INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 16 No. 3 (2024) 166-180
https://publisher.uthm.edu.my/ojs/index.php/ijie

MobUNet: Utilizing Deep Learning for Segmenting Cucumber Leaves

Nurul Amirah Mashudi[1]*, Norulhusna Ahmad[1], Suriani Mohd Sam[1], Norliza Mohamed[1], Hazilah Mad Kaidi[1], Norliza Mohd Noor[1]

[1] *Faculty of Artificial Intelligence,*
Universiti Teknologi Malaysia, Kuala Lumpur, 54100, MALAYSIA

*Corresponding Author: amirahmashudi@gmail.com
DOI: https://doi.org/10.30880/ijie.2024.16.03.016

Article Info

Received: 2 August 2024
Accepted: 11 October 2024
Available online: 3 November 2024

Keywords

Deep learning, image segmentation, cucumber leaf segmentation, U-Net, MobileNetV2

Abstract

Plant image segmentation is challenging due to overlapping leaves and complex image backgrounds. Consequently, the segmentation model has some challenges recognizing the leaves, further affecting the segmentation performance. This study proposes a deep learning method called MobUNet, using U-Net, and employs MobileNetV2 as an encoder to overcome the problems for cucumber leaf segmentation. Around 145 leaf images with complex backgrounds are collected at the cucumber farm and annotated for ground truth data. The experiment uses the ratio of 80:20 for training and testing sets, and some hyperparameters are modified to achieve a good segmentation result. The segmentation results are subject to several metrics: accuracy, Dice score, IoU, Dice loss, Jaccard distance, and Hausdorff distance. The experimental results for segmentation accuracy, Dice score, and IoU were 93.23%, 91.30%, and 85.03%, respectively. An analysis was conducted to create a benchmark in segmentation performance, utilizing the U-Net baseline, MobileNetV1, and MobileNetV2, which use the same dataset. Despite the complex background, MobUNet can successfully segment the cucumber leaf images compared to the other models. The MobUNet showed the closest Hausdorff distance value to the origin point, measuring at 0.0001; hence, it demonstrates high quality and accuracy in the segmentation.

1. Introduction

Cucumbers have been essential for human diets for thousands of years, providing a refreshing and nutritious snack. In addition, cucumber plants are an essential agricultural commodity due to their intense farming style (fast-growing plant) that yields a harvest that can provide a substantial income to the farmers if done correctly. However, the plant is quickly succumbing to various diseases that will undoubtedly affect farmers' income. Innovative and long-lasting solutions are required to address the problems associated with farming, such as pests, diseases, and changing climates. Cucumber leaves play a crucial role in the plant's growth and photosynthesis. Conventional methods for monitoring and analyzing the structure and health of cucumber leaves, on the other hand, are time-consuming and frequently subjective [1]. Hence, image analysis and computer vision provide more accurate and efficient leaf segmentation. Moreover, deploying a drone or a mobile phone to capture, analyze, and monitor leaf images can indicate the effectiveness of edge computing.

The development of imaging methods has led to their widespread use in various fields, including geoscience [2], smart farming [3], and remote sensing [4]. Non-destructive plant phenotyping techniques based on computer vision hold great promise, automatically capturing features with minimal human intervention [5]. On the other

hand, image analysis of various plant organs is the primary focus of computer vision-based research projects to track plant development, study plant anatomy, and identify plant diseases. Leaves, more than any other organ, reveal the progress of vegetation and allow us to track its developmental phases. Parameters like leaf area, leaf shape, leaf count, and others provide insight into important aspects of plant biology and physiology, such as respiration, nutrition, and photosynthesis [6], [7].

Recent plant morphological trait analyses have been widely employed to learn how different species react to biotic and abiotic environmental conditions. As a result of several ongoing initiatives, information on morphological traits continues to be transcribed into various forms useful in ecology and earth system studies [8]. Most past procedures used to determine morphological traits have been labour-intensive and time-consuming, requiring specialized tools and experts. Today, several applications, such as LeafJ [9] and Easy Leaf Area [10], can analyze digital images of leaves, identify the outlines, and classify the shapes.

Deep learning is a cutting-edge technology that can automatically identify different aspects of an image, making it useful for tasks such as segmentation. However, plant inherent and environmental limitations make leaf segmentation challenging [5]. Environmental factors such as changing illumination, shadows, and blurring from the wind can increase the difficulties that exist in plants, such as textural variation, changes in leaf shapes and sizes, a prominent vein on the leaf, and overlapping leaves [11]. Thus, most researchers developed deep learning models based on convolutional neural networks (CNN) architecture to tackle these challenges.

This study focuses on the limited dataset initially captured by the researchers using a mobile device at a cucumber farm in Malaysia. Hence, it has not been utilized to train any deep learning models. The acquisition of this cucumber leaf data is of utmost importance to obtain significant conclusions and facilitate well-informed decision-making. Identifying cucumber plant components, like its leaf, necessitates using robust object detection and image segmentation techniques [12], [13].

Despite the significance of Hausdorff distance in providing a robust, global, and boundary-sensitive evaluation of image segmentation quality, it has received relatively less attention from researchers. Several studies conducted in the domain of segmentation quality evaluation tend to prioritize measures such as intersection over union (IoU) and Dice coefficient, which are well-acknowledged and understandable. However, the limited use of Hausdorff distance might be attributed to its high computing burden and perceived complexity despite its ability to provide crucial data regarding the spatial accuracy and shape fidelity of segmented regions.

This paper aims to develop a deep learning model based on U-Net architecture and MobileNetV2 as the backbone to support the cucumber leaf segmentation process. The significant contributions of the paper are as follows,

- This paper proposed a segmentation method, MobUNet, for cucumber leaf segmentation using a MobileNetV2 encoder with U-Net architecture.
- The MobUNet validates the performance of the primary dataset captured at the cucumber farm in Banting, Selangor, Malaysia.
- The MobUNet evaluates the performance based on segmentation accuracy, Dice coefficient score, and IoU and measures the segmentation quality using Hausdorff distance, Jaccard distance, and Dice loss.

The following sections organize the paper: Section 2 presents the related work of leaf segmentation and detection using deep learning methods. Section 3 explains the proposed method for cucumber leaf segmentation and the materials used to perform the analysis. Section 4 discusses the experimental results on the performance of segmentation, including the quality of the segmentation of the proposed method. Lastly, Section 5 concludes the overall paper and future works.

2. Related Works

Some researchers have recently produced numerous leaf segmentation and identification systems. Deep Leaf's new deep learning method performed leaf detection and pixel-wise instance segmentation using Mask Region CNN (MRCNN) [8]. The study has improved the architecture of ResNet-50 and ResNet101 to achieve the robustness of Deep Leaf. A novelty of semantic segmentation based on a modified U-Net using VGG was proposed in [14] to improve the weed segmentation from the soil and crops. The image labelling process was one of the challenges in the study due to the complexity of the weeds. Hence, image augmentation is needed to produce more weed images.

A leaf disease is essential to tackle the problems in vegetation systems. A combination of U-Net and DeepLabV3+ was proposed for a complex segmentation of cucumber leaf disease [15]. The purpose of DeepLabV3+ is to remove the complex background on the leaf images using the feature fusion technique. Another hybrid method using CNN and Watershed was developed based on pixel-wise instance segmentation [16] to detect and identify leaves in greenery settings. The proposed method was compared with MRCNN to achieve a significant image segmentation benchmark.

A modified U-Net (MU-Net) based on the MultiResUNet structure was proposed to calibrate leaf disease images [17]. Combining Resblock with Respath increases network depth and expression. The author combines

these two structures to work with leaf segmentation efficiently. Bhagat *et al.* have proposed Eff-Unet++ by employing EfficientNet-B4 as an encoder and decoder to analyze the leaf segmentation and counting [5]. The improved skip connections reduced the computational complexity. In addition, the lateral output layer aggregates the decoder's low-level to high-level features for better segmentation.

The MRCNN was used as a segmentation method to remove the background and extract features of the overlapping leaves on the images [18]. First, the study analyzed the leaf images based on the maximum epoch, learning rate, momentum, and training time. Then, the author implemented VGG-16 for classification to achieve classification accuracy. A study in [19] developed a lightweight U-Net based on the conventional U-Net, called a lightweight multi-scale extended U-Net (LWMSDU-Net), for leaf disease image segmentation. Generally, the method has subnetworks for encoding and decoding, which the sub-network encodes using multi-scale extended convolution and decodes using a deconvolution model. Then, it fuses the input image's shallow and deep features using the residual link between the two modules.

Guo *et al.* in [20] have employed the MRCNN as proposed in [8], [16], [18] based on dual-attention guided mask (DAG-Mask), mask assembly, and mask refining modules. The study achieved significant Dice score results for leaf segmentation using the LSC dataset. Ngugi *et al.* [21] have proposed a new deep learning method, KijaniNet, which uses a multi-scale encoder based on SegNet and U-Net. SegNet was presented for distributed mobile apps that perform inference on the mobile device, whereas KijaniNet is best for centralized processing. An unsupervised image segmentation fusion was adopted in [22] to tackle the problems of plant leaf segmentation. The authors adopted three algorithms: k-means, self-organizing map (SOM), and fuzzy c-means (FCM) based on the g-calculus and maximum mutual information to find the Dice, Jaccard, Manhattan, and fusion time. Table 1 summarizes the research gaps concerning methods used and limitations in the state-of-the-art. In addition, several limitations are considered for the method used in this study.

The U-Net architecture was introduced by Ronneberger *et al.* [23], contributing significantly to semantic image segmentation. The U-Net architecture includes contracting and expanding paths with skip connections. It has proven highly successful in capturing fine details and general context, especially in medical image analysis. Conversely, the MobileNetV2 model [24] has played a crucial role in efficient mobile and embedded vision applications. The depthwise separable convolutions of this architectural design, which effectively minimize computational complexity without compromising performance, render it highly suitable for environments with limited resources.

MobUNet takes advantage of the weaknesses of U-Net in high memory consumption during the training phase. Therefore, MobUNet employs MobileNetV2, which was developed specifically for efficient mobile applications that lower power consumption in the training phase and improve the performance of cucumber leaf segmentation. Both models have inherent advantages and disadvantages, and the selection between them is contingent upon the specific needs of the application and the computing resources that are accessible.

Table 1 *Summary of the research gaps from the literature*

Authors	Method	Advantages	Limitations
Triki *et al.* [8]	MRCNN, ResNet-50, ResNet-101	The model employed the Mish activation function instead of ReLU, improving information propagation and reducing training error.	Excluding leaves with missing parts from the analysis. Limited data resources.
Zou *et al.* [14]	Modified U-Net: VGG&	The model was simplified to eliminate redundant layers, improving segmentation accuracy, network speed, and efficiency.	Longer training time for segmentation. Limited data resources.
Wang *et al.* [15]	DeepLabV3 and U-Net	Combining DeepLabV3+ and U-Net reduced parameters significantly. Only 6.67% of the DeepLabV3+ network's parameters were merged, making it suitable for limited resources mobile devices.	Segmentation errors occur, especially for small targets. Training time took longer and needed optimization.
Vayssade *et al.* [16]	CNN + Watershed	The method supported mixed-species plant images and natural light images. A vegetation index and watershed algorithm	No contributions to each block in the network. Data augmentation is needed to increase the number of images.

		improved segmentation output.	
Zhang *et al.* [17]	MU-Net	MU-Net segmentation was enhanced with Resblocks and Respaths, which boosts training and feature extraction.	The model needs additional testing and optimization to be lightweight.
Bhagat *et al.* [5]	Eff-Unet++	The modified UNet++ architecture's redesigned skip connections preserve information while reducing computational demands.	The model is not suitable for maximum leaf occurrences in unknown data.
Yang *et al.* [18]	MRCNN	The pixel-level recognition and extraction of object regions from the background generate Mask R-CNN for leaf segmentation, which requires exact contour and shape data.	The model is not working for real-time applications.
Xu *et al.* [19]	LWMSDU-Net	The VGG16 model used with Mask R-CNN has fewer parameters and depth than VGG19 and Inception ResNetV2, making it better for training with limited datasets. The model outperformed Grabcut and the Otsu segmentation algorithm in leaf image segmentation with a misclassification error (ME) of 1.15%.	Some optimizations are needed for the edge-computing version.
Guo *et al.* [20]	LeafMask	Capturing global and local features in the DAG-Mask branch optimizes data expression and segmentation for small and large leaves. More accurate leaf masks, especially at the edges, are produced by this module's adaptive selection of points at leaf borders and computation of sharp boundaries to prevent noise effects.	The model works only for one type of plant phenotype.
Ngugi *et al.* [21]	KijaniNet	Despite its better performance, KijaniNet used less memory than U-Net as it does not require extra memory for copy and concatenate operations.	The background removal needs to be worked out. The method does not work for lightweight devices.
Nikbakhsh *et al.* [22]	Clustering: k-means, SOM, FCM	The method worked effectively for segmenting plant leaves from complex backgrounds. The method effectively integrates the results of several segmentation methods, including fuzzy c-	The study focused on the lighting variations only, not the leaf shapes and sizes.

	means, SOM, and k-means in various colour spaces and parameters, improving segmentation robustness.

3. Materials and Method

3.1 Image Acquisition

This study selects cucumber leaf images captured at the KMK Agro Global Sdn Bhd farm in Banting, Selangor, Malaysia. The images were captured in early 2022 using two types of mobile devices: iPhone 13 Pro Max and Huawei Nova 4e. The images captured with these devices were saved in JPG format with a resolution of 3024 × 4032 and 2448 × 3264, respectively. Moreover, only 145 images were captured using both devices, and no data augmentation techniques were employed for segmenting cucumber leaves to ensure the evaluation was based only on the original dataset. Figure 1 shows the image collection of cucumber leaves in various forms with complex backgrounds. The cucumber leaf images have disease spots on some leaves, which can be used for leaf disease segmentation and classification. However, other researchers can employ this dataset to identify the leaf shapes and sizes.

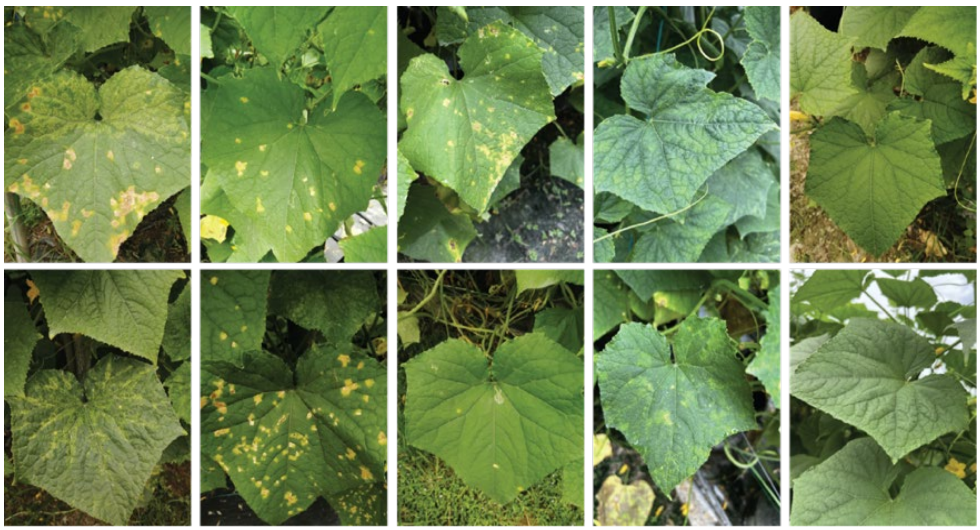

Fig. 1 *A collection of cucumber leaves*

3.2 Image Annotation

The dataset must be labelled before performing MobUNet segmentation. This study used RoboFlow to create annotations on leaf images. RoboFlow is a cloud-based environment used as a pipeline for developing artificial intelligence to help researchers or scientists develop training and testing images [25], [26], [27]. RoboFlow gives users some alternatives for pre-processing, augmentation, and exporting. The annotation processes in RoboFlow are straightforward: using a bounding box to draw a rectangle annotation, a freeform polygon tool to get a precise shape or a smart polygon tool that works as an intelligent assistant. This study employed a smart polygon tool, as in Figure 2, to annotate leaf images containing complex backgrounds. Image annotation aims to create a ground truth dataset for leaf segmentation.

3.3 Model Training

This study uses Google Colaboratory Pro, or Colab Pro for short, as a software platform to execute the MobUNet. Colab is a research initiative that uses a graphic processing unit (GPU) and tensor processing unit (TPU) for machine learning and data analysis using Python language without complex configuration [28]. Some libraries are available in Colab, such as Keras, PyTorch, Apache MxNet, OpenCV, XGBoost, GraphViz, and fastai. All dataset is stored in Google Drive and mounted on the drive to retrieve via Colab Notebooks. Colab Pro aims to work faster with high GPU and RAM in this study. Thus, it can avoid any interruption during the analysis. The hypermeters used in this study are an input size of 512 × 512, batch size of 24, a thousand of buffer size, 60 epochs, and Adam optimizer.

Fig. 2 *Image annotation on leaf images using RoboFlow*

3.4 Segmentation Performance Evaluation

The Dice coefficient score validates the pixel-wise consistency between a predicted segmentation and ground truth and measures the similarity of the leaves based on the predicted mask and ground truth mask. It divides the total of the two objects by the size overlapping in the segmentation. The Dice coefficient score computed in Equation 1 shows that two binary vectors, *A* and *B*, represent the truth and the classification result.

$$Dice(A,B) = \frac{2|A \cdot B|}{|A||B|} \tag{1}$$

Furthermore, this study also measures Dice loss. Dice loss solves the imbalance between foreground and background but ignores the imbalance between easy and hard instances, negatively impacting learning model training. The formula is stated as follows,

$$DiceLoss = 1 - Dice \tag{2}$$

Accuracy is used in this study to determine the performance of leaf segmentation. Accuracy examines all image pixels and assigns a score of 1 if the pixel is correctly predicted and 0 otherwise. Thus, the indicator function can quantitatively express it in Equation 3, where A_i is all elements considered, and B_i is the accuracy equal to 1.

$$Accuracy = \frac{1}{n}\sum_{i=0}^{n} 1_{(A_i = B_i)} \tag{3}$$

The Jaccard score, also known as IoU, is similar to the Dice coefficient score, where the area of the overlap divides the sum of the sample sets. It is a formal measurement concept that emphasizes similarities between limited sample sets. This study also calculates Jaccard distance as an optimization metric of iris segmentation. Equation 4 and Equation 5 state the formula of IoU and Jaccard distance.

$$IoU(A,B) = \frac{|A \cdot B|}{|max(A,B)|} = \frac{|A \cdot B|}{|A| + |B| - |A \cdot B|} \tag{4}$$

$$JD = (1 - IoU) \times 100 \tag{5}$$

The Hausdorff distance quantifies how close each point in a model set is to an image set and vice versa [29]. Therefore, this distance can be used to evaluate the similarity of two overlapping objects. Millimetres or voxels measure the average Hausdorff distance between the ground truth and segmentation voxel sets. The following formula shows the average Hausdorff distance for image segmentation,

$$AHD(A,B) = (\frac{A_x}{A} + \frac{B_x}{B}) \div 2 \quad (6)$$

A_x is from ground truth to predicted segmentation, B_x from the predicted segmentation to ground truth, A denotes the number of voxels in the ground truth, and B denotes the number of voxels in the segmentation.

4. System Model

Figure 3 shows the system flowchart implemented for the MobUNet segmentation method. The process begins with input leaf images for data labelling, performed using RoboFlow as described in Section 3.2. This study employed semantic segmentation to provide more accurate and comprehensive cucumber leaf recognition. The MobUNet was trained by processing the pixel label image. It offers data storage for training MobUNet using ground truth data as input. The training data contribute to 80% of the total image, while the remaining 20% of the leaf images are allocated for testing. By allocating 80% of the leaf image for training, MobUNet has sufficient information to learn from, resulting in improved generalization. The remaining 20% offers enough data to test the MobUNet, providing its statistical validity.

Alternative splits, as compared to an 80:20 split, possess drawbacks. Although a 90:10 split yields a larger training set, it also reduces the test set, resulting in less accurate evaluation results and an overestimation of the performance of MobUNet. On the other hand, a 70:30 split provides a smaller training set, which may lead to underfitting, but it also provides a larger test set, improving the robustness of performance evaluation. Therefore, the 80:20 split is preferred because it achieves a compromise between having enough training data to create a robust model and maintaining a large test set for accurate performance evaluation. It also serves to reduce the risks of overfitting and underfitting. This study examines the performance evaluation of cucumber leaf segmentation using metrics such as the Dice score, IoU, and accuracy. In addition, the Dice loss and Hausdorff distance were computed as supplementary metrics to evaluate the performance of MobUNet.

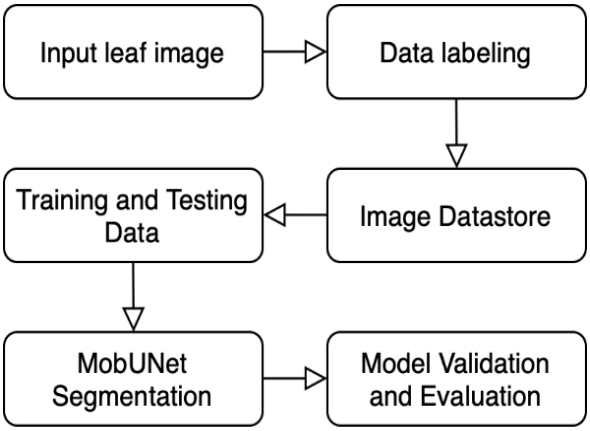

Fig. 3 *System flowchart of the MobUNet segmentation method*

4.1 U-Net

U-Net is a fully convolutional network (FCN)-inspired semantic segmentation network with the same network architecture. Ronneberger *et al.* proposed the U-Net [23], in which the image can either be a single channel or three channels with an input size of 512 × 512 to the network. Two good options for constructing the whole network are an encoding–decoding architecture or a contraction path with an expanding path. Each step of the contraction path relies on a pair of feature-extracting with 3 × 3 convolutions. The feature map is matched and fused with the contraction path at each step of the expansion path, which involves up-sampling the feature map.

Specifically, the U-Net networks employ higher-resolution, shallower layers to address the pixel positioning problem, while the network's deeper layers are employed to address the pixel classification problem. Hence, U-Net can reliably segment pixels. However, a detection method is required for each area centred on a pixel. As a result, redundant activities arise when major areas overlap, which may cause a cascade of issues, including poor performance and quality.

4.2 MobileNetV2

MobileNetV2 was proposed in [24] to incorporate inverted residual with linear bottleneck modules to improve upon its predecessor. MobileNet was built using depthwise convolution as its foundation [30]. Figure 4 shows that

the conventional 2D convolution employs depth-based channel convolving to process all input channels directly and generate a single output channel. This approach facilitates a comprehensive analysis of the input data by integrating information from all input channels. The depthwise convolution works by decomposing the input image and the filter into a series of channels and then convolving each with the matching filter channel. Filtered output channels are generated and then stacked back.

To merge the stacked output channels into one channel, the separable depthwise convolution first filters the stacked output channels using a 1 × 1 convolution, also known as pointwise convolution. The convolution operation (Conv) comprises moving a kernel filter over the input data to generate feature maps. With convolution, local patterns and spatial structures can be found in data, like edges, textures, and objects in images. The depthwise separable convolution generates the same result as the regular convolution but is more efficient due to fewer parameters. MobileNetV1 features 28 convolutional layers, with an output size of 7 × 7 × 1280 pixels, considering the depthwise and pointwise convolution of two different layers. This study applied the Rectified Linear Unit (ReLU) to provide computational efficiency and non-linearity to the MobUNet, which can be computed as $f(x) = \max(0, x)$.

Figure 4 presents MobileNetV2, which contains several convolution blocks with a skip layer connection. Skip connections offer additional paths for the gradient to propagate through a neural network. During the backpropagation process, these paths facilitate the transmission of gradients across the network, hence helping to adjust the weights associated with earlier layers.

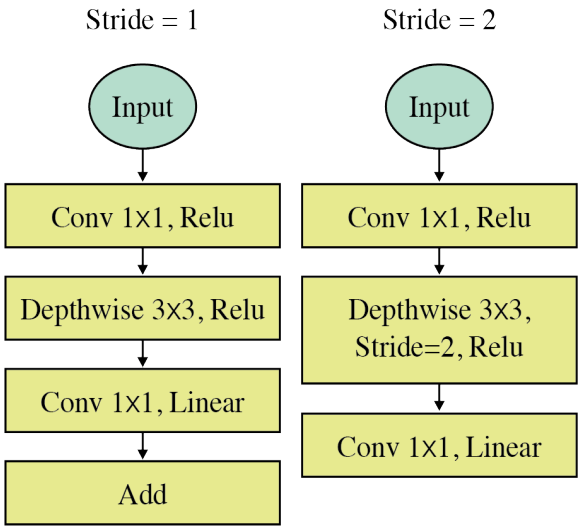

Fig. 4 *MobileNetV2 layers*

Figure 5 illustrates the standard convolution operation. Let *I* represent the input image with dimensions a × b, and let x denote the number of channels. For RGB images, x equals 3, whereas for grayscale images, x equals 1. Let y represent the number of filters with dimensions a_1 × b_1 According to [31], after performing the standard convolution operation, the image size will be a_2 × b_2, with a depth of y. The multiplications required for standard convolution can be represented as C_n.

In the context of depthwise separable convolution, the convolution operation is partitioned into two separate operations: depthwise convolutions and pointwise convolutions. Figure 6 illustrates the depthwise separable convolution operation. Each filter is applied to a different channel of the input image one at a time. It creates an intermediate image, that is a_2 × b_2 and has a x-depth. The multiplications required for depthwise convolution can be represented as C_d.

The pointwise convolution operation takes the intermediate image as its input. The convolution operation in pointwise convolution employs a filter size of 1 × 1. When a certain number of filters with dimensions 1 × 1 are employed, it produces an output image with dimensions a_2 × b_2× y. The number of multiplications required for pointwise convolution can be represented as C_p.

$$C_n = a_1 \times b_1 \times x \times a_2 \times b_2 \times y \tag{7}$$

$$C_d = a_1 \times b_1 \times x \times a_2 \times b_2 \tag{8}$$

$$C_p = a_2 \times b_2 \times x \times 1 \times 1 \times y \tag{9}$$

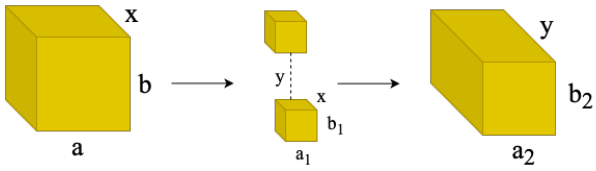

Fig. 5 *Standard convolution*

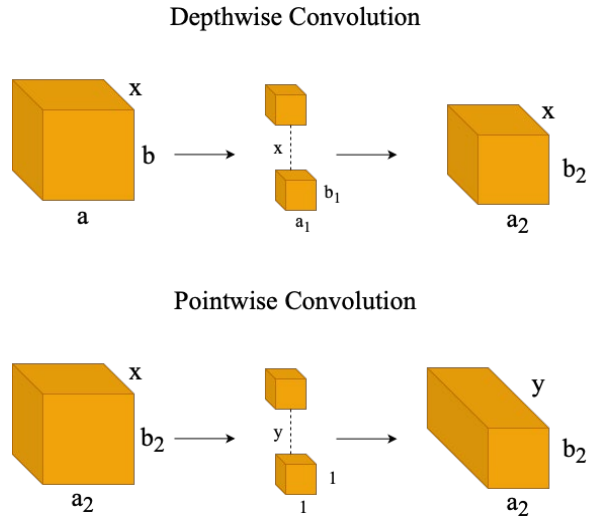

Fig. 6 *Depthwise and pointwise convolution*

The MobileNetV2 model belongs to the MobileNet family and is known for its improved speed compared to the MobileNetV1 model. MobileNetV2 has two extra functions than MobileNetV1. MobileNetV1 and MobileNetV2 allow 224 × 224 × 3 images. The input images in the dataset are thus resized and cropped to 224 × 224 pixels. After the initial convolution layer with 32 filters, MobileNetV2 adds nineteen inverted residual bottleneck layers, and the entire process is capped off with a pointwise convolution that yields a 7 × 7 × 1280 pixel output. Residual blocks use skip connections to send information to the network's deeper layer.

As a result, the beginning and ending layers of a standard residual block often have more channels than the middle. In contrast, the inverted residual block used in MobileNetV2 has much fewer parameters than the standard residual block since the connected layers have fewer channels than the middle layers.

4.3 Proposed Model: MobUNet

This study carries out cucumber leaf segmentation using the proposed method: MobUNet, in which MobileNetV2 was employed as an encoder or contracting path for the baseline of U-Net. Figure 7 presents the MobUNet architecture, which downsamples MobileNetV2 and then upsamples the U-Net. The upsampling path combines the feature map with the skip connection generated during the downsampling path. These skip connections upsample local information to global information. Based on the explanations of U-Net architecture in the previous section, U-Net consists of three parts: the contracting path (downsampling), the bottleneck, and the expanding path (upsampling).

Table 2 summarizes the MobUNet network layers, in which the input and output feature maps are 512 × 512 × 3. The first step, depthwise convolution, convolves each channel to generate an intermediate result separately. Hence, the pointwise is the depthwise convolution shape multiplied by the number of filters. Each feature vector is then mapped to the required number of classes in the final layer using a 1 × 1 convolution. MobileNetV2 is less complex to fine-tune because it has fewer parameters in the model. The model converges substantially more quickly when a pre-trained encoder is used instead of a non-pre-trained model. A pre-trained encoder achieves higher performance than a model without a pre-trained encoder.

Traditionally, U-Net models have utilized heavy backbones such as VGG16 or ResNet, which, while powerful, require substantial computational resources and are less practical for deployment in environments with limited processing capabilities. MobUNet stands out by incorporating MobileNetV2, a lightweight network designed for efficiency. MobileNetV2 uses depthwise separable convolutions, which split the standard convolution into two distinct operations: depthwise convolution, applying a single filter per input channel, and pointwise convolution,

which combines these channels. This approach significantly reduces the number of parameters and the overall computational load, making the model much more efficient.

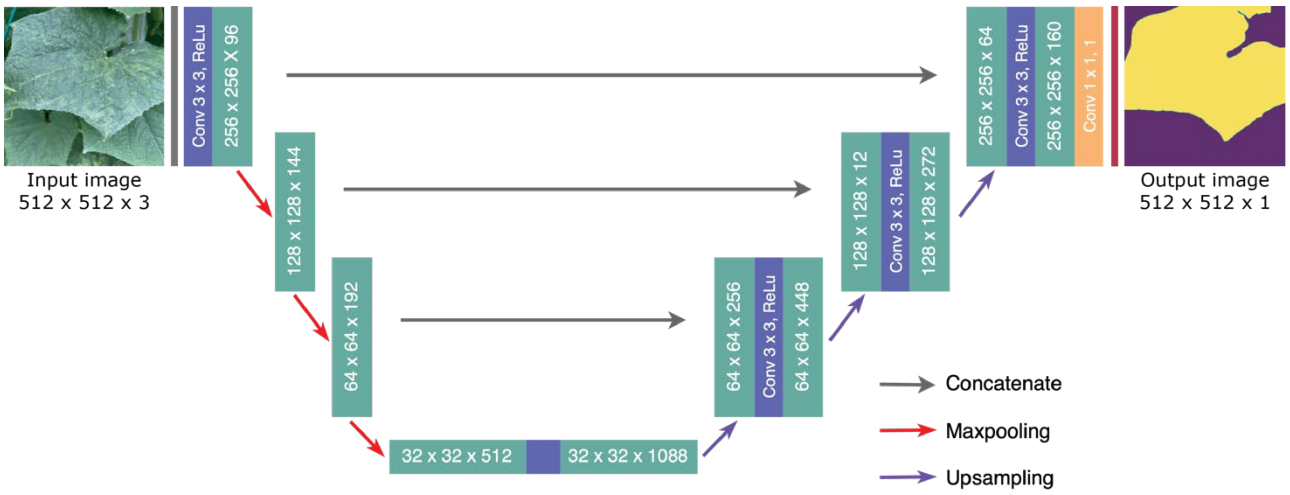

Fig. 7 *Proposed MobUNet architecture for cucumber leaf segmentation*

Table 2 *MobUNet network layers*

Layer (Type)	Output Shape	Parameter	Connected to
input (InputLayer)	512 × 512 × 3	0	[]
model (Functional)	256 × 256 × 96 128 × 128 × 144 64 × 64 × 192 32 × 32 × 576 16 × 16 × 320	1,841,984	['input[0][0]']
sequential (Sequential)	32 × 32 × 512	1,476,608	['model[0][4]']
concatenate (Concatenate)	32 × 32 × 1088	0	['sequential[0][0]', 'model[0][3]']
sequential_1 (Sequential)	64 × 64 × 256	2,507,776	['concatenate[0][0]']
concatenate_1 (Concatenate)	64 × 64 × 448	0	['sequential_[0][0]', 'model[0][2]']
Sequential_2 (Sequential)&	128 × 128 × 12	516,608	['concatenate_1[0][0]']
concatenate_2 (Concatenate)&	128 × 128 × 272	0	['sequential_2[0][0]', 'model[0][1]']
sequential_3 (Sequential)	256 × 256 × 64	156,928	['concatenate_2[0][0]']
concatenate_3 (Concatenate)&	256 × 256 × 160	0	['sequential_3[0][0]', 'model[0][0]']
conv2d_transpose_4 (Conv2DTranspose)	512 × 512 × 1	1441	['concatenate_3[0][0]']

The integration of MobileNetV2 in MobUNet allows the model to maintain the high segmentation accuracy characteristic of U-Net while being optimized for low-power, real-time applications. This is a clear departure from previous works that utilized U-Net with more computationally intensive backbones, as MobUNet achieves similar, if not better, segmentation performance with a fraction of the computational resources. This makes MobUNet particularly suitable for deployment on mobile devices or in resource-constrained environments, where traditional U-Net models would struggle to operate efficiently.

5. Results and Discussion

Table 3 summarizes the simulation parameters used in this study to train MobUNet and other deep learning techniques. All experiments were performed on Colab Pro with NVIDIA T4 Tensor Core GPUs. Python was the primary programming language for training the models, with PyTorch and Keras serving as the library packages. The cucumber leaf dataset has 145 raw and 145 ground truth images stored in Google Drive. This study considers cucumber leaf segmentation using MobUNet to find the segmentation performance based on several metrics: Dice

coefficient score, IoU, Hausdorff distance, Dice loss, and Jaccard distance. The cucumber leaf images use 80% of the training and 20% of the testing sets. The MobUNet analyzed the performance based on 60 epochs, considering 30 initial and 30 fine-tuned epochs.

Table 3 *Summary of the simulation parameter*

Parameter	Description
Dataset	Cucumber Leaf: 145 images
Development Platform	Google Colaboratory Pro
GPU	NVIDIA T4 Tensor Core
Programming Language	Python (PyTorch, Keras)
Segmentation Metrics	Accuracy, Dice coefficient, IoU, Dice loss, Jaccard distance, Hausdorff distance
Dataset Ratio	Training: 80 Testing: 20
Epochs	60 Initial: 30 Fine-Tuned: 30

5.1 Training and Validation

Figure 8 illustrates the training process of MobUNet: training loss and validation loss, as well as training accuracy and validation accuracy. The difference expanded rapidly as the training progressed. However, the validation accuracy remained constant, and there was no further decrease in validation loss. The MobUNet network is pointed to overfitting and insufficient training samples. Therefore, it is necessary to perform data augmentation to increase the number of cucumber leaf images.

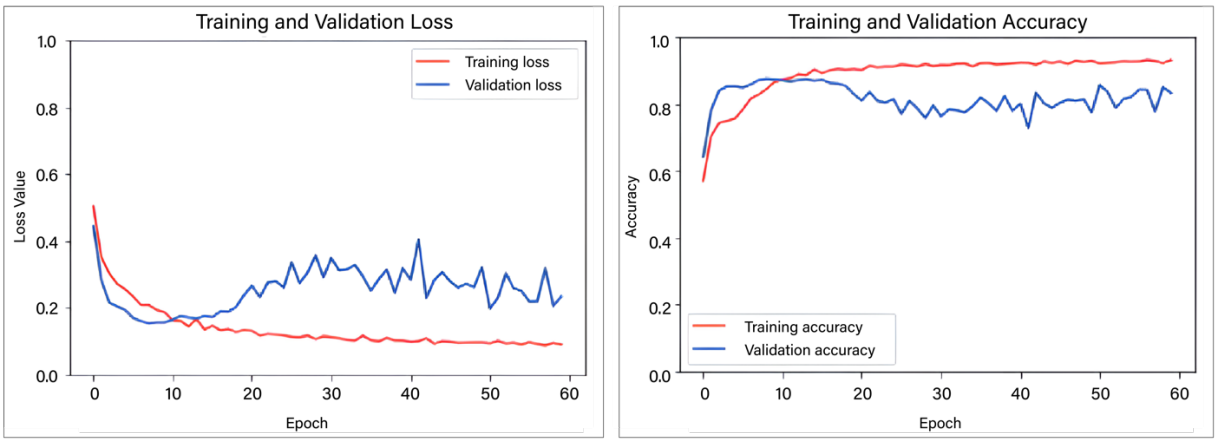

Fig. 8 *The loss and accuracy during the training process*

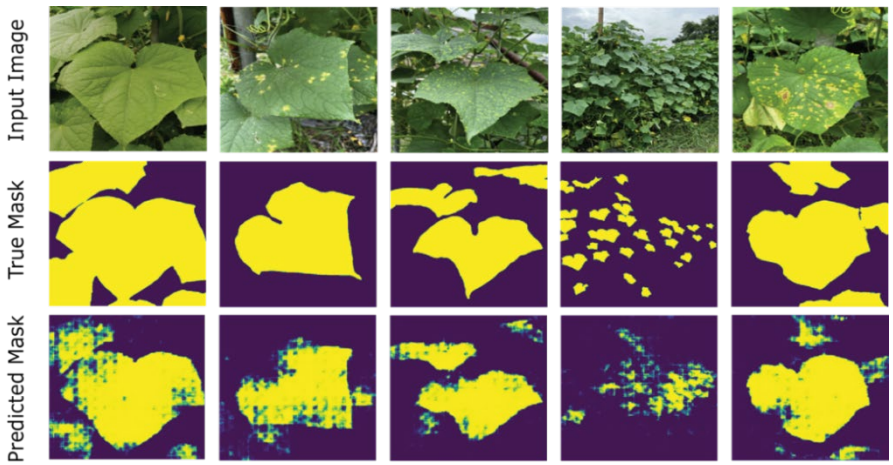

Fig. 9 *Segmentation results: input image, true mask, and predicted mask*

The input image results in Figure 9 showed the input image, the true mask based on the ground truth images created using RoboFlow, and the predicted mask of MobUNet. Leaf images have a complex background, and many researchers adopted Mask RCNN [32] to address the challenge of identifying the leaf shapes, textures, colours, and sizes. The MobUNet can identify the cucumber leaf images; however, some unnecessary masks were identified in the predicted images due to the complicated backgrounds of the cucumber leaf image. This challenge degraded the segmentation performance, yet it can be improved during the pre-processing by exploiting threshold and edge detection.

5.2 Comparison Based on the Metrics

This study analyzed several methods, including U-Net baseline, MobileNetV1, and MobileNetV2, to compare with MobUNet, as presented in Table 4. The objective was to develop a benchmark by evaluating the accuracy, Dice score, and IoU performance. Despite the limited number of cucumber leaf images in the dataset, the findings indicated that MobUNet outperformed other methods in accuracy, achieving a rate of 93.23%. Moreover, MobUNet showed a Dice score of 91.30% and an IoU of 85.03%. On the other hand, the U-Net baseline model yielded the lowest accuracy results, with a value of 51.02%. Similarly, the Dice score and IoU metrics were relatively low, measuring 66.11% and 50.62%, respectively.

Table 4 *Performance comparison of U-Net baseline, MobileNetV2, and MobUNet on accuracy, Dice, and IoU*

Methods	Accuracy (%)	Dice (%)	IoU (%)
U-Net baseline	51.02	66.11	50.62
MobileNetV1	75.74	74.47	61.55
MobileNetV2	81.75	79.16	68.16
MobUNet	93.23	91.30	85.03

The comparative analysis conducted in this study demonstrates that MobileNetV2 outperformed MobileNetV1 in several aspects, as evidenced by previous research [33], [34]. Specifically, MobileNetV2 indicates significant improvements over its predecessor in the context of cucumber leaf segmentation. The MobileNetV1 model did not achieve an accuracy rate of over 80%, and its Intersection over Union (IoU) score is only 61.55%. In comparison, MobileNetV2 demonstrates a 6.61% improvement in performance.

The accuracy of MobileNetV2 was seen to increase by 30% to reach a value of 81.75%. However, it should be noted that this accuracy rate falls below that achieved by MobUNet, which surpasses 90%. The findings of this study indicate that adopting MobileNetV2 as an encoder in the U-Net architecture can improve segmentation performance. However, data augmentation and pre-processing methods are needed to improve the segmentation performance. Furthermore, the complexity of leaf images made the segmentation process challenging. Thus, the encoder of MobUNet needs to be simplified to fit the features of the segmentation problem better.

Table 5 presents the segmentation quality results based on Dice loss, Jaccard distance, and Hausdorff distance for U-Net baseline, MobileNetV1, MobileNetV2, and MobUNet. The Dice loss seen in MobUNet indicated a moderate amount of data loss at 0.0870 due to the imperfect overlap between the two sets of cucumber leaf images, which prevented the achievement value of zero. However, the data loss of MobUNet is considerably better than U-Net baseline (0.3398), MobileNetV1 (0.2553), and MobileNetV2 (0.2084) because MobUNet combines the advantageous features of U-Net, which excels in collecting fine details and contextual data, resulting in improved segmentation quality.

Table 5 *Dice loss, Jaccard distance, and Hausdorff distance of U-Net baseline, MobileNetV2 and MobUNet*

Methods	Dice Loss	Jaccard Distance	Hausdorff Distance
U-Net baseline	0.3398	50.1248	0.5162
MobileNetV1	0.2553	39.3989	0.2650
MobileNetV2	0.2084	33.2269	0.1970
MobUNet	0.0870	15.5950	0.0001

In contrast, the Jaccard distance yields a dissimilarity value of 15.5950, which indicates a significant deviation from perfect overlap, as the ratio is far from zero. The ratio should closely approximate zero to achieve a near-perfect overlapping of two sets of cucumber leaf images. Nevertheless, when MobileNetV2 is not utilized as an encoder for the U-Net model, the Jaccard distance of the U-Net baseline exhibits a much higher dissimilarity value of 50.1248 in comparison to MobileNetV1 (39.3989), MobileNetV2 (33.2269), and MobUNet. Integrating the MobUNet model resulted in a considerable improvement in the values.

Hausdorff distance is mainly used for medical image segmentation. However, Hausdorff distance was employed in this study to find the nearest point in the other cucumber leaf images. Based on the findings, it can be observed that MobUNet exhibited the closest Hausdorff distance value to the origin point, measuring at 0.0001. In contrast, the U-Net baseline recorded a Hausdorff distance of 0.5162, MobileNetV1 reached 0.2650, and MobileNetV2 achieved 0.1970, indicating that the models were comparatively farther from the origin point. The findings indicate that MobUNet significantly enhanced its structural aspects, demonstrating high quality and accuracy in the segmentation. Achieving a low Hausdorff distance indicates that the segmented regions closely match the reference or ground truth data.

5.3 Comparison with the State-of-the-art Methods

The MobUNet was compared with the state-of-the-art methods in Section 2. In addition, some state-of-the-art methods were performed for leaf disease segmentation and classification. Fig. 11 shows the performance of MobUNet with the state-of-the-art methods, considering the accuracy, IoU, and Dice scores. Based on the results, Modified U-Net: VGG [14] showed excellent accuracy at 98.24% compared to other methods, 93.71% for LWMSDU-Net [19], 93.27% for DeepLabV3 and U-Net [15], including MobUNet at 93.23%, respectively. Although the MobUNet has the lowest accuracy compared to other methods, the total training time for 60 epochs is only 13 minutes compared to Modified U-Net, which took an hour for 40 epochs, LWMSDU-Net took 5.17 hours, and DeepLabV3 and U-Net took 7.14 hours for 300 epochs.

The improvement in MobUNet, which combines U-Net with MobileNetV2, is not only due to adding more CNN layers. However, the improvement is mainly influenced by complementary integration and optimization methodologies specifically designed for this architecture. The integration between U-Net and MobileNetV2 offers the practical spatial context preservation of U-Net with the efficient feature extraction of MobileNetV2. For applications such as image segmentation, it enables easy implementation of robust feature fusion techniques at various sizes. Furthermore, utilizing specific optimization techniques and customized training methodologies improves the convergence and generalization of the model. Numerical findings support these assertions by showing that MobUNet outperforms the separate models in segmentation tasks by expertly combining the advantages of both architectures. Future research might focus on refining these integration systems and investigating more architectural modifications to continue improving the performance capabilities of MobUNet.

Modified U-Net employed the VGG network as the backbone architecture for weed images. The upsampling and combining of the feature layers from the backbone allowed U-Net to recover the details of the missing results. However, U-Net segmentation durations are longer than other networks due to the complexity of the underlying architecture. Therefore, the duration of the segmentation process was prolonged with time. However, the Modified U-Net: VGG showed a slightly lower IoU score at 92.91% compared to KijaniNet [21] at 97.66%.

MobUNet produced a poor IoU at 85.03% compared to other segmentation methods due to a lack of training and testing data. The performance of MobUNet can increase if data augmentation is implemented in the dataset. The Deep leaf using Mask RCNN [8] showed 90.5%, the second lowest IoU due to a similar limitation with this study: a limited number of leaf images. Despite the low result of IoU, the MobUNet showed a better performance using Dice score compared to DeepLabV3 and U-Net [15], CNN+Watershed [16], Eff-Unet++ [5], and LeafMask [20]. The Dice scores for the state-of-the-art methods were 69.14%, 76.08%, 78.27%, and 90.09%, respectively.

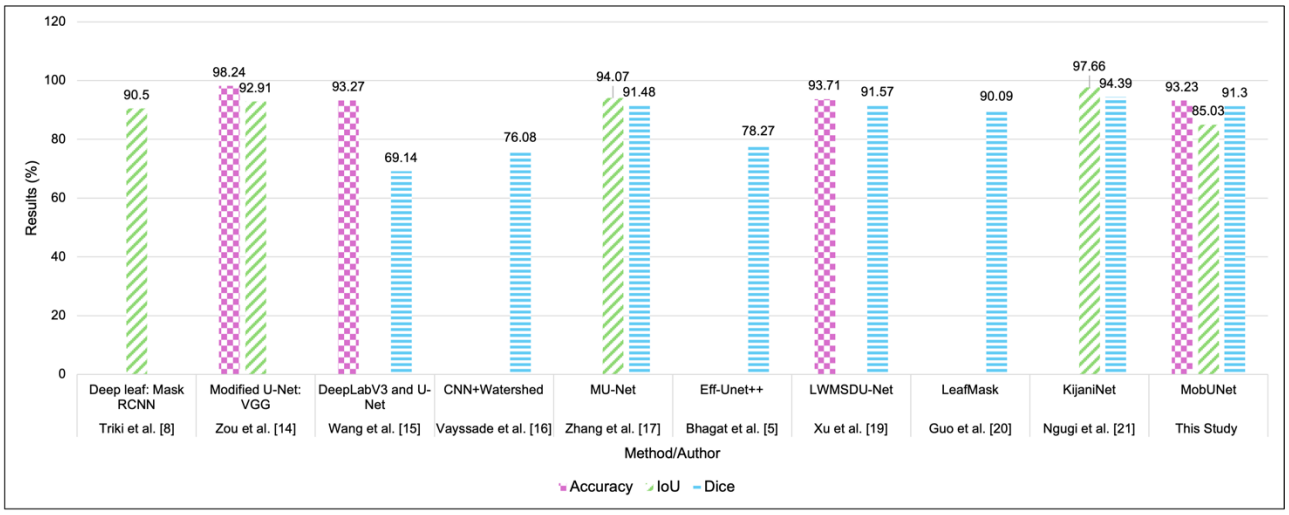

Fig. 10 *Segmentation results with state-of-the-art methods*

6. Conclusion

This study proposes a cucumber leaf segmentation based on U-Net by employing the MobileNetV2 structure as an encoder. An essential preliminary step is image annotation to achieve better performance. Image annotation was conducted before applying the MobUNet segmentation method to cucumber leaf images. Based on the findings, MobUNet achieved better accuracy, Dice score, and IoU compared to the other segmentation methods in the literature and other models conducted in this study: U-Net baseline, MobileNetV1, and MobileNetV2. In addition, the Hausdorff distance yielded the highest segmentation quality since it showed the closest value to the zero point.

From an agricultural point of view, integrating cucumber leaf segmentation with monitoring systems can work as an early warning system for undesirable conditions, such as drought stress or nutrient deficiencies. Damage to crops can be reduced if farmers receive early warnings and can take preventative actions. MobUNet can also be utilized to detect diseases in plants. The precise segmentation of cucumber leaves enables farmers to promptly detect diseases during the early stage, such as discolouration, lesions, or wilting. This early detection allows for specific treatment, preventing diseases from spreading to an entire crop and decreasing crop losses.

In future work, the MobUNet model can capture leaf images using drones to identify the leaf shapes, size, diseases, and leaf count. In addition, data augmentation and pre-processing methods could be studied to address the problems on the limited dataset and enhance the segmentation accuracy. The amount of training data can be expanded to include a more significant number of images and enhance the overall segmentation performance. Hence, it is also accessible for other computer vision applications.

Acknowledgement

This work was supported by the Ministry of Higher Education Malaysia under Fundamental Research Grant Scheme (FRGS/1/2021/ICT09/UTM/02/1) and the output of ASEAN IVO project (http://www.nict.go.jp/en/asean_ivo/index.html) entitled "Agricultural IoT based on Edge Computing" (S.K130000.7656.4X796).

Conflict of Interest

The authors declare that there is no conflict of interest regarding the paper's publication.

Author Contribution

The authors confirm their contribution to the paper as follows: ***study conception and design:*** *Nurul Amirah Mashudi, Norulhusna Ahmad, Suriani Mohd Sam;* ***data collection:*** *Norulhusna Ahmad, Norliza Mohd Noor, Hazilah Mad Kaidi;* ***analysis and interpretation of results:*** *Nurul Amirah Mashudi, Norulhusna Ahmad, Norliza Mohd Noor;* ***draft manuscript preparation:*** *Nurul Amirah Mashudi, Norulhusna Ahmad, Norliza Mohamed. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Lu, J., Tan, L.,and Jiang, H. (2021). Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification, *Agriculture*, *vol. 11, p. 707*.

[2] Li, S., Xiong, L., Tang, G., & Strobl, J. (2020). Deep Learning-based Approach for Landform Classification from Integrated Data Sources of Digital Elevation Model and Imagery, *Geomorphology*, *vol. 354, p. 107045*.

[3] Shumack, S., Hesse, P., & Farebrother, W. (2020). Deep Learning for Dune Pattern Mapping with the AW3D30 Global Surface Model, *Earth Surface Processes and Landforms, vol. 45, pp. 2417–2431*.

[4] Li, S., Hu, G., Cheng, X., Xiong, L., Tang, G., & Strobl, J. (2022). Integrating Topographic Knowledge into Deep Learning for the Void-Filling of Digital Elevation Models, *Remote Sensing of Environment*, *vol. 269, p. 112818*.

[5] Bhagat, S., Kokare, M., Haswani, V., Hambarde, P., & Kamble, R. (2022). Eff-UNet++: A Novel Architecture for Plant Leaf Segmentation and Counting, *Ecological Informatics, vol. 68, p. 101583*.

[6] Ward, D., Moghadam, P., & Hudson, N. (2018). Deep Leaf Segmentation Using Synthetic Data, *British Machine Vision Conference (BMVC) 2018 Proceedings*, *pp. 1–13*.

[7] Dami´an, X., Ochoa-L´opez, S., Gaxiola, A., Fornoni, J., Dom´ınguez, C. A., & Boege, K. (2020). Natural Selection Acting on Integrated Phenotypes: Covariance Among Functional Leaf Traits Increases Plant Fitness, *New Phytologist, vol. 225, pp. 546–557*.

[8] Triki, A., Bouaziz, B., Gaikwad, J., & Mahdi, W. (2021). Deep leaf: Mask R-CNN based Leaf Detection and Segmentation from Digitized Herbarium Specimen Images, *Pattern Recognition Letters, vol. 150, pp. 76–83*.

[9] Maloof, J. N., Nozue, K., Mumbach, M. R., & Palmer, C. M. (2013). LeafJ: An ImageJ Plugin for Semi-Automated Leaf Shape Measurement, *Journal of Visualized Experiments, p. e50028*.

[10] Easlon, H. M., & Bloom, A. J. (2014). Easy Leaf Area: Automated Digital Image Analysis for Rapid and Accurate Measurement of Leaf Area, *Applications in Plant Sciences, vol. 2, p. 1400033*.

[11] Akhtar, M. S., Zafar, Z., Nawaz, R., & Fraz, M. M. (2024). Unlocking plant secrets: A systematic review of 3D imaging in plant phenotyping techniques. *Computers and Electronics in Agriculture, 222, 109033*.
[12] Sinha, A., & Shekhawat, R. S. (2020). Review of Image Processing Approaches for Detecting Plant Diseases, *IET Image Processing, vol. 14, no. 8, pp. 1427–1439*.
[13] Singh, V., & Misra, A. K. (2017). Detection of Plant Leaf Diseases Using Image Segmentation and Soft Computing Techniques, *Information processing in Agriculture, vol. 4, no. 1, pp. 41–49*.
[14] Zou, K., Chen, X., Wang, Y., Zhang, C., & Zhang, F. (2021). A Modified U-Net with a Specific Data Argumentation Method for Semantic Segmentation of Weed Images in the Field, *Computers and Electronics in Agriculture, vol. 187, p. 106242*.
[15] Wang, C., Du, P., Wu, H., Li, J., Zhao, C., & Zhu, H. (2021). A Cucumber Leaf Disease Severity Classification Method Based on the Fusion of DeepLabV3+ and U-Net, *Computers and Electronics in Agriculture, vol. 189, p. 106373*.
[16] Vayssade, J.-A., Jones, G., G´ee, C., & Paoli, J.-N. (2022). Pixelwise Instance Segmentation of Leaves in Dense Foliage, *Computers and Electronics in Agriculture, vol. 195, p. 106797*.
[17] Zhang, S., & Zhang, C. (2023). Modified U-Net for Plant Diseased Leaf Image Segmentation, *Computers and Electronics in Agriculture, vol. 204, p. 107511*.
[18] Yang, K., Zhong, W., & Li, F. (2020). Leaf Segmentation and Classification with a Complicated Background Using Deep Learning, *Agronomy, vol. 10, p. 1721*.
[19] Xu, C., Yu, C., & Zhang, S. (2022). Lightweight Multi-Scale Dilated U-Net for Crop Disease Leaf Image Segmentation, *Electronics, vol. 11, p. 3947*.
[20] Guo, R., Qu, L., Niu, D., Li, Z., & Yue, J. (2021). LeafMask: Towards Greater Accuracy on Leaf Segmentation, *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 1249– 1258*.
[21] Ngugi, L. C., Abdelwahab, M., & Abo-Zahhad, M. (2020). Tomato Leaf Segmentation Algorithms for Mobile Phone Applications Using Deep Learning, *Computers and Electronics in Agriculture, vol. 178, p. 105788*.
[22] Nikbakhsh, N., Baleghi, Y., & Agahi, H. (2021). A Novel Approach for Unsupervised Image Segmentation Fusion of Plant Leaves based on G-Mutual Information, *Machine Vision and Applications, vol. 32, p. 5*.
[23] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference. Springer, pp. 234–241*.
[24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520*.
[25] Alexandrova, S., Tatlock, Z., & Cakmak, M. (2015). RoboFlow: A Flow-based Visual Programming Language for Mobile Manipulation Tasks, *2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 5537–5544*.
[26] Hantos, G. B., Simon, G., Hejda, M., Bernassau, A. L., & Desmulliez, M. P. Y. (2021). Automated Particle and Cell Phenotyping Using Object Recognition and Tracking Based on Machine Learning Algorithms, *2021 IEEE International Ultrasonics Symposium (IUS), pp. 1–4*.
[27] Lin, Q., Ye, G., Wang, J., & Liu, H. (2022). RoboFlow: A Data-Centric Workflow Management System for Developing AI-Enhanced Robots, *Proceedings of the 5th Conference on Robot Learning, vol. 164, pp. 1789–1794*.
[28] Bisong, E. (2019). Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners. *Apress*.
[29] Huttenlocher, D., Klanderman, G., & Rucklidge, W. (1993). Comparing Images Using the Hausdorff Distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 850–863*.
[30] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv preprint. arXiv:1704.04861.
[31] Wang, G., Yuan, G., Li, T., & Lv, M. (2018). An Multi-scale Learning Network with Depthwise Separable Convolutions, *IPSJ Transactions on Computer Vision and Applications, vol. 10, no. 1, pp. 1–8*.
[32] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, pp. 386–397*.
[33] Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A Real Time DNN-based Face Mask Detection System Using Single Shot Multibox Detector and MobileNetV2, *Sustainable cities and society, vol. 66, p. 102692*.
[34] Sutaji, D., & Yıldız, O. (2022). LEMOXINET: Lite Ensemble MobileNetV2 and Xception Models to Predict Plant Disease, *Ecological Informatics, vol. 70, p. 101698*.