Regular Article

Prune and Quantize Semantic Segmentation Network for Aerial Objects Recognition

Gia-Vuong Nguyen, Thien Huynh-The

Department of Computer and Communications Engineering, HCM City University of Technology and Education

Correspondence: Thien Huynh-The, thienht@hcmute.edu.vn

Communication: received 17 August 2024, revised 12 September 2024, accepted 14 September 2024
Online publication: 16 September 2024, Digital Object Identifier: 10.21553/rev-jec.380

Abstract– **Semantic segmentation of aerial and satellite images is crucial for applications in environmental management, urban planning, and traffic safety. While deep learning techniques with convolutional neural networks (CNNs) and attention mechanisms have achieved superior accuracy compared to traditional methods, they often struggle with model complexity and resource constraints. This paper introduces two novel techniques - pruning and quantization - to enhance the efficiency of semantic segmentation models for remote sensing images (RSIs) by reducing computational complexity while preserving accuracy. Pruning reduces model complexity by eliminating less significant weights, while quantization decreases memory usage by converting weights into a more compact format. We applied these techniques to the DeepLabV3+ model with ResNet18 and ResNet50 backbones and assessed their performance across multiple RSI datasets. Our results show that pruning and quantization effectively reduce computational efficiency but still achieve a mean IoU of** 81.24% **with a memory footprint of** 135.19 **MB for pruning, and** 81.04% **mean IoU with a memory footprint of** 33.79 **MB for quantization on the ISPRS Vaihingen dataset. These methods offer a promising solution for deploying semantic segmentation models on resource-constrained hardware.**

Keywords– **Image segmentation, network pruning, network quantization, object recognition, remote sensing.**

1 Introduction

Semantic segmentation, which involves classifying every pixel in an image, is both a critical and computationally intensive task in image-based remote sensing applications. The primary goal is to accurately assign each pixel in a remote sensing image to its corresponding semantic category. This task has gained importance with the advent of very high-resolution and large-scale images, impacting various applications related to land cover observations. Consequently, extensive research has been conducted, broadly categorized into two approaches: (i) traditional methods relying on handcrafted feature extraction with machine learning (ML) support; and (ii) deep learning (DL)-based methods, particularly using convolutional neural networks (CNNs) and attention mechanisms.

Traditional methods for remote sensing image (RSI) segmentation primarily involve handcrafted feature extraction techniques. These methods use manually designed features and classifiers, with empirical line methods (ELM) based on image characteristics being a notable example. As highlighted in [1], accuracy improves when empirical calibration is applied to processed rasters rather than raw images. Although ELM is intuitive and easy to implement, its effectiveness diminishes significantly if the number of ground calibration targets is insufficient to extract. To address this, some studies have explored ML-based methods to support feature extraction. For instance, Radman et al. [2] investigated a robust model combining histograms of oriented gradients (HOG) and support vector machines (SVM) as feature descriptors and classifiers, respectively, using GrowCut segmentation to capture handcrafted features. Recent research has employed deep neural networks (DNNs) to enhance feature understanding through input variables and hidden layer nodes, thereby reducing the reliance on human interpretation of RSI characteristics. Notably, backpropagation neural network (BPNN) is one of the DNNs-based methods that demonstrate the feasibility and potential of DNN-based methods for radiometric correction of unmanned aerial vehicle (UAV) multispectral images, showing promise compared to previous studies [3]. However, the complex spectral characteristics of very high-resolution and large-scale remote sensing images often lead to increased intra-class variance and reduced inter-class variance, posing challenges to the effectiveness of traditional methods.

Building on the success of DL in various computer vision tasks, DL-based semantic segmentation techniques have made notable progress in both natural and remote sensing domains. However, unlike natural images captured at close range, RSIs present unique challenges due to scale differences. Small-scale land cover features are often lost as spatial resolution decreases, which can lead to reduced segmentation accuracy. To tackle this issue, research has concentrated on developing deep networks with architectures that facilitate multi-scale feature aggregation [4, 5]. Notably, the study in [6] introduced dilated convolutions to enhance context information during feature aggregation, combining multi-scale features from various network layers to boost feature representation and learning efficiency. Additionally, to improve boundary detection in RSIs, the authors in [7] recently proposed a boundary attention module (BA-module) designed to capture land-cover boundary information through hierarchical feature aggregations.

Research in remote sensing image segmentation has increasingly turned to DL models to enhance performance, marking a shift from traditional approaches to more advanced methods. Semantic segmentation models firstly based on CNNs have introduced key innovations that have significantly impacted the field. One such innovation is atrous convolution, which expands the receptive field to capture more contextual information without increasing the computational complexity [8]. Building on this, the development of atrous spatial pyramid pooling (ASPP) in [9] further advanced the capabilities of CNNs. ASPP leverages atrous convolution to extract features at multiple scales, enabling better segmentation across a variety of object sizes, which is particularly important in remote sensing. In addition to these techniques, the introduction of the adaptive feature selection (AFS) module marked a significant improvement in feature extraction [10]. AFS acts as dynamically learning the importance of features at different scales, which enhances the model's ability to capture high-level abstractions from the data. This not only improves the extraction of relevant features but also enhances the decoding process, resulting in more accurate and reliable segmentation outcomes. These advancements established CNNs as a highly effective solution for remote sensing image segmentation tasks.

However, the pursuit of greater accuracy did not stop with CNNs. Researchers began exploring alternative architectures, particularly the self-attention mechanism within Transformer models [11]. Transformers excel at capturing long-range dependencies and contextual relationships, making them well-suited for remote sensing segmentation tasks. This exploration led to the development of numerous semantic segmentation models based on Transformers, including the discovery and integration of Swin blocks [12–15], significantly improving segmentation performance. Furthermore, the combination of Transformers and CNNs resulted in hybrid models that capitalize on the strengths of both architectures [16, 17]. These models leverage the spatial hierarchies inherent in CNNs and the global context awareness of Transformers, achieving remarkable segmentation accuracy. The resulting models have consistently outperformed earlier approaches, highlighting the substantial potential of these advanced techniques for semantic segmentation in remote sensing imagery.

Despite the advancements of DL-based methods in RSI semantic segmentation, several significant challenges remain. One major issue is the increasing demand for hardware efficiency, particularly in UAVs, which imposes constraints on deploying highly complex models. The high computational and memory requirements of these advanced models often make them unsuitable for UAVs and other low-resource environments. To address this, researchers have started developing models with lighter architectures that are more compatible with low-cost hardware. For example, in [18], a lightweight CNN is proposed, featuring fewer channels per layer, which significantly reduces the number of parameters and, consequently, the computational load. Additionally, other networks have focused on constructing bilateral architectures [19, 20], leveraging two-branch structures to exploit hardware's parallel processing capabilities effectively. It can be observed that the field of DL for semantic segmentation of remote sensing images is witnessing an increasing number of breakthroughs. To summarize the review of existing related works in this interested topic, we present their technical highlights, advantages, and limitations in Table I. While the trend toward optimizing model complexity remains strong, it is not without challenges. Firstly, developing innovative methods is inherently difficult and requires significant time to train new models to adapt to the RSIs data. Secondly, balancing model accuracy with complexity is a persistent challenge. This trade-off often results in low-cost methods being associated with lower accuracy, highlighting the difficulty in achieving both high performance and efficiency simultaneously.

To address these challenges, we propose semantic segmentation methods for RSIs by employing pruning and quantization techniques. We begin by selecting a CNN model, which offers an optimization between accuracy and complexity in the field of semantic segmentation. Specifically, we utilize DeepLabV3+ [9] with various backbones, such as ResNet18 and ResNet50 [21], as the standard framework for our approach. To optimize the model's performance in terms of both accuracy and complexity, we leverage Taylor approximation to evaluate the trained model, selectively disabling the least significant weights in the network, thereby implementing our pruning method. Additionally, we convert the network's weights to a new data type with a lower memory footprint, reducing memory usage and enhancing the model's adaptability to low-cost hardware.

In summary, the main contributions of our work are as follows:

- We introduce a novel approach to reducing the number of weights in a remote sensing image segmentation model by estimating and removing unnecessary weights, resulting in a model we refer to as pruning.
- We propose a method for optimizing memory usage by converting network weights to a more resource-efficient data type, leading to a model referred to as quantization.
- Through extensive simulations, both pruning and quantization demonstrate superior performance in terms of accuracy and complexity across diverse RSI datasets, highlighting the effectiveness of the proposed network architecture.

The remainder of this paper is organized as follows: Section 2 describes the details of our proposed method. Section 3 conducts the experimental validation

Table I
The Summarization of Technical Highlights, Advantages, and Limitations of Related Works in the Field of Semantic Segmentation

Ref	Technical highlights	Advantages	Limitations
[9]	Atrous convolution and atrous spatial pyramid pooling for multi-scale feature extraction.	Archives high accuracy and efficient segmentation performance.	High computational complexity, challenges in resource-constrained hardware deployment.
[10]	Adaptive feature selection with attention mechanism module.	Enhances segmentation of multi-scale remote sensing data.	An improved method comes with increasing the complexity.
[12]	Swin blocks, a global context fusion module and a gate convolution module for refined information processing.	Improves segmentation performance on complex targets and boundaries.	Increased model complexity and potential computational overhead.
[13]	Adaptive transformer fusion and foreground saliency guided loss for enhanced saliency modeling.	Effectively handles large-scale variation, complex backgrounds, and imbalanced distributions.	Performance is potentially impacted by foreground complexity and image resolution variability.
[14]	A group transformer, group convolution, and a cross-feature fusion module to integrate local and global features.	Enhances segmentation performance by capturing global contextual information from large-scale remote sensing images.	The architecture is very complex to integrate with different models.
[15]	A transformer and encoder-decoder structure with a Swin backbone and a class-guided Transformer block in the decoder.	Effectively captures long-range dependencies for improved performance.	The architecture may potentially increase complexity compared to CNN-based methods.
[18]	An light-weight architecture with fewer parameters and channels.	Reduces computational cost while maintaining competitive performance.	Limitations in capturing fine details due to reduced model capacity.
[19]	A bilateral segmentation network help better parallel processing.	Balances high segmentation performance with real-time inference speed.	Limitations in handling extremely fine details or very large-scale variations.
[20]	A shelf-shaped structure consisting of multiple branch pairs.	Archives faster inference speed compared to non-real-time methods.	The shelf-shaped architecture may be more complex to implement.

of the proposed method with comprehensive discussions, along with a comparative assessment against other methodologies. Finally, we discuss the conclusions of the paper in Section 4.

2 Methodology

2.1 DeepLabV3+ Architecture

In this subsection, we intend to utilize a CNN for remote sensing image segmentation. Currently, several CNN architectures have proven effective in semantic segmentation, such as U-Net [22], Mask R-CNN [23], and particularly the various versions of DeepLab [8], specifically DeepLabV3+ [9], which have made significant advances and have become state-of-the-art methods in semantic segmentation tasks. In addition to using pre-trained backbones such as ResNet18 and ResNet50 [21] to enhance recognition performance, the contributions of DeepLabV3+ can be summarized through its use of atrous convolutions and the atrous spatial pyramid pooling (ASPP) architecture.

To learn features from a larger receptive field while maintaining computational efficiency, DeepLab networks have been introduced by incorporating multiple atrous convolution layers. These layers allow for the extraction of spatial features with a larger receptive field. Atrous convolution layers compute features to produce output according to the following formula

$$\mathbf{Y}_a[i,j] = \sum_{u,v} \mathbf{X}_a[i + r_H u, j + r_W v] \times \mathbf{W}_a[u,v], \qquad (1)$$

where, the filter $\mathbf{W}_a$ has a size of $u \times v$ to extract local features from the input $\mathbf{X}_a$, r_H and r_W denote the dilation rates in the height and width dimensions, respectively, and $\mathbf{Y}_a$ is the output of the atrous convolution. Interestingly, conventional convolution layers are defined with $r_H = r_W = 1$.

Along with atrous convolution layers, the ASPP module is designed to facilitate feature extraction at multiple scales using atrous convolution layers with different dilation rates r, meaning $r_H = r_W = r$ in DeepLabV3+ [9]. Thus, ASPP enriches the spatial context information of feature maps by expanding the receptive field (the area on the feature map that a convolutional layer can scan at one time) without increasing the number of weights or computational costs, thereby making the model more suitable for segmentation tasks. The output of ASPP is a synthesis of multiple feature maps obtained from atrous convolution layers with different dilation rates through a depthwise concatenation layer as follows

$$\mathbf{F} = \langle \mathbf{A}^{1\times1}_{1,1}(\mathbf{X}), \mathbf{A}^{3\times3}_{1,6}(\mathbf{X}), \mathbf{A}^{3\times3}_{1,12}(\mathbf{X}), \mathbf{A}^{3\times3}_{1,18}(\mathbf{X}) \rangle, \qquad (2)$$

where, $\mathbf{A}^{n\times n}_{s,r}$ denotes a sequential operation, including an atrous convolution layer (with the filter size $n \times n$, stride s, and dilation rate r), batch normalization (BN), and ReLU (rectified linear unit) activation function. Here, $\mathbf{X}$ and $\mathbf{F}$ are the input and output of ASPP, respectively, and $\langle\cdot\rangle$ denotes a depthwise concatenation of feature maps.

Although DeepLabV3+ with atrous convolution and ASPP has achieved significant advancements in semantic segmentation, it still faces a drawback related to model complexity. The high complexity of DeepLabV3+ makes it unsuitable for implementation on low-power hardware resources, necessitating solutions that optimize the model while maintaining accuracy.

2.2 Pruning Techniques

Network pruning emerges as a technique to address the challenge of reducing the number of weights (a.k.a., network parameters or learnable parameters) while preserving model performance. In detail, this approach operates iteratively, during each iteration, weights with minimal influence on the model's output are identified and subsequently removed. Specifically, suppose a singular weight in the weight matrix $\mathcal{D}$ denoted by w_i is being considered for pruning. The impact of this weight on the loss function $\mathcal{L}$ can be expressed by the following equation

$$|\Delta\mathcal{L}(\mathcal{D}, w_i)| = |\mathcal{L}(\mathcal{D}, w_i = 0) - \mathcal{L}(\mathcal{D}, w_i)|, \qquad (3)$$

where, $\mathcal{L}(\mathcal{D}, w_i)$ represents the value of the loss function evaluated on the weight matrix $\mathcal{D}$ with w_i have not been pruned, and $\mathcal{L}(\mathcal{D}, w_i = 0)$ is the value of the loss function when w_i is disabled. Thus, $|\Delta\mathcal{L}(\mathcal{D}, w_i)|$ is the change in the loss function's value before and after pruning a weight.

To approximate $\mathcal{L}(\mathcal{D}, w_i)$, a first-order Taylor expansion is used. For a function $f(x)$, the Taylor expansion at $x = a$ is

$$f(x) = \sum_{n=0}^{f^{(n)}(a)} (x-a)^n + R_n(x), \qquad (4)$$

where, $f^{(n)}(a)$ denotes the n-th derivative of f evaluated at the point a, and $R_n(x)$ is the remainder term of the n-th order expansion, which is typically very small (approximately zero) and can be neglected. Therefore, the first-order Taylor expansion can be written as

$$f(x) = f(a) + f^{(1)}(a)(x-a). \qquad (5)$$

Applying the first-order Taylor approximation represented by Equation 5, the value of the loss function evaluated on the weight matrix $\mathcal{D}$ with w_i has not been pruned, $\mathcal{L}(\mathcal{D}, w_i)$ at $w_i = 0$, can be approximated as follows

$$\mathcal{L}(\mathcal{D}, w_i) = \mathcal{L}(\mathcal{D}, w_i = 0) + \left.\frac{\delta\mathcal{L}}{\delta w_i}\right|_{w_i=0} w_i, \qquad (6)$$

where, $\left.\frac{\delta\mathcal{L}}{\delta w_i}\right|_{w_i=0}$ represents the value of the first derivative of the loss function with respect to w_i, evaluated at $w_i = 0$.

By substituting Equation 6 into Equation 3, we can obtain the change in the loss function value before and after pruning a weight as follows

$$|\Delta\mathcal{L}(\mathcal{D}, w_i)| = \left| \left.\frac{\delta\mathcal{L}}{\delta w_i}\right|_{w_i=0} w_i \right|. \qquad (7)$$

Thus, the change in the loss function resulting from pruning a specific weight can be easily computed. Leveraging this, we aim to identify weights that have minimal impact on the output, with the goal of optimizing the network in terms of both accuracy and complexity. However, pruning individual weights by setting their values to zero is impossible, because even zero weights contribute to the output computation, making the reduction in computational cost nearly meaningless. Therefore, we extend the pruning process from singular weights to the filters in each layer, using the concept of average objective function evaluation. The comprehensive objective function for evaluating the loss of all weights in a filter are pruned can be mathematically represented as follows

$$\Phi(W_i) = \left| \frac{1}{M} \sum_{m}^{M} \left.\frac{\delta\mathcal{L}}{\delta w_{i,m}}\right|_{w_i=0} w_{i,m} \right|, \qquad (8)$$

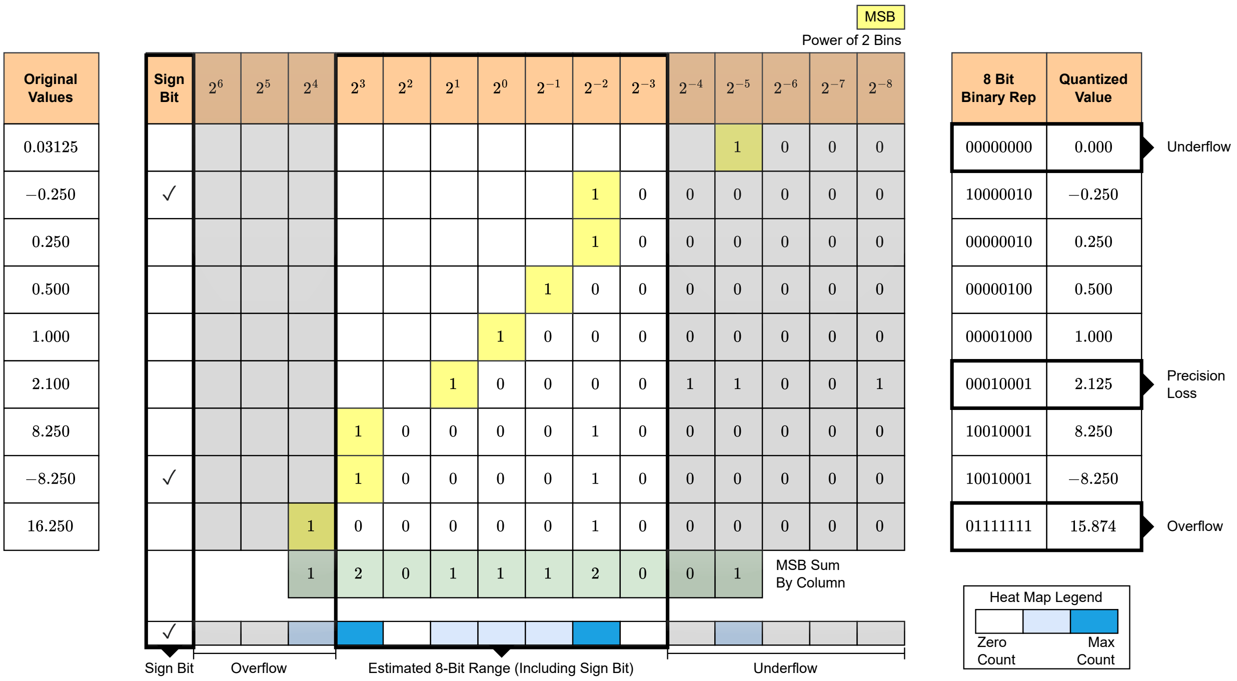
where, $\Phi(W_i)$ denotes the objective function calculated over all weights of the filter W_i, M and is the total number of weights within the filter W_i. Thus, network pruning becomes more manageable as it allows for the easy identification of filters with minimal impact on the model's output and their removal to optimize computational cost. For this reason, we propose integrating the improved weight pruning method into the base network and define this improvement as pruning.

Pruning implementation: To apply pruning as a deep learning approach for image semantic segmentation, particularly in the context of remote sensing images, involves several key steps as follows. At first, a deep convolutional neural network, specifically the DeepLabV3+ model, is trained to ensure that its trainable parameters adapt effectively to the image data. During the training process, a loss function is utilized to direct and optimize the model's weights, enabling it to generate outputs that closely approximate real-world ground truth (a.k.a. segmentation mask). After training, pruning is then applied to the trained DeepLabV3+ network, using Taylor expansion to assess the impact of each weight in the trained network on the loss function, as described in the preceding section. In this step, an iterative process consisting of 60 loops is employed, wherein each loop, the filters with the least influence on the model are identified and removed. Following each iteration, the network is recalibrated to remove the pruned filters, and this process is repeated for the subsequent iterations until the loop end. Finally, the number of weights in the network is significantly reduced, which inevitably impacts the model's segmentation performance to some extent. To address this, we propose a retraining procedure to allow the weights in the pruned network to better adapt to the data, thereby enhancing the segmentation performance.

To effectively utilize pruning techniques in semantic segmentation-based applications, it is crucial to consider the relationship between the complexity of input images and the number of parameters in the CNN employed. Specifically, when input images are highly complex—characterized by high resolution or the dense presence of multiple objects with varying sizes, pruning may not be optimal for CNNs with limited parameters. Conversely, pruning demonstrates its effectiveness for excessively large and complex models designed to handle such inputs.

2.3 Quantizing Techniques

In digital hardware, data is stored in binary words, which are fixed-length sequences of bits (0's and 1's). The data type determines how these sequences are interpreted by hardware components or software functions. Typically, numerical data is represented in two principal forms: 8-bit scaled integer (often referred to as fixed-point) or 16-bit floating-point data types. Most



Original Values	Sign Bit	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	8 Bit Binary Rep	Quantized Value	
0.03125													1	0	0	0	00000000	0.000	Underflow
−0.250	✓									1	0	0	0	0	0	0	10000010	−0.250	
0.250										1	0	0	0	0	0	0	00000010	0.250	
0.500									1	0	0	0	0	0	0	0	00000100	0.500	
1.000								1	0	0	0	0	0	0	0	0	00001000	1.000	
2.100							1	0	0	0	0	1	1	0	0	1	00010001	2.125	Precision Loss
8.250					1	0	0	0	0	1	0	0	0	0	0	0	10010001	8.250	
−8.250	✓				1	0	0	0	0	1	0	0	0	0	0	0	10010001	−8.250	
16.250				1	0	0	0	0	0	1	0	0	0	0	0	0	01111111	15.874	Overflow
				1	2	0	1	1	1	2	0	0	1	MSB Sum By Column					
	✓																		

Figure 1. Visualization of the dynamic ranges for select sample values, including weights and biases from the convolutional layers, as well as activations across all layers within the network.

pre-trained neural networks utilize floating-point data types. Even relatively small neural networks require substantial memory and hardware capable of performing floating-point arithmetic. These requirements can limit the deployment of DL capabilities to low-power microcontrollers due to their constrained resources. To address this limitation, we leverage the quantization technique to convert 16-bit floating-point data to 8-bit scaled integer data types. Although the network quantization solution may lead to inaccuracies in model predictions, it contributes to a significant reduction in complexity, so in this work, we apply a quantization network as a solution named quantization.

To achieve this, we start by analyzing the logged values of certain parameters during network operations. For instance, some logged original values can be observed in Figure 1. Next, we determine the optimal binary representation for each recorded parameter value. This is typically represented by the most significant bit (MSB), which is the left-most bit in the binary sequence and has the greatest impact on the value. In our implementation, each MSB is highlighted in yellow for clarity. By aligning these binary sequences, we can observe the bit distribution for the logged parameter values. Summing the MSBs in each column, highlighted in green, provides a collective view of the logged values. A heat map displays the MSB counts for each bit position, with darker blue areas indicating higher MSB counts at those positions. Based on the distributions of MSB shown as a heat map, we determine the estimated 8-bit range (including the sign bit), which must encompass the majority of MSB bits within our selected range. Once the range is assigned, any bits beyond this type are discarded.

Notably, assigning a smaller fixed-length data type may lead to issues such as precision loss, overflow, or underflow for values that the data type cannot fully represent. For instance, a value of 0.03125 experiences underflow, resulting in a quantized value of 0. This implies that all parameter values lower than 0.125 or higher than −0.125 (the smallest positive and largest negative value represented by our estimated range) will be set to 0. On the other hand, a value of 2.1 incurs a precision loss, leading to a quantized value of 2.125, due to the lack of number of bits representing the value. Furthermore, a value of 16.250 surpasses the maximum representable value of the data type, resulting in an overflow. This occurs because its MSB falls outside the estimated range, leading to a situation where all bits, except the sign bit, are set to 1, yielding a saturated quantized value of 15.874.

Quantization implementation: Similar to pruning, applying quantization as a solution for DL algorithms in image semantic segmentation also employs the training of DeepLabV3+ network to learn features from images. However, rather than using iterative loops to remove filters that contribute minimally to the model's output, as in pruning, quantization uses a statistical approach to convert the weights of the trained network from 16-bit to 8-bit. This conversion substantially reduces the memory required to store and process the model while preserving most of the information, as no filters are removed. Consequently, quantization does not necessitate an additional retraining phase.

To optimize quantization, it can be configured to only apply to weights where the difference before and after quantization is negligible, demonstrating its potential to reduce model complexity while maintaining segmentation performance. However, this approach may introduce heterogeneity in the data types of weights within the DL model, complicating image processing computations. Therefore, in our study, we propose applying quantization to all weights in the trained network to ensure ease of implementation while exploring the potential of quantization.

3 Results and Discussions

3.1 Datasets

UAVid [24]: This dataset is a significant resource for aerial semantic segmentation, offering high-resolution imagery of urban scenes captured by unmanned aerial vehicles. UAVid contains 42 image sequences, divided into three subsets: 20 sequences for training, 7 for validation, and 15 for testing. Moreover, the dataset includes eight object categories: *Building*, *Road*, *Static car*, *Moving car*, *Tree*, *Low vegetation*, *Human*, and *Background clutter*. Both images and their segmented labels are initially provided at a 4K resolution and are downsampled by a factor of four for computational efficiency in our implementation. In detail, all images and labels will be resized to a resolution of 540 × 960.

Vaihingen[1]: This dataset comprises 33 high-resolution image patches, each with an average resolution of around 2500 × 2000 pixels. The ground truth data includes five object classes: *Impervious surfaces*, *Buildings*, *Low vegetation*, *Trees*, and *Cars*, in addition to *Background clutters*. In our experiments, we use only the IRRG bands, excluding the digital surface model (DSM) information. Notably, 11 image patches $(1,3,5,7,13,17,21,23,26,32,37)$ are used for training, while the remaining five patches $(11,15,28,30,40)$ are reserved for testing, following previous works [18, 25]. Notably, we also resized the images and labels in this dataset to a resolution of 512 × 512 to suitably optimize hardware resources.

3.2 Implementation Details

Training configurations: For the training configuration, we employ weight initialization following the pre-trained parameters on the ImageNet dataset [26]. During the training phase, all weights are iteratively updated using stochastic gradient descent with momentum (SGDM) with a momentum parameter $\beta = 0.9$ and the cross-entropy loss function. The initial learning rate is set to 1^{-3} and decays by a factor of 0.3 every 10 epochs. Additionally, L2 regularization with a coefficient of 0.001 is applied to mitigate model overfitting. All network models are trained for 80 epochs to ensure model convergence, with a mini-batch size of 8. Both the training and evaluation phases are implemented in MATLAB R2023b and trained on an RTX2080 GPU.

For the pruning configurations, we perform 60 iterations to conduct network review, evaluation, and pruning. The cross-entropy loss function value is computed using the training set for both datasets. Following pruning, we implement a retraining process to enhance the pruned model's adaptation to the dataset and improve its performance. This retraining process uses the same configurations as the initial training, with the exception that the number of epochs is reduced to 20, as most weight values have already adapted well during the initial training phase.

For the quantization configurations, our implementation involves converting all training weights from 16-bit floating-point to 8-bit scaled integer data types. This approach is adopted to streamline the process and assess the effectiveness and possibility of the quantization technique.

Data augmentation: As previously mentioned, images are resized to a predetermined resolution of 540 × 960 for the UAVid dataset and 512 × 512 for the ISPRS Vaihingen dataset to ensure compatibility with the low-power hardware resource. This preprocessing simplifies feature extraction and reduces computational complexity.

For data augmentation, we apply random left-right reflection, randomly flipping the image horizontally along its vertical axis within a range of $[-10, 10]$ pixels. This augmentation enables the model to learn more diversified features independent of the object's orientation in the image [27, 28]. In real-world scenarios, objects can be viewed from various angles, and flipping helps the model generalize better to unseen orientations.

Evaluation metrics: The segmentation results of all models are evaluated using commonly adopted metrics in the field of semantic segmentation [29]. Specifically, we employ global accuracy, mean intersection-over-union (IoU), and mean boundary-F1-score (BFScore) to assess model accuracy.

For comparisons of the complexity of the model, we use the number of parameters (Params) and estimated memory usage metrics (Memory). All results represent the average of each metric value over 5 executions. Our complete segmentation results are based on the metric values obtained during the evaluation process on the validation set for the UAVid dataset (due to the absence of a testing set) and on the testing set for the ISPRS Vaihingen.

3.3 Ablation Study

In our first simulations, we provide a comprehensive evaluation of improvement strategies implemented including pruning and quantization. The evaluation was conducted on an RTX2080 GPU, with a particular focus on comparing different backbones, ResNet18 and ResNet50. Table II highlights the performance in terms of global accuracy, mean IoU, mean BFScore, memory usage, and the number of parameters.

Ablation for pruning: For the backbone ResNet18, when only the strategy pruning was activated, there was a noticeable decline in performance metrics, with global accuracy decreasing by 5.97%, mean IoU by 3.77%, and mean BFScore by 5.00%, compared to the standard DeepLabV3+ model without any improvement strategies. This reduction in performance, while significant, was offset by a substantial gain in memory efficiency. The memory usage decreased to 24.67 MB,

[1] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx

Table II
Detailed Performance Comparison of Improvement Strategies in Our Proposed Method. **P**: Pruning and **Q**: Quantization. The Evaluation Is Implemented on an RTX2080 GPU

Backbone	P	Q	Global Acc	Mean IoU	Mean BFScore	Memory (MB)	Params (M)
ResNet18	✗	✗	81.09	53.06	62.19	59.75	20.6
	✓	✗	76.25	51.06	59.08	24.67	9.2
	✗	✓	75.63	51.02	58.97	14.96	20.6
	✓	✓	72.14	50.27	58.88	6.24	9.2
ResNet50	✗	✗	84.65	59.26	69.11	148.32	43.9
	✓	✗	84.79	59.53	69.47	127.85	37.8
	✗	✓	84.23	59.41	69.13	37.08	43.9
	✓	✓	83.96	58.54	68.88	31.94	37.8

due to the reduction of the number of parameters to 9.2 millions. This demonstrates the trade-off inherent in applying pruning to the model, while it enhances memory efficiency, it simultaneously compromises the model's segmentation accuracy. When pruning and quantization were combined, pruning must be applied before quantization because quantization first alters the weight values within the network, leading to inaccuracies in Taylor estimation calculations and subsequently reducing network performance. The model's performance in this configuration showed a reduction in the number of parameters by up to 55.34% and in memory usage by 58.29% compared to the model with only the quantization strategy applied. However, this reduction in complexity came with a significant drop in global accuracy, mean IoU, and mean BFScore fell to 72.14%, 50.27%, and 58.88%, respectively. Thus, while pruning is effective in reducing model complexity and memory usage, it also leads to a severe decline in performance, particularly when applied to networks that already have low complexity and a small number of parameters.

To further explore the effectiveness of pruning, we implemented it in a standard DeepLabV3+ model with a more complex backbone, ResNet50. The ResNet50 backbone generally demonstrated superior performance in accuracy metrics compared to ResNet18, though this came at the cost of higher memory consumption and a larger number of parameters. Interestingly, applying pruning to this configuration led to a slight improvement in term of accuracy, while also moderately reducing memory usage to 127.85 MB and decreasing the number of parameters to 37.8 million. The improvement in accuracy can be attributed to the retraining process associated with pruning, which allows the pruned network to better adapt to the data. However, when both pruning and quantization were combined to further minimize the model's memory requirements, there was a substantial decline in performance. Global accuracy dropped to 83.96%, mean IoU to 58.54%, and mean BFScore to 68.88%. This combination, however, resulted in a significant reduction in memory usage to 31.94 MB, with the number of parameters decreasing to 37.8 millions. Despite the reduction in memory usage, the combination of these strategies did not yield further improvements in accuracy. The reduction in complexity in this configuration was huge so the model's accuracy could no longer be sustained.

In summary, the advantages and disadvantages of the pruning strategy are as follows:

- Reduces the number of parameters and decreases memory usage.
- Particularly effective for high-complexity models (e.g., ResNet50), with the potential for a slight accuracy improvement after retraining.
- Leads to substantial degradation in model performance (e.g., accuracy, mean IoU) when applied to low-complexity models (e.g., ResNet18).
- When combined with quantization, pruning must be performed first to prevent inaccuracies in weight estimation, which may significantly impact model performance.

Ablation for quantization: Activating the quantization strategy for the ResNet18 backbone led to further reductions in performance metrics, but it also resulted in a substantial decrease in memory usage to 14.96 MB, while the number of parameters remained constant at 20.6 million. This outcome is attributed to the focus of quantization on reducing the memory required to store the network by minimizing the memory footprint of each individual weight. Consequently, the number of parameters in the network remains unchanged, but the overall memory required for storage is significantly reduced. Pruning, which targets the complete elimination of unnecessary weights, not only reduces the number of parameters but also lowers memory usage, however, this reduction is not as pronounced as that achieved through quantization. While quantization optimizes memory usage, it also significantly impacts the model's segmentation performance. As shown in Table II, the performance of the model with quantization alone is approximately 75.63% for global accuracy and 51.02% for mean IoU, which is lower than that of both the pruning-based approach and the standard network. When combining quantization with pruning, the segmentation performance drops further, reaching about 72.14% in global accuracy and 50.27% in mean IoU. However, these results are still competitive when considering the deployment of the model on low-performance hardware, where the trade-off between reduced complexity, minimal memory requirements, and adequate accuracy is crucial.

For the ResNet50 backbone, the application of quantization results in a slight reduction in global accuracy, about 0.49%, but an interesting observation is that the performance in terms of mean IoU and mean BFScore slightly improves. Although these increases are marginal (approximately 0.25% for mean IoU and 0.02% for mean BFScore compared to the standard network), they demonstrate the potential benefits of employing quantization in more complex models. This suggests that quantization can be effectively leveraged in scenarios where memory efficiency is critical, without substantially compromising model performance.

To summarize the discussion on the effectiveness of the quantization strategy, the key points are as follows:

- Substantially reduces memory usage without altering the number of parameters, making it suitable

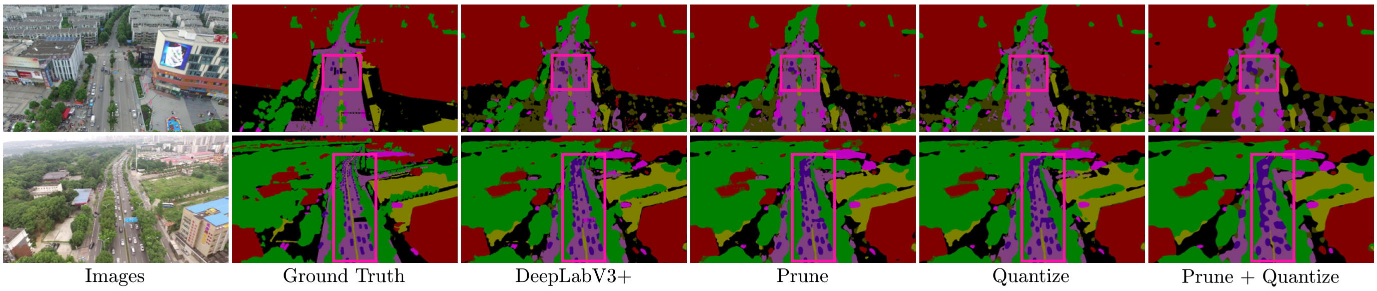


Figure 2. Visualization results (as segmentation masks) of the proposed methods based on the UAVid dataset. The pink rectangles are used to emphasize the regions that show a clear different in segmentation accuracy. Prune and Quantize denote the standard DeepLabV3+ with the combination of pruning and quantization, respectively.

for deployment on low-performance hardware.

- For complex models (e.g., ResNet50), it may result in slight improvements in mean IoU and mean BFScore.
- Reduces model performance (e.g., accuracy, mean IoU) when applied independently, with a more pronounced decline when combined with pruning.
- Less appropriate when maintaining high model performance is critical.

The results from our ablation study demonstrate a trade-off between accuracy and resource efficiency. The ResNet18 backbone shows higher efficiency in terms of memory usage and the number of parameters, particularly when the improvement strategies are enabled. However, this comes at the cost of reduced performance metrics such as global accuracy, mean IoU, and mean BFScore due to the lack of the number of parameters. In contrast, the ResNet50 backbone generally provides better accuracy but requires significantly more memory and parameters. The choice of configuration thus depends on the specific requirements of the application, balancing the need for high performance with resource constraints.

Visualization results: To further illustrate the impact of these strategies on segmentation accuracy, we present the segmentation results as image overlays in Figure 2. These results compare various configurations of the model architecture, including the standard DeepLabV3+ network, and its variants with pruning, quantization, and both strategies combined. Our analysis highlights the effectiveness of each model in producing accurate segmentation results, particularly in the pink rectangle of the visualizations, where the complete version of the model (with both pruning or quantization) is slightly less accurate than the another version. Notably, despite the application of pruning and quantization separately, the segmentation accuracy remains close to that of the standard DeepLabV3+ network, even though the model complexity is significantly reduced. However, when both pruning and quantization are applied together, there is a noticeable decline in segmentation accuracy, particularly in small classes such as vehicles and minor structures, where misclassification becomes more prevalent. This observation suggests that while combining these strategies is effective in reducing model complexity and resource usage, it may not be ideal for tasks that require high precision in identifying small or detailed objects.

3.4 Comparison Results

The performance of our proposed methods is thoroughly evaluated by analyzing each component and comparing it against state-of-the-art image segmentation models. This comparison is conducted on the UAVid and ISPRS Vaihingen datasets, utilizing standard quantitative metrics such as global accuracy, mean IoU, mean BFScore, and model size (i.e., the memory usage and total number of trainable parameters). Furthermore, class-wise performance analysis discussions of the experimental results are provided to offer deeper insights and analysis of our work.

Overall performance analysis: In our implementation, both of the compared models are trained to full convergence to ensure the reliability of our approach. To further validate the flexibility of our model across different datasets, we present accuracy and complexity metrics evaluated on both the UAVid and ISPRS Vaihingen datasets with varying image sizes. As shown in Table III, our models consistently achieve superior segmentation accuracy compared to other existing models on the UAVid dataset, while also demonstrating competitive performance on the ISPRS Vaihingen dataset. This analysis underscores the effectiveness of pruning and quantization strategies in accurately segmenting objects within aerial and satellite imagery.

Firstly, the results from the UAVid dataset highlight that the pruning technique excels in key performance metrics related to accuracy, achieving a global accuracy of 84.79%, a mean IoU of 59.53%, and a mean BFScore of 69.47% (as shown in Table III). When combined with other state-of-the-art methods, our approach surpasses other models such as BiSeNet (which attained a global accuracy of 83.02%, a mean IoU of 54.42%, and a mean BFScore of 68.71%), ShelfNet (which recorded a global accuracy of 84.24%, a mean IoU of 57.62%, and a mean BFScore of 68.91%), and others, all while maintaining lower memory usage and parameter counts. Notably, several studies focus on developing methods suited for real-time applications, such as LWN-A-F and BANet, despite their focus on efficiency, these methods still require a similar memory footprint to our quantization approach, yet do not achieve the same level of accuracy. This performance emphasizes the effectiveness of pruning and quantization in accurately segmenting a variety

Table III
Method Comparison Between the Proposed Models and Other State-of-the-Art Models in Terms of Segmentation Performance and Model Size

UAVid dataset at resolution of 540 × 960

Method	Global Acc	Mean IoU	Mean BFScore	Memory (MB)	Params (M)
LWN-A-F [18]	82.69	56.28	65.12	32.48	15.0
BANet [30]	81.98	55.18	65.87	39.22	**10.5**
BiSeNet [19]	83.02	54.42	68.71	51.34	49.0
ShelfNet [20]	84.24	57.62	68.91	129.05	35.6
DeepLabV3-AFS [10]	77.02	51.76	61.91	154.23	27.7
PSPNet-AFS [10]	76.89	51.64	61.64	167.92	34.9
CG-Swin [15]	83.72	57.18	66.27	886.12	197.1
Pruning	**84.79**	**59.53**	**69.47**	127.85	37.8
Quantization	83.96	58.54	68.88	**31.94**	37.8

ISPRS Vaihingen dataset at resolution of 512 × 512

Method	Global Acc	Mean IoU	Mean BFScore	Memory (MB)	Params (M)
DANet [31]	90.31	81.13	89.21	262.01	68.5
GFFNet [14]	**91.26**	**81.82**	89.93	417.23	74.3
RSSFormer [13]	90.94	81.57	89.40	512.21	72.9
STDSNet [12]	90.24	81.76	89.80	762.18	130.1
SwinCNN [16]	90.02	81.12	89.42	985.21	235.8
ST-UNet [17]	89.95	80.89	89.28	923.16	208.4
Pruning	91.18	81.24	**90.13**	135.19	**39.1**
Quantization	90.08	81.04	89.09	**33.79**	43.9

of objects like buildings, roads, vehicles, and people in UAV imagery, while also keeping the model's complexity manageable. It's also noteworthy that our proposed models in the UAVid dataset are built on the ResNet50 backbone, indicating that ResNet50 is particularly well-suited for semantic segmentation tasks involving aerial imagery.

The second experiment, conducted on the ISPRS Vaihingen dataset, reconfirms the strong performance of pruning and quantization (as detailed in Table III). The trends observed in the UAVid dataset are consistent with those seen in the ISPRS Vaihingen dataset, where pruning achieves high scores in global accuracy (91.18%), mean IoU (81.24%), and mean BFScore (90.13%). While the accuracy of both pruning and quantization-based methods is slightly lower compared to some advanced models that utilize Transformer architectures, such as GFFNet (which achieves a global accuracy of 91.26%, a mean IoU of 81.82%, and a mean BFScore of 89.93%). Pruning and quantization still demonstrate their efficiency by achieving the lowest memory usage (33.79 MB) with quantization and the lowest number of parameters (37.8 million) with pruning.

These results further solidify pruning and quantization as leading methods for semantic segmentation in diverse aerial and satellite imagery applications. Additionally, the table emphasizes the continued use of our proposed network with ResNet50, highlighting its adaptability and effectiveness across various architectures.

The discussion of our overall performance analysis can be summarized as follows:

- Pruning and quantization surpassed other state-of-the-art models in terms of memory usage and the number of parameters, demonstrating their feasibility for implementation in image segmentation applications on resource-constrained hardware.
- On the UAVid dataset, our proposed strategies outperformed models such as BiSeNet and ShelfNet, indicating their effectiveness for overhead image segmentation compared to other CNN models.
- On the ISPRS Vaihingen dataset, our model achieved comparable segmentation performance to more complex models based on transformers, underscoring its suitability and modernity for segmenting different types of remote sensing images.

Class-wise performance analysis: To demonstrate more detail about our proposed model capacity in individual class recognition, we provide comprehensive results about the comparison of mean IoU between our method against other state-of-the-art methods.

UAVid dataset: In Table IV, the mean IoU metric provides a detailed evaluation of accuracy across various categories. Among all methods evaluated, pruning stands out as the top performer in mean IoU across most classes in the UAVid dataset. Specifically, pruning achieves the highest mean IoU for several critical classes, including *Background Clutter* with 56.86%, *Trees* with 75.99%, *Low Vegetation* with 67.97%, *Moving Car* with 62.87%, and *Static Car* with 45.89%. The only class where pruning's performance is slightly lower than CG-Swin is *Road*. Notably, pruning also attains the highest mean IoU for *Buildings* at 89.67%, surpassing both quantization, ShelfNet, CG-Swin, and LWN-A-F, while other methods yield significantly lower results, ranging from 81.10% to 86.05%. This indicates that the model excels in segmentation performance, demonstrating its strong adaptation to large-sized objects. The model is thus well-suited for object segmentation tasks in remote sensing images, particularly those related to land observation. However, the pruning performs at only 5.11% mean IoU for the *Humans* class. While this represents the best performance compared to other state-of-the-art methods, it still shows limited improvement in segmenting small objects, revealing a notable weakness in pruning capability for high-precision recognition of small objects. Quantization also exhibits notable performance, frequently surpassing other models in segmentation accuracy and closely following pruning in effectiveness. It achieves high mean IoU values of 88.95% for *Buildings*, 75.53% for *Trees*, and 67.11% for *Low Vegetation*. This underscores the effectiveness of both pruning and quantization in segmenting a variety of objects within the UAVid dataset's aerial imagery. However, similar to pruning, quantization struggles with small objects, achieving only 4.69% mean IoU for the *Human* class. The results on the UAVid dataset confirm the efficiency of our proposed methods for remote sensing segmentation tasks.

ISPRS Vaihingen dataset: Table V offers a comprehensive class-wise performance comparison between our proposed methods, pruning and quantization, and several leading models on the ISPRS Vaihingen dataset. Interestingly, the results illustrate that the current state-of-the-art models excel in specific categories: GFFNet leads in the *Impervious Surface* class with a mean IoU of 87.39%, STDSNet tops the *Building*, *Low Vegetation*,

Table IV
Class-Wise Performance Comparison Between the Proposed Models and Other State-of-the-Art Models on the UAVid Dataset

Method	Bac. Clu.	Building	Road	Tree	Low Veg.	Mov. Car	Sta. Car	Human
LWN-A-F [18]	52.32	87.38	71.73	65.43	62.25	61.71	44.42	4.92
BANet [30]	52.88	86.05	70.12	62.71	60.33	61.18	43.56	4.68
BiSeNet [19]	50.32	84.38	69.33	63.01	61.73	60.15	43.23	3.12
ShelfNet [20]	56.09	88.17	70.18	72.54	65.26	61.16	43.52	4.01
DeepLabV3-AFS [10]	47.53	82.95	64.68	59.02	57.12	56.79	42.74	3.38
PSPNet-AFS [10]	48.67	81.10	63.74	59.66	55.87	58.64	41.75	3.77
CG-Swin [15]	53.66	87.51	**72.92**	66.32	64.38	62.90	45.54	4.30
Pruning	**56.86**	**89.67**	71.86	**75.99**	**67.97**	**62.87**	**45.89**	**5.11**
Quantization	55.67	88.95	70.44	75.53	67.11	61.76	44.22	4.69

Table V
Class-Wise Performance Comparison Between the Proposed Models and Other State-of-the-Art Models on the ISPRS Vaihingen Dataset

Method	Imp. Surf.	Buil.	Low Veg.	Tree	Car
DANet [31]	86.33	91.20	71.63	79.91	**76.75**
GFFNet [14]	**87.39**	92.11	71.71	80.68	77.23
RSSFormer [13]	86.75	91.81	71.18	80.42	77.53
STDSNet [12]	86.09	**92.19**	**72.81**	**81.18**	76.56
SwinCNN [16]	85.46	91.54	72.65	80.71	75.17
ST-UNet [17]	85.25	91.55	72.34	80.91	74.38
Pruning	86.14	90.98	72.01	80.53	76.56
Quantization	86.10	90.95	71.76	79.96	76.41

and *Tree* classes with mean IoUs of 92.19%, 72.81%, and 81.18%, respectively, while DANet performs best in the *Car* class with a mean IoU of 76.75%. Pruning achieves a mean IoU of 86.14% for *Impervious Surface*, 90.98% for *Building*, 72.01% for *Low Vegetation*, 80.53% for *Tree*, and 76.56% for *Car*. Although it does not top any individual class, pruning delivers consistently strong results, particularly in Low Vegetation, where it ranks among the highest scores. Similarly, quantization achieves mean IoUs of 86.10% for *Impervious Surface*, 90.95% for *Building*, 71.76% for *Low Vegetation*, 79.96% for *Tree*, and 76.41% for *Car*. While slightly trailing pruning in *Impervious Surface* and *Building*, quantization remains competitive, especially in *Low Vegetation* and *Tree*, though with slightly lower scores compared to the leading models.

Overall, pruning and quantization demonstrate robust and consistent performance across all classes, underscoring their effectiveness in semantic segmentation tasks on the ISPRS Vaihingen dataset. This comparison highlights the competitive standing of our proposed methods in relation to the latest state-of-the-art models, particularly in terms of class-wise performance.

In short, the discussion of our class-wise performance analysis can be summarized as follows:

- Pruning and quantization demonstrate strong performance in segmenting large objects. Notably, on the UAVid dataset, pruning leads to improved performance in some classes.
- However, the proposed solution shows limitations in recognizing small-sized objects. While this is a common challenge for existing DL models, the proposed model particularly struggles with this issue, highlighting significant potential for developing more effective solutions.

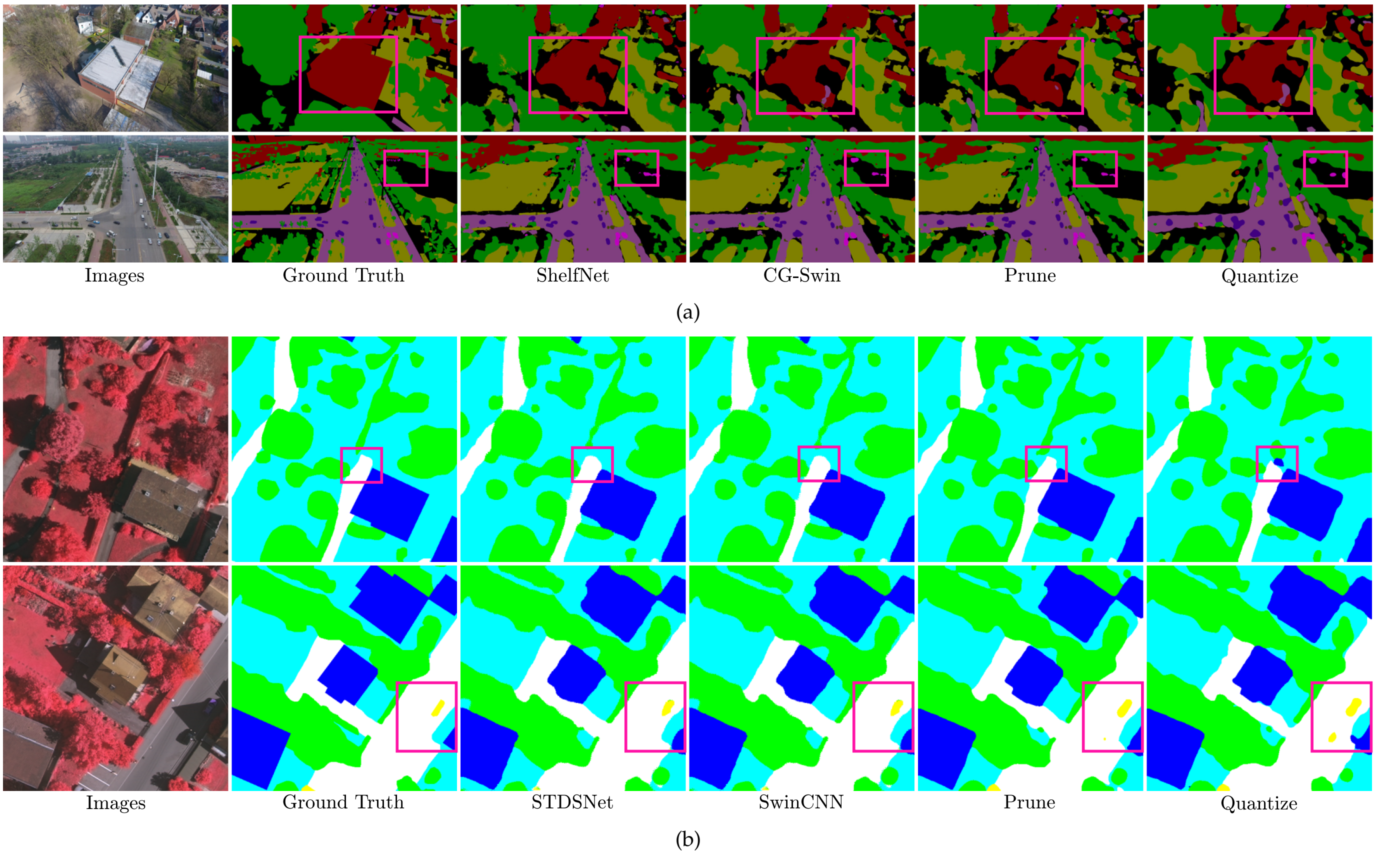
Visualization analysis: Building on the strong performance of pruning and quantization as demonstrated by quantitative metrics, we further explore their segmentation capabilities through visual comparisons. We evaluate the segmentation outputs of our methods against state-of-the-art models that have shown high performance in previous analyses.

For the UAVid dataset, we compare the segmentation results of pruning and quantization with those of ShelfNet and CG-Swin. Similarly, for the ISPRS Vaihingen dataset, we present segmentation masks generated by STDSNet and SwinCNN alongside those produced by our pruning and quantization methods. These comparisons are visually represented in Figure 3. Each dataset includes two sets of images, focusing on the ground truth and the segmentation masks produced by the compared models. While methods like ShelfNet and CG-Swin achieve effective object segmentation, they often struggle with accurately delineating boundaries, particularly for closely situated objects. In the UAVid dataset, although our proposed models have lower complexity compared to ShelfNet and CG-Swin, they maintain a segmentation accuracy comparable to other state-of-the-art models. However, there are instances where large fields are misclassified, as indicated by the pink rectangle, highlighting the challenges in developing efficient methods that adapt well to remote sensing images.

In the ISPRS Vaihingen dataset, even advanced models like STDSNet and SwinCNN, which integrate convolutional layers with transformer self-attention mechanisms, encounter difficulties in achieving fine-grained boundary delineation between high-resolution objects. This issue is particularly evident in the pink rectangular areas, where these models show reduced accuracy. Interestingly, pruning and quantization follow closely behind the compared models in terms of segmentation accuracy. However, there remains room for improvement, particularly in handling small-sized object classes, where further refinement is needed to achieve sharper segmentation mask borders.

4 Conclusion

In this paper, we have presented novel semantic segmentation methods for RSIs, focusing on reducing computational complexity while preserving accuracy. By leveraging pruning and quantization techniques, we have demonstrated that it is possible to significantly reduce model complexity and memory usage without compromising segmentation accuracy. Our proposed



(a)

(b)

Figure 3. Visualization results (as segmentation masks) of our proposed methods and other state-of-the-art models based on different datasets: (a) UAVid and (b) ISPRS Vaihingen. The pink rectangles are used to emphasize the regions that show a clear improvement in segmentation accuracy. Prune and Quantize denote the standard DeepLabV3+ with the combination of pruning and quantization, respectively.

pruning method effectively identifies and eliminates less significant weights in the network, leading to a more efficient model that maintains high performance. The quantization approach further optimizes memory utilization by converting network weights into a more compact data type, making our methods well-suited for deployment on low-cost hardware.

Extensive experiments on diverse RSI datasets, including UAVid and ISPRS Vaihingen, have shown that our methods achieve superior performance compared to state-of-the-art models, particularly in scenarios where computational resources are limited. The results highlight the effectiveness of our approaches in both urban and natural landscapes, reinforcing the potential of pruning and quantization in practical remote sensing applications. Future work could explore further optimizations, such as integrating more advanced pruning strategies or combining our methods with other lightweight architectures. Additionally, expanding the application of these techniques to other types of RSIs or integrating them into real-time processing pipelines could further enhance their utility in operational settings.

Acknowledgment

This work is supported by Ho Chi Minh City University of Technology and Education (HCMUTE) under Grant No. SV2024-164.

References

[1] A. M. Poncet, T. Knappenberger, C. Brodbeck, M. Fogle, J. N. Shaw, and B. V. Ortiz, “Multispectral UAS Data Accuracy for Different Radiometric Calibration Methods,” *Remote Sensing*, vol. 11, no. 16, Aug 2019.

[2] A. Radman, N. Zainal, and S. A. Suandi, “Automated segmentation of iris images acquired in an unconstrained environment using HOG-SVM and GrowCut,” *Digital Signal Processing*, vol. 64, pp. 60–70, Feb 2017.

[3] Y. Zhang, Q. Hu, H. Li, J. Li, T. Liu, Y. Chen, M. Ai, and J. Dong, “A Back Propagation Neural Network-Based Radiometric Correction Method (BPNNRCM) for UAV Multispectral Image,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 112–125, Nov 2023.

[4] G.-V. Nguyen and T. Huynh-The, “Enhancing Aerial Semantic Segmentation With Feature Aggregation Network for DeepLabV3+,” *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, Jul 2024.

[5] T. Huynh-The, S. N. Truong, and G.-V. Nguyen, “HBSeNet: A Hybrid Bilateral Network for Accurate Semantic Segmentation of Remote Sensing Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 14 179–14 193, Aug 2024.

[6] K. Nogueira, M. Dalla Mura, J. Chanussot, W. R. Schwartz, and J. A. Dos Santos, “Dynamic multicontext segmentation of remote sensing images based on convolutional networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 10, pp. 7503–7520, Jun 2019.

[7] A. Li, L. Jiao, H. Zhu, L. Li, and F. Liu, “Multitask semantic boundary awareness network for remote sensing image segmentation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, Feb 2021.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr 2018.

[9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Proceedings of the European Conference on Computer Vision. (ECCV)*, Munich, Germany, Sep. 2018, pp. 833—-851.

[10] S. Xiang, Q. Xie, and M. Wang, "Semantic Segmentation for Remote Sensing Images Based on Adaptive Feature Selection Network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Jan. 2022.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proceedings of the Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. CA, USA: Curran Associates, Inc., 2017.

[12] X. Zhou, L. Zhou, S. Gong, S. Zhong, W. Yan, and Y. Huang, "Swin Transformer Embedding Dual-Stream for Semantic Segmentation of Remote Sensing Imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 175–189, 2024.

[13] R. Xu, C. Wang, J. Zhang, S. Xu, W. Meng, and X. Zhang, "RSSFormer: Foreground Saliency Enhancement for Remote Sensing Land-Cover Segmentation," *IEEE Transactions on Image Processing*, vol. 32, pp. 1052–1064, Jan. 2023.

[14] Y. Cao, C. Huo, S. Xiang, and C. Pan, "GFFNet: Global Feature Fusion Network for Semantic Segmentation of Large-Scale Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 4222–4234, 2024.

[15] X. Meng, Y. Yang, L. Wang, T. Wang, R. Li, and C. Zhang, "Class-Guided Swin Transformer for Semantic Segmentation of Remote Sensing Imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Oct. 2022.

[16] C. Zhang, W. Jiang, Y. Zhang, W. Wang, Q. Zhao, and C. Wang, "Transformer and CNN Hybrid Deep Neural Network for Semantic Segmentation of Very-High-Resolution Remote Sensing Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–20, 2022.

[17] X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao, and Y. Xue, "Swin Transformer Embedding UNet for Remote Sensing Image Semantic Segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[18] S. Liu, J. Cheng, L. Liang, H. Bai, and W. Dang, "Light-Weight Semantic Segmentation Network for UAV Remote Sensing Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8287–8296, Aug. 2021.

[19] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 325–341.

[20] J. Zhuang, J. Yang, L. Gu, and N. Dvornek, "ShelfNet for Fast Semantic Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Seoul, Korea (South), 2019, pp. 847–856.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, Jun 2016, pp. 770–778.

[22] T. B. Olaf Ronneberger, Philipp Fischer, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Cham, 2015, pp. 234–241.

[23] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct 2017, pp. 2980–2988.

[24] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang, "UAVid: A semantic segmentation dataset for UAV imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 165, pp. 108–119, Jul 2020.

[25] R. Niu, X. Sun, Y. Tian, W. Diao, K. Chen, and K. Fu, "Hybrid Multiple Attention Network for Semantic Segmentation in Aerial Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, Mar. 2022.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, pp. 248–255.

[27] R. Li, S. Zheng, C. Zhang, C. Duan, J. Su, L. Wang, and P. M. Atkinson, "Multiattention Network for Semantic Segmentation of Fine-Resolution Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

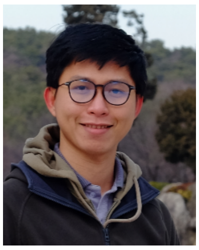
[28] L. Ding, J. Zhang, and L. Bruzzone, "Semantic Segmentation of Large-Size VHR Remote Sensing Images Using a Two-Stage Multiscale Training Architecture," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 8, pp. 5367–5376, 2020.

[29] G. Csurka, D. Larlus, and F. Perronnin, "What is a good evaluation measure for semantic segmentation?" in *Proceedings of the 24th British Machine Vision Conference 2013 (BMVC)*, Bristol, 2013, pp. 32.1–32.11.

[30] L. Wang, R. Li, D. Wang, C. Duan, T. Wang, and X. Meng, "Transformer Meets Convolution: A Bilateral Awareness Network for Semantic Segmentation of Very Fine Resolution Urban Scene Images," *Remote Sensing*, vol. 13, no. 16, Aug. 2021.

[31] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3146–3154.

Gia-Vuong Nguyen received the B.S. degree in Embedded Systems and Internet of Things from Ho Chi Minh City University of Technology and Education, Vietnam, in 2024. He was a recipient of the REV-ECIT Best Paper Award in 2023. His current research interests include digital image processing, radio signal processing, computer vision, machine learning, and DL.



Thien Huynh-The (IEEE Senior Member) received the Ph.D. degree in Computer Science and Engineering from Kyung Hee University (KHU), South Korea, in 2018. He was a recipient of the Superior Thesis Prize awarded by KHU. From March 2018 to August 2018, he was a Postdoctoral Researcher with Ubiquitous Computing Laboratory, KHU. From September 2018 to May 2022, he was a Postdoctoral Researcher with ICT Convergence Research Center, Kumoh National Institute of Technology, South Korea. He is currently a Lecturer in Department of Computer and Communications Engineering, HCMUTE, Vietnam. He was a recipient of Golden Globe Award 2020 for Vietnamese Young Scientist and IEEE ATC Best Paper Award in 2023. His current research interests include digital image processing, radio signal processing, computer vision, wireless communications, IoT applications, machine learning, and DL. He is currently serving as an Editor of IEEE COMML.