Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 10 No. 1 (2026) 113 - 126 e-ISSN: 2580-0760

Multi-Class Semantic Segmentation of Oil Palm Areas Using a VGG-19 U-Net Improvement

Maura Widyaningsih[1], Tri Kuntoro Priyambodo[2]*, Moh Edi Wibowo[3], Muhammad Kamal[4]
[1]Department of Informatica Technic, STMIK Palangkaraya, Kalimantan Tengah, Indonesia
[2,3]Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia
[4]Department of Computer Science and Electronics, Faculty of Geography, Universitas Gadjah Mada, Yogyakarta, Indonesia
[1]maurawidyaningsih@mail.ugm.ac.id, [2*]mastri@ugm.ac.id, [3]mediw@ugm.ac.id, [4]m.kamal@ugm.ac.id

Abstract

UAV imagery-based semantic segmentation is crucial for mapping tropical agricultural areas such as oil palm plantations. The main challenges are overlapping vegetation objects, unclear boundaries, and spectral similarities between classes, which reduce the accuracy of conventional models. This study proposes a modified U-Net architecture with a VGG-19 backbone, achieved through hyperparameter tuning (M7) and the integration of residual blocks (M8), to enhance multi-class segmentation performance. Experiments were conducted on aerial imagery with two resolutions (512×512 and 256×256) using four-class and three-class scenarios. The results show that M7 and M8 consistently outperform the baseline model (M2) in terms of accuracy, precision, recall, and average Intersection over Union (IoU). In the 512x512 four-class scenario, M8 achieved the highest accuracy (87.40%), precision (88.32%), recall (86.32%), and MIoU (0.132). M7 reached similar accuracy (>86%) but trained significantly faster than the baseline. In the 256x256 scenario, M8 maintained strong performance with 86.44% accuracy and 0.302 MIoU. For the three-class experiment, M8 reached a top MIoU of 0.178. Accuracy, precision, and recall were all above 87%, showing improved recognition of minority classes such as waterways. Confusion matrix analysis confirmed that M8 provided more balanced class predictions. It also reduced false negatives for oil palm vegetation. M7 showed slight fluctuations, suggesting possible overfitting. These findings support M8 as a robust solution for UAV-based oil palm mapping and large-scale monitoring.

Keywords: multi-class; oil palm; residual block; semantic segmentation; U-Net; VGG-19

How to Cite: [T. K. Priyambodo, M. Widyaningsih, M. E. Wibowo, and M. Kamal, "Multi-Class Semantic Segmentation of Oil Palm Areas Using a VGG-19 U-Net Improvement ", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 10, no. 1, pp. 113 - 126, Feb. 2026.
*Permalink/*DOI: https://doi.org/10.29207/resti.v10i1.7062

Received: August 8, 2025
Accepted: February, 3 2026
Available Online: February 23, 2026

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Accurate mapping of vegetation in an area of land cover classes is crucial in monitoring oil palm plantations, particularly through remote sensing image analysis [1]. Remote sensing data, particularly from drone images, presents challenges due to its wide spatial coverage and spectral similarities between land surface features [2]. Such similarities often complicate the mapping process, particularly in segmentation and classification tasks, due to factors such as vegetation density, color similarity, and object overlap. These issues can cause object boundary ambiguity and complicate pixel-level labeling [3]. Although the processing time for generating map images can be lengthy, advances in deep learning computation continue to provide opportunities to address these challenges effectively, especially for pixel-level analysis in complex tropical plantation environments.

Semantic segmentation, a subfield of Deep Learning (DL), has emerged as a powerful approach to addressing this problem by assigning a class label to each pixel in an image [4]. The biggest challenge in semantic segmentation is assigning each pixel to a specific category, due to variations in plant structure, biochemistry, and physiology [5]. These challenges clearly demonstrate the inherent difficulties in providing and processing UAV-based plantation imagery, given the scarcity of annotated data and the high intra-class similarity. It is essential to develop segmentation models that consistently perform with

limited or noisy labels by optimizing the architecture rather than relying on larger datasets or intricate post-processing.

Among segmentation methods, U-Net has become widely recognized due to its encoder-decoder structure [6], skip connections[4], and robustness across biomedical and environmental imaging applications [7], [8]. Still, its performance may be limited in complex settings such as tropical oil palm plantations, where class imbalance, spectral similarity, and unclear boundaries pose challenges. These challenges point to the need for architectural improvements and encoder–decoder refinements, rather than simply switching backbone networks.

Building upon these limitations, the novelty of this study lies in the systematic refinement of a VGG-19-based U-Net architecture tailored for UAV-based oil palm semantic segmentation. Unlike existing studies that mainly focus on backbone substitution or general-purpose segmentation frameworks, this research emphasizes controlled architectural modification at the decoder level through hyperparameter optimization and residual connection integration. These modifications are designed to improve feature reuse, gradient stability, and class balance in multi-class segmentation tasks under limited data conditions.

This study adopts, develops, and applies a modified U-Net architecture based on the VGG-19 Backbone to improve semantic segmentation of oil palm areas. This improvement stems from previous research demonstrating the effectiveness of model compression techniques (e.g., pruning, layer reduction, residual connections) in improving segmentation accuracy while maintaining computational efficiency [9], [10]. Such efficiency is critical for UAV-based monitoring systems, where onboard computation and processing resources are constrained. The primary objective of this research is to implement and evaluate enhanced variants of a VGG-19-based U-Net model for multi-class semantic segmentation of oil palm plantation components.

This study used UAV aerial imagery from an oil palm plantation in Palangka Raya City, Central Kalimantan Province. The dataset comprised 200 image–mask pairs resized to 512 × 512 pixels and 400 pairs at 256 × 256 pixels. Mask images were manually annotated for two segmentation scenarios with three and four land-cover classes, respectively. After acquisition, the data were preprocessed with resizing, enhancement, and normalization. Segmentation learning was performed using a VGG-19-based U-Net under two scenarios: hyperparameter tuning and integrating residual connections in the decoder. Model performance was assessed using confusion matrix analysis and segmentation metrics.

The segmentation experiments were conducted using two class configurations. The first configuration involved three classes: roads/drainage/other objects, oil palm vegetation, and non-oil palm areas. The second configuration involved four classes: roads/land, oil palm vegetation, non-oil palm vegetation, and drainage channels**.** This multi-scheme evaluation was designed to assess the model's adaptability and robustness across varying segmentation complexities and labeling granularity.

Overall, this study contributes by demonstrating that decoder-level architectural refinement in a VGG-19-based U-Net can significantly improve multi-class semantic segmentation performance for UAV-based oil palm imagery. The main contributions of this research are as follows: (1) integrating hyperparameter tuning and residual connections to enhance segmentation robustness and class balance; (2) optimizing computational efficiency for UAV-based plantation monitoring applications; and (3) benchmarking the proposed models against established baseline architectures, including VGG-16 [11], VGG-19 [12], SegNet [13], Fully Convolutional Network 8-stride (FCN-8s) [14], ResNet [15], and AtrousDenseUDeconvNet (ADUDN) [4].

2. Methods

Segmentation learning is still under development and requires algorithm improvements to address persistent problems in learning image data [16], [17]. Deep learning (DL) helps train high-complexity images to achieve optimal results [18], [19], processes large datasets efficiently [4], and provides a low-cost approach [20]. Segmentation-based architectures reduce training time [21], recognize detailed features [22], and handle problematic image annotations [23].

U-Net is a convolutional neural network (CNN) architecture specifically designed for image segmentation. U-Net belongs to the DL approach, which differs from conventional machine learning (ML)-based classification methods [17]. The U-Net architecture is relatively easy to modify and integrate with other models, making it flexible for various image segmentation needs [23], [24].

2.1 U-Net Modification

The baseline U-Net VGG-19 has a large parameter size, which results in longer processing time and increased computational cost. In this study, a modified version of the U-Net VGG-19 architecture is implemented by applying hyperparameter tuning and incorporating residual functions to improve model performance.

Tuning key hyperparameters significantly improves segmentation accuracy, reduces overfitting, and accelerates the training process. The addition of residual functions, especially in the decoder layer, helps mitigate feature degradation and overcomes the vanishing gradient problem commonly found in deep architectures [10]. Furthermore, adjustments to the convolution kernels in the encoder, bridge, and decoder

layers are performed to better handle heterogeneous aerial imagery data.

The modified architecture, as illustrated in Figure 1, retains the original U-Net encoder-decoder structure with a VGG-19-based encoder consisting of five convolutional blocks (C1–C5) and a central bridge layer (B1). The decoder uses transpose convolution layers with skip connections from each corresponding encoder block. The improvements include hyperparameter tuning and integration of residual connections to improve feature learning and network stability [25].

The modified architecture, as illustrated in Figure 1, retains the original U-Net encoder-decoder structure with a VGG-19-based encoder consisting of five convolutional blocks (C1–C5) and a central bridge layer (B1). The decoder uses transpose convolution layers with skip connections from each corresponding encoder block. The improvements include hyperparameter tuning and integration of residual connections to improve feature learning and network stability [20].

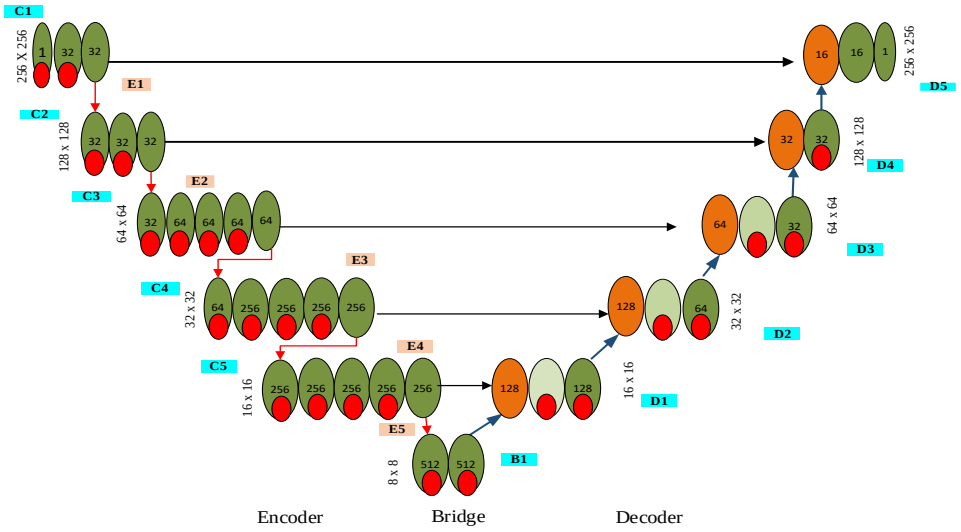


Figure 1. Hyperparameter tuning modification on U-Net VGG19

Figure 2 introduces the design of the modified Figure 1 with the addition of residual functions, to support better convergence and generalization. These functions are mainly integrated in the decoder layers D1, D2, D3, and D4 to retain important feature information.

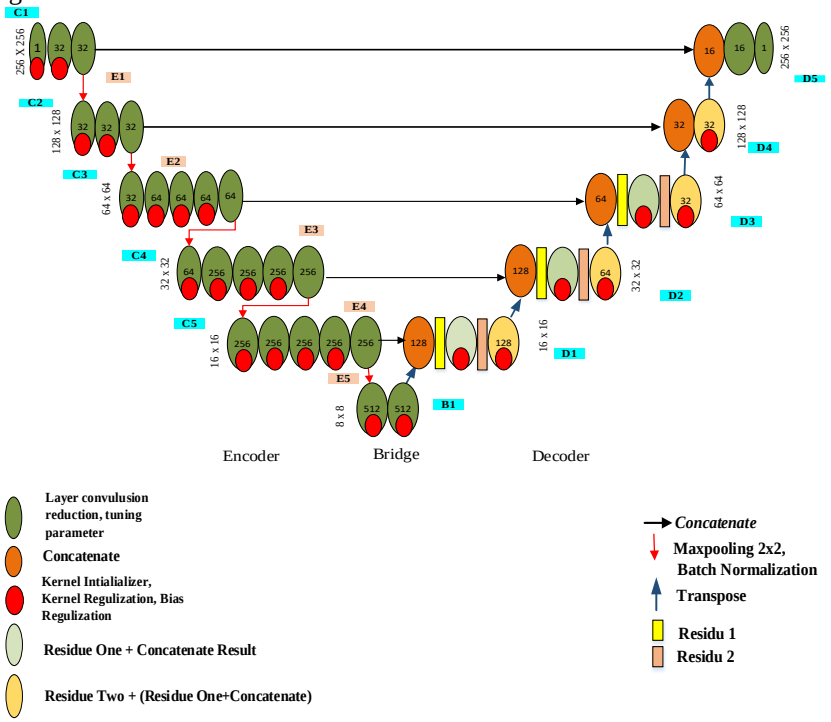
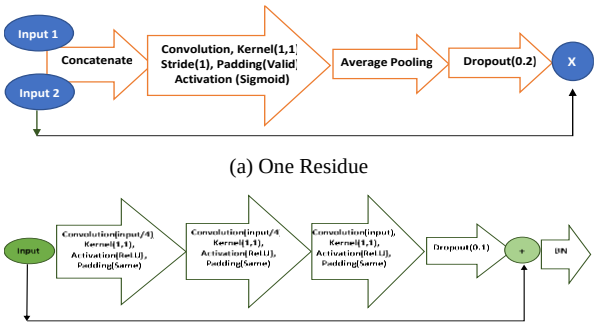


Figure 2. Addition of Residual function + hyperparameter tuning on U-Net VGG19

The introduction of residual functions, as shown in Figure 3, supports better convergence and generalization. These functions are primarily integrated in the decoder layers D1, D2, D3, and D4 to retain important feature information, while residual blocks help stabilize network training and improve per-pixel classification performance.

Residual Module One, shown in Figure 3(a), receives two concatenated 2-input tensors and passes them through a convolutional layer with a 1×1 kernel, stride 1, valid padding, sigmoid activation, and then average pooling with a Dropout of 0.2. The output is then generated by multiplying the result by the second input data. This module also serves as a mechanism that allows the model to focus on weighted feature integration.

Residual Module Two in Figure 3(b) consists of three consecutive 1×1 convolutional layers with ReLU activation and the same padding. After convolution, a Dropout of 0.1 is applied. The final output is added to the original input (residual connections) and then followed by batch normalization, which maintains the stability of the activation distribution and improves gradient propagation.



(a) One Residue

(b) Two Residue

Figure 3. Block residue

These architectural improvements contribute to more efficient training, higher segmentation accuracy, and better handling of class boundaries in complex oil palm area images.

2.2 Research work stages

The workflow of the network applied to oil palm plantation imagery, as shown in Figure 4, consists of several stages of semantic segmentation.

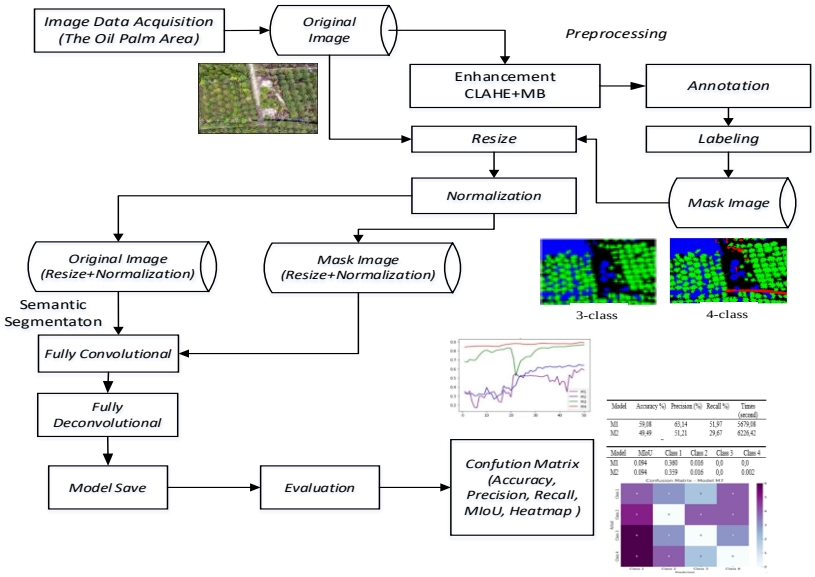


Figure 4. Semantic Segmentation processing stages

Aerial imagery data was acquired in a palm oil plantation area using a drone (UAV). The area obtained from the drone data with an RGB sensor is approximately 126 hectares. The drone's height is 100 meters (from the ground), the resulting resolution is 16.9 centimeters per pixel, and the resolution of each photo is 4056x3040 pixels. The length and width of each photo are in meters: L = 154 meters and W = 116 meters. The digital data format is processed in the form of a ".png" file.

Preprocessing is crucial in machine learning and data analytics, as it significantly impacts accuracy and effectiveness. It enhances image quality to improve segmentation analysis [24-26]. Proper preprocessing ensures consistent object separation in predictions, especially when dealing with significant intra-class variability and large-scale objects, which often lowers segmentation accuracy. In this study, preprocessing was performed by combining Median Blur (MB) and Contrast Limited Adaptive Histogram Equalization (CLAHE). The goal is to improve image quality by reducing noise, such as salt-and-pepper interference [27], [28]. Additional enhancement processes, in addition to MB and CLAHE, include adjusting the YUV color space and then returning it to the RGB color space. This helps restore cloud-free images and sharpen pixel intensities [3]. The results of these three enhancement processes will then be continued to data

annotation using the ENVI tool, which helps in creating data masks as targets for dataset learning.

Object annotation is a crucial data preparation step in segmentation learning, often using a region-based concept [29] to assign numerical values to each class. Annotation helps separate classes based on the areas to be used. There are two annotation techniques used here: 3-class and 4-class. Annotation was performed on objects of 3 classes for the Road/land/gutter/other area: black, oil palm vegetation: green, non-oil palm vegetation: blue) and 4 classes for the Road/land area: black, oil palm vegetation: green, non-oil palm vegetation: blue, and ditch/gutter: red. Sample data from a portion of the area representing 4 classes, consisting of 200 datasets for 512x512 resize, 400 datasets for 256x256 size.

Segmentation is used to delineate regions and improve data quality [30], as well as aid object detection [31]. Segmentation divides an image into coherent parts based on regional homogeneity, natural color, texture [23], or semantic meaning for classification. A major challenge in segmentation is distinguishing vegetation types due to their irregular shapes, patterns, and colors [3], as well as the complexity of high-dimensional data [32]. The semantic segmentation process involves two main phases: fully convolutional and fully deconvolutional.

In the fully convolutional phase, the encoder part of the U-Net processes the image by extracting hierarchical features, capturing patterns such as shape and texture. These features are important for distinguishing different objects in the image. In the fully deconvolutional phase, the decoder up-samples/ transposes the feature maps, restoring the original image dimensions, and ensuring precise pixel-by-pixel segmentation. This process allows the network to accurately reconstruct segmented areas based on the learned features.

The VGG-19 concept uses deeper layers, trained on large datasets, and effectively learns low-level features such as spatial information, edges, rotation, lighting, shape, and transferable features for knowledge transfer and feature extraction [12]. Convolution-Deconvolution functions as an encoder-decoder, with the bridge representing the image and the classification map, with the encoder and decoder facilitating the translation between these representations [33]. The transition formulas in a convolutional network are shown in Table 1.

The model's working process begins with the encoder section, which extracts image features through a convolution operation using a W_m filter with a specific kernel size and depth. A bias term, b_m, is added to produce an H_m feature map. This process follows Equation 1, where $H_m = W_m\ H_m - 1 + b_m$ for $0<m\leq M$, while $H_0 = X$ represents the input image. Here, W_m denotes the convolution kernel at layer m, H_{m-1} is the previous feature map, and b_m is the bias term. This formulation explains how hierarchical feature extraction is performed in the encoder of the VGG-19 U-Net architecture. The kernel size, padding, and stride determine the output size of each layer. The spatial dimension of the output feature map is computed using Equation 2, which defines the relationship between input size, kernel size, padding, and stride to ensure consistent dimensional transitions across layers. Each convolution result is then passed through a nonlinear activation function, such as ReLU or Sigmoid, to add complexity to the feature representation. As described in Equation 3, the activation function $g(\cdot)$ is applied to the linear transformation output to introduce non-linearity, enabling the network to learn complex feature patterns. The ReLU and Sigmoid functions defined in Equation 4 are used to prevent vanishing gradients and to generate probabilistic pixel outputs for segmentation tasks. Batch Normalization is applied after activation to stabilize the data distribution during training, and pooling is used to reduce dimensions, preserving the most significant features. Equation 5 integrates Batch Normalization (BN) into the convolution-activation pipeline to normalize intermediate feature distributions, while Equation 6 formalizes the Max Pooling operation that reduces spatial resolution while retaining dominant features. These operations improve training stability and computational efficiency.

The features extracted by the encoder are passed to the bottleneck layer, which serves as a connector and stores important information before the decoder reconstructs. The bottleneck (bridge) layer serves as the transition between encoder and decoder by maintaining high-level semantic representations before up-sampling. In the decoder, a deconvolution or up-sampling operation is performed to restore the spatial resolution to its original size, enabling precise pixel-level segmentation. This reconstruction stage reverses the dimensionality reduction process performed in the encoder, ensuring accurate localization in semantic segmentation.

The model training process is controlled by several hyperparameters, including filter size, stride, padding, number of neurons, kernel initialization (e.g., He Normal), and *L2* regularization to prevent overfitting. The L2 regularization term, defined in Equation 7 as $L2= \lambda . W_m^2$, penalizes large weight magnitudes to reduce overfitting and improve model generalization. Kernel bias is added to help the model learn feature relationships better and to avoid unwanted negative activations [34]. Equation 8 explains how kernel weights and bias contribute to each convolution output through weighted summation across the receptive field, ensuring that each feature map reflects both spatial correlation and bias adjustment.

The model's performance is evaluated quantitatively by measuring inference time and model size, which helps identify areas for further optimization. It is essential to carefully evaluate and validate the model after each modification to ensure that its performance remains sufficient. Optimization should always aim to improve efficiency without sacrificing segmentation quality.

The loss function $L(\Theta)$ is used to solve multi-class classification problems by applying categorical Cross-Entropy (CE) [35]. As formulated in Equation 9, the CE loss computes the negative log-likelihood between the actual probability distribution $p(k)$ and the predicted probability $\acute{p}(k)$, measuring the degree of misclassification at the pixel level. This function measures how far the model's predictions deviate from the expected output, or in other words, the degree of misclassification. In the CE function, the negative logarithm is used to calculate the probability of each prediction based on the actual probability $p(k)$. When the model is applied to regression problems or continuous value prediction, the loss function typically used is the Mean Squared Error (MSE), which calculates the squared difference between the predicted value $\hat{y}k$ and the actual value yk. The MSE formulation in Equation 9 evaluates the average squared prediction error, making it suitable for continuous-value estimation tasks.

These modifications include additional hyperparameters to strengthen weights, thereby improving its performance. Features in the segmentation process were also enhanced to increase accuracy and reduce processing time. Modifications include activation functions, additional Batch Normalization (BN), Kernel Initializer (KI) with He-normal for Encoder, Bridge, and Decoder, Kernel Regularization (KR) with $L2$ = 0.00005, and Bias Regularization (BR) with $L2 = 10^{-4}$ for the same layers.

Table 1. Summary of Formulas and Parameter Descriptions in VGG-19 U-Net Architecture for Semantic Segmentation

Number	Components	Hyperparameter Explanation	Formula/Definition
1	Encoder Layer [36]	H_m: output of layer m; X: input image; W_m: convolutional filter/kernel with dimensions (w_x(m), w_y(m), w_z(m), $w_d(m)$); b_m: bias vector for layer m, m: layer index.	$H_m = \begin{cases} X & if\ m = 0 \\ W_m H_{m-1} + b_m & if\ 0 < m \leq M \end{cases}$ (1)
2	Output size Feature Map [37]	$h_x^{(m)}$, $h_y^{(m)}$: spatial dimensions of output in x and y at layer m; $w_x^{(m)}$, $w_y^{(m)}$: kernel size; $p_x^{(m)}$, $p_y^{(m)}$: padding; $st_x^{(m)}$, $st_y^{(m)}$: stride.	$h_x^{(m)} = \left[\frac{h_x^{(m-1)} - w_x^{(m)} + 2p_x^{(m)}}{st_x^{(m)}}\right] + 1;$ $h_y^{(m)} = \left[\frac{h_y^{(m-1)} - w_y^{(m)} + 2p_y^{(m)}}{st_y^{(m)}}\right] + 1$ (2)
3	Nonlinear Activation [37]	$g(\cdot)$: element-wise activation function such as ReLU or Sigmoid.	$H_m = \begin{cases} X & if\ m = 0 \\ g(W_m H_{m-1} + b_m) & if\ 0 < m \leq M \end{cases}$ (3)
4	Activation Functions [37]	ReLU introduces nonlinearity and prevents vanishing gradients; Sigmoid produces probabilistic output.	$ReLU(x) = \begin{cases} x, & if\ x \geq 0 \\ 0, & if\ x < 0 \end{cases}$; $\text{Sigmoid}(x) = \begin{cases} 1, & if\ x \geq 0 \\ \frac{1}{1+e^{-x}}, & if\ x < 0 \end{cases}$ (4)
5	Batch Normalization [37]	BN normalizes layer output to maintain stable distributions and accelerate training convergence.	$H_m = \begin{cases} X & if\ m = 0 \\ g(BN(W_m H_{m-1} + b_m)) & if\ 0 < m \leq M \end{cases}$ (5)
6	Max Pooling [37]	Pooling reduces spatial dimensions by selecting the maximum value in a window.	$H_m = \begin{cases} X & if\ m = 0 \\ pool(g(BN(W_m H_{m-1} + b_m))) & if\ 0 < m \leq M \end{cases}$ (6)
7	Bridge (Bottleneck) [36]	Connects encoder and decoder; retains key features for reconstruction.	—
8	Decoder Layer [36]	Performs deconvolution/up-sampling to restore original spatial dimensions and reconstruct segmentation output.	—
9	L2 Regularization [33][36]	λ : regularization coefficient; prevents overfitting by reducing filter weights (W_m).	$L2 = \lambda . W_m^2$ (7)
10	Kernel Bias [37]	b_i: bias at kernel index i ; $w_{j,k}$: kernel weights at position (j,k); $x_{i+j-1,k}$: input at corresponding position.	$b_i = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{j,k} \cdot x_{i+j-1,k} + b_i$ (8)
11	Loss Function (Cross-Entropy dan MSE) [38]	$p(k)$: actual probability; $\acute{p}(k)$: predicted probability; yk, $\hat{y}k$: actual and predicted values.	$\text{CE}(\Theta) = -\sum_{k=1}^{K} (\log \acute{p}(k)) . p(k)$ $\text{MSE}(\Theta) = \frac{1}{p} \sum_{k=1}^{P} (\hat{y}_k - y_k)^2$ (9)
12	Optimizer (Adam) [34]	θ_t : model parameter at iteration t ; α : learning rate = 1e^{-3}; $\acute{m}_t$, $\tilde{V}_t$: first and second moment estimates; ϵ : small constant for numerical stability.	$\acute{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$; $\tilde{V}_t \leftarrow \frac{v_t}{1-\beta_2^t} \theta_{t+1} \leftarrow \theta_t - \frac{\alpha}{\sqrt{\tilde{V}_t + \epsilon}} . \acute{m}_t$ (10)

This study uses weight and bias optimization with the Adam function, which combines momentum-based learning techniques (first moment) and RMSProp (second moment). The parameter update mechanism follows Equation 10, where model parameters θ_{t+1} are updated using bias-corrected first- and second-moment estimates ($\acute{m}_t$, $\tilde{V}_t$) with learning rate α, ensuring adaptive gradient adjustment during training. This optimizer is

configured with parameters $\beta_1 = 0.9$, $\beta_2 = 0.9999$, and $\epsilon = 10^{-8}$ to allow for adaptive learning rate adjustment and accelerate convergence during training [34]. To ensure reproducibility, this study reports the key training configuration: a batch size of 42 and the default Adam optimizer and learning rate from TensorFlow/Keras.

After training the U-Net network on the dataset, a model is created and saved for future use. This model enables inference (prediction) for mapping image areas or automating the process of segmenting plantation image areas. The saved model adapts to the type of scenario used for learning, including data size, resolution, and class. The dataset learning saves the model after a certain number of epochs are reached. The model is saved in ".hdf5" format and is developed using Python.

Model performance is evaluated using a confusion matrix to assess accuracy, precision, recall, and the average Intersection over Union (IoU) across all categories. As defined in Equations 11-14, these metrics are derived from the confusion matrix components as True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) for each class *k* across *N* classes. Overall Accuracy (OA) measures the proportion of correctly classified samples over all predictions. Precision (P) evaluates the reliability of positive predictions, while Recall (R) reflects the model's sensitivity in detecting actual positives. The Mean IoU (MIoU), computed as the average of $TP_k / (TP_k+FP_k+FN_k)$ across all classes, provides a pixel-level overlap evaluation that is particularly relevant for semantic [39], [40].

The loss function measures how well the model predicts the expected output from the input data. In multi-class classification, the CE function is used to estimate the probability of misclassifying the correct class during training [3].

$$\mathrm{OA} = \frac{\sum_{k=1}^{N} TP_k}{\sum_{k=1}^{N} TP_k+FP_k+FN_k+TN_k} \quad (11)$$

$$\mathrm{P} = \frac{\sum_{k=1}^{N} TP_k}{\sum_{k=1}^{N} TP_k+FP_k} \quad (12)$$

$$\mathrm{R} = \frac{\sum_{k=1}^{N} TP_k}{\sum_{k=1}^{N} TP_k+FN_k} \quad (13)$$

$$\mathrm{MIoU} = \frac{1}{N}\left(\frac{\sum_{k=1}^{N} TP_k}{\sum_{k=1}^{N} TP_k+FP_k+FN_k}\right) \quad (14)$$

3. Results and Discussions

Experiments were conducted on a primary dataset with previously annotated labeled image targets. Dataset learning was performed using baseline U-Net VGG-16 and U-Net VGG-19 architectures and compared with the proposed models that extend the U-Net VGG-19 architecture. This experimental design enables a controlled analysis of how architectural refinements, rather than primary structural replacements, influence segmentation performance under data-limited conditions. Testing was conducted using both 3-class and 4-class masks across two image resolutions to evaluate robustness and scalability.

The Hold-Out Validation method was applied by dividing the dataset into 75% training data and 25% testing data. This strategy was selected due to the limited dataset size, which is constrained by manual mask annotation complexity and computational capacity. Under such constraints, model robustness and generalization capability are more important than absolute accuracy, justifying the use of MIoU and confusion-matrix-based evaluation.

3.1 Four-class dataset training

Table 2 presents the models that will be used for network training on the dataset, comprising four models: two baseline models and two modified U-Net VGG-19 models, along with their respective network parameter sizes. Table 2 summarizes the model architectures and parameter sizes. It can be observed that the proposed models (M7 and M8) significantly reduce parameter complexity compared to baseline models, while maintaining or improving segmentation performance. This reduction is particularly important for UAV-based systems, where computational efficiency directly impacts deployment feasibility.

Table 2. Model Parameter Comparation

Model	Type	Parameter
M1	VGG-16 Keras	23,751,556
M2	VGG-19 Keras	29,061,252
M3	Segnet	38,108
M4	FCN 8s	1,332,420
M5	ResNet 101	19,757,024
M6	AtrousDenseUDeconvNet	51,599,623
M7 (Purposed)	VGG-19 Modification with Parameter Tuning	9,346,148
M8 (Purposed)	VGG-19 Modification with Parameter Tuning +Residue	10,525,708

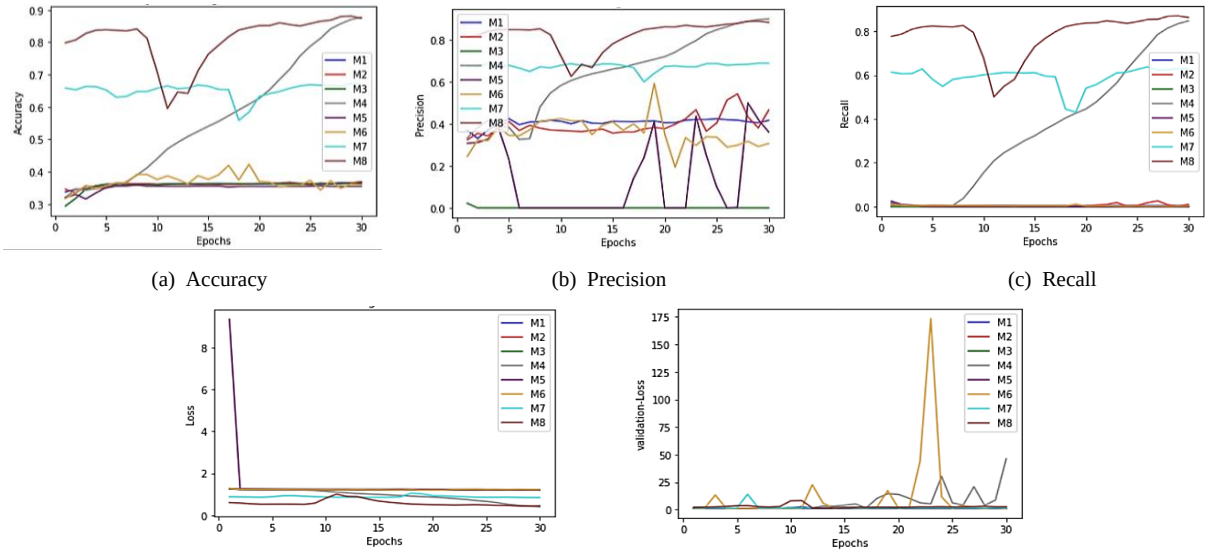
Table 3 shows the comparative results of 8 models from Table 2. Testing was conducted to see the capability of the proposed model compared to the state-of-the-art U-Net method using 30 epochs. As shown in Table 3, most baseline models (M1, M2, M3, M5, and M6) exhibit poor performance in accuracy, precision, and recall. Standard encoder–decoder architectures struggle to separate spectrally similar land-cover classes in high-resolution UAV imagery. In contrast, M4 and the proposed model M8 perform strongly. They achieve the highest recall and competitive accuracy while training significantly faster.

Table 3. Comparison test results of U-Net with 512x512 input size

Model	Accuracy (%)	Precision (%)	Recall (%)	Times (second)
M1	36,79	41,66	00,30	8230,80
M2	36,79	46,62	00,99	9233,55
M3	36,31	0.00	0.00	1704,89
M4	87,68	89,97	84,72	6032,39
M5	35,45	36,11	6.61e-07	8627,14
M6	35,90	30,64	00,19	7002,25
M7	67,06	68,88	62,78	3189,43
M8	87,40	88,32	86,32	3656,79
M7 : M2	+30,27	+22,26	+61,79	-6044,12
M8 : M2	+50,61	+41,7	+85,33	-5576,76

The baseline model M2, compared with the proposed model M7, shows an increase in accuracy of 30.37%, precision of 22.26%, and recall of 61.79%, with training time reduced by 6044.12 seconds. The M7 model demonstrates a notable improvement over M2, confirming that hyperparameter optimization enhances learning efficiency. Likewise, the M8 model outperforms the baseline and prior models in terms of differences in accuracy, precision, and recall. The M8 model consistently outperforms M7 by achieving higher recall and more stable training behavior. This improvement is due to residual connections in the decoder. These connections facilitate gradient propagation and preserve spatial information during up-sampling. As a result, M8 generalizes better, especially in identifying minority classes.

Figure 5 is a comparison graph of 200 data training results with a size of 512x512, with Epoch 30.



(a) Accuracy (b) Precision (c) Recall

(d) Loss (e) Validation Loss

Figure 5. Comparison graph dataset testing resize 512x512

Figure 5 illustrates the training curves. M8 consistently outperformed other models, achieving accuracy above 90% with stable precision and recall throughout training. M4 remained competitive, with accuracy and recall approaching M8. M7 showed moderate gains but remained less stable.

Loss and validation-loss curves indicated that M8 was the most stable. M8 is stable across accuracy, precision, and recall curves. This suggests that residual connections reduce overfitting and improve convergence behavior. In contrast, M4 exhibits occasional fluctuations, indicating weaker generalization, while M7 shows moderate gains but less stability.

Table 4 shows the results of class distribution from the segmentation process based on MIoU and scores per class on eight U-Net models.

Table 4 reveals a critical insight: high overall accuracy does not necessarily correspond to effective semantic segmentation performance. Although M7 achieves accuracy above 67%, its MIoU remains low (0.091), indicating that predictions are heavily biased toward the majority class (Class 1). This highlights a common pitfall in segmentation evaluation, where accuracy masks poor minority-class recognition.

In contrast, M8 improves MIoU to 0.132 by distributing predictions more evenly across Classes 1, 2, and 4. This improvement shows that decoder-level residual connections enhance feature reuse and boundary reconstruction. These effects improve discrimination between minority classes. However, the persistent low performance for Class 3 indicates that spectral similarity between water channels and bare land remains a challenging problem.

Table 4. Comparison of training results of test data class distribution against 512x512 resized

Model	MIoU	Class 1	Class 2	Class 3	Class4
M1	0.094	0.360	0.016	0,0	0,0
M2	0.094	0.359	0.016	0,0	0.002
M3	0.091	0.363	0,0	0,0	5.53e-06
M4	0.095	0.076	0.307	0,0	0,0
M4	0.087	0.060	0.287	0,0	0,0
M5	0.091	0.363	0,0	0,0	0.001
M6	0.096	0.362	0.00	0,0	0.021
M7	0.091	0.363	0,0	0,0	0,0
M8	0.132	0.327	0.176	0,0	0.026

The results of class mapping from the M7 and M8 proposal models from the distribution of the 50 data test class can be seen in the heatmap Figure 6. Figure 6 shows the confusion matrices for M7 and M8. In M7, frequent misclassification occurs between Class 2 (non-oil palm vegetation) and Class 1 (land/road), and between Class 3 (ditch/water) and Class 1. This indicates a strong bias toward the dominant land class. M7 fails to capture subtle spectral and textural differences. Furthermore, Class 4 (oil palm vegetation) is often predicted as either Class 1 or Class 2,

suggesting that M7 has difficulty distinguishing oil palm vegetation from surrounding vegetation.

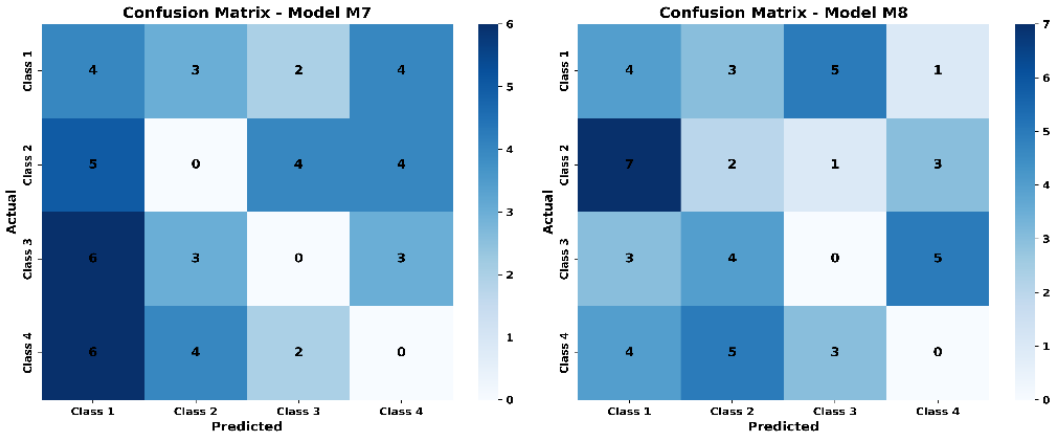


Figure 6. Heatmap 4-class test results 50 data with 512x512 resize

By contrast, M8 significantly reduces false negatives for Class 4 (oil palm vegetation). This improvement is critical for plantation monitoring applications, as misclassification of oil palm areas directly affects productivity estimation and land management decisions. Although some confusion persists, M8 demonstrates a more balanced and reliable segmentation behavior.

The next experiment used a dataset of 400 data, with a resize of 256x256 with 30 epochs. Table 5 shows that the overall model performance varies significantly in terms of accuracy, precision, recall, and training time efficiency.

Table 5. Comparison test results of U-Net resize 256x256

Model	Accuracy %)	Precision (%)	Recall %)	Times (second)
M1	59.08	63.14	51.97	5679.08
M2	49.49	51.21	29.67	6226.42
M3	49.26	47.76	12.06	1434.35
M4	94.73	94.73	94.12	3892.33
M5	49.26	49.26	16.71	5478.30
M6	77.18	79.39	79.39	5728.67
M7	86.62	86.62	84.97	2481.98
M8	86.44	87.72	85.02	2831.70
M7 : M2	+37.13	+35.41	+55.3	-3744.44
M8 : M2	+36.95	+36.51	+55.35	-3394.72

The M4 model stands out as the highest-performing, most accurate model across all classes. Following the M4 model, the proposed M7 and M8 models also demonstrate excellent performance, as evidenced by high accuracy, precision, and recall results, above 84%. The main advantage of the M7 and M8 over the M4 is their training time efficiency. While the M4 achieves the highest accuracy, the proposed M7 and M8 models offer a better balance between performance and computational efficiency. This makes them more suitable for developing real-time or large-scale UAV deployment scenarios.

Compared to the baseline M2 model, the M7 model shows an increase in accuracy of +37.13%, precision of +35.41%, and recall of +55.3%, with a faster training time of 3744.44 seconds. Meanwhile, the M8 model also outperforms the M2 in all metrics by a similar margin, demonstrating the superiority of the modified architecture.

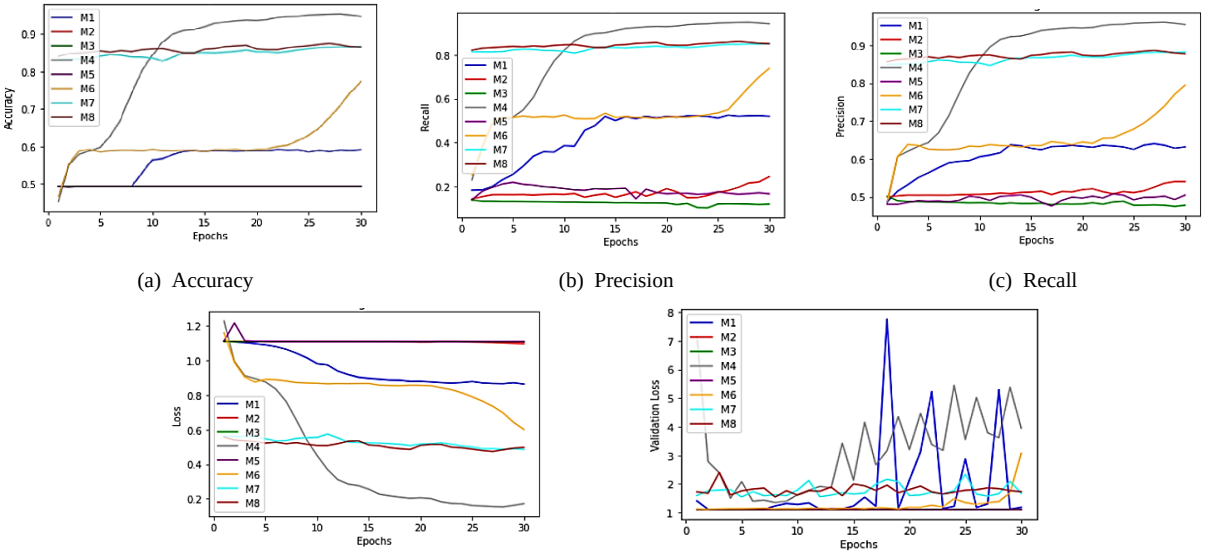
Further experiments were conducted using a dataset of 400 samples with an input size of 256×256 over 30 epochs. The learning curves are presented in Figure 7. Model M4 consistently outperforms the other models across accuracy, precision, and recall, while also achieving the lowest training loss and stable validation behavior, indicating strong convergence and generalization capability.

Model M8 demonstrates substantial improvement and emerges as a strong alternative to M4, exhibiting relatively stable learning dynamics. Model M6 remains competitive with moderate and stable performance, whereas M7 shows gradual but consistent improvement. Overall, the results suggest that architectures incorporating residual connections contribute to enhanced representational stability and more reliable convergence.

Table 6 shows a comparison of the class distribution results from the MIoU-based segmentation process and the scores per class on the test data for eight U-Net models with 256x256 image sizes.

Table 6 shows that M4 model performed best and was optimal in recognizing all classes, while the M7 and M8 excelled in most classes, performing well but with distinct weaknesses. Meanwhile, the M1, M2, M3, and M5 models performed poorly and were considered unsuitable for use without further architectural improvements or data balancing on the research data.

M8 achieves the highest MIoU (0.302), slightly outperforming M7. Unlike M7, which nearly fails to recognize Class 3, M8 demonstrates improved sensitivity toward minority classes, confirming its superior generalization capability.



(a) Accuracy (b) Precision (c) Recall

(d) Loss (e) Validation Loss

Figure 7. Comparison graph dataset testing resize 256x256

Table 6. Comparison of training results of test data class distribution against 256x256 resized images

Model	MIoU	Class 1	Class 2	Class 3	Class 4
M1	0.124	0.0	0.0	0.0	1.0e-6
M2	0.123	0.0	0.0	0.0	0.123
M3	0.123	0.0	0.0	0.0	0.188
M4	0.311	0.463	0.177	0.177	0.537
M5	0.123	0.0	0.0	0.0	0.493
M6	0.085	0.106	0.208	0.0	0.027
M7	0.300	0.473	0.473	4.9e-5	0.621
M8	0.302	0.319	0.137	0.137	0.563

The results of class mapping from the M7 and M8 proposal models from the distribution of the 100 data resize 256x256 test class can be seen in the heatmap Figure 8.

M7 performed poorly in Class 3, with predictions approaching zero, indicating that the model almost completely failed to recognize its features. This suggests that M7 struggled to distinguish water areas from visually similar land areas, leading to frequent misclassifications as Class 1.

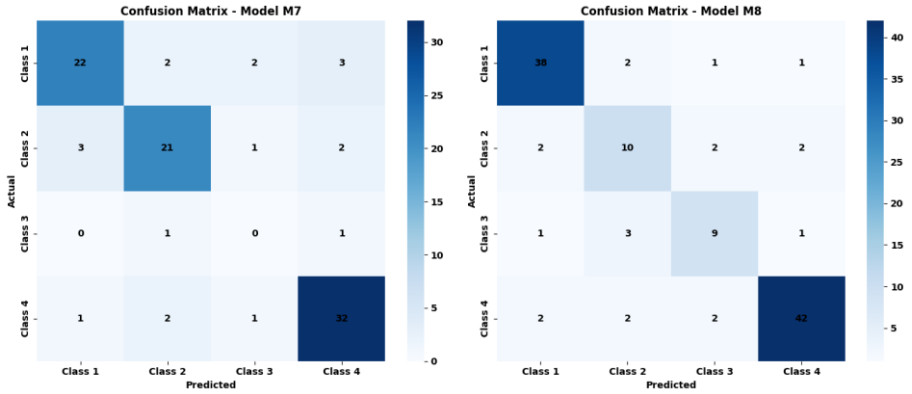


Figure 8. Heatmap 4-class test results 100 data 256x256 resize

In contrast, M8 showed a more balanced distribution across all four classes, improving recognition of Classes 2 and 3 while maintaining high performance in Classes 1 and 4. Although M8 demonstrated its ability to capture Class 2 and Class 3, this has yielded significant improvements in land cover mapping accuracy, as it better generalizes across minority classes.

Figure 8 further highlights that M8 produced a more balanced class distribution, particularly improving recognition of waterways (Class 3), which are often confused with land due to spectral similarity. This suggests that architectural refinements do not eliminate complexity, and that the primary contributor is improved performance over the baseline model.

3.2 Three-class network training with 512 x 512 resize

To evaluate the performance of the proposed models in multi-class semantic segmentation of oil palm areas, a comparison was conducted against baseline architectures. Table 7 presents the training results of four models using a dataset consisting of 3 classes, with input images resized to 512×512 pixels and trained over 50 epochs. This experiment used three classes (1:

land/road/ditch/water puddle, 2: non-oil palm vegetation, 3: oil palm vegetation).

The proposed models M7 and M8, which are modified versions of the VGG-19 architecture with hyperparameter tuning and the addition of a residual function, demonstrate significantly superior performance. M7 achieves 85.16% accuracy, 86.63% precision, and 83.44% recall, while M8 records further improvements with 88.01% accuracy, 88.84% precision, and 87.12% recall. Furthermore, both models also demonstrate better training time efficiency, with 848.10 seconds for M7 and 974.52 seconds for M8.

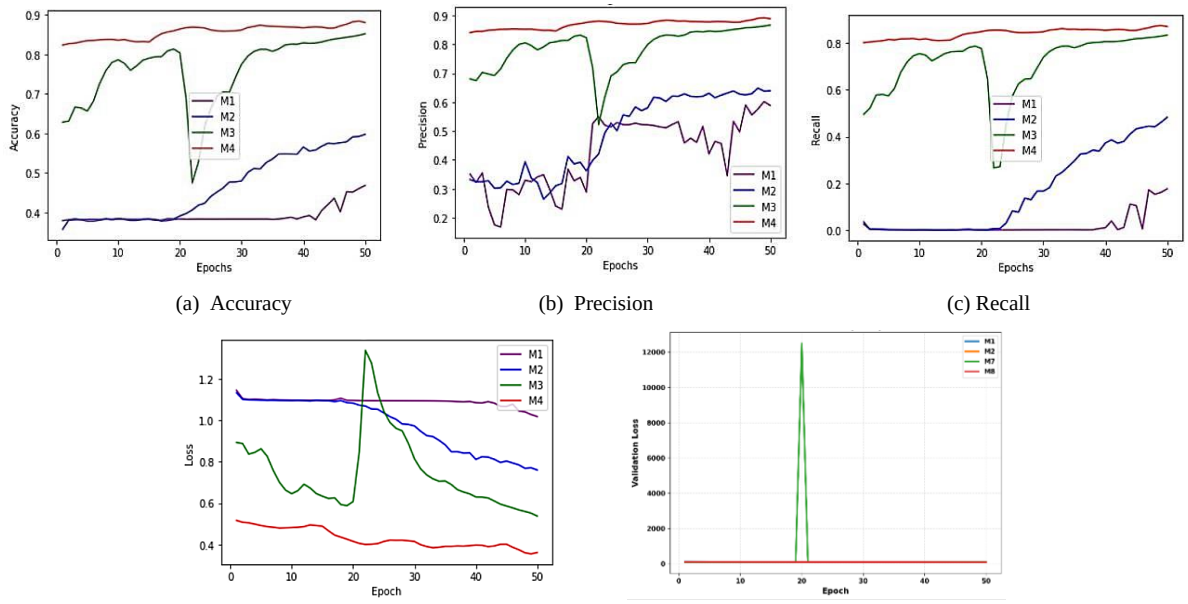
A direct comparison shows that M7 and M8 significantly outperform M2 in both accuracy (+25–28%), precision (+22–25%), and recall (+35–39%), with significantly shorter training times (1200–1300 seconds faster).

Table 7. Comparison results of network training with 3-class 512x512 resize

Model	Accuracy (%)	Precision (%)	Recall (%)	Times (second)
M1	46,84	58,83	17,63	1947.28
M2	59,76	63,89	48,30	2207.66
M7	85,16	86,63	83,44	848,10
M8	88,01	88,84	87,12	974.52
M7 : M2	+ 25,4	+ 22,74	+ 35,14	-1359,56
M8 : M2	+ 28,25	+ 24,95	+ 38,82	-1233,14

This M8 was the most stable, accurate, and efficient model throughout training, achieving high, stable accuracy, precision, and recall. In this 3-class scenario, both M7 and M8 significantly outperform baseline models. The reduction in class complexity allows the architectural advantages of residual connections to be more clearly expressed, resulting in stable and high segmentation accuracy.

Figure 9 illustrates the learning results from Table 7.



(a) Accuracy (b) Precision (c) Recall

(d) Loss (e) Validation-Loss

Figure 9. Graph comparation model baseline with proposed for 3-class resize 512x512

The evaluation graph shows that the M8 model has the best and most stable performance across all metrics, with high accuracy, precision, and recall (around 0.88) and low and consistent loss and validation loss for up to 50 epochs. The M7 model experienced overfitting with sharp spikes in epochs 15 and 22, but managed to recover and stabilize above 0.83–0.85. Meanwhile, M1 and M2 showed low and stagnant performance, with minimal improvement and high validation loss, reflecting an inability to generalize and slow convergence. M8 consistently demonstrates the best balance between accuracy, precision, recall, and training stability. Its low and stable loss curves indicate strong convergence and reduced overfitting, as confirmed in Figure 9.

Table 8 shows the results of the comparative test of the baseline method on the distribution of 3 classes based on MIoU and scores per class.

Table 8. Comparison classification results of test data for 3 classes

Model	MIoU	Class 1	Class 2	Class 3
M1	0.150	0.365	0.073	0.0116
M2	0.102	0.00673	0.299	1.08e-4
M7	0.168	0.169	0.261	0.072
M8	0.178	0.104	0.172	0.252

Model M8 performed best with the highest MIoU of 0.178, and a relatively balanced IoU distribution across all classes, particularly Class 3 (oil palm vegetation). Model M7 followed with an MIoU of 0.168, indicating primary strength in Class 2 (non-oil palm vegetation). Model M1 performed best in Class 1 (land/road/ditch), but was very low in the other classes, while the baseline model M2 performed the lowest overall, with dominance only in Class 2 and almost zero in the other classes. MIoU analysis in Table 8 further validates this result. M8 achieves the most balanced IoU distribution across all classes, particularly for oil palm vegetation

(Class 3), which is the primary target class in plantation monitoring.

The heatmap results of the class distribution of Models M7 and M8 against the test data sample are shown in Figure 10.

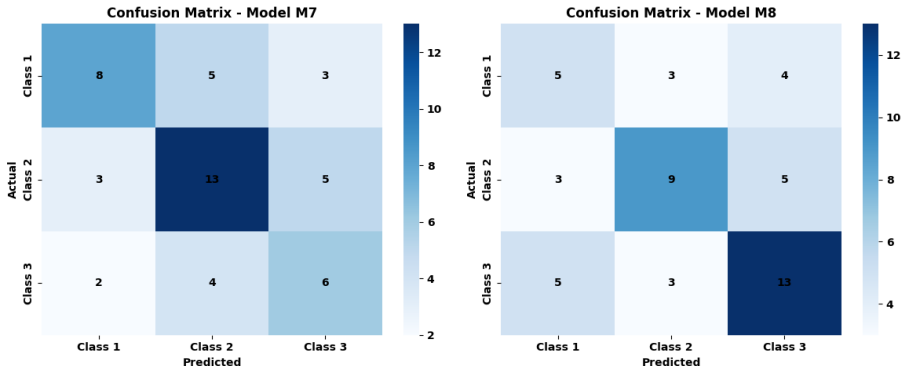


Figure 10. Heatmap 3-class test results 50 data resize 512x512

The heatmap results for the three-class dataset show that Model M7 performs better in recognizing Class 2, while Model M8 is more accurate for Class 3. Both models still face challenges distinguishing Class 1 from the others, but M7 produces a slightly more balanced prediction distribution. The choice of the best model depends on which class is prioritized in real-world applications. Figure 10 shows that while both M7 and M8 still face challenges in distinguishing Class 1, M8 exhibits superior discrimination of oil palm vegetation, reinforcing its suitability for real-world deployment.

Overall, the experimental results demonstrate that architectural refinement at the decoder level plays a more critical role than hyperparameter tuning alone in multi-class semantic segmentation of UAV imagery. Although hyperparameter tuning in M7 improves training efficiency and overall accuracy, the model remains sensitive to class imbalance, as indicated by low MIoU values and uneven class distribution in the confusion matrix. This suggests that parameter optimization alone is insufficient to address the challenges of spectral similarity and minority-class representation in oil palm plantation imagery.

In contrast, integrating residual connections into the decoder layer (M8) consistently improves segmentation stability and class balance across different resolutions and class configurations. Residual connections enhance gradient propagation and preserve spatial features during up-sampling, thereby improving boundary reconstruction and reducing bias toward majority classes. This effect is particularly evident in improved recognition of oil palm vegetation and drainage classes, which are often misclassified as surrounding land cover. Therefore, the superior performance of M8 is attributable not only to higher accuracy but also to its enhanced generalization capability and balanced class-wise predictions, which are essential for practical plantation monitoring applications.

Despite demonstrating good segmentation performance, this study is limited by the relatively small dataset, primarily due to the complexity of UAV data acquisition and manual annotation. Consequently, the trained model may not fully capture the variability of oil palm plantation structure across different locations, growth stages, or seasonal conditions. This limitation underscores the importance of architectural robustness and balanced class learning, as evidenced by the M8 model's superior performance under limited data conditions.

4. Conclusions

The VGG-19-based U-Net was enhanced through hyperparameter tuning (M7) and decoder-level residual integration (M8). While hyperparameter tuning improved training efficiency and convergence, residual connections significantly strengthened segmentation robustness and class balance. As a result, the modified architectures, particularly M8, achieved superior multi-class segmentation performance on UAV oil palm imagery, with higher accuracy, precision, recall, and consistently improved MIoU compared to baseline models.

For four-class segmentation, M8 achieved strong performance at both resolutions, reaching 87.40% accuracy, 88.32% precision, and 86.32% recall (MIoU = 0.132) at 512×512, and 86.44% accuracy, 87.72% precision, and 85.02% recall at 256×256. These results show that decoder-level residual connections effectively reduce majority-class dominance while maintaining stable accuracy and recall, demonstrating robustness and adaptability to lower spatial resolutions.

At 512×512 resolution for three-class segmentation, M8 achieved the highest accuracy (88.01%) with high precision (88.84%), recall (87.12%), and the most balanced MIoU across classes, demonstrating strong generalization under limited data. Although M7 showed competitive performance and faster training, it remained prone to minority-class misclassification and validation loss fluctuations, indicating potential overfitting.

Decoder-level residual connections significantly

enhance feature extraction and spatial representation in UAV imagery, making the M8 architecture a reliable solution for balanced multi-class semantic segmentation in oil palm monitoring. While M7 is suitable for computational efficiency, M8 provides a better overall balance of accuracy, stability, and generalization.

Despite these promising results, the limited dataset size remains a constraint for large-scale deployment and may restrict generalization to environments with different spatial or illumination conditions. Nevertheless, the robustness of M8 across multiple experimental settings underscores the effectiveness of decoder-level residual refinement.

Future work will focus on expanding the dataset through multi-location and multi-season UAV acquisitions, as well as exploring advanced techniques such as adaptive loss functions, targeted data augmentation, and cross-dataset validation. These efforts aim to further enhance generalization and support scalable, reliable automated monitoring of oil palm plantations.

Acknowledgements

The authors acknowledge the financial support from the Directorate of Research, Technology, and Community Service, Directorate General of Higher Education, Research and Technology, Ministry of Education, under the Doctoral Dissertation Research Funding Scheme (PDD) No. 0557/E5.5/AL.04/2023 (2023). Additional support was provided by the Department of Computer Science and Electronics through laboratory facilities and by STMIK Palangkaraya. The authors declare no conflict of interest.

References

[1] R.A. Perdana, A.M. Arimurthy and Risnandar. "Remote Sensing Scene Classification using ConvNeXt-Tiny Model with Attention Mechanism and Label Smoothing," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* , vol. 5 number 2, pp. 389 – 400, 2024, doi: 10.29207/resti.v8i3.5731.

[2] R. Li, L. Wang, C. Zhang, C. Duan, S. Zheng, "A2-fpn for semantic segmentation of fine-resolution remotely sensed images," *International Journal of Remote Sensing*, 43, 1131–1155, 2022, doi : 10.1109/LGRS.2021.3052886.

[3] M. Widyaningsih, T.K. Priyambodo, M.E. Wibowo, M. Kamal, "Optimization contrast enhancement and noise reduction for semantic segmentation of oil palm aerial imagery," *International Journal of Intelligent Engineering and Systems*,16, 597–609, 2023, doi : 10.22266/ijies2023.0228.51.

[4] Z. Su, W. Li, Z. Ma, R. Gao, "An improved u-net method for the semantic segmentation of remote sensing images," *Applied Intelligence*, 52, 3276–3288, 2021, doi: 10.1007/s10489-021-02542-9.

[5] B. Xu, W. Wang, G. Falzon, P.W.H. Kwan, L. Guo, Z. Sun, and C. Li, "Livestock classification and counting in quadcopter aerial images using Mask R-CNN," *International Journal of Remote Sensing*, 41, 8121 – 8142, 2020, doi: 10.1080/01431161.2020.1734245.

[6] M. Lan, Y. Zhang, L. Zhang, and B Du, "Global context based automatic road segmentation via dilated convolutional neural network," *Information Sciences*, 535, 1–254, 2020, doi : 10.1016/j.ins.2020.05.062.

[7] A. Tashk, J. Herp, T. Bjørsum-Meyer, A. Koulaouzidis, and E.S. Nadimi, "AID-U-Net: An Innovative Deep Convolutional Architecture for Semantic Segmentation of Biomedical Images". *Diagnostics*, *12*, 2022, doi: 10.3390/diagnostics12122952.

[8] N. Islam, M.F. Hossain, and M. Hossain, "Semantic Segmentation in Satellite Imagery: An Attentive U - Net Approach," *2024 2nd International Conference on Information and Communication Technology (ICICT)*, 259-263. 2024, doi : 10.1109/ICICT64387.2024.10839725.

[9] M. Liang, and X. Wang, "A bidirectional semantic segmentation method for remote sensing image based on super-resolution and domain adaptation," *International Journal of Remote Sensing*, 44, 666–689, 2023, doi : 10.1080/01431161.2023.2169593.

[10] B. Cui, X. Chen, and Y. Lu, "Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection," *IEEE Access*, 8, 116744–116755, 2020, doi : 10.1109/ACCESS.2020.3003914.

[11] H. Toba, H. Bunyamin, J.E. Widyaya, C. Wibisono, and L.S. Haryadi, "Masking preprocessing in transfer learning for damage building detection," *IAES International Journal of Artificial Intelligence* (IJ-AI), 2022, doi: 10.11591/ijai.v12.i2.pp552-559.

[12] S. Reza, O.B. Amin, and M.M. Hashem, "TransResUNet: Improving U-Net Architecture for Robust Lungs Segmentation in Chest X-rays", *IEEE Region 10 Symposium (TENSYMP)*, 1592-1595, 2020, doi : 10.1109/ TENSYMP50017. 2020.9230835.

[13] M.I. Ahmed, M. Foysal, M.D. Chaity, and A.B. Hossain, "DeepRoadNet: A deep residual based segmentation network for road map detection from remote aerial image", *IET Image Process.*, *18*, 265-279, 2020, doi : 10.1049/ipr2.12948.

[14] N. Ali, A. Z. Ijaz, R. H. Ali, Z. Ul Abideen and A. Bais, "Scene Parsing Using Fully Convolutional Network for Semantic Segmentation," *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Regina, SK, Canada, 2023, pp. 180-185, 2023, doi : 10.1109/CCECE58730.2023.10288934.

[15] S. Landgraf, M. Hillemann, K. Wursthorn, and M. Ulrich, "U-CE:Uncertainty-aware Cross-Entropy for Semantic Segmentation". *ArXiv*, 2023, doi : 10.48550/arXiv.2307.09947.

[16] S.I. Hussain and E. Toscano, "An Extensive Investigation into the Use of Machine Learning Tools and Deep Neural Networks for the Recognition of Skin Cancer: Challenges, Future Directions, and a Comprehensive Review," *Symmetry*, 16, 366. 2024, doi : 10.3390/sym16030366.

[17] F.K. Alarfaj, I. Malik, H.U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-art Machine Learning and Deep Learning Algorithms," *IEEE Access*, PP, 1-1, 2022, doi: 10.1109/ACCESS.2022.3166891.

[18] E. Moliner, L.S. Romero, and V. Vilaplana, "Weakly supervised semantic segmentation for remote sensing hyperspectral imaging", in: *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), p. 2273–2277. 2020, doi :10.1109/ICASSP40776.2020.9053384.

[19] X. Yang, R.B. Bist, S. Subedi, L. Chai, "A deep learning method for monitoring spatial distribution of cage-free hens," *Artificial Intelligence in Agriculture*, 8, 20–29, 2023, doi : 10.1016/j.aiia.2023.03.003.

[20] N. Shabrina, R.A. Lika, and S. Indarti, "Deep learning models for automatic identification of plant-parasitic nematode," *Artificial Intelligence in Agriculture*, 7, 1–12, 2023, doi : 10.1016/j.aiia.2022.12.002.

[21] S. Xu, S. Cheng, X. Min, N. Pan, dan H. Hu, "Left Ventricle Segmentation Based on a Dilated Dense Convolutional Networks," *IEEE Access*, 8, 214087-214097, 2020, doi : 10.1109/ACCESS.2020.3040888.

[22] S. Li, Y. Guo, Z. Pang, W. Song, A. Hao, B. Xia, and H. Qin, "Automatic Dental Plaque Segmentation Based on Local-to-Global Features Fused Self-Attention Network," *IEEE*

Journal of Biomedical and Health Informatics, *26*, 2240-2251, 2022, doi: 10.1109/JBHI.2022.3141773.

[23] C. Ma and Z. Wang, "Semi-Mamba-UNet: Pixel-Level Contrastive Cross-Supervised Visual Mamba-based UNet for Semi-Supervised Medical Image Segmentation". *ArXiv*, 2024, doi: 10.48550/arXiv.2402.07245.

[24] H.A. Tedja, and O.W. Purbo, "Performance and Efficiency Comparison of U-Net and Ghost U-Net in Road Crack Segmentation with Floating Point and Quantization Optimization," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2024, doi : 10.29207/resti.v8i6.6089.

[25] Z. Zhehan, F. Wei, D. Lili, Q. Yanli, Z. Dongying, and D. Guoshen, "Semantic segmentation of remote sensing image based on encoder decoder convolutional neural network," *Acta Opt. Sin.*, vol. 40, no. 3, p. 0310001, 2020, doi: 10.3788/AOS202040.0310001.

[26] D. Tribuana, Hazriani, and A.L. Arda, "Image Preprocessing Approaches Toward Better Learning Performance with CNN," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2024, doi : 10.29207/resti.v8i1.5417.

[27] G. Ulutaş, and B. Ustubioglu, "Underwater image enhancement using contrast limited adaptive histogram equalization and layered difference representation," *Multimedia Tools and Applications*, 80, 15067–15091, 2021, doi : 10.1007/s11042-020-10426-2.

[28] U. Erkan, S. Enginoğlu, D.N.H. Thanh, and L.M. Hieu, "Adaptive frequency median filter for the salt and pepper denoising problem," *IET Image Process*, 14, 1291–1302. 2020, doi : 10.1049/iet-ipr.2019.0398.

[29] B. Xu, W. Wang, G. Falzon, P.W.H. Kwan, L. Guo, Z. Sun, and C. Li, "Livestock classification and counting in quadcopter aerial images using Mask R-CNN," *International Journal of Remote Sensing*, 41, 8121 – 8142, 2020, doi: 10.1080/01431161.2020.1734245.

[30] J. Yu, J. Wang, Z. Xu, D. Li, and W. Lin, "A similarity-based method for evaluating single-stroke regions," *Other Conferences*, 2024, doi : 10.1117/12.3049017.

[31] N.S.M. Asri, H. Sakidin, M. Othman, A.N. Matori, and A. Ahmad, "Analysis of the hydrocarbon seepage detection in oil palm vegetation stress using unmanned aerial vehicle (uav) multispectral data," in: *Proceedings of the 27th National Symposium on Mathematical Sciences* (SKSM27), 2020, doi: 10.1063/5.0018054.

[32] Y. Muralimohanbabu and K. Radhika, "Multi spectral image classification based on deep feature extraction using deep learning technique," *Int. J. Bioinform Res*. Appl., 17, 250-261. 2021, doi : 10.1504/IJBRA.2021.117169.

[33] Z. Xia, "Overfitting of CNN model in cifar-10: Problem and solutions," *Applied and Computational Engineering*. 2024, doi: 10.54254/2755-2721/37/20230511.

[34] S. Dereich, and A. Jentzen, "Convergence rates for the Adam optimizer", *ArXiv*, 2024, doi: 10.48550/arXiv.2407.21078abs/ 2407.21078.

[35] J. Kong, Y. Cheng, K. Li, and J. Xing, "Dsam: A distance shrinking with angular marginalizing loss for high performance vehicle re-identificatio," *ArXiv*, 2020, doi: 10.48550/ arXiv.2011.06228.

[36] Y. Chen, B. Zou, Z. Guo, Y. Huang, F. Qin, Q. Li, and C. Wang, "SCUNet++: Swin-UNet and CNN Bottleneck Hybrid Architecture with Multi-Fusion Dense Skip Connection for Pulmonary Embolism CT Image Segmentation", *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 7744-7752, 2024, doi: 10.1109/ WACV57701.2024.00758.

[37] C. Ranjan, "Understanding Deep Learning: Application in Rare Event Prediction," *J. Chem. Inf. Model.*, vol. 1, no. 1, pp. 1–426, 2021, https://www.researchgate.net/publication/348077077_ Understanding_Deep_Learning_Application_in_Rare_Event_Prediction.

[38] J. Kong, Y. Cheng, K. Li, and J. Xing, "Dsam: A distance shrinking with angular marginalizing loss for high performance vehicle re-identificatio," *ArXiv*, 2020, doi: 10.48550/ arXiv.2011.06228.

[39] P. Fränti and R, Mariescu-Istodor, "Soft precision and recall," *Pattern Recognition Letters*, 167, 115–121, 2023, doi : 10.1016/ j.patrec.2023.02.005.

[40] M. Widyaningsih**,** T.K. Priyambodo, M.E. Wibowo, M. Kamal, "Modification U-Net VGG-19 to Improve Semantic Segmentation Using Aerial Imagery of Oil Palm," *Ssrn*, 2023, doi : 10.2139/ssrn.4596338.