Enhancing Leaf Area Segmentation by Using Attention Gates and Knowledge Distillation in UNet Architecture

A. Shamim Banu[1,2] and S. Deivalakshmi[1]

[1]*National Institute of Technology, Tiruchirappalli, Tamilnadu, India,*
[2]*Government Polytechnic College, Tiruchirappalli, Tamilnadu, India*

https://doi.org/10.26636/jtit.2025.3.2079

Abstract — Accurate segmentation of leaf regions plays a vital role in plant phenotyping and agricultural analysis. This paper presents AKDUNet, a lightweight UNet-based architecture that integrates attention gates and knowledge distillation to improve segmentation performance while minimizing computational complexity. The architecture replaces traditional skip connections with attention gates to focus on salient spatial features and employs a two-stage training pipeline, where a compact student model learns from a deeper teacher model using a tailored distillation loss function. AKDUNet is evaluated on two benchmark datasets (CWFID and Sunflower) and outperforms a range of state-of-the-art models, including UNet++, Inception UNet, VGG-based UNets, SDUNet, INSCA UNet, and SegFormer. Ablation studies confirm the advantages of attention modules, and qualitative analyses using Grad-CAM visualizations reveal the model's ability to effectively focus on crucial leaf structures. The results demonstrate that AKDUNet is not only computationally efficient but also highly accurate, making it suitable for real-time deployment in resource-constrained agricultural environments.

Keywords — attention gate, knowledge distillation, modified light weight UNet, semantic segmentation

1. Introduction

Plant phenotyping has recently gained more attention from researchers due to its potential to enhance high-yield plant capabilities and augment food security. It is a key tool for understanding plant genetics, plant-environment interactions, and various traits [1]. It also involves creating new technologies to improve plant yields and address the aforementioned issues. Furthermore, plant phenotyping is crucial for examining plant growth, yield, and internal structure.

Plant image analysis is a key technique for plant phenotyping, facilitating the evaluation of plant traits, growth forecasts, and spatial details of plants. Manual measurements of visual characteristics are costly. Hence, there is a need for automated solutions. Recent studies suggest that deep learning (DL) techniques, a contemporary AI approach, are becoming increasingly important in plant phenotyping due to their advanced features [2].

In plant phenotyping, leaf area segmentation is crucial for analyzing plant growth. However, the task becomes challenging when dealing with small leaves or when many leaves overlap. Furthermore, the effectiveness of leaf area segmentation can be significantly influenced by factors such as image capture angles and lighting conditions [3].

The proposed work introduces an AKDUNet model, designed to enhance leaf area segmentation. It boosts segmentation accuracy while keeping the model simple, by relying on a smaller number of parameters. It is based on the UNet architecture, known for its effective leaf segmentation, and is compared with other known segmentation models, in example UNet++, Inception UNet, SDUNet, and INCSA UNet, as well as pre-trained deep learning models such as VGG16-UNet, VGG19-UNet, ResNet-UNet, and SegFormer. The performance of the AKDUNet model is assessed using the crop weed field image dataset (CWFID) and the Sunflower data set for plant phenotyping. Both qualitative and quantitative analyses demonstrate that AKDUNet outperforms current leaf segmentation techniques.

The key contributions of this work are as follows.

- **Knowledge distillation.** This method trains a smaller student model P^s to replicate the performance of a larger teacher model P^t. By transferring the core knowledge from the teacher to the student, this approach enables the smaller model to achieve similar performance with fewer parameters, thus reducing the computational burden and memory requirements.
- **Attention gate mechanism.** Traditional skip connections are replaced with an attention gate mechanism. This technique allows the model to focus on the most relevant parts of the input data and ignore less important information. By emphasizing significant features and filtering out irrelevant details, attention gates enhance the model's efficiency and accuracy without significantly increasing computational demands.
- **Performance evaluation.** The effectiveness of the AKDUNet model is evaluated in comparison to advanced models using metrics such as IoU score, F1 score, and Dice coefficient loss, providing a thorough evaluation of its performance.
- **Assessment of computational complexity.** This research evaluates the computational efficiency of the proposed ap-

proach by analyzing floating-point operations per second (FLOPs), the number of model parameters, and the inference time, and thus providing a comprehensive assessment of both performance and resource utilization.

The structure of the remaining sections is as follows. Section 2 reviews related work on semantic segmentation, attention modules, and knowledge distillation learning. Section 3 provides a detailed description of the key components of the AKDUNet model, including its architecture, implementation, loss functions, and training algorithm. Section 4 presents an in-depth analysis of the experimental setup and results. Section 5 discusses the major advancements and limitations of the approach and provides the conclusion.

2. Related Works

The field of knowledge distillation continues to evolve with several advances that enhance the efficiency and effectiveness of computer vision models. Various techniques such as channel attention, transformers, self-attention, and novel feature distillation modules mature to optimize performance by reducing model size and complexity. Each of these contributions reflects ongoing efforts to balance the trade-off between model performance and computational resources, which is crucial for deploying advanced computer vision technologies in mobile and embedded systems.

The authors of [4] introduced a method that uses channel attention and feature maps within a transformer framework to facilitate knowledge transfer between a large teacher model P^t and a smaller student model P^s. Paper [5] pioneered the application of KD frameworks, specifically in the field of salient object detection (SOD), which involves identifying the most important objects in an image. In [6], a method for efficient semantic segmentation was proposed that combines self-attention mechanisms with self-distillation. The authors of [7] developed a common feature distillation module designed to consolidate multi-stream information into a spatially coherent single-stream representation.

Semantic segmentation involves assigning each pixel in an image to a specific category. Modern RGB-D semantic segmentation methods often rely on deep learning techniques which have significantly advanced the field due to their strong automatic learning and feature extraction capabilities. In such a context, the authors of [8] improved deep learning models by incorporating channel attention mechanisms to enhance features, while in [9], self-attention modules were introduced to refine the features extracted by the encoder.

In [10], an RFNet was developed which balances performance and speed for real-time applications. [11] proposed an ESANet which dynamically adjusts feature space representations by weighting the outputs of each encoding block. In [12], high-level features were applied to dynamically adjust the decoding structure of deep learning networks.

Several adaptations of the traditional UNet architecture have been developed to improve segmentation tasks. The authors of [13] presented an improved UNet++ design which adds deconvolution blocks to the skip connections. This modification enriches the semantic information in the decoder, enabling deeper supervision. In [14], the UNet architecture was introduced, incorporating inception layers and combining binary cross-entropy, the Dice coefficient, and intersection over union to boost performance.

The authors of [15] developed a SDUNet which features structured dropout in all UNet layers. This approach helps prevent overfitting by eliminating some semantic details from the network. [16] explored the INCSA UNet architecture which integrates inception blocks with spatial attention mechanisms. This model uses parallel and sequential layers to effectively extract key features.

In [17], a compressed version of a UNet customized for plant disease segmentation was presented. This streamlined model is more storage-efficient and performs faster than the original UNet. Paper [18] proposed SegFormer, a transformer-based semantic segmentation model that combines hierarchical encoding with a lightweight MLP decoder. The model eliminates the need for positional encoding, improving robustness to varying input resolutions.

Traditional semantic segmentation methods frequently face challenges that can result in inaccurate predictions. To address these issues, this paper proposes an AKDUNet model that integrates attention mechanisms and knowledge distillation with the UNet architecture. This approach effectively reduces the number of parameters compared to that used by conventional UNet models, thereby reducing computational complexity. Despite the reduction in the number of parameters, the AKDUNet model aims to enhance performance by utilizing attention mechanisms to focus on important features and employing knowledge distillation to preserve essential knowledge from more complex models. This customized solution specifically addresses the limitations of existing methods.

3. Proposed Methodology

This section presents a detailed overview of the AKDUNet model, including its architecture, implementation, loss functions, and training algorithm for knowledge distillation.

3.1. Implementation of Framework

Knowledge distillation is a deep learning technique that enables the transfer of knowledge from a large to a smaller model. It is particularly useful for deploying models in resource-constrained environments, where computational power and memory are limited. To enhance segmentation accuracy, the proposed method transfers knowledge from a larger teacher model P^t to a smaller student model P^s. Figure 1 illustrates the model's architecture, which includes two deep convolutional networks such as the student model P^s with weights θ_s and the teacher model P^t with weights θ_t.

The work uses a UNet backbone [19] for both the teacher model P^t and the student model P^s, which helps to effectively capture and maintain detailed features. The network configurations are as follows: teacher model P^t (depth, width);

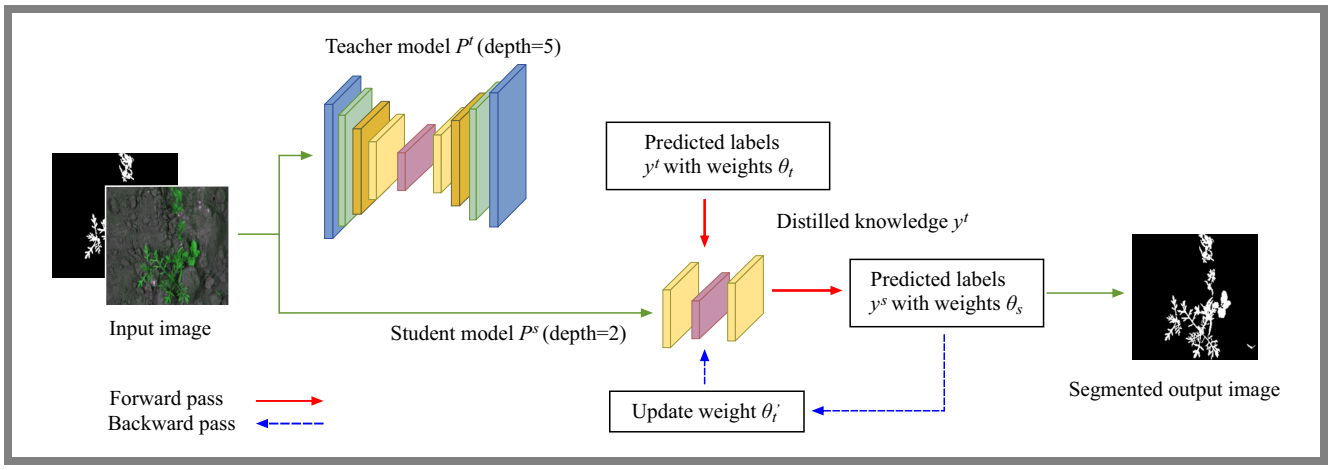

Fig. 1. Architecture of the AKDUNet model for leaf segmentation.

$n \in \{5, 3\}$ and student UNet P^s (depth, width); $n \in \{2, 3\}$, where depth denotes the number of layers and width denotes the number of convolutions in each layer which are depicted in Fig. 1. These variations are intended to explore the trade-offs between detailed feature extraction, the number of parameters, and overall model complexity.

Table 1 presents a detailed architectural comparison between the teacher UNet (depth 5) and the student UNet (depth 2). The teacher model follows a deeper architecture with more convolutional blocks and a higher parameter counts of 2 354 785 to ensure rich feature extraction and accurate segmentation.

On the contrary, the student model is a lightweight version with significantly fewer layers and 13 681 parameters, designed for faster inference and deployment in resource-constrained environments. Despite its simplicity, the Student UNet retains the core structural elements of UNet, including convolution, pooling, up-sampling, and skip connections, allowing it to perform efficient segmentation with reduced computational cost.

The number of parameters in a 2D convolutional layer is given by the formula:

$$CONV_{Par} = (K_H \times K_W \times C_{IN} + 1) \times C_{OUT} \,, \qquad (1)$$

where: K_H – kernel height, K_W – kernel width, C_{IN} – number of input channels, C_{OUT} – number of output filters.

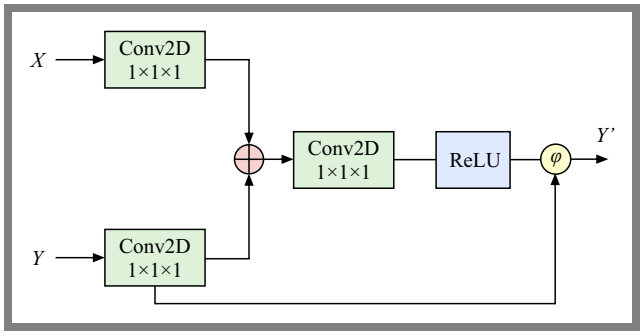

Fig. 2. Attention gate module.

Instead of directly passing features from the encoder to the decoder through conventional skip connections, the proposed method incorporates attention gates within the skip connections to enhance feature selection, as illustrated in Fig. 2. In this design, the encoder feature map is used as one input X to the attention gate, while the gating signal from the decoder's previous stage is used as the second input Y. The attention gate computes attention coefficients that selectively highlight relevant regions in the feature maps while suppressing less informative activations. This is mathematically represented as:

$$Y' = ReLU(Y) \odot Y \,, \qquad (2)$$

where: $\odot$ – element-wise multiplication and Y' – output after applying the attention.

The multiplication with Y ensures that only the most significant features of the decoder contribute to the subsequent layers. By adaptively focusing on important spatial regions, these attention gates help improve segmentation accuracy by refining the information passed from the encoder to the decoder. The training process is carried out in two stages. In the first stage, teacher model P^t is designed with an encoder decoder architecture of depth 5. The model is trained using the Dice coefficient loss function, with the objective of optimizing its performance based on F1 and IoU scores. Once training has been completed, the parameters of the teacher model, including weights θ_t, are frozen to ensure that they remain fixed during the subsequent distillation phase. The trained teacher model P^t generates fixed predictions y^t, which will be used as ground truth in the second stage.

In the second stage, a smaller student model P^s with a depth of 2 is used. The student model has fewer parameters than the teacher model, allowing for a more compact architecture. During this stage, the outputs of teacher model P^t serve as ground truth y^t for calculating the knowledge distillation loss L_{KD}, which quantifies the discrepancy between the teacher's ground truth predictions y^t and the student model's predictions y^s. It is also advantageous to train the student model with ground truth labels y to get predictions y^s produced by the student model itself to calculate standard loss L_s. This loss is derived using the Dice coefficient loss, which quanti-

Tab. 1. Architecture and parameters of teacher and student UNet models.

Stage	Teacher UNet (depth 5)	Student UNet (depth 2)
Input	128 × 128 × 3	128 × 128 ×1
Block 1	3 × Conv2D (16 filters ReLU) MaxPool (2 × 2) Dropout (0.5)	3 × Conv2D (16 filters ReLU) MaxPool (2 × 2) Dropout (0.5)
Block 2	3 × Conv2D (32 filters ReLU) MaxPool (2 × 2) Dropout (0.5)	–
Block 3	3 × Conv2D (64 filters ReLU) MaxPool (2 × 2) Dropout (0.5)	–
Block 4	3 × Conv2D (128 filters ReLU) MaxPool (2 × 2) Dropout (0.5)	–
Bottleneck	2 × Conv2D (256 filters ReLU)	2 × Conv2D (32 filters ReLU)
Up block 4	UpSample (2 × 2) Conv2D (128) Concat 2 × Conv2D Dropout (0.5)	–
Up block 3	UpSample (2 × 2) Conv2D (64) Concat 2 × Conv2D Dropout (0.5)	–
Up block 2	UpSample (2 × 2) Conv2D (32) Concat 2 × Conv2D Dropout (0.5)	–
Up block 1	UpSample (2 × 2) Conv2D (16) Concat 2 × Conv2D Dropout (0.5)	UpSample (2 × 2) Conv2D (16) Concat 2 × Conv2D Dropout (0.5)
Output	3 × Conv2D (labels 1 × 1) Activation (Sigmoid/Softmax)	3 × Conv2D (labels 1 × 1) Activation (Sigmoid/Softmax)
Total parameters	2 354 785	13 681

fies the predictions of similarity between student model y^s and ground truth y.

This total distillation loss combines both standard loss L_s and knowledge distillation loss L_{KD}, ensuring that the student model learns to approximate both teacher predictions y^t and ground truth y effectively. The combination of these losses is used to update the weights of student model θ_s, resulting in an optimized student model with updated parameters θ'_s.

Loss functions play a vital role in image segmentation as they guide the model to segment images into distinct regions. The choice of loss function depends on the specific requirements of the segmentation task and the nature of the segmentation process. To evaluate the proposed knowledge distillation approach, three key loss functions such as knowledge distillation loss L_{KD}, student loss L_s, and overall distillation training loss L_T are considered.

3.2. Knowledge Distillation Loss

In knowledge distillation, knowledge is transferred from teacher model P^t to student model P^s by minimizing the discrepancy between the predictions of teacher labels $y^s \in R^{h\times w}$ and student labels $y^t \in R^{h\times w}$, where y^s and y^t represent predictions of student model P^s and teacher model P^t respectively, with h and w referring to the height and width of the input image. The equations used to derive knowledge distillation loss L_{KD} are as follows:

$$p^t = \text{softmax}\left(\frac{y^t}{\tau}\right), \tag{3}$$

$$p^s = \text{softmax}\left(\frac{y^s}{\tau}\right), \tag{4}$$

$$L_{KD} = \frac{2}{\tau}\left(\sum_{i=1}^{B}\sum_{j=1}^{C} p_{ij}^t \log\frac{p_{ij}^t}{p_{ij}^s}\right), \tag{5}$$

where B is the batch size, C is the number of classes, and τ is the temperature factor that typically ranges between 1 and 5 and controls the smoothness of the probability distributions p^t and p^s.

The purpose of Eqs. (3) and (4) is to convert the raw logits y^t and y^s from the teacher model and student model into probability distributions. The softmax function is applied to the logits to normalize them, transforming them into values between 0 and 1 that sum up to 1 across all classes. This ensures that both teacher model P^t and student model P^s outputs are in the form of probabilities, allowing for a meaningful comparison of their predictions.

The equation for L_{KD} shown in Eq. (5) quantifies how much the student model probability distribution differs from the teacher model P^t distribution. In Eq. (5), p^t and p^s are probabilistic prediction values of teacher model P^t and student model P^s, respectively, while $\log\frac{p_{ij}^t}{p_{ij}^s}$ is the Kullback-Leibler divergence which measures the difference between the teacher and the student probability distributions. The logarithm computes how far the student model P^s distribution is from the teacher model P^t distribution.

This loss function can also be interpreted as the cross-entropy between the distributions and encourage the student to match the teacher distribution. By doing this, student model P^s learns not only the prediction of the final class, but also the relative likelihoods of different classes, enabling it to better approximate the decision-making process.

3.3. Standard Loss

It is beneficial to train the student together with ground truth labels y and y^s labels predicted by student model P^s to obtain standard loss L_s. The Dice coefficient loss is utilized to get the standard loss which is:

$$L_S = 1 - 2\,\frac{|y \cap y^s|}{|y + y^s|}, \tag{6}$$

where $|y \cap y^s|$ is the number of pixels that are correctly predicted by student model P^s with respect to ground truth y, $|y + y^s|$ represent the total number of pixels in ground truth y and predicted labels y^s of student model P^s.

3.4. Total Distillation Loss

Equation (7) represents the overall training loss as a weighted combination of standard loss L_s and knowledge distillation loss L_{KD}. The equation is essential for achieving two key objectives during model training, i.e. minimizing L_s and reducing L_{KD}.

$$L_T = \lambda\, L_s + (1 - \lambda)\, L_{KD}\ . \tag{7}$$

Parameter λ is the weight factor, typically assuming a value between 0 and 1, that acts as a hyperparameter controlling the trade-off between standard loss and knowledge distillation loss.

3.5. Backpropagation

To update the weights of student model P^s using backpropagation, the gradients of total loss L_T with respect to student model weights θ_s are calculated in the following form:

$$\nabla\theta_s = \frac{L_s}{\theta_s}\ , \tag{8}$$

$$\nabla\theta_{KD} = \frac{L_{KD}}{\theta_s}\ , \tag{9}$$

$$\nabla\theta_T = \alpha\, \nabla\theta_s + (1 - \alpha)\, \nabla\theta_{KD}\ , \tag{10}$$

where $\nabla\theta_s$ is the gradient of standard loss L_s with respect to student weights θ_s, $\nabla\theta_{KD}$ is the gradient of L_{KD} with respect to student weights θ_s.

Once $\nabla\theta_T$ gradients are calculated, propagate these gradients backward through student model P^s from the output layer to the input layer, which updates the weights of each layer according to:

$$\theta'_s = \theta_s - \eta\, \nabla\theta_T\ , \tag{11}$$

where θ'_s is the updated weight and η is the learning rate.

3.6. Training Algorithm

The Algorithm 1 outlines the knowledge distillation process during training, where the weight of a student model is updated by backpropagation through gradient calculation.

Algorithm 1 Training process.

Start

Input

1: Training data $D = \{x_i^t, y_i^t\}$, where x_i^t is the input image, y_i^t is the corresponding ground truth label and initialize the hyper parameters such as optimizer, learning rate η, batch size B, number of classes C

2: Train teacher model P^t and obtain corresponding weights θ^t

Backpropagation

3: Initialize student model P^s with the hyper parameters and load teacher model P^t with weight θ^t

Forward pass

4: Set weight factor λ, temperature factor τ and employ student model P^s and teacher model P^t

5: **for** each mini batch B and input $DB = \{x_b^t, y_b^t\}$ **do** forward propagation and compute s weight θ_s, knowledge divergence loss L_{KD}, student loss L_s and total distillation loss L_T using Eqs. (1)–(5)

6: **end for**

Backward pass

7: Compute gradients $\nabla\theta_T$ as defined by Eqs. (6)–(8)

Update

8: Update the weight of student model weight θ_s using the calculated gradients $\nabla\theta_T$ and obtain updated student model weight θ'_s using Eq. (9)

9: Increment the iteration count i

Termination

10: Repeat steps 5 to 8 until the model converges or until a maximum number of iterations is reached

End

4. Experimental Setup and Results

The proposed methodology is evaluated using two benchmark datasets that present diverse and challenging conditions, the CWFID dataset [3] and the Sunflower dataset [20]. The CWFID dataset consists of 60 high-resolution images with detailed pixel-level annotations, collected by the *Bonirob* agricultural robot on an organic carrot farm. The images show carrot plants at the early true leaves, where dense plant clusters with complex double-compound leaves and secondary structures create significant challenges for segmentation due to frequent overlapping and occlusions.

Additionally, the Sunflower dataset, a publicly available resource, is used for crop and weed segmentation experiments. This dataset was acquired by an agricultural robot in sunflower fields in Jesi, Italy. It comprises 500 scene images organized into three subsets representing different stages of crop growth: emergence, intermediate growth, and the final stage before chemical treatment. The images were captured over various days and times to include natural variations in lighting and field conditions.

To facilitate evaluation, the datasets are divided into 80% for training and 20% for testing, with sample images and corresponding ground truth masks shown in Fig. 3.

To measure the effectiveness of the AKDUNet model, performance metrics such as Dice coefficient loss, F1 score, and IoU score are considered in the following manner:

$$Dice = 1 - \frac{2 \times TP}{(TP + FP) + (TP + FN)}\ , \tag{12}$$

$$Precision = \frac{TP}{TP + FP}\ , \tag{13}$$

$$Recall = \frac{TP}{TP + FN}\ , \tag{14}$$

$$\text{F1 score} = \frac{2 \times Precision \times Recall}{Precision + Recall}\ , \tag{15}$$

$$\text{IoU score} = \frac{TP}{TP + FN + FP}\ , \tag{16}$$

where TP, TN, FN, and FP are true positives, true negatives, false negatives, and false positives, respectively.

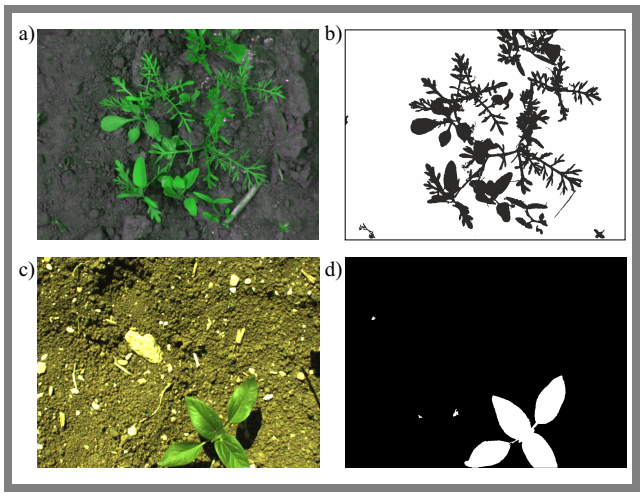

Fig. 3. Sample images and corresponding ground truth masks.

Tab. 2. AKDUNet model parameters.

Parameters	Teacher UNet P^t	Student UNet P^s
Depth	5	2
Filters	16, 32, 64, 128, 256	16, 32
Kernel size	3 × 3	2 × 2
Activation function	ReLU	ReLU
No. of parameters	2.35 M	0.013 M
Inference time [ms]	168 ms	65 ms

4.1. Model Parameters and Implementation Details

The UNet architecture is used as the backbone for both the teacher and student models. To evaluate the AKDUNet model, Google Colab is used equipped with NVIDIA GPUs and 12 GB of RAM to handle the necessary computations. The input images, originally 1296 × 966 pixels in resolution, are resized to 128 × 128 pixels for the evaluation process.

Table 2 presents the key parameters of the two models. The teacher model P^t, with a deeper architecture, employs filters of increasing size and a 3 × 3 kernel size. It has 2.35 m parameters and requires 168 ms for inference of the test image, reflecting its more complex design and computational demand. In contrast, the student model P^s, with a shallower architecture, uses fewer filters, a smaller 2 × 2 kernel size, and has significantly fewer parameters (0.29 M). This model achieves a faster inference time of 65 ms, making it more efficient and computationally powerful compared to the teacher model. Despite the difference in complexity, both models utilize the ReLU activation function, ensuring similar activation behavior across both architectures.

Table 3 describes the hyper-parameters used for the proposed model design. The learning rate η is set to 0.0001, allowing the model to make small, controlled updates to its weights during training. The Adam optimizer is employed to adaptively adjust the learning rate based on gradients, which helps the model to converge efficiently. A temperature factor τ of 4 softens the teacher model output probabilities in the knowledge distillation process. The weight factor λ is 0.5, balancing the contribution of the hard target loss (ground truth) and the soft target loss (teacher's predictions). With a batch size B of 8, the model processes 8 samples per iteration and it is set to perform binary segmentation with two classes C, representing the target leaf area and background in the segmentation task.

4.2. Ablation Study

This ablation study explores the effect of adding an attention module by comparing the models with and without it. The results show that adding the attention layer significantly improves performance, making it a valuable enhancement. Table 4 presents a comparative analysis of three models: student, teacher, and AKDUNet, evaluated on three key metrics such as F1 score, IoU score, and Dice coefficient. The models were evaluated using 5-fold cross-validation, ensuring that the evaluation is robust and not subject to over-fitting. The evaluation is conducted under two conditions, without the attention layer and with the attention layer. The student model performs poorly on all metrics, with an F1 score of 66.79%, an IoU score of 50.16%, and a loss of Dice coefficient of 0.1319.

At baseline (without attention), teacher model P^t outperforms the student model, achieving an F1 score of 90.06% and an IoU score of 81.98%, indicating its strong performance in segmentation tasks. However, the AKDUNet model shows

Tab. 3. Hyperparameters.

Parameters	Value
Learning rate η	0.0001
Optimizer	Adam
Temperature factor τ	4
Weight factor λ	0.5
Batch size B	8
Number of classes C	2

Tab. 4. Comparison of the models' performance with and without the attention layer.

Model	F1 score	IoU score	Dice coefficient loss
Without attention layer			
Student model	66.79%	50.16%	0.1319
Teacher model	90.06%	81.98%	0.0815
AKDUNet model	91.79%	85.82%	0.0414
With attention layer			
Student model	85.86%	75.22%	0.0773
Teacher model	94.69%	89.94%	0.0699
AKDUNet model	96.46%	93.16%	0.0227

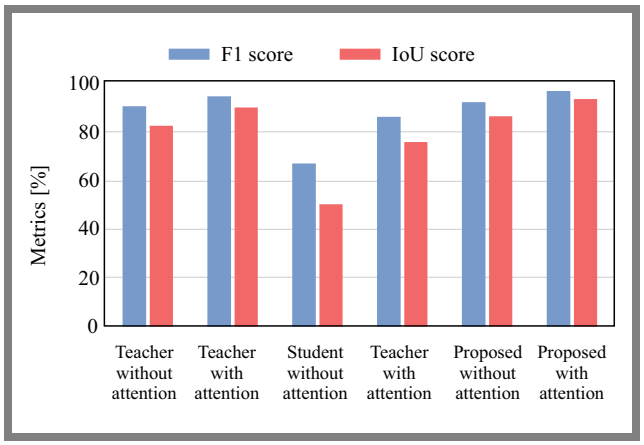

Fig. 4. Comparison of F1 and IoU scores for models with and without the attention layer.

a notable improvement with an F1 score of 91.79% and an IoU score of 85.82%, along with the lowest Dice coefficient loss of 0.0414, suggesting its superior ability to minimize coefficient losses.

Subsequently, an attention module is integrated into all the above-mentioned models and is evaluated for its effectiveness. The inclusion of the attention layer results in significant improvements in all three models. The AKDUNet model demonstrates the most significant performance improvement, achieving an F1 score of 96.46%, an IoU score of 93.16%, and a reduced Dice coefficient loss of 0.0227. The teacher model P^t also benefits from the addition of an attention layer, with an F1 score of 94.69%, an IoU score of 89.94%, and a decrease in the loss of Dice coefficient to 0.0699. Although student P^s model remains the least performing, it still shows improvements, with an F1 score of 85.86%, an IoU score of 75.22% and a Dice coefficient loss of 0.0773. These results show that incorporating an attention layer consistently improves segmentation performance across all models, as illustrated in Fig. 4.

Figure 5, which shows the Dice coefficient loss with and without the attention layer, highlights a significant reduction in prediction error when the attention layer is included. The teacher model P^t shows an improvement, with its loss of the Dice coefficient reducing from 0.0815 to 0.0699, indicating that attention helps minimize errors even in more complex models.

The student model P^s, while showing an improvement by reducing the loss of Dice coefficient from 0.1319 to 0.0773, still has the highest Dice coefficient loss among all models, highlighting its relative difficulty in minimizing prediction errors compared to the two remaining models. The AKDUNet model achieves the lowest loss of Dice coefficients in both conditions, decreasing from 0.0414 (without attention) to 0.0227 (with attention), showcasing its superior efficiency in reducing the error rate. Overall, the graph confirms that adding the attention layer contributes to a significant reduction in Dice coefficient loss, particularly for the proposed model, which achieves the lowest error across all conditions.

Tab. 5. Comparison of the proposed method with other methods using the CWFID data set.

Method	F1 score [%]	IoU score [%]	Loss
UNet++ [21]	77.56	63.37	0.2274
Inception UNet [22]	62.31	45.26	0.6211
SDUNet [15]	83.97	72.39	0.1650
INSCA UNet [16]	94.34	89.30	0.0604
VGG16 UNet [23]	90.70	82.99	0.0954
VGG19 UNet [24]	94.19	85.16	0.0634
ResNet UNet [25]	60.72	75.50	0.1730
SegFormer [18]	95.77	91.85	0.0456
Teacher UNet	94.69	89.94	0.0699
Student UNet	85.86	75.22	0.0773
AKDUNet model	96.46	93.16	0.0227

4.3. Comparison with Other Methods

The effectiveness of the proposed AKDUNet model in segmenting leaf regions from agricultural images was evaluated using two benchmark datasets, namely CWFID and Sunflower. The proposed method was benchmarked against various advanced UNet architectures, including UNet++, Inception UNet, SDUNet, and INSCA UNet, SegFormer, and pre-trained UNet models such as VGG16 UNet, VGG19 UNet, and ResNet UNet. As shown in Tab. 5, which presents results for the CWFID dataset, AKDUNet achieves a remarkable F1 score of 96.46%, IoU score of 94.85%, and the lowest loss value of 0.0208, outperforming all other compared methods.

These results indicate that AKDUNet provides highly accurate and consistent segmentation of leaf regions, which is crucial for downstream tasks such as disease detection, leaf counting, and plant phenotyping. While transformer-based models such as SegFormer also show strong performance with an F1 score of 95.77% and IoU of 91.85%, the superior metrics of AKDUNet highlight the benefits of integrating convolutional feature learning with attention-guided knowledge distillation, enabling the model to capture fine-grained leaf boundaries and spatial details more effectively.

Models such as INSCA UNet with an F1 score of 94.34% and VGG19 UNet with 94.19%, also demonstrate competitive

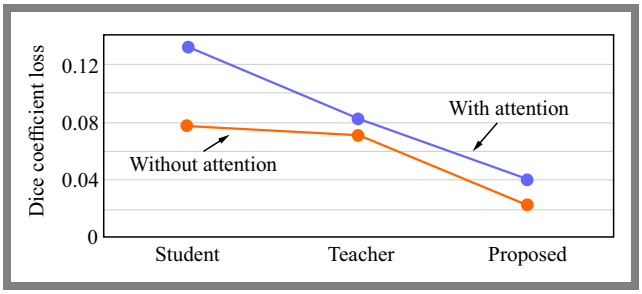

Fig. 5. Dice coefficient loss for models with and without the attention layer.

Tab. 6. Comparison of the proposed method with other methods using the CWFID data set.

Method	F1 score [%]	IoU score [%]	Loss
UNet++ [21]	70.04	69.38	0.0890
Inception UNet [22]	94.31	84.99	0.0040
SDUNet [15]	66.75	50.09	0.0220
INSCA UNet [16]	62.94	46.07	0.0238
VGG16 UNet [23]	90.90	82.35	0.0127
VGG19 UNet [24]	94.78	87.37	0.0064
ResNet UNet [25]	91.00	83.62	0.0152
SegFormer [18]	93.66	82.87	0.0042
Teacher UNet	90.44	77.99	0.0079
Student UNet	85.79	68.70	0.0119
AKDUNet model	95.16	88.17	0.0037

performance, yet were surpassed by AKDUNet in terms of both IoU and loss. This indicates that while attention mechanisms and deep CNN backbones aid in segmentation, the interlaced attention and knowledge transfer mechanism of AKDUNet allows for a more robust and precise extraction of leaf regions. Traditional models like UNet++, Inception UNet, and ResNet UNet show comparatively lower performance, with F1 scores below 78%, suggesting that they may struggle to delineate leaf boundaries in complex backgrounds or overlapping structures.

With the Sunflower dataset, as shown in Tab. 6, a similar pattern emerges. AKDUNet achieves the best performance with an F1 score of 95.16%, an IoU score of 88.17%, and the lowest loss of 0.0037, demonstrating its generalizability across different types of leaf structures and lighting conditions. Competing models such as VGG19 UNet with an F1 score of 94.78% and Inception UNet with 94.31% deliver strong results, but fall short in overall consistency and overlap accuracy compared to AKDUNet.

Interestingly, SegFormer, while highly effective on the CWFID dataset, shows a drop in performance here with an F1 score of 93.66% and IoU of 82.87%, suggesting that AKDUNet's convolutional attention hybrid design is more robust for diverse leaf morphology and field conditions. Moreover, the performance gap between the teacher UNet and the student UNet illustrates the impact of knowledge distillation, whereas AKDUNet significantly exceeds both, validating the advantage of its attention-enhanced distillation strategy.

Analysis performed with the use of both CWFID and Sunflower datasets clearly demonstrates the effectiveness and generalizability of the proposed AKDUNet model for leaf region segmentation in agricultural images. AKDUNet consistently outperforms all other benchmarked methods in terms of F1 score, IoU, and loss, indicating its ability to accurately extract leaf regions with minimal prediction error. Strong performance in both datasets, despite differences in leaf types, image complexity, and lighting conditions, highlights the robustness of the model's architecture.

4.4. Model Complexity Comparison

To assess computational complexity, the AKDUNet model uses floating point operations per second (FLOPs), which consists of number of addition, subtraction, multiplication, and division operations involved during the training process, and the formula to calculate FLOPs is shown in Eq. (17). In addition, the number of parameters for each method is also considered. To ensure fairness in comparison, all methods including the proposed model are evaluated with the same input size of 128 × 128 × 3.

For a Conv2D layer:

$$FLOPs_S = 2 \times H_{out} \times W_{out} \times C_{in} \times K_H \times K_W \times C_{out} \ , \quad (17)$$

where H_{out}, W_{out} – output height and width of the image, C_{in} – input channels of the image, K_H, K_W – kernel height and width of the filter, C_{out} – output channels.

Table 7 presents a comparative analysis of various segmentation models in terms of the number of parameters and computational complexity (FLOPs), while Fig. 6 offers a visual comparison based on parameter count (in millions) and FLOPs (in giga FLOPs).

The proposed AKDUNet model stands out with an exceptionally low parameter count of just 0.013 M and a minimal 0.17 GFLOPs, demonstrating remarkable efficiency in both memory usage and computational cost. Despite its lightweight architecture, AKDUNet achieves a superior segmentation accuracy of 93.16%, surpassing more complex models such as VGG16 UNet with 25.85 M parameters, 32.26 GFLOPs, and VGG19 UNet with 16.24 M parameters, 41.15 GFLOPs, which require significantly more resources. The reduced complexity enables faster execution and lower resource consumption, making it suitable for deployment in resource-constrained environments without compromising segmentation performance.

In comparison, models such as SDUNet with 0.498 M parameters and 5.22 GFLOPs as well as ResNet UNet with 24.29 M parameters and 3.12 GFLOPs offer a trade-off be-

Tab. 7. Comparison of model complexity and computational requirements.

Model	Parameters [M]	FLOPs [G]
UNet++	0.149	4.89
VGG16 UNet	25.85	32.26
VGG19 UNet	16.24	41.15
ResNet UNet	24.29	3.12
Inception UNet	0.177	2.90
SDUNet	0.498	5.22
SegFormer	0.124	5.2
Teacher UNet	2.354	2.67
Proposed AKDUNet	0.013	0.17

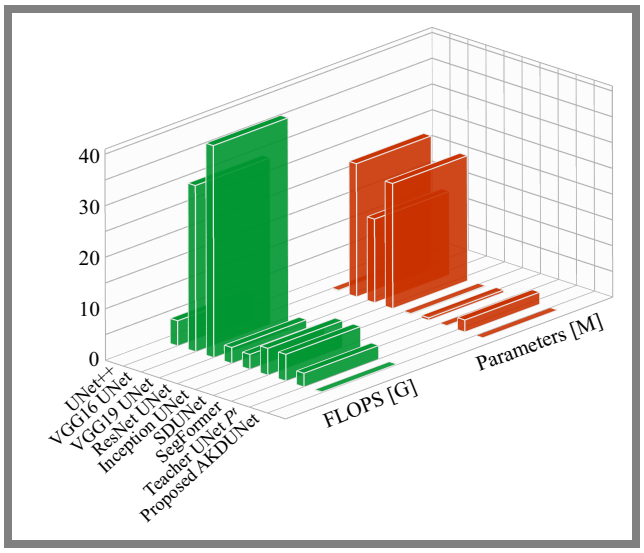

Fig. 6. Comparison of computational complexity with existing models.

tween complexity and accuracy, but fall short in both segmentation precision and computational efficiency. Similarly, the teacher UNet model, with 2.354 M parameters and 2.67 GFLOPs, delivers competitive results, but does not outperform AKDUNet in either metric. The SegFormer model, despite its lightweight nature with only 0.124 M parameters, incurs a relatively high computational cost of 5.2 GFLOPs due to its transformer-based architecture, limiting its usability for real-time deployment on edge devices.

The ability of AKDUNet to achieve superior segmentation accuracy with a significantly lower computational footprint highlights its effectiveness and efficiency, especially when both IoU score and resource demands are critical considerations.

4.5. Qualitative Analysis

Grad CAM visualizations were also generated for the student model P^s, the teacher model P^t, and the AKDUNet model to assess the effectiveness of the distillation process, as shown in Fig. 7.

The results revealed that while both the teacher model P^t and the student model P^s showed attention to relevant features, the AKDUNet model demonstrated superior focus on the critical regions of the input images. This suggests that, while the student model P^s achieves reasonable accuracy, it is not as interpretable or capable of making significant decisions as the teacher model P^t. Specifically, the grad CAM heat maps of the AKDUNet model closely aligned with those of the teacher model P^t, indicating that the student model P^s successfully learned to prioritize the most significant areas through the distillation process.

To evaluate the effectiveness of the proposed architecture, a qualitative analysis was conducted for all models, and the performance was compared for scenarios with and without knowledge distillation. As shown in Fig. 8, segmentation maps from the student model P^s trained from scratch without knowledge distillation lack clear separation between the

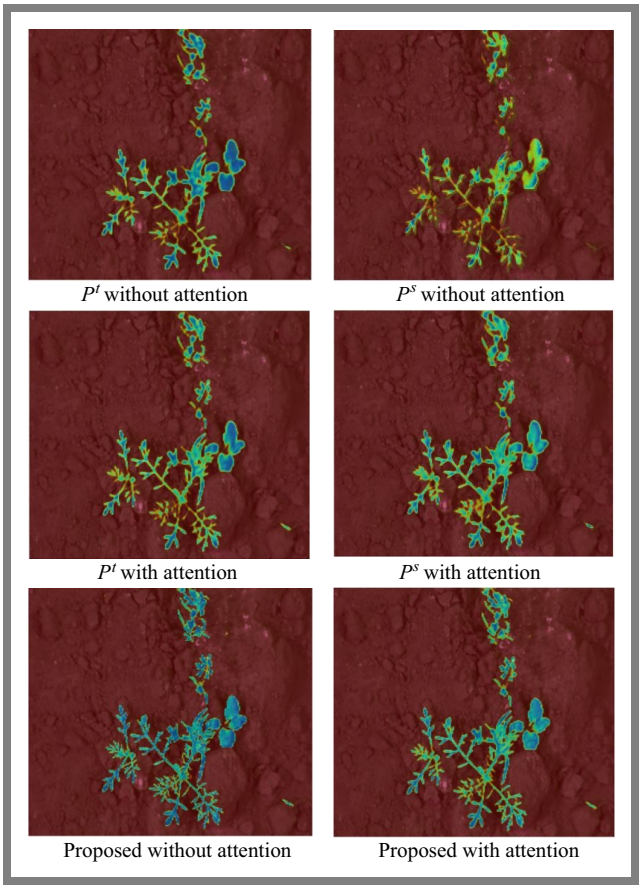

Fig. 7. Grad CAM visualizations of all models.

secondary leaflets and do not present sharp boundaries. This model produces an incomplete segmentation map with many leaf regions incorrectly classified as background.

In contrast, the segmentation map produced by the proposed model with an attention gated module is much clearer and more accurately aligns with the ground truth map, as shown in Fig. 8. This analysis demonstrates that combining knowledge distillation with an attention layer significantly enhances segmentation performance. This improvement is made possible by the transfer of knowledge from the teacher model P^t, whose segmented output outperforms the student model's output image.

Figures 9, 10 show the original images, ground truth, and segmentation outputs for a variety of models, including pretrained UNet architectures such as VGG16 UNet, VGG19 UNet, and ResNet UNet, along with advanced models like UNet++, Inception UNet, SDUNet, and INCSA UNet. By integrating knowledge distillation techniques and attention mechanisms, AKDUNet achieves highly accurate segmentation, even in challenging areas such as secondary leaflets and finer details of leaf structures.

The attention module enhances the model's ability to focus on critical regions, while KD effectively transfers knowledge from the teacher model, improving the student model's generalization and segmentation accuracy. The qualitative results clearly demonstrate that AKDUNet significantly outperforms existing segmentation approaches. It delivers sharper and

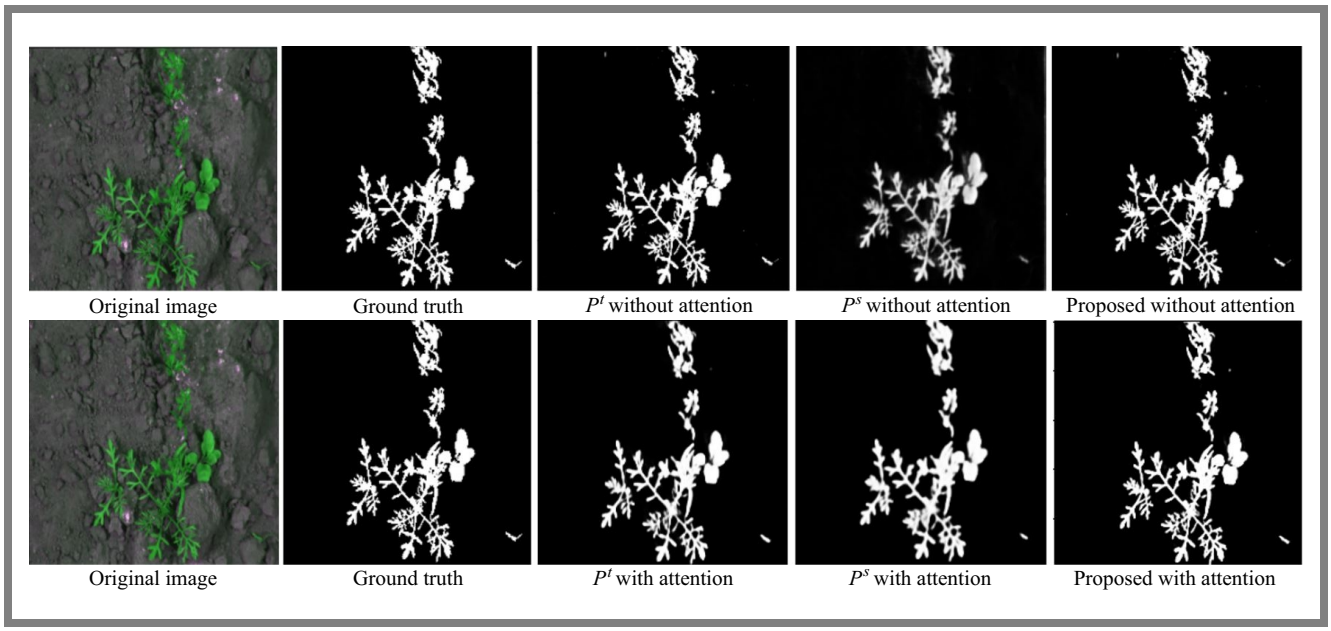

Fig. 8. Leaf area segmentation results with and without the attention module.

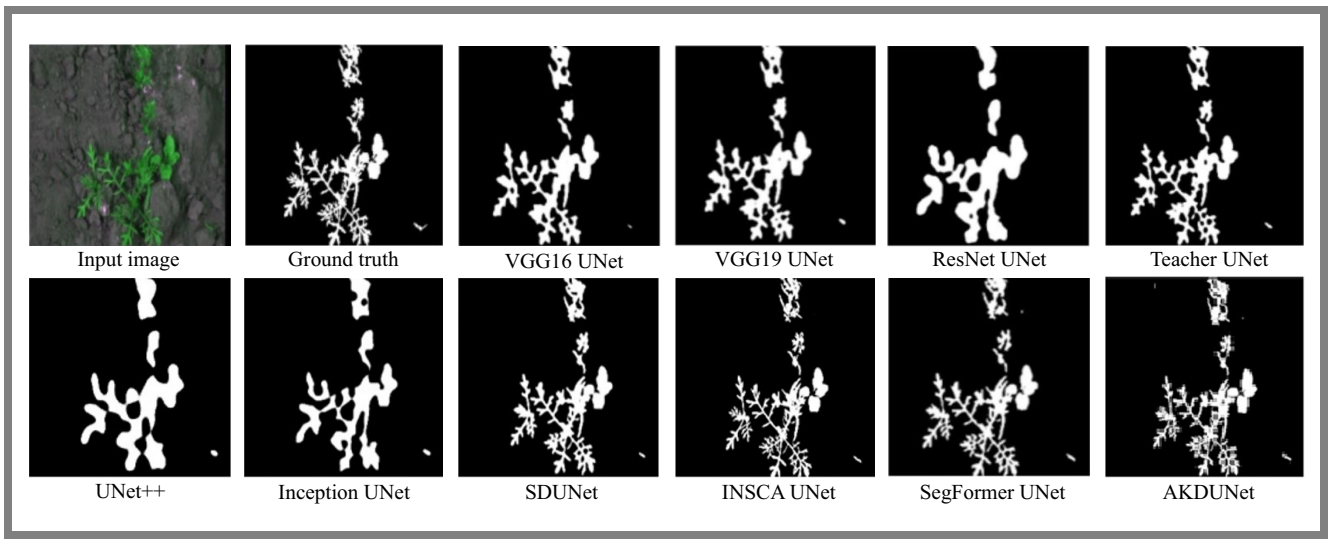

Fig. 9. Results of segmentation – area of the CWFID carrot leaf.

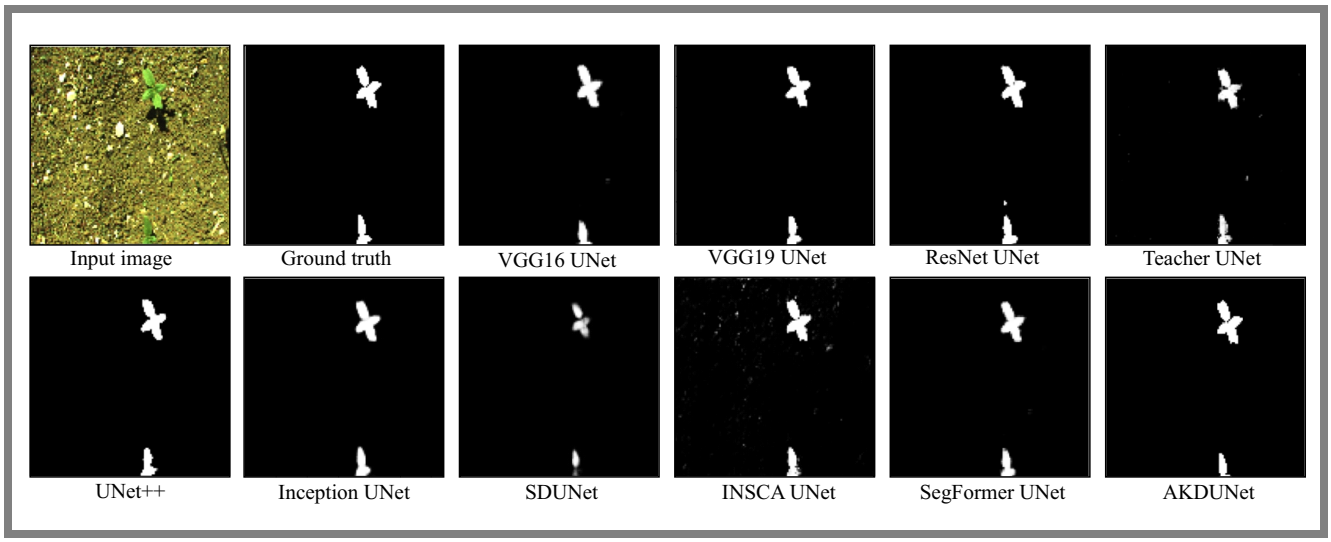

Fig. 10. Results of segmentation – sunflower leaf area.

more precise boundaries, particularly in complex areas, when compared to UNet, VGG16 UNet, VGG19 UNet, ResNet UNet, UNet++, Inception UNet, SDUNet, and INCSA UNet.

5. Conclusions

This study demonstrates the performance of the proposed AKDUNet model while segmenting the leaf region for precision agriculture. Across the CWFID carrot leaf and Sunflower data sets from CWFID, AKDUNet consistently outperformed several state-of-the-art architectures, including SegFormer, UNet++, Inception UNet, SDUNet, INSCA UNet, and pre-trained UNet variants with VGG16, VGG19 and ResNet backbones. The model achieved higher F1 scores and IoU values while maintaining minimal loss, validating its robustness in handling variations in leaf morphology and imaging conditions.

Its significantly lower computational complexity is a key advantage of AKDUNet. Although only 0.013 million parameters and 0.17 GFLOPs were used, the segmentation results were comparable to or better than those of much larger models. Such a high degree of efficiency positions AKDUNet as a strong candidate for real-time deployment on resource-constrained devices such as drones, mobile phones, or edge computing platforms used in agricultural monitoring systems.

The model's architecture, which integrates attention mechanisms and knowledge distillation from a deeper teacher network, enables it to focus on critical regions and segment fine details with a high degree of accuracy. Its ability to generalize across datasets suggests that it can adapt well to diverse plant species and imaging conditions without the need for extensive retraining.

However, there are certain limitations to this work. First, the evaluation was conducted on a limited number of datasets, primarily focused on leaf structures from specific crops. While AKDUNet showed strong generalization across these, its performance on more complex agricultural scenes with overlapping plant parts, occlusions, or mixed crop types has not yet been tested. Second, although the model is lightweight, it still relies on supervised learning and labeled data, which can be costly and time-consuming to obtain at scale. Finally, the knowledge distillation strategy, while effective, may require careful tuning of teacher-student dynamics to ensure stable training across various domains.

Acknowledgments

The datasets used in the study are publicly available in the following repository: https://github.com/cwfid/dataset.

References

[1] A. Walter and U. Schurr, "The Modular Character of Growth in Nicotiana Tabacum Plants under Steady-state Nutrition", *Journal of Experimental Botany*, vol. 50, pp. 1169–1177, 1999 (https://doi.org/10.1093/jxb/50.336.1169).

[2] A.L. Chandra, S.V. Desai, W. Guo, and V.N. Balasubramanian, "Computer Vision with Deep Learning for Plant Phenotyping in Agriculture: A Survey", *ArXiv*, 2020 (https://doi.org/10.48550/arXiv.2006.11391).

[3] S. Haug and J. Ostermann, "A Crop/weed Field Image Dataset for the Evaluation of Computer Vision Based Precision Agriculture Tasks", *Proc. of Computer Vision – ECCV 2014*, pp. 105–116, 2014 (https://doi.org/10.1007/978-3-319-16220-1_8).

[4] R. Liu *et al.*, "TransKD: Transformer Knowledge Distillation for Efficient Semantic Segmentation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, 2024 (https://doi.org/10.1109/TITS.2024.3455416).

[5] J. Zhang, Q. Liang, and Y. Shi, "KD-SCFNet: Towards More Accurate and Efficient Salient Object Detection via Knowledge Distillation", *ArXiv*, 2022 (https://doi.org/10.48550/arXiv.2208.02178).

[6] S. An, Q. Liao, Z. Lu, and J.-H. Xue, "Efficient Semantic Segmentation via Self-attention and Self-distillation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 15256–15266, 2022 (https://doi.org/10.1109/TITS.2021.3139001).

[7] T. Zhang *et al.*, "Efficient RGB-T Tracking via Cross-modality Distillation", *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, 2023 (https://doi.org/10.1109/CVPR52729.2023.00523).

[8] X. Hu, K. Yang, L. Fei, and K. Wang, "ACNet: Attention Based Network to Exploit Complementary Features for RGBD Semantic Segmentation", *IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019 (https://doi.org/10.1109/ICIP.2019.8803025).

[9] W. Zhou, J. Yuan, J. Lei, and T. Luo, "TSNet: Three-stream Self-attention Network for RGB-D Indoor Semantic Segmentation", *IEEE Intelligent Systems*, vol. 36, pp. 73–78, 2021 (https://doi.org/10.1109/MIS.2020.2999462).

[10] W. Zhou, E. Yang, J. Lei, and L. Yu, "FRNet: Feature Reconstruction Network for RGB-D Indoor Scene Parsing", *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 677–687, 2022 (https://doi.org/10.1109/JSTSP.2022.3174338).

[11] J. Zhou *et al.*, "ESA-Net: A Network with Efficient Spatial Attention for Smoky Vehicle Detection", 2021 *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Glasgow, UK, 2021 (https://doi.org/10.1109/I2MTC50364.2021.9460078).

[12] P. Chavan, P.P. Chavan, and A. Chavan, "Hybrid Architecture for Crop Detection and Leaf Disease Detection with Improved U-Net Segmentation Model and Image Processing", *Crop Protection*, vol. 190, art. no. 107117, 2025 (https://doi.org/10.1016/j.cropro.2025.107117).

[13] M.K. Surehli, N. Aggarwal, G. Joshi, and H. Nayyar, "Semantic Segmentation of Plant Structures with Deep Learning and Channel-wise Attention Mechanism", *JTIT*, vol. 99, pp. 56–66, 2025 (https://doi.org/10.26636/jtit.2025.1.1853).

[14] A. Das *et al.*, "Deep Learning-based Classification, Detection, and Segmentation of Tomato Leaf Diseases: A State-of-the-art Review", *Artificial Intelligence in Agriculture*, vol. 15, pp. 192–220, 2025 (https://doi.org/10.1016/j.aiia.2025.02.006).

[15] C. Guo *et al.*, "SD-UNet: A Structured Dropout U-Net for Retinal Vessel Segmentation", *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, Athens, Greece, 2019 (https://doi.org/10.1109/BIBE.2019.00085).

[16] I. Delibasoglu, "INCSA-UNET: Spatial Attention Inception UNET for Aerial Images Segmentation", *Computing and Informatics*, vol. 40, pp. 1244–1262, 2022 (https://doi.org/10.31577/cai_2021_6_1244).

[17] M. Agarwal, S.K. Gupta, and K.K. Biswas, "Plant Leaf Disease Segmentation Using Compressed UNet Architecture", *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 9–14, 2021 (https://doi.org/10.1007/978-3-030-75015-2_2).

[18] E. Xie *et al.*, "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers", *ArXiv*, 2021 (https://doi.org/10.48550/arXiv.2105.15203).

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Munich, Germany, 2015 (https://doi.org/10.1007/978-3-319-24574-4_28).

[20] M. Fawakherji *et al.*, "Multi-spectral Image Synthesis for Crop/weed Segmentation in Precision Farming", *Robotics and Autonomous Systems*, vol. 146, art. no. 103861, 2021 (https://doi.org/10.1016/j.robot.2021.103861).

[21] Z. Zhou, M.R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A Nested UNet Architecture for Medical Image Segmentation", in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Lecture Notes in Computer Science*, Springer Cham, pp. 3–11, 2018 (https://doi.org/10.1007/978-3-030-00889-5_1).

[22] N.S. Punn, S. Agarwal, "Inception U-Net Architecture for Semantic Segmentation to Identify Nuclei in Microscopy Cell Images", *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 1, pp. 1–15, 2020 (https://doi.org/10.1145/3376922).

[23] Y. Miao *et al.*, "CT Image Segmentation of Foxtail Millet Seeds Based on Semantic Segmentation Model VGG16-UNet", *Plant Methods*, vol. 20, art. no. 169, 2024 (https://doi.org/10.1186/s13007-024-01288-y).

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition", *ArXiv*, 2014 (https://doi.org/10.48550/arXiv.1409.1556).

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016 (https://doi.org/10.1109/CVPR.2016.90).

A. Shamim Banu, Research Scholar
Department of ECE
https://orcid.org/0009-0002-2329-4235
E-mail: shamiece@gmail.com
National Institute of Technology, Tiruchirappalli, Tamilnadu, India
https://www.nitt.edu
Government Polytechnic College, Tiruchirappalli, Tamilnadu, India
http://gptctrichy.com

S. Deivalakshmi, Ph.D., Associate Professor
Department of ECE
https://orcid.org/0000-0002-7019-9807
E-mail: deiva@nitt.edu
National Institute of Technology, Tiruchirappalli, Tamilnadu, India
https://www.nitt.edu