee for Evaluation of Egyptian Space Program of the National Authority for Remote Sensing and Space Sciences—Ministry of Scientific Research. He has been a member of the Association for Computing Machinery (ACM), USA, since 2000; the Software Engineering Competence Center (SECC), since 2004; the Information Technology Academic Collaboration (ITAC), since 2005; and the E-Learning Committee Board, since 2008. He is also the honorary chairperson of many international conferences and the chairperson of several IT sector committees in Egypt.