Energy-Efficient Real-Time Crop and Weed Segmentation for UAVs using Resource-Aware Dynamic Pruning

Aditya Vidiyala, Jashwanth Aravapalli, Sri Sreshta, Taruni
Department of Computer Science (AI & ML)
Keshav Memorial Engineering College
Hyderabad, India
Mentor: Naga Hari Babu KV
https://github.com/adityavidiyala/ReViT-CropAndWeed

***Abstract*—Unmanned Aerial Vehicles (UAVs) play a pivotal role in precision agriculture, enabling site-specific weed management through real-time semantic segmentation. However, deploying deep learning models on these edge devices is difficult due to strict limits on onboard energy and computing power. Existing solutions often rely on static, lightweight architectures that sacrifice too much accuracy for speed, failing to adapt when mission conditions change.**

This paper presents an energy-efficient, real-time crop and weed segmentation system using a Resource-Aware Dynamic (RDD) Pruning framework. Unlike static models, our approach introduces a "self-aware" mechanism that dynamically adjusts the network architecture in real-time based on available computational resources and energy constraints. We implement a dual-pruning strategy that combines dynamic encoder block skipping with decoder channel pruning to optimize the trade-off between segmentation fidelity and system efficiency.

Experimental results demonstrate that our system effectively manages this balance, reducing computational costs from 25.24 GFLOPs to 20.61 GFLOPs and decreasing latency from 178 ms to 104 ms. This achieves a 1.7x speedup while maintaining a competitive mean Intersection over Union (mIoU) of 85.6% in eco-mode. Simulations confirm that this dynamic adaptation extends drone mission duration without compromising critical segmentation tasks.

***Index Terms*—mIoU - Mean Intersection Over Union, GFLOPS - Giga Floating Point Operations**

I. Introduction

Feeding a growing global population—projected to reach nearly 10 billion by 2050—without destroying the environment is one of the biggest challenges of our time [1]. To meet this demand, traditional agriculture has relied heavily on the application of chemical herbicides. While effective in the short term, this "spray everything" approach comes at a steep cost: it contaminates soil and water, degrades biodiversity, and has accelerated the evolution of herbicide-resistant "superweeds" [2].

In response, the industry is shifting toward **Precision Agriculture (PA)**, specifically **Site-Specific Weed Management (SSWM)**. Unlike traditional methods that treat entire fields as uniform, SSWM treats every square meter—or even every plant—individually [3]. This paradigm shift has given rise to autonomous robotic systems, including Unmanned Aerial Vehicles (UAVs) and ground-based rovers, capable of targeting weeds with micro-doses of herbicide or mechanical removal tools.

However, the hardware is only as good as its software. For a robot to intervene effectively, it must first perceive its environment. This places the research squarely in the domain of **Computer Vision for Agricultural Robotics**, specifically **Semantic Segmentation**. The success of these autonomous systems hinges on a single, critical capability: the ability to instantly and accurately distinguish a crop from a weed at the pixel level, regardless of lighting conditions or plant density [4]. While early methods relied on simple color indices, modern approaches have increasingly turned to Deep Learning (DL) to handle the complex, unstructured nature of real-world agricultural environments [5].

The Reality of "Edge" Computing: While deep learning has revolutionized computer vision in the lab, bringing it to a muddy agricultural field is a different story. Agricultural robots and drones don't have the luxury of running on powerful cloud servers. They operate on "edge devices"—small, onboard computers with limited battery life and processing power [6].

This creates a frustrating trade-off for researchers. You can use a heavy, complex model that catches every weed but drains the robot's battery in an hour [6]. Or, you can use a lightweight model that runs all day but misses the small, tricky weeds that actually cause crop damage [7].

The primary objectives of this research are to develop a system that addresses the trade-off between segmentation accuracy and computational efficiency on edge devices. Specifically, this study aims to:

- **Develop a Resource-Dependent Dynamic (RDD) pruning framework** capable of reducing computational costs (GFLOPs) and latency by adapting to real-time resource constraints.
- **Enhance the standard SegFormer architecture** with an **ExG-Guided Attention** module to improve the seg-

Identify applicable funding agency here. If none, delete this.

mentation accuracy of vegetation against complex backgrounds.
- **Integrate a Detail-Preserving Decoder** to resolve inter-class similarities and ensure the detection of early-stage weeds at the cotyledon growth stage

This work introduces a novel, resource-aware semantic segmentation system tailored for agricultural UAVs. The main contributions include the development of a **Resource-Dependent Dynamic (RDD)** inference strategy that dynamically adjusts the model architecture based on real-time hardware constraints, achieving a 1.7x speedup over static baselines. Additionally, we propose a specialized **ExG-Guided Attention module** and a **Detail-Preserving Decoder** that specifically target the spectral characteristics of vegetation. This approach significantly improves the detection of morphologically similar crops and weeds compared to standard vision transformers, ensuring reliable performance even under varying field conditions

The rest of the paper is organized as follows: Section 2 reviews the evolution of segmentation architectures and their limitations. Section 3 presents the proposed RDD framework and architectural enhancements. Section 4 discusses the experimental results and ablation studies, and Section 5 concludes the study.

II. Literature Survey

For the past decade, Convolutional Neural Networks (CNNs) have served as the backbone of agricultural robotics. Architectures such as U-Net, Mask R-CNN, and DeepLabV3+ became the industry standard for tasks ranging from fruit counting to weed detection [11]. These models rely on convolution operations, which are effective at extracting local texture features. However, recent studies indicate that CNNs suffer from a limited "receptive field" [8]. In simpler terms, a CNN looks at an image through a small window. It might recognize a green leaf, but it often struggles to understand the broader context—such as the spacing pattern of a crop row versus the random clustering of weeds. This lack of global context often leads to misclassifications in complex, unstructured fields [12].

This limitation drove the adoption of Vision Transformers (ViTs). Unlike CNNs, ViTs utilize self-attention mechanisms to model global context, allowing every pixel to compare itself to every other pixel [6]. While this provides a complete understanding of the scene, standard ViTs come with two major drawbacks for robotics. First, the self-attention mechanism is computationally expensive with quadratic complexity [6]. As image resolution increases—which is necessary to see small weeds—the processing time explodes, making them too slow for real-time drone flight. Second, standard ViTs process images in a "columnar" fashion, outputting features at a single, low resolution [6]. This is disastrous for segmentation, which needs high-resolution details to draw precise boundaries around leaves. Consequently, they miss the fine edges that CNNs capture effectively.

SegFormer, introduced by Xie et al. [9], emerged as a dominant architecture in this domain to bridge this gap. It combines the hierarchical feature extraction of CNNs with efficient attention mechanisms to capture both fine details and global context. While SegFormer significantly improves segmentation accuracy in challenging lighting conditions, it introduces a high computational burden, often making it unsuitable for battery-powered edge devices [13]. Furthermore, even SegFormer retains critical inefficiencies when applied to precision agriculture. Primarily, it utilizes **static inference**, processing a clear image of bare soil with the exact same computational effort (GFLOPs) as a complex image full of tangled weeds [6]. It lacks the intelligence to "relax" when the task is easy. Additionally, despite its hierarchical design, the standard encoder struggles with **inter-class similarity**, often blurring the fine boundary lines between crops and morphologically similar weeds in dense vegetation [10]. Most importantly, effective weed management requires intervention at the early growth stages (cotyledon stage) [7]. Standard SegFormer architectures often lose the tiny, high-frequency details of these small plants during the down-sampling process. If the model cannot detect these minute weeds, the robotic intervention fails exactly when it is most needed.

To address these computational constraints, researchers have proposed various acceleration techniques, which generally fall into three categories:

Static Model Compression Initial efforts focused on static compression techniques such as Quantization (reducing weight precision) and Knowledge Distillation (training smaller student models) [14]. While effective at reducing model size, these methods produce a permanently static architecture. They cannot adapt to real-time changes in mission conditions; a quantized model offers the same performance whether the drone is critically low on battery or fully charged.

Input-Dependent "Early-Exit" Mechanisms A more adaptive approach involves "Early-Exit" networks, such as MSDNet or DynaBERT, which insert classifiers at intermediate layers to stop processing "easy" images early [15]. However, Venkatesan et al. [6] argue that these methods are ill-suited for semantic segmentation. Segmentation requires high-resolution feature reconstruction from the final decoder layers to delineate object boundaries; exiting early often results in coarse, unusable masks for weeding tasks.

Resource-Dependent Dynamic (RDD) Inference The most relevant advancement for this work is Resource-Dependent Dynamic (RDD) inference, which tailors the model architecture to specific hardware constraints rather than input data. Venkatesan et al. [6] conducted a comprehensive profiling of Vision Transformers (specifically SegFormer and Swin Transformer) and identified a **"FLOPs Paradox"**. They found that while FLOPs are traditionally used to estimate latency, computational bottlenecks in modern Transformers often lie in the convolutional decoder head rather than the self-attention mechanism. Specifically, the `Conv2DFuse` layer in the SegFormer decoder accounts for nearly 62% of the total FLOPs, far outweighing the cost of attention blocks. Based on these findings, they proposed a dual-strategy for RDD:

- **Dynamic Encoder Block Skipping:** This involves se-

lectively bypassing transformer blocks in the encoder to reduce attention costs.

- **Decoder Channel Pruning:** This involves dynamically reducing the input channels to the heavy `Conv2DFuse` layer to alleviate the primary computational bottleneck.

Their experiments demonstrated that SegFormer is surprisingly resilient to this pruning. By switching between these dynamically pruned configurations, they achieved a 28% reduction in energy consumption with only a 1.4% drop in accuracy on the ADE20K dataset. Importantly, they highlighted that "pruning without retraining" is a viable strategy for moderate resource constraints, offering significant flexibility for real-time edge deployments

Despite the promise of RDD inference, existing frameworks remain generic. The pruning strategies proposed in [6] are mathematical and "blind" to the agricultural context. They do not prioritize features specific to vegetation, such as the Excess Green (ExG) index. Furthermore, current dynamic models do not explicitly address the **inter-class similarity** problem between crops and weeds at the cotyledon stage. When these models prune themselves to save energy, they risk losing the high-frequency details required to distinguish a young crop from a weed. This paper aims to bridge this gap by integrating **ExG-Guided Attention** into the dynamic pruning framework.

III. Dataset

To evaluate the proposed architecture under challenging real-world conditions, we utilized the **CropAndWeed** dataset [16], a large-scale benchmark for precision agriculture. Unlike synthetic or controlled-environment datasets, CropAndWeed captures highly variable field conditions, including diverse soil types, varying moisture levels, and complex lighting scenarios such as direct sunlight and diffuse shadows.

For this study, we specifically selected the CropAndWeed2 variant of the dataset. While the original dataset contains 74 distinct semantic classes, CropAndWeed2 aggregates these into a binary segmentation task consisting of two super-classes:Crop and Weed. This mapping aligns with the operational requirements of autonomous weeding robots, which primarily need to distinguish between "plants to preserve" and "plants to remove." After processing valid image-mask pairs, our experimental dataset consists of **7,704 images**, providing a substantial foundation for learning robust vegetative features.

- **Crop Class:** This category aggregates multiple economic crops across various growth stages to ensure the model generalizes to different plant morphologies. It includes **Maize** (from two-leaf to eight-leaf stages), **Sugar Beet**, **Soybean**, **Sunflower**, **Potato**, **Pea**, **Common Bean**, **Faba Bean**, and **Pumpkin**.
- **Weed Class:** This category encompasses a wide diversity of unwanted vegetation, including **Grasses** (e.g., Cockspur grass, Annual meadow grass), **Broadleaf weeds** (e.g., Thistles, Goosefoot, Knotweed), and other common agricultural intruders.

By training on this binary variant, the system learns robust features that separate valuable crops from a heterogeneous background of weeds and soil, regardless of specific species sub-types.

Fig. 1. Images from the Dataset[16]

IV. Proposed Work

This section details the development of a resource-aware segmentation system tailored for precision agriculture. Our approach builds upon the **SegFormer B2** architecture (originally pre-trained on ADE20K at 512x512 resolution) by integrating domain-specific modules and a dynamic inference mechanism.

A. 4.1 Overview of Original Segformer Architecture

Before detailing our modifications, it is essential to understand the baseline SegFormer framework [9]. Unlike standard Vision Transformers (ViT) that generate single-scale, low-resolution features, SegFormer employs a hierarchical Mix Transformer (MiT) encoder. This encoder generates multi-scale feature maps across four stages (F_1, F_2, F_3, F_4) with resolutions of $1/4, 1/8, 1/16$, and $1/32$ of the original image, respectively. This pyramidal structure allows the model to capture both high-resolution details (in early stages) and coarse semantic context (in deep stages).The standard SegFormer decoder is a lightweight "All-MLP" design. It projects multi-level features to a uniform channel dimension, upsamples them to a common resolution ($1/4$), concatenates them, and fuses them using a single 1×1 convolutional layer (often referred to as Conv2DFuse). While efficient, Venkatesan et al. noted that this fusion layer can become a computational bottleneck, and the simple aggregation strategy often blurs fine boundary details required for agricultural segmentation.

B. 4.2 Architectural Enhancements

To address the limitations of standard Vision Transformers in agricultural settings—specifically the loss of high-frequency details and the inability to prioritize vegetation—we introduce two novel architectural modifications: the **ExG-Guided Attention Module** and the **Parallel Detail-Preserving Decoder**.

C. 4.2.1 ExG-Guided Feature Gating

The proposed architecture incorporates a lightweight **ExG-Guided Attention Module** that computes an Excess Green (ExG) probability mask directly from the raw input image, establishing a strong prior for vegetation. Standard models typically treat all pixels with equal initial importance, which is inefficient in agricultural scenes often dominated by soil. By explicitly boosting 'green' features, this module suppresses the background and significantly improves the detection of crops and weeds.

First, we compute the ExG index for the input image $I \in \mathbb{R}^{H \times W \times 3}$. Let Red (r), Green (g), and Blue (b) represent the normalized color channels. The ExG index is calculated as:

$$ExG_{raw} = 2 \cdot g - r - b$$

To convert this raw index into a usable attention map, we apply a Sigmoid activation function, squashing the values to a [0, 1] probability range:

$$M_{ExG} = \sigma(ExG_{raw})$$

The SegFormer encoder produces hierarchical feature maps, with the final stage (F4) containing the richest semantic information but the lowest spatial resolution (1/32 of input). To prioritize vegetation, we downsample the M_{ExG} mask to match the resolution of F4. We then apply element-wise multiplication (gating):

$$S4_{refined} = F4 \otimes \text{Downsample}(M_{ExG})$$

This operation explicitly dampens feature activations in non-green regions, ensuring the decoder receives "clean" semantic features focused purely on plant matter.

D. 4.2.2 Parallel Detail-Preserving Decoder

Standard decoder used in the original segformer architecture often lose fine boundary details during feature aggregation. To address the "cotyledon detection" challenge, we designed a **Parallel Detail-Preserving Decoder** that relies on concatenation and explicit residual injection.

The encoder outputs four stages of features $[F1, F2, F3, F4_{refined}]$ with varying channel depths (e.g., $[64, 128, 320, 512]$). First, 1×1 "lateral convolutions" project all stages to a common embedding dimension C (256). Next, the lower-resolution maps $(F2, F3, F4_{refined})$ are bilinearly upsampled to match the resolution of the high-detail Stage 1 map $(F1)$.

Instead of summing features which can wash out details, we concatenate the upsampled maps along the channel dimension, creating a unified feature tensor of depth $4 \times C$. A fusion layer then compresses this information back to dimension C:

$$F_{fused} = \text{GeLU}\left(\text{BN}\left(\text{Conv}_{fuse}\left(\text{Concat}(F_1, F_2, F_3, F_4)\right)\right)\right) \quad (1)$$

To guarantee that the fine edges captured in the earliest layer are not lost during fusion, we apply a hard residual connection. We explicitly add the projected F1 features back to the fused output:

$$F_{final} = F_{fused} + F1_{projected}$$

This connection forces the model to retain high-frequency spatial information (leaf edges and very smaller crops and weeds) even while processing deep semantic context, directly addressing the inter-class similarity problem.

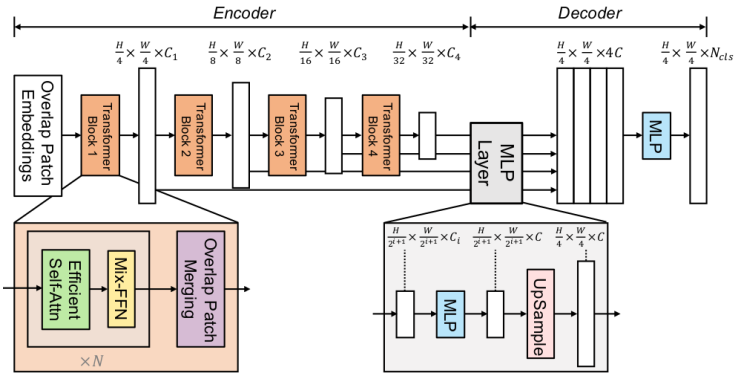

Fig. 2. Original Segformer Architecture [3]

E. 4.3 Resource Dependent Dynamic (RDD) Integration

Drones and other robotic systems needs to adapt to the fluctuating energy constraints such as diminishing battery levels or thermal throttling. We integrated a Resource Dependent Dynamic inference framework. Unlike "input-dependent" methods that exit early based on image difficulty, our RDD system adjusts the model's architectural depth and width based on available hardware resources. Building on the principles established by Venkatesan et al.[6], we implemented a **Dual-Pruning Strategy** that targets two specific computational levers within the network.

*a) **Dynamic Encoder Block Skipping**:* The Mix Transformer (MiT) encoder consists of four stages with varying depths (e.g., the B2 backbone has [3, 4, 6, 3] transformer blocks per stage). While deep blocks refine semantic features, they are computationally expensive due to the quadratic complexity of self-attention. We introduced a **Dynamic Block Wrapper** around each encoder stage. This wrapper accepts a binary mask at runtime, allowing specific transformer blocks to be bypassed entirely during the forward pass. Taking an example, when the model is in low-power mode(Eco Mode), we selectively skip the deepest blocks in Stage 3 and Stage 4. Since the ExG-Gating mechanism (Section 4.2.1) is applied at the *end* of the encoder, skipping internal blocks does not break the feature flow. It merely provides slightly less refined features to the gating mechanism. The strong biological prior from the ExG mask compensates for this reduction in depth, maintaining segmentation robustness even when encoder capacity is reduced.

*b) **Dynamic Decoder Channel Pruning**:* Unlike the baseline SegFormer analyzed in, where the decoder constituted the primary computational bottleneck, our architecture shifts the "heavy lifting" to the **Encoder** through the ExG-Gating mechanism. This ensures that the system prioritizes feature extraction over complex feature aggregation.

However, the `linear_fuse` layer in our decoder still receives a high-dimensional concatenated tensor (4 x 256 channels). While this is no longer a performance bottleneck in terms of latency, it remains a source of redundancy. To maximize energy efficiency in Eco-modes without compro-

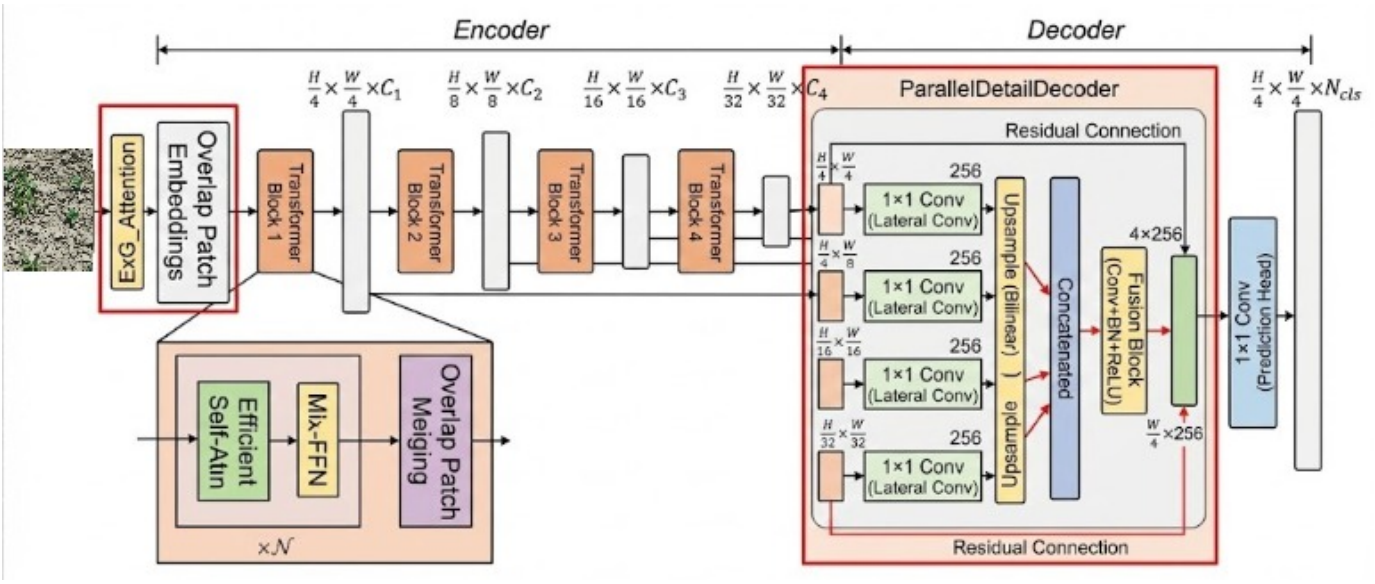

Fig. 3. The proposed architecture with ExG_Module and Parallel Detail Decoder

mising the critical encoder operations, we apply **Dynamic Channel Slicing** to this fusion layer. Instead of retraining separate models, we perform runtime weight slicing:

$$W_{active} = W_{fusion}[:, : K, :, :]$$

Where K is the number of active input channels (e.g., reducing from 1024 to 768). Since our system relies on the **Encoder** for its primary detection logic (ExG priors), the deep, abstract features in the decoder fusion layer become less critical for simple segmentation tasks. In constrained scenarios, we prune these channels (K), effectively reducing the memory bandwidth of the decoder while leaving the computationally dense Encoder fully active (or lightly pruned) to maintain high-fidelity vegetation detection. This strategy is uniquely enabled by our Parallel Detail-Preserving Decoder. Because we enforce a hard residual connection ($F_{final} = F_{fused} + F_1$), the system retains the high-frequency spatial details from the earlier layers even when the fusion layer is pruned. This allows us to aggressively reduce the decoder's footprint for energy savings without "breaking" the segmentation mask.

TABLE I
SEGFORMER B2 MODEL CONFIGURATIONS

Config	E1	E2	E3	E4	Channels
B2 Full	3	4	6	3	1024
B2$_a$	2	4	6	3	980
B2$_b$	2	4	6	3	920
B2$_c$	2	3	6	3	900
B2$_d$	2	3	5	3	880
B2$_e$	2	3	5	2	832
B2$_f$	2	3	5	2	768

*c) **RDD Configurations and Integrations**:* Unlike standard dynamic inference systems that rely solely on input complexity, our proposed framework operates as a "Self-Aware" Control System. It continuously monitors three critical telemetry vectors—**State of Charge (SoC)** i.e Current battery level percentage , **Thermal Status** T_{GPU}, and **Flight Velocity** v—to determine the optimal model configuration $\mathcal{C}_{opt}$ in real-time.

The decision logic is formulated as a hierarchical constraint satisfaction problem:

1. Energy Budget (Battery Constraint): The system first establishes an upper bound on computational complexity (GFLOPs) based on the remaining battery level. We define discrete energy tiers to ensure mission endurance: *Performance Tier* ($SoC \geq 80\%$): Permits maximum complexity ($\approx$ 26 GFLOPs), enabling the B2_Full configuration for high-fidelity segmentation.

Efficiency Tier ($40\% \leq SoC < 60\%$): Enforces a "Balanced" mode ($\leq$ 23 GFLOPs), triggering the skipping of Stage 4 encoder blocks.

Critical Tier ($SoC < 20\%$): Restricts the budget to the minimum floor ($\approx$ 20.7 GFLOPs), forcing the system into B2_Eco mode to maximize survival time.

2. Motion Constraint (Velocity Logic): To ensure complete field coverage without coverage gaps or motion blur, the inference latency (τ) must decrease as drone velocity (v) increases. We enforce a dynamic latency limit defined by the ground sampling requirement:

$$\tau_{limit} = \frac{\kappa}{v}$$

Where κ is a coverage constant derived from the camera's field of view and overlap requirements (experimentally set to 300 in our simulation). If the drone accelerates to high speeds ($v > 8$ m/s), τ_{limit} tightens, forcing the selection of low-latency configurations (e.g., B2_f) regardless of battery level.

3. Thermal Throttling (Safety Override): To prevent hardware damage or thermally induced voltage sags, the system monitors the GPU temperature (T_{GPU}).

Warning State ($T_{GPU} \geq 70°$C): The system caps the complexity budget to 22.0 GFLOPs.

Critical State ($T_{GPU} \geq 80°$C): A hard override is triggered, forcing the lowest-power configuration (B2_f) immediately to mitigate thermal runaway.

4. Final Configuration Selection: The system filters the look-up table for configurations satisfying all active constraints:

$$C_{valid} = \{c \in C_{all} \mid \text{GFLOPs}(c) \leq \text{Budget}_{energy} \wedge \text{Latency}(c) \leq \tau_{limit}\}$$

From this valid set, the selection policy adapts to the context: Normal Operation: The system selects the configuration with the highest mIoU (Accuracy). Critical/High-Thermal Operation: The system selects the configuration with the lowest Latency (Speed) to minimize active duty cycles.

V. Results

This section details the fine-tuning and evaluation of the proposed resource-aware segmentation system on the **CropAndWeed2**.All experiments were conducted on an **NVIDIA RTX 5060** GPU to simulate the deployment environment of modern edge computing platforms. The analysis proceeds in three stages: a preliminary architectural comparison, a detailed analysis of the training dynamics for the enhanced model, and a final quantitative evaluation of the RDD inference configurations.

A. 5.1 Impact of Architectural Enhancements

To strictly isolate the benefits of the proposed **ExG-Guided Attention** and **Parallel Detail-Preserving Decoder**, we first conducted a controlled "rapid-convergence" experiment. We compared the standard SegFormer B2 baseline against our Enhanced Architecture under identical initial conditions.

Both models were initialized with ADE20K pre-trained weights and fine-tuned on the CropAndWeed2 dataset for **only 1 epoch**. This constraint was chosen to evaluate how quickly and effectively each architecture adapts to the agricultural domain before extensive optimization.

Even with this minimal training, the Enhanced Architecture demonstrated superior feature extraction capabilities.

Qualitative Gain: As shown in Figure 5, the standard baseline struggled to differentiate small, scattered weeds from the soil background after just one epoch, often producing noisy, disconnected masks. In contrast, the Enhanced Architecture—leveraging the ExG prior—immediately locked onto vegetative regions, producing sharper and more coherent masks for cotyledon-stage weeds.

Quantitative Gain: The performance gap was statistically significant. While the baseline SegFormer B2 achieved a validation mIoU of **0.7233** and a mean accuracy of **0.8314**, our Enhanced model reached a validation mIoU of **0.7721** and a mean accuracy of **0.8614** under the same conditions.

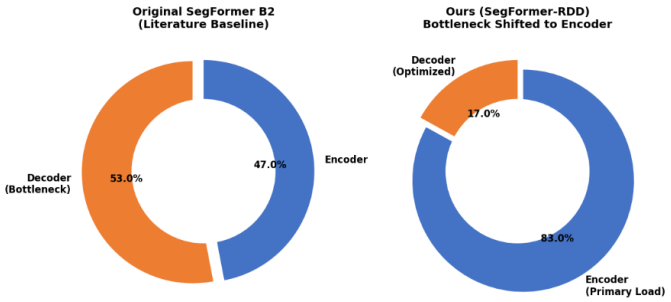

Fig. 4. Enter Caption

This substantial improvement (≈5% in mIoU) confirms that the hard residual connections in the parallel decoder facilitate faster gradient flow and more efficient learning of high-frequency details compared to the standard MLP decoder.

A critical finding of this study is the fundamental shift in the model's computational profile compared to standard Vision Transformers. Previous profiling by Venkatesan et al. [6] identified a "FLOPs Paradox" in the standard SegFormer architecture, reporting that the decoder accounted for approximately 53–62% of the total FLOPs. This bottleneck was primarily driven by the heavy Conv2DFuse layer, which processes high-dimensional features from all four encoder stages.

In contrast, our proposed architecture effectively resolves this decoder bottleneck. By replacing the standard MLP decoder with our efficient Parallel Detail-Preserving Decoder and enhancing the encoder with ExG-Gating, we have inverted the computational distribution.

As quantified in our experiments (and illustrated in Figure 4), the load distribution in our baseline B2_Full configuration (Total: 25.24 GFLOPs) is as follows:

Encoder Load: The enhanced encoder consumes approximately 21.0 GFLOPs, representing 83% of the total computational cost.

Decoder Load: The optimized parallel decoder consumes only 4.24 GFLOPs, representing just 17% of the total cost.

This shift from a "Decoder-Heavy" to an "Encoder-Heavy" profile provides the theoretical justification for our RDD strategy. Since the vast majority of the compute budget is now allocated to feature extraction, our primary energy-saving lever must be Dynamic Encoder Block Skipping. Pruning the encoder yields maximum returns, whereas the decoder is already sufficiently lightweight that further pruning serves primarily to reduce memory bandwidth rather than raw computation.

The Enhanced Architecture was fine-tuned on the CropAndWeed2 dataset for a total of 7 epochs to achieve optimal convergence. We utilized the AdamW optimizer with a learning rate of 6×10^{-5} and a weight decay of 0.01. A batch size of 4 was employed to accommodate the memory requirements of the high-resolution (512×512) tensor operations on the NVIDIA RTX 5060.

Fig 7 illustrates a robust learning trajectory characterized by rapid initial adaptation. By Epoch 2, validation mIoU surged to

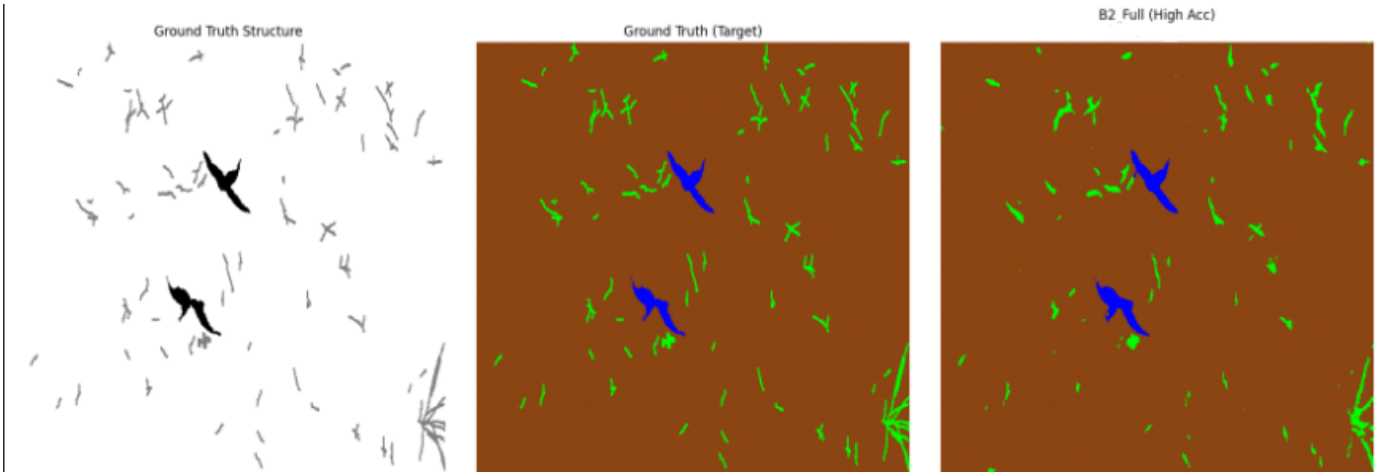

Fig. 5. Visual comparison of Segformer B2 (Fine-Tuned on CropAndWeed) + ExG_module + ParallelDecoderAttention: Ground Truth Structure (Left), Target Mask (Middle), and Fine-Tuned Prediction (Right).

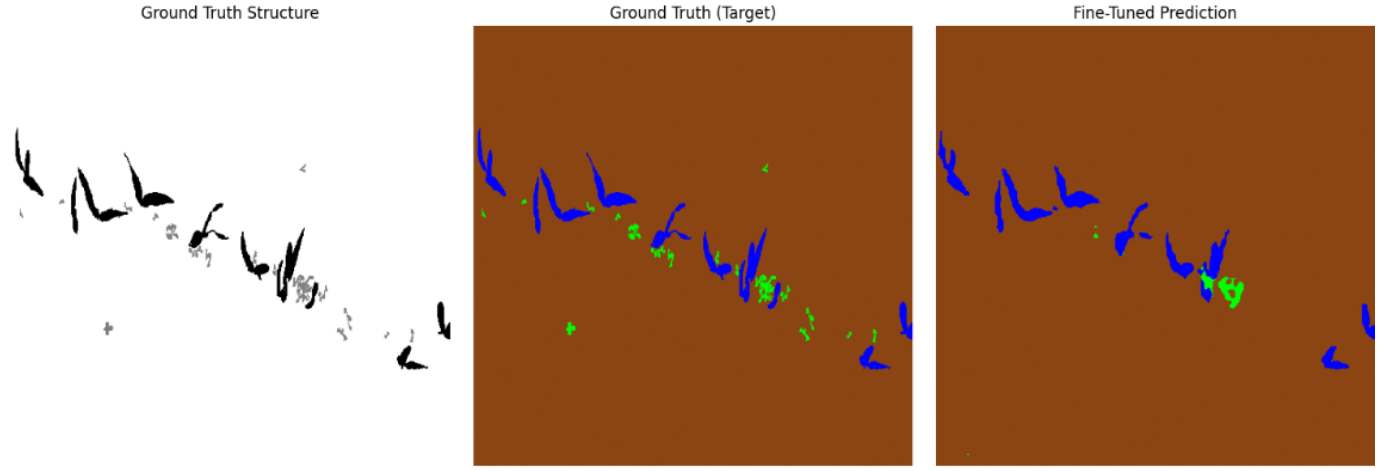

Fig. 6. Visual comparison of Segformer B2(FineTuned On CropAndWeed): Ground Truth Structure (Left), Target Mask (Middle), and Fine-Tuned Prediction (Right).

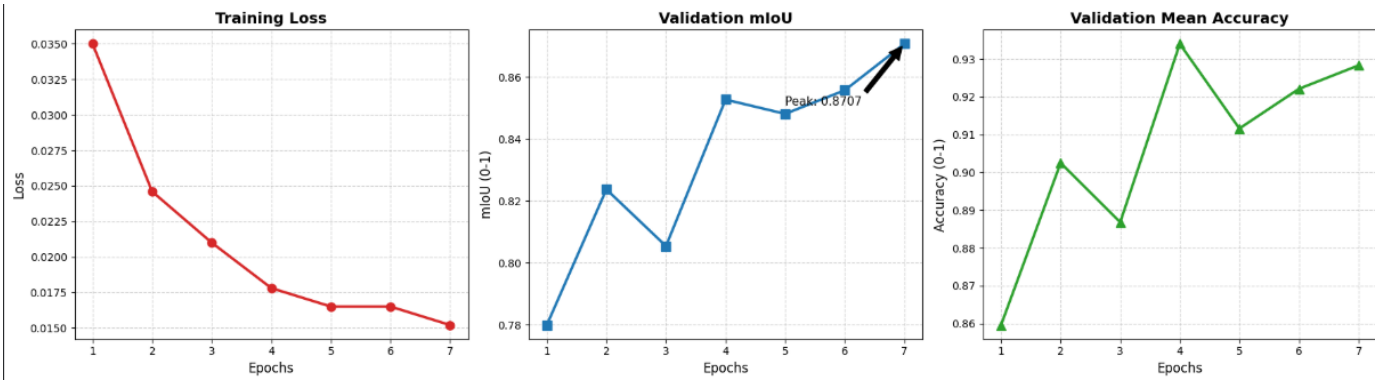

Fig. 7. (a) Training Loss minimized to 0.0152; (b) Validation mIoU peaked at 0.8707; (c) Mean Pixel Accuracy exceeded 92% in the final stages.

0.8237 as the ExG priors successfully guided the encoder toward vegetative features. Despite minor fluctuations in Epoch 3—typical of aggressive learning rate schedules—the model stabilized quickly, reaching a peak mean accuracy of **93.40%** in Epoch 4. Convergence was achieved by Epoch 7,with training loss minimizing to **0.0152** and validation mIoU peaking at **87.07%**. This efficient performance confirms the effectiveness of fine-tuning ADE20K-initialized weights Segformer B2 with

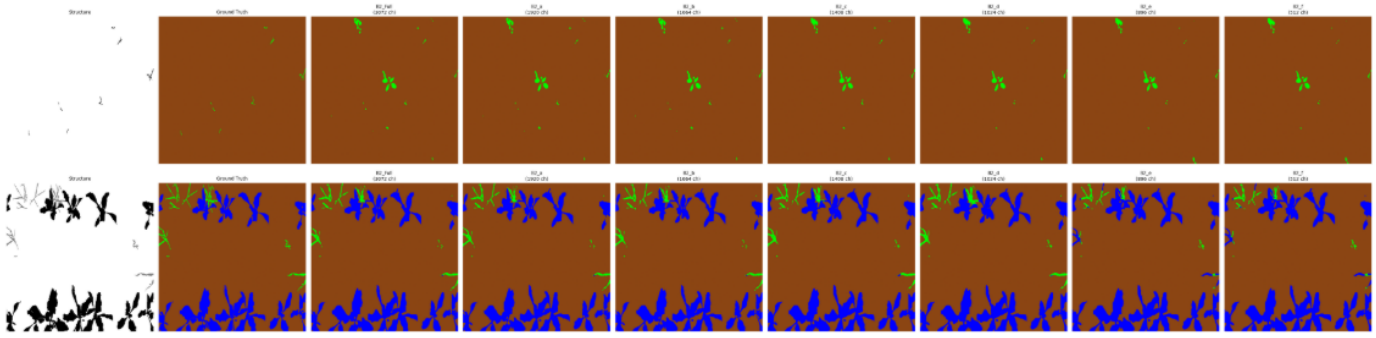

Fig. 8. **(a)** Original Image Structure, **(b)** Ground Truth Mask, **(c)** B2_Full (Baseline, 100% compute), **(d-h)** Pruned configurations B2_a through B2_e, and **(i)** B2_f

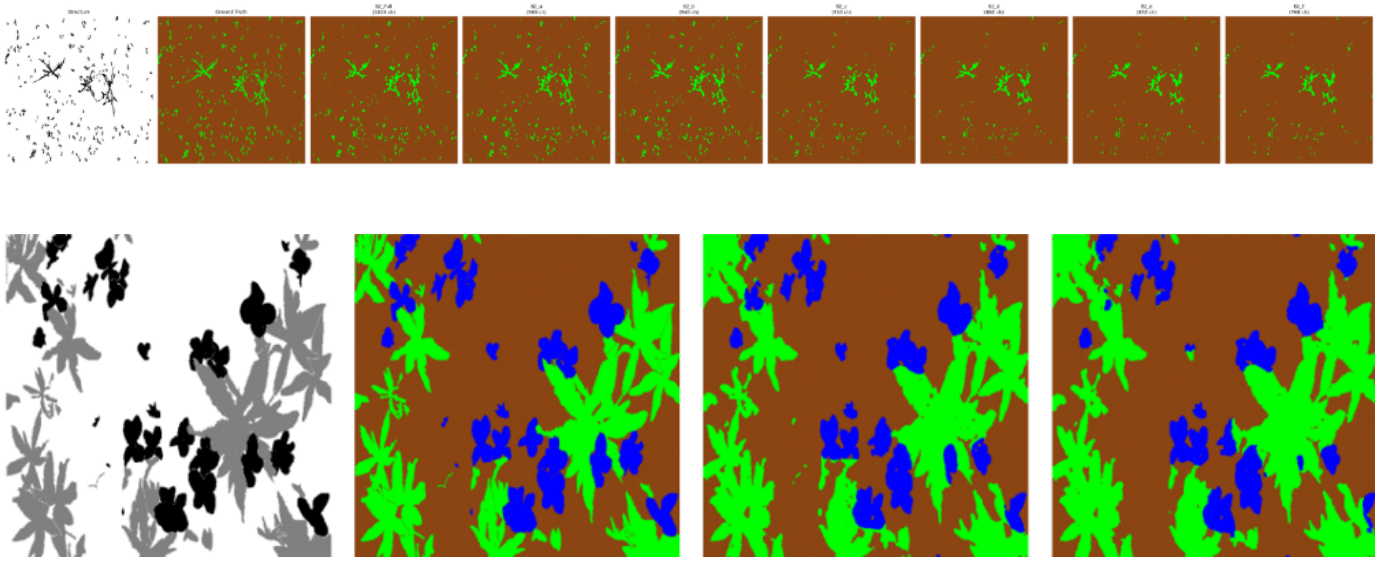

Fig. 9. **(a)** Original Image Structure, **(b)** Ground Truth Mask, **(c)** B2_Full (Baseline, 100% compute), **(b)** B2_f

our domain-specific architectural enhancements.

B. 5.1 RDD Inferencing Analysis

We evaluated the spectrum of RDD configurations defined previously, ranging from maximum fidelity (`B2_Full`) to maximum efficiency (`B2_f`). Table I summarizes the quantitative trade-offs between accuracy (mIoU), theoretical complexity (GFLOPs), and real-world speed (Latency) on the NVIDIA RTX 5060. **Note that for this specific dynamic profiling, the mIoU metrics were computed on a representative subset of 200 images from the validation dataset.**

A detailed analysis of the results reveals a significant discrepancy between theoretical and practical efficiency gains. Transitioning from the **PERF** mode (`B2_Full`) to the **CRITICAL** mode (`B2_f`) results in an **18.34% reduction in theoretical GFLOPs** (from 25.24 to 20.61). However, the real-world latency improves by **41.6%** (from 178.6 ms to 104.2 ms), achieving a **1.71x speedup**.

This finding confirms that our dual-pruning strategy successfully targeted memory-bound bottlenecks rather than just computational ones. By aggressively pruning the decoder channels in conjunction with encoder block skipping, we reduced the memory bandwidth requirements of the fusion layer, yielding speed improvements that far exceed what is predicted by FLOPs reduction alone.

TABLE II
QUANTITATIVE ANALYSIS OF RDD CONFIGURATIONS

Config	mIoU(%)	GFLOPS	Latency	SpeedUp
B2 Full	89.09	25.24	178.6	1.00x
$B2_a$	88.19	23.92	112.4	1.59x
$B2_b$	88.19	23.92	108.7	1.64x
$B2_c$	86.35	22.88	107.5	1.66x
$B2_d$	85.55	21.49	106.4	1.68x
$B2_e$	85.40	20.61	105.3	1.70x
$B2_f$	85.05	20.61	104.3	1.71x

Despite the aggressive pruning in the **CRITICAL (Survival)** i.e `B2_f` mode—which skips 25% of the encoder depth and prunes 25% of the decoder channels—the system demonstrates remarkable resilience. The segmentation accuracy (mIoU) drops by only **3.5%** (from 89.08% to 85.57%).

This stability is attributed to the strong biological prior established by the ExG-Gating mechanism, which ensures that even when the encoder capacity is reduced, the remaining features stay focused on vegetation, preventing the model from collapsing into soil-dominated predictions.

VI. Conclusions

This paper presented a novel **Resource-Aware Dynamic (RDD) Pruning** framework for real-time crop and weed segmentation on energy-constrained UAVs. By fundamentally restructuring the standard SegFormer architecture, we addressed the "FLOPs Paradox" identified in prior literature, shifting the computational bottleneck from the memory-bound decoder to the feature-rich encoder. Our proposed **Parallel Detail-Preserving Decoder** and **ExG-Guided Attention** mechanism allowed us to aggressively prune the network depth and width without compromising the detection of critical vegetative features.

Experimental results on the CropAndWeed2 dataset (validated on an NVIDIA RTX 5060) confirm the efficacy of this approach. The system achieved a **1.71x speedup**, reducing inference latency from **178 ms** to **104 ms**. Notably, this **41.6% reduction in latency** was achieved with only an **18.3% reduction in theoretical GFLOPs**, validating our hypothesis that relieving memory bandwidth bottlenecks yields performance gains exceeding theoretical arithmetic reductions. Even in the most constrained "Critical Survival" mode (`B2_f`), the system maintained a competitive mIoU of **85.57%**, dropping only **3.5%** from the baseline while extending operational viability.

Looking ahead, this framework lays the groundwork for several critical advancements in "self-aware" agricultural robotics. Future work will extend the validation from simulation to physical embedded platforms (e.g., NVIDIA Jetson Orin Nano) to quantify the direct impact of dynamic pruning on battery discharge rates and thermal throttling. Furthermore, we aim to replace the current rule-based controller with a **Deep Reinforcement Learning (DRL)** agent capable of learning optimal switching policies

VII. References

[1] (FAO Population Report): OECD/FAO, OECD-FAO Agricultural Outlook 2023-2032. OECD Publishing, Paris, 2023.

[2] (Herbicide Resistance): I. Heap, "The International Herbicide-Resistant Weed Database," Online, 2024. Available: www.weedscience.org

[3] (Precision Ag Adoption): R. Finger, S. M. Swinton, N. El Benni, and A. Walter, "Precision Farming at the Nexus of Agricultural Production and the Environment," Annual Review of Resource Economics, vol. 11, pp. 97-135, 2019.

[4] (Crop-Weed Classification): P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss, "Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018.

[5] (Deep Learning in Ag): A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70-90, 2018.

[6] (Vision Transformer Computation and Resilience for Dynamic Inference): R. Venkatesan, K. Sreedhar, J. Clemons, and S. W. Keckler, "Vision Transformer Computation and Resilience for Dynamic Inference," arXiv preprint arXiv:2212.02687, 2024.

[7] (Small Weeds/Cotyledon): P. Lottes, et al., "Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming," IEEE/RSJ IROS, 2018. (Or similar paper on early-stage detection).

[8] (Receptive Field): W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," NeurIPS, 2016.

[9] (SegFormer Original): E. Xie, et al., "SegFormer: Simple and efficient design for semantic segmentation with transformers," NeurIPS, 2021.

[10] (Similarity/Ag Challenges): A. Olsen, et al., "Deep learning for computer vision in agriculture: A survey," Electronics, 2019.

[11] (CNNs in Ag): A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, 2018.

[12] (Static Models): A. Bhujel, et al., "A Survey of Robotic Weed Control Systems," Computers and Electronics in Agriculture, 2020.

[13] S. Khan, et al., "Transformers in Vision: A Survey," ACM Computing Surveys, 2022.

[14] L. Deng, et al., "Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey," Proceedings of the IEEE, 2020.

[15] Y. Wang, et al., "DynaBERT: Dynamic BERT with Adaptive Width and Depth," NeurIPS, 2020.

[16] D. Steininger, A. Trondl, G. Croonen, J. Simon, and V. Widhalm, "The CropAndWeed Dataset: A Multi-Modal Learning Approach for Efficient Crop and Weed Manipulation," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 3729-3738, 2023.