RMAE-ProGRess: Advancing Semantic Segmentation in Unstructured Environments

Manish Bhurtel
Howard University
Washington, DC, USA
manish.bhurtel@bison.howard.edu

Danda B. Rawat
Howard University
Washington, DC, USA
danda.rawat@howard.edu

Abstract

Semantic segmentation in unstructured environments presents unique challenges due to irregular terrain, occlusions, and complex spatial layouts. While structured settings (e.g., urban scenes) have been widely studied, segmentation in unstructured settings remains relatively underexplored, both in terms of standardized benchmarking and architectural design. In this work, we propose a encoder-decoder based semantic segmentation architecture that integrates a Reduced Masked Autoencoder (RMAE) as the encoder, a Feature-to-Pyramid (F2P) neck, and a novel decoder called ProGRess. The ProGRess decoder introduces Progressive Leapwise Fusion (PLF) for top-down multi-scale fusion of non-contiguous feature maps, a Lightweight Channel Attention gate with Residuals (LCAR) module, and a Bottleneck Feature Fusion (BFF) block for compact refinement. We establish comprehensive baselines by benchmarking state-of-the-art CNN and transformer-based models on challenging unstructured environment datasets viz. RELLIS-3D, it's coarse-grained variant, and RUGD. Our architecture achieves the state-of-the-art performance with 57.41% mIoU on RELLIS-3D, 45.63% mIoU on RUGD, 78.95% mIoU on RELLIS-3DC datasets while maintaining competitive parameter-count and vRAM usage. Code is available at https://gitlab.com/coeaiml/rmae-progress.

1. Introduction

Semantic segmentation has become a crucial task in computer vision, enabling pixel-level scene understanding essential for autonomous navigation, robotics, and environmental perception. While significant progress has been made in structured environments like urban driving scenes (e.g., Cityscapes [8], ADE20K [47]) through the use of powerful encoder-decoder architectures, semantic segmentation in *unstructured environments* remains significantly underexplored. Unstructured environments, characterized by off-road terrain, uneven landscapes, ambiguous boundaries and a lack of geometric consistency, present unique challenges in designing an accurate and efficient segmentation model. Despite their importance to critical domains such as battlefield scenarios, search-and-rescue missions, defense robotics, and planetary exploration, unstructured environments have received limited attention in terms of both model design and standardized benchmarking. Moreover, unstructured environments are often contested and resource-constrained, necessitating models that are not only *high-performing* but also *computationally balanced*. Therefore, this work is motivated by two core research gaps: (1) the lack of standardized benchmarks and comparative evaluations for semantic segmentation in unstructured settings, and (2) the need for high-performing architectures with reasonable computational efficiency. We address these by revisiting the unstructured environment datasets, RUGD [34], RELLIS-3D [20] and its coarse-grained variant RELLIS-3DC [14], and introducing a semantic segmentation framework that balances accuracy, computational efficiency, and design modularity.

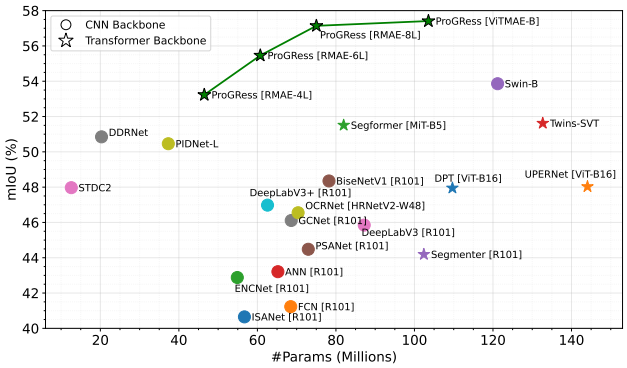

Figure 1. **Comparison of mIoU vs parameter count in RELLIS-3D dataset.** Our proposed RMAE-ProGRess models achieves state-of-the-art performance with balanced parameter count.

To address the gaps, we begin by comprehensively benchmarking a suite of strong CNN and transformer-based segmentation models, thus, establishing strong and reproducible baselines for both RELLIS-3D and RUGD datasets. We then design **Reduced MAE (RMAE)**, a lightweight ViT-based [9] encoder, to enhance the computational efficiency while preserving the accuracy. This design enables strong representation learning while significantly reducing computation. Next, we employ a standard **Feature-to-Pyramid (F2P)** neck network to build hierarchical spatial features. Finally, we develop a novel **ProGRess decoder**, a lightweight and modular architecture composed of three key components discussed in section 3.3. Our full **RMAE-ProGRess** pipeline achieves state-of-the-art performance on RELLIS-3D, RELLIS-3DC, and RUGD datasets while maintaining a balanced computational efficiency.

Our contributions are summarized as follows:

- **Reduced MAE encoder (RMAE)**: We design a compact encoder by reducing the depth of the ViTMAE-Base model, with Masked Autoencoder (MAE) pretrained weights while significantly lowering parameter count. We show that even shallow variants outperform heavier models on RELLIS-3D.
- **A lightweight segmentation decoder**: We introduce *ProGRess*, a modular decoder featuring Progressive Leapwise Fusion (PLF), Lightweight Channel Attention with Gate with Residual connections (LCAR), and a Bottleneck Feature Fusion (BFF) for accurate semantic segmentation in unstructured environments.
- **Comprehensive benchmarking on RELLIS-3D and RUGD**: Along with the performance reports in the literature, we train and provide a systematic evaluation of CNN and transformer-based models on the RELLIS-3D and RUGD datasets, establishing standardized baselines for future work in offroad segmentation.
- **State-of-the-art performance with balanced compute**: Our RMAE-ProGRess model achieves 57.41% mIoU on original RELLIS-3D, 78.95% on it's coarse-grained variant RELLIS-3DC, and 45.63% mIoU on RUGD datasets establishing the state-of-the-art performance. The parameter count, vRAM usage, and FLOPs are competitive as compared to the general-purpose segmentation baselines.

2. Related Work

2.1. Semantic segmentation in unstructured settings

Although datasets like Cityscapes [8] and ADE20K [47] contain a few off-road classes (e.g., soil, vegetation, river), these categories are sparse and do not reflect the complexity, variability, or boundary ambiguity present in real offroad settings. To address this gap, the U.S. Army Research Laboratory (ARL) introduced RUGD [34] and RELLIS-3D [20] datasets, captured in offroad settings using autonomous platforms. Several works have used these datasets, ranging from group-wise attention for navigability [14] to memory modules for illumination shifts [21], active learning [12], and boundary-enhanced segmentation [26]. Despite these efforts, unstructured semantic segmentation remains underexplored from a model-design and benchmarking perspective. To bridge this gap, we perform a comprehensive evaluation of CNN and transformer-based architectures on these datasets, prioritizing accuracy while maintaining balanced computational cost.

2.2. Lightweight vision transformer encoders

While CNN-based backbones like ResNet [15] are lightweight and widely used, Vision Transformers (ViTs) have shown superior performance across various visual tasks [9]. This has motivated the design of lightweight transformer architectures suitable for deployment in resource-constrained settings. Hybrid models such as MobileViT [25] combine MobileNet efficiency with transformer expressiveness, while ViT-Tiny and MAE-Tiny [13, 16] reduce embedding dimensions and model width. However, these approaches typically retain the full depth of ViT-Base model, 12 layers, even in their compact variants. In contrast, we propose a *depth-reduced* version of ViTMAE-Base, which maintains the original embedding dimensions but reduces the number of transformer layers.

2.3. Decoder networks for semantic segmentation

Decoder design plays a crucial role in semantic segmentation, particularly in aggregating multi-scale features and recovering spatial precision from deep encoder outputs. Traditional encoder-decoder frameworks like U-Net [29] and SegNet [1] utilized symmetric decoding with skip connections to merge high-resolution spatial features with semantic-rich encoder outputs. DeepLabV3+ [5] improved upon this by integrating atrous spatial pyramid pooling (ASPP) and a simplified decoder to better capture multi-scale context. Similarly, UPerNet [37] extended the Feature Pyramid Network (FPN) [23] by combining lateral connections and pyramid pooling, enabling top-down semantic refinement. Other approaches like OCRNet [42] and GC-Net [3] introduced global context modules for object-level understanding. Despite their strong performance on structured benchmarks, these decoder designs remain largely untested in unstructured environments. In this work, we conduct a comprehensive evaluation of these decoder architectures with various encoder backbones and compare them directly against our proposed RMAE-ProGRess model.

3. Methodology

An overview of our full architecture is illustrated in Fig. 2. The system comprises three main components: a lightweight RMAE encoder that provides non-contiguous

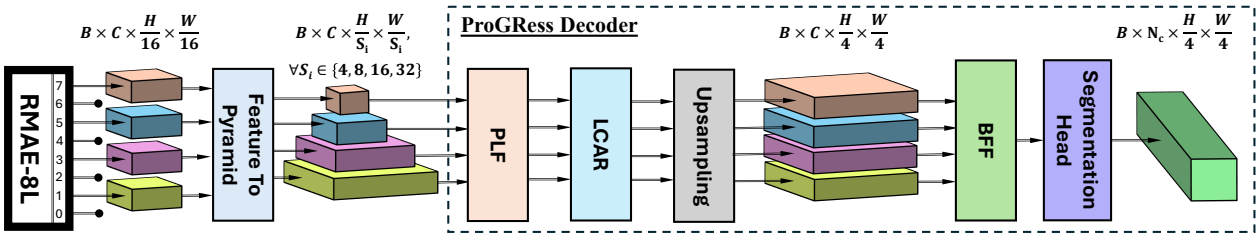

Figure 2. **Overview of proposed RMAE-ProGRess Architecture.** It consists of RMAE encoder (RMAE-8L in the diagram), which provides 4 non-contiguous feature maps, Feature To Pyramid neck network that converts the feature maps into multiple spatial dimensions, and a ProGRess Decoder to decode the hierarchical feature maps. Channel dimension C is consistent throughout the architecture.

coarse and fine semantic features; a Feature-to-Pyramid neck that resizes and aligns the feature maps across spatial scales; and our proposed ProGRess decoder, which performs progressive fusion, attention-based refinement, and final segmentation.

3.1. RMAE: Reduced Encoder Design

To reduce computational overhead while maintaining strong representational capacity, we adopt a truncated ViT-Base encoder with Masked Autoencoder (MAE) [16] pretrained weights. Standard ViT-Base encoders use 12 transformer layers; we reduce this to 8, 6 and 4 layers, forming our **Reduced MAE (RMAE)** encoders.

Table 1. Different RMAE variants and the standard ViTMAE-Base encoder. Out Index refers to the indices of selected feature levels

Model	Layers	Heads	Params (M)	Out Index
ViTMAE-Base	12	12	86.0	{3, 5, 7, 11}
RMAE-8L	8	12	58.5	{1, 3, 5, 7}
RMAE-6L	6	12	44.2	{1, 2, 4, 5}
RMAE-4L	4	12	29.9	{0, 1, 2, 3}

In transformer-based encoders like ViTMAE, adjacent layers often produce highly correlated features due to incremental token mixing [9, 16]. Sampling from every layer introduces redundancy, increases memory usage, and complicates the fusion process without significant gains in representation quality. Inspired by prior work [2, 33, 38], we extract features at regular intervals, capturing early structure, mid-level patterns, and deep semantics. As shown in Table 1, our RMAE variants significantly reduce parameter count compared to the original ViTMAE-Base model.

Let a batch of input images of shape (C', H, W), denoted as $I \in \mathbb{R}^{B\times C'\times H\times W}$, be passed into the RMAE encoder, then the output feature maps are $f_i = \{f_1, f_2, f_3, f_4\}$, where $f_i \in \mathbb{R}^{B\times C\times \frac{H}{16}\times \frac{W}{16}}$, B is the batch size, and C is the channel dimension. Note that, each output feature maps corresponds to the out index for the selected RMAE encoder variant in Table 1. Next, the set of feature maps f_i is passed to the Feature to Pyramid (F2P) module.

3.2. Feature to Pyramid (F2P)

We employ a standard F2P neck that resizes the 4 feature maps into a multi-scale pyramid, enabling the decoder to operate over features of varying spatial resolution, thereby capturing both coarse context and fine details. In F2P, the feature maps f_i are rescaled by a corresponding factor $r_i \in \{4, 2, 1, 0.5\}$ to generate pyramid outputs $F_i = \mathcal{R}_{r_i}(f_i)$ where $\mathcal{R}_{r_i}(\cdot)$ is a rescaling operator defined as:

$$F_i = \mathcal{R}_{r_i}(f_i) = \begin{cases} \mathrm{C}^{\mathrm{T}}\big(\phi(\mathrm{BN}(\mathrm{C}^{\mathrm{T}}(f_i)))\big), & \text{if } r_i = 4 \\ \mathrm{C}^{\mathrm{T}}(f_i) & \text{if } r_i = 2 \\ f_i, & \text{if } r_i = 1 \\ \mathrm{MaxPool}(f_i), & \text{if } r_i = 0.5 \end{cases}$$

where C^{T} is the Transpose Convolution function, ϕ is the $\mathrm{GELU}(.)$ activation function and BN denotes Batch Normalization. Here, $\mathrm{MaxPool}$ uses a pooling size of 2, C^{T} uses the stride of 2, and C^{T} weights are independent for each rescaling operation. Finally, F2P outputs rescaled features: $F_i = \{F_1, F_2, F_3, F_4\}$, with spatial scales $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, and $\frac{H}{32} \times \frac{W}{32}$ respectively.

The F2P module upsamples early-layer features (e.g., output from out indices 1 and 3 in RMAE-8L) to larger resolutions to capture fine-grained spatial details, maintains mid-layer features (out index 5) at the original resolution, and downsamples deep-layer features (out index 7) to emphasize high-level semantics, thus creating a balanced pyramid while preserving the original channel dimension C.

3.3. ProGRess Decoder

We introduce the lightweight and modular **ProGRess decoder** that progressively aggregates and refines multi-scale features. The ProGRess decoder is composed of three primary modules: (i) *Progressive Leapwise Fusion (PLF)* (ii) *Lightweight Channel Attention with Residuals (LCAR)* and (iii) *Bottleneck Feature Fusion (BFF)*.

3.3.1. Progressive Leapwise Fusion (PLF)

The pyramid features $F_i = \{F_1, F_2, F_3, F_4\}$ serve as the inputs to the PLF module. Each pyramid feature corresponds

to the feature maps coming from four non-contiguous indices of the encoder (e.g., out indices 1, 3, 5, and 7 for RMAE-8L). To fuse these leapwise features, PLF operates top-down on these non-contiguous feature maps, progressively fusing semantically deep but spatially coarse features with shallower and high-resolution features, thus combining global context with local detail.

Following the F2P module, we preserve the channel dimension C throughout the decoder. PLF fuses features by successively upsampling deeper features and merging them with higher-resolution counterparts. This fusion forms a cascade since F_i represents the leapwise feature maps. Here, we also apply self-fusion to F_4 to enhance the semantic representations in the deeper feature map.

$$\tilde{F}_4 = \text{Fuse}(F_4, F_4) \quad \text{(self-fusion)} \tag{1}$$

$$\tilde{F}_3 = \text{Fuse}(F_3, \tilde{F}_4) \tag{2}$$

$$\tilde{F}_2 = \text{Fuse}(F_2, \tilde{F}_3) \tag{3}$$

$$\tilde{F}_1 = \text{Fuse}(F_1, \tilde{F}_2) \tag{4}$$

where,

$$\text{Fuse}(F_i, F_j) = \phi\left(\text{BN}\left(\mathbf{W}_{ij} * [F_i \,\|\, \mathcal{U}(F_j, \text{size of} F_i)]\right)\right) \tag{5}$$

where $\mathcal{U}(a, \text{size of } b)$ is the interpolation operation (nearest neighbor chosen through ablation studies) that upsamples a feature map a to the spatial dimension of b; $\|$ represents concatenation operation; and $\mathbf{W}_{ij} \in \mathbb{R}^{C\times(C_i+C_j)\times1\times1}$ denotes a learnable 1×1 convolution kernel that projects the concatenated features to C channels, with separate parameters for each fusion pair (F_i, F_j).

A critical challenge in multi-scale feature fusion is ensuring effective flow of information across all pyramid levels. While PLF shares some similarities with FPN-style methods [23, 37], it differs in its recursive fusion design. A detailed discussion is provided in the supplementary (Section 12). We clarify the information flow in PLF through a simple property of its recursive formulation (Proposition 1).

Proposition 1 (Information Preservation)**.** *The PLF mechanism preserves information from all pyramid levels through recursive fusion.*

Proof. We prove by construction that each $\tilde{F}_i$ contains information from all levels $j \geq i$.

Base case: $\tilde{F}_4$ contains information from F_4 only (self-fusion).

Inductive step: Assume $\tilde{F}_{k+1}$ contains information from $\{F_{k+1}, F_{k+2}, \ldots, F_4\}$. Then:

$$\tilde{F}_k = \phi(\text{BN}(\mathbf{W}_k * [F_k \| \mathcal{U}(\tilde{F}_{k+1})])) \tag{6}$$

By the concatenation operation, $\tilde{F}_k$ explicitly contains F_k and $\tilde{F}_{k+1}$. By the inductive hypothesis, $\tilde{F}_{k+1}$ contains $\{F_{k+1}, \ldots, F_4\}$. Therefore, $\tilde{F}_k$ contains information from $\{F_k, F_{k+1}, \ldots, F_4\}$. □

3.3.2. Lightweight Channel Attention with Residuals

Given the heterogeneity of visual cues in unstructured settings (e.g., foliage, debris, shadows), simple fusion may not sufficiently highlight relevant spatial structures. LCAR addresses this by applying a per-pixel per-channel gating mechanism that emphasizes or suppresses channels based on local importance. In contrast to general channel attention [18], which compress spatial information via global average pooling and fully connected layers, we employ a lightweight, pooling-free 1×1 convolution to generate per-pixel, per-channel attention maps, thereby preserving spatial detail with minimal overhead.

Lightweight Channel Attention (LCA). For an input X, LCA computes:

$$\text{LCA}(X) = X \odot \sigma(\mathbf{W}_c * X) \tag{7}$$

where $\mathbf{W}_c \in \mathbb{R}^{C\times C\times1\times1}$ is a 1×1 convolution kernel that mixes channels at a constant dimension C; $\sigma(\cdot)$ is the sigmoid function; $\odot$ denotes element-wise multiplication.

Attention with Residual Connection. The LCAR module is then defined as:

$$\text{LCAR}(X) = \text{LCA}(X) + \alpha X = X \odot \sigma(\mathbf{W}_c * X) + \alpha X \tag{8}$$

where $\alpha \in \{0, 1\}$ is a binary indicator for residual connection at each level. The value of α is chosen via empirical validation. Applying Eq. 8 across all feature levels F_i where $i \in \{1, 2, 3, 4\}$, we get:

$$\hat{F}_i = \text{LCAR}(\tilde{F}_i) = \tilde{F}_i \odot \sigma(\mathbf{W}_i * \tilde{F}_i) + \alpha_i \tilde{F}_i \tag{9}$$

Empirically, we obtain the best performance with $\alpha_1 = \alpha_2 = \alpha_3 = 0$ and $\alpha_4 = 1$, i.e., the residual connection is applied only at the deepest level ($\tilde{F}_4$). Residual connections are known to stabilize optimization and improve gradient flow in deeper levels where representations are semantically richer but more fragile due to reduced resolution [15]. Accordingly, the residual pathway in LCAR helps maintain stable training dynamics.

3.3.3. Bottleneck Feature Fusion (BFF)

Spatial Alignment. BFF serves as a compact yet expressive final aggregation block. All LCAR features are upsampled to the target resolution $\frac{H}{4} \times \frac{W}{4}$:

$$\bar{F}_i = \mathcal{U}(\hat{F}_i, \text{ size of } \tfrac{H}{4} \times \tfrac{W}{4}), \quad i \in \{1, 2, 3, 4\} \tag{10}$$

Multi-scale Aggregation. The BFF module aggregates features at all scales through:

$$\mathbf{Z} = \phi(\text{BN}(\mathbf{W}_{bff} * [\bar{F}_1 \| \bar{F}_2 \| \bar{F}_3 \| \bar{F}_4]))$$

where $\mathbf{W}_{bff} \in \mathbb{R}^{C\times4C\times1\times1}$ is a 1×1 convolution kernel that reduces channel dimension from $4C$ to C.

3.3.4. Final Prediction

The segmentation output is obtained via:

$$\mathbf{Y} = \text{softmax}(\mathbf{W}_{cls} * \mathbf{Z}) \tag{11}$$

where $\mathbf{W}_{cls} \in \mathbb{R}^{N \times C \times 1 \times 1}$ is the 1×1 convolution kernel converting resultant channels C into number of classes N. The softmax produces per-pixel class probabilities, from which the final label is obtained via an $\arg\max$ operation.

4. Experiments

4.1. Datasets and evaluation metrics

We conduct comprehensive evaluations of our proposed model across multiple off-road datasets and their variants to assess both fine-grained semantic understanding and cross-domain generalization capabilities as shown in Table 2. Note that, we include the validation sets in the training set during the benchmarking and evaluation.

Table 2. Summary of all datasets

Dataset Name	Training	Test	Classes
RELLIS-3D	4562	1672	18
RELLIS-3DC	4562	1672	6
RUGD	5512	1924	22
RUG-REL-COMMON	-	1924	14

RELLIS-3D. This is the popular offroad semantic segmentation dataset [20]. Following the original dataset protocol [20], we evaluate on 18 classes, excluding the *void* and *dirt* categories due to their sparse annotations.

RELLIS-3DC. This is a coarse-grained variant of RELLIS-3D introduced by [14], which we denote as RELLIS-3DC. This dataset groups the fine-grained RELLIS-3D classes into 6 broader categories viz. *smooth*, *rough*, *bumpy*, *forbidden*, *obstacle* and *background* classes.

RUGD. We extend our evaluation to RUGD dataset [34]. Although RUGD defines 25 classes, we report results on 22 classes, excluding *void*, *bicycle*, and *bridge* classes due missing annotations in either training or test splits.

RUG-REL-Common. For zero-shot domain generalization (RELLIS-3D $\rightarrow$ RUGD), we construct a subset of the RUGD test set containing 14 classes common with RELLIS-3D, denoted as RUG-REL-Common. Pixels corresponding to classes absent in RELLIS-3D are masked as *void* (assigned value 0). This setup enables direct evaluation of cross-domain transfer, where models trained solely on RELLIS-3D are tested on RUGD without fine-tuning.

Evaluation Metrics. Following [14, 38], we use 4 standard semantic segmentation evaluation metrics: per-class Intersection over Union (IoU), mean IoU (mIoU), mean pixel accuracy (mAcc), and average pixel accuracy (aAcc).

Table 3. Comparison with existing literature on RUGD and RELLIS-3D test sets.

Dataset	Method	Encoder	mIoU ↑
RUGD	UperNet [37]	ResNet50	31.95
	PSPNet [44]	ResNet-50	32.07
	Deeplabv3 [4]	ResNet50	32.81
	TrSeg [22]	ResNet50	33.91
	Memory-based Deeplabv3 [21]	ResNet50	37.71
	ProGRess (Ours)	**ViTMAE-Base**	**45.63**
RELLIS-3D	PSPNet [44]	ResNet18	38.52
	DeepLabv3 [4]	MobileNetV2	38.67
	DeepLabv3 [4]	ResNet50	43.97
	DeepLabv3 with memory [21]	ResNet50	45.61
	GSCNN [32]	-	52.92
	ProGRess (Ours)	**ViTMAE-Base**	**57.41**

4.2. Comparison with existing literature

Table 3 presents a comparison of our proposed method against existing literature on RUGD and RELLIS-3D datasets. The performance on RUGD are reported from [21, 22, 34] and the performance on RELLIS-3D are reported from [20, 21]. There are some other works in the literature concerning the performance benchmarking on these datasets; however, the performance metrics were not clear and are hence discarded from the comparison. On RUGD, prior works using ResNet-50 backbones achieve mIoU scores ranging from 31.95% to 37.71%, with Memory-based DeepLabv3 [21] representing the previous best result. Our ViTMAE-ProGRess model demonstrate substantial improvement achieving 45.63% mIoU. On RELLIS-3D, the literature reports include GSCNN [20, 32] at 52.92% mIoU as the best performing model. Our ViTMAE-ProGRess method sets a new state-of-the-art on this dataset as well achieving 57.41% mIoU outperforming all methods in the literature. While these results demonstrate clear improvements over existing literature, the limited scope and inconsistent evaluation protocols across prior works motivate the need for a comprehensive benchmark. We therefore conduct an extensive evaluation of 16 state-of-the-art semantic segmentation methods, as detailed in subsection 4.3.

4.3. Comprehensive benchmarking and comparison with the state-of-the-art

We conduct comprehensive training and benchmarking of state-of-the-art semantic segmentation models on the RELLIS-3D and RUGD datasets. For each method, hyperparameters are tuned to ensure strong and reproducible baselines across both CNN and transformer architectures in unstructured outdoor settings. All experiments are implemented using the MMSegmentation framework [7]. Models are trained for 160K iterations, and we report the best mIoU achieved during training. We evaluate 16 widely used methods from the MMSegmentation library. Full training

Table 4. Comparison of proposed methods with all benchmarked models on RELLIS-3D and RUGD test sets. Bold values are the instances where our model outperforms other models, underscore indicates the competitive performance as compared to other models and * indicate transformer-based models.

Methods	Backbone	Aux. Head	Params ↓ (M)	vRAM ↓ (MB)	RELLIS-3D [512 × 512]			RUGD [512 × 512]		
					FLOPS (G) ↓	mIoU ↑	mAcc ↑	FLOPS (G) ↓	mIoU ↑	mAcc ↑
ISANet [19]	ResNetV1c-101	FCN	56.7	227.4	227.8	40.65	49.86	455.6	21.63	32.50
FCN [30]	ResNetV1c-101	FCN	68.49	270.8	275.7	41.23	50.94	551.4	23.53	34.04
EncNet [43]	ResNetV1c-101	FCN	54.88	219.4	218.8	42.88	53.03	437.5	24.61	36.37
ANN [48]	ResNetV1c-101	FCN	65.22	257.8	263.3	43.21	51.56	526.6	25.81	36.11
Segmenter* [31]	ViT-B16	-	102.4	401.0	126.5	44.19	52.72	253.1	41.08	52.99
PSANet [45]	ResNetV1c-101	FCN	72.96	290.0	273.6	44.48	52.03	-	-	-
DeepLabV3 [4]	ResNetV1c-101	FCN	87.21	343.9	347.9	45.85	52.60	-	-	-
GCNet [3]	ResNetV1c-101	FCN	68.62	273.7	275.7	46.11	54.65	551.4	22.10	32.98
OCRNet + FCN [42]	HRNetV2-W48	-	70.37	282.1	162.8	46.55	54.58	325.9	27.16	37.56
DeepLabV3+ [5]	ResNetV1c-101	FCN	62.58	249.7	254.4	46.98	56.12	508.9	29.97	44.78
DPT* [28]	ViT-B16	-	109.67	441.4	217.4	47.95	57.32	-	-	-
ViT-UPerNet* [9]	ViT-B16	FCN	144.07	564.3	442.6	48.03	56.17	885.3	41.21	53.39
SETR-PUP* [46]	ViT-L16	SETRUP	317.15	1255.5	417.0	51.12	59.27	-	-	-
Segformer* [38]	MiT-B5	-	81.97	331.4	74.6	51.51	60.66	149.3	43.69	56.45
Twins-UPerNet* [6]	Twins-SVT	FCN	132.68	524.2	296.1	51.62	62.77	592.3	31.59	45.10
Swin-UPerNet* [24]	Swin-B-W7	FCN	121.18	473.8	298.3	53.86	63.33	596.7	41.76	55.19
ProGRess (Ours)*	RMAE-4L	-	**46.47**	221.9	128.1	53.23	63.04	256.3	37.30	50.31
ProGRess (Ours)*	RMAE-6L	-	60.74	296.0	145.9	**55.46**	**66.02**	291.8	37.67	50.81
ProGRess (Ours)*	RMAE-8L	-	75.02	365.0	163.6	**57.14**	**68.53**	327.3	41.34	53.73
ProGRess (Ours)*	ViTMAE-Base	-	103.56	509.4	199.1	**57.41**	**69.21**	398.3	**45.63**	**57.80**

and implementation details are provided in the supplementary material (Section 7). We also include comparisons with real-time segmentation models, with details in the supplementary material (Section 10).

Table 4 compares our models with the benchmarked models. On RELLIS-3D, all ProGRess variants rank at the top, with ViTMAE-Base achieving 57.41% mIoU and 69.21% mAcc, significantly outperforming the strongest baseline, Swin-UPerNet (53.86% mIoU, 63.33% mAcc). Notably, the lightweight RMAE-4L variant (46.47M parameters) attains 53.23% mIoU with competitive efficiency (128.1 GFLOPs), surpassing almost all baselines. We discuss the complete breakdown and analysis of FLOPS in the supplementary material (Section 9). On RUGD, ViTMAE-Base with our ProGRess decoder again achieves state-of-the-art performance with 45.63% mIoU and 57.80% mAcc, outperforming SegFormer (43.69% mIoU, 56.45% mAcc). These consistent gains across datasets, combined with competitive compute, demonstrate the effectiveness of our approach for unstructured semantic segmentation. Fig. 1 illustrates the accuracy–parameter trade-off. We report the per-class IoU results in the supplementary material (Section 8), which shows notable improvements on rare classes and those with irregular geometry or ambiguous boundaries. Qualitative results are also included in the supplementary material (Section 11).

Table 5 shows the comparison of our ProGRess decoder with RMAE-8L encoder against the state-of-the-art models as reported in [14]. Our ProGRess model with the RMAE-8L encoder outperforms the previous methods by a significant margin of more than 4% mIoU. Furthermore, our method claims the best per-class IoU performance on the *Smooth*, *Forbidden*, *Obstacle*, and *Background* classes, and competitive performance on the *Rough* and *Bumpy* classes. Refer to supplementary material for reproducibility notes.

Table 5. Comparison with the state-of-the-art on RELLIS-3DC dataset. Values for each class are the per-class IoU. Bold indicates the best performance and * indicate transformer-based models.

Method	Smooth	Rough	Bumpy	Forbidden	Obstacle	Background	mIoU↑	aAcc↑
Segmenter [31]	51.67	78.4	19.38	42.61	66.04	92.05	58.36	82.16
PSANet [45]	64.06	75.29	17.08	47.45	61.74	94.31	59.89	83.17
CGNet [36]	62.84	74.17	49.57	45.41	68.88	94.31	60.85	81.04
PSPNet [44]	69.21	80.99	8.89	53.7	60.7	94.67	61.36	86.01
DeepLabv3+ [5]	65.76	79.84	19.72	47.52	64.88	95.92	62.27	85.84
BiseNetv2 [41]	65.56	73.24	39.35	48.17	71.91	93.78	65.33	83.03
SETR* [46]	65.37	78.64	40.89	52.59	63.8	91.87	65.53	83.59
DANet [11]	72.93	85.18	13.10	60.60	70.53	95.95	66.38	89.11
OCRNet [42]	74.67	83.04	27.76	60.44	70.22	92.58	66.81	86.95
DPT* [28]	5.42	76.65	47.13	54.87	62.74	85.5	66.88	81.61
FastFCN [35]	70.51	79.15	49.72	51.37	63.9	94.82	68.24	84.1
Segformer* [38]	60.28	79.78	53.35	53.78	66.82	85.37	68.62	85.37
FastSCNN [27]	67.06	77.6	**56.49**	49.76	70.31	94.43	69.27	84.51
GA-Nav-r32* [14]	76.73	86.86	23.49	71.58	71.24	94.65	70.76	90.44
GA-Nav-r16* [14]	78.37	85.58	28.63	67.55	73.82	95.73	71.62	90.26
GA-Nav-r8* [14]	78.5	**88.25**	37.28	72.34	74.75	96.07	74.44	91.69
Progress [RMAE-8L]	**85.41**	86.67	46.65	**74.41**	**83.48**	**97.09**	**78.95**	**92.59**

4.4. Ablation Studies

4.4.1. Effectiveness of decoder across multiple encoders

To isolate the contribution of our ProGRess decoder and validate it's effectiveness, we conduct comprehensive ablation experiments comparing it against other state-of-the-art decoder architectures paired with multiple CNN-based and transformer-based encoder backbones as shown in Table 6. ProGRess achieves the highest mIoU and mAcc across all four transformer-based backbones and a com-

petitive performance across the ResNet101-V1c encoder backbone. ProGRess with ViT-B16 achieves 55.68% mIoU compared to 48.03% for UPerNet (FPN-based decoder), representing a gain of 7.65%. Similarly, with our proposed RMAE-4L encoder, ProGRess achieves 53.23% mIoU and 63.04% mAcc, outperforming DeeplabV3+ by 4.93% and UPerNet by 4.74% in mIoU. These consistent performance gains across diverse encoder architectures validate the effectiveness of the ProGRess decoder.

Table 6. **Ablation studies on the effectiveness of the decoder across multiple encoders in RELLIS-3D dataset.** Bold values indicate the best performance. Note that the reported parameter counts correspond only to the decoder, and they vary depending on the dimensionality of the input features produced by each encoder.

Encoder	Decoder	Params (M) ↓	mIoU ↑	mAcc ↑
ResNet101-V1c	ISANet	11.81	40.65	49.86
	EncNet	10.0	42.88	53.03
	DeeplabV3+	17.7	**46.98**	**56.12**
	ProGRess	**5.91**	42.39	51.39
Swin Transformer	UPerNet	33.25	51.01	61.23
	ProGRess	**4.41**	**52.16**	**62.34**
MiT-B5	SegFormer	**0.53**	51.51	60.66
	ProGRess	1.12	**54.11**	**62.99**
ViT-B16	DPT	23.87	47.95	57.32
	UPerNet	32.27	48.03	56.17
	ProGRess	**4.86**	**55.68**	**67.68**
RMAE-4L	UPerNet	67.86	48.03	56.17
	DeepLabV3+	14.4	49.30	58.30
	ProGRess	**9.46**	**53.23**	**63.04**

Beyond accuracy, ProGRess remains parameter-efficient across most configurations. With ViT-B16, it uses only 4.86M parameters compared to UPerNet's 32.27M (85% reduction) while achieving higher accuracy. Decoder parameter counts vary with the encoder due to differences in feature dimensionality. The only exception in parameter efficiency is MiT-B5 with SegFormer (0.53M parameters); however, ProGRess still achieves higher accuracy (54.11% vs 51.51% mIoU) with only 1.12M parameters. These consistent gains across ResNet, Swin, MiT, and ViT backbones highlight the architectural advantages of ProGRess. Overall, the results indicate that the proposed decoder effectively captures multi-scale context while remaining versatile and efficient across diverse encoder designs.

4.4.2. Impact of each decoder components

To analyze the contribution of each component in our ProGRess decoder, we conduct systematic ablation experiments on the RELLIS-3D test set using the RMAE-8L encoder under both frozen and fine-tuned settings. For this ablation, we retain the same implementation settings discussed in supplementary materials (Section 7), with two key modifications. First, we replace the *nearest* interpolation with *bicubic* interpolation to reduce probable aliasing artifacts and promote smoother feature transitions when removing decoder blocks. Second, we adopt the learning rates from the first 8 layers of ViTMAE-B layer-wise learning-rate configuration and use it in the RMAE-8L encoder. The layerwise learning-rate is discussed in the supplementary materials (Section 7). These adjustments are made to maintain the stability of the gradients and ensure safe optimization of the partially pruned architectures.

Table 7. **Ablation studies on impact of each component of ProGRess decoder.** We train and see the performance for both Frozen vs Fine-tuned RMAE-8L encoder on the RELLIS-3D test set. Note that these layers are added on top of the segmentation head.

BFF	PLF	Self Fusion	LCAR	Frozen RMAE-8L		Fine-tuned RMAE-8L	
				mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑
✗	✗	✗	✗	49.02	59.20	52.52	60.55
✓	✗	✗	✗	49.01	58.82	54.56	66.68
✓	✓	✗	✗	52.60	62.09	56.15	66.35
✓	✓	✓	✗	52.81	62.15	56.56	67.01
✓	✓	✓	✓	53.18	62.36	56.90	68.25
✗	✓	✓	✗	51.51	61.46	56.78	66.85
✗	✗	✗	✓	51.03	60.69	54.51	63.15

Table 7 presents the results as we progressively add components on top of the segmentation head. The baseline without any decoder components achieves 49.02% mIoU (frozen) and 52.52% (fine-tuned). Adding BFF yields small improvement in the frozen setting (49.01%) but provides a notable gain of +2.04 point mIoU when fine-tuned (54.56%). The addition of PLF on top of BFF leads to substantial improvements, reaching 52.60% (frozen) and 56.15% (fine-tuned). Further incorporating Self Fusion results in consistent gains, achieving 52.81% and 56.56% mIoU, respectively. The full ProGRess decoder, comprising BFF, PLF, Self Fusion, and LCAR blocks, achieves the best performance, with 53.18% mIoU and 62.36% mAcc in the frozen setting, and 56.90% mIoU and 68.25% mAcc when fine-tuned. Isolating PLF or LCAR leads to noticeable performance degradation, as shown in the final two rows. Overall, these results highlight the benefits of each block and validate the effectiveness of the overall ProGRess decoder design.

4.4.3. Choice of interpolation method

Interpolation can influence feature upsampling and our decoder uses upsampling at two key stages (Eqs. 5 and 10). We therefore compare three interpolation methods viz. bicubic, bilinear, and nearest neighbor using the RMAE8L-ProGRess model as shown in Table 8.

All methods yield nearly identical results, with differences under 0.42 mIoU and 0.38 mAcc, indicating that ProGRess is largely insensitive to the interpolation choice. However, nearest neighbor interpolation achieves the highest mIoU (57.14%) while offering significantly lower com-

Table 8. Interpolation Vs Accuracy with RMAE8L-ProGRess.

Interpolation	mIoU↑	mAcc↑
Bicubic	57.10	68.42
Bilinear	56.72	**68.80**
Nearest	**57.14**	68.53

putational cost compared to bilinear and bicubic interpolation, as it requires no mathematical operations beyond index mapping. This finding demonstrates that nearest neighbor interpolation can effectively replace bilinear or bicubic methods in our framework, providing a computationally efficient alternative without sacrificing accuracy.

4.5. Zero-shot domain generalization

Table 9 presents the zero-shot domain generalization results of our proposed ProGRess model (using the RMAE-8L encoder) on the RUG-REL-Common dataset. For comparison, we selected the top-performing CNN-based and transformer-based models from Table 4 and evaluated them alongside our method. All models were trained exclusively on RELLIS-3D and tested on RUG-REL-Common **without any fine-tuning**, making this a strict zero-shot evaluation.

Table 9. Zero-shot domain generalization on RUG-REL-Common dataset. The methods use the same encoder as discussed in Table 4 and we use RMAE-8L variant with ProGRess decoder.

Methods	grass	tree	pole	water	sky	vehicle	object	asphalt	building	log	person	fence	bush	concrete	mIoU↑
DeepLabV3+	10.57	56.07	2.25	0.0	64.22	9.51	0.2	0.01	4.93	0.03	0.0	**0.58**	2.21	21.8	12.31
OCRNet	13.7	58.79	7.02	0.0	66.1	13.51	0.04	0.0	**10.26**	0.06	0.0	0.01	2.43	14.09	13.29
Segformer	45.51	39.93	6.0	**0.11**	60.28	9.72	**1.1**	0.72	1.63	3.07	0.0	0.01	4.4	**63.55**	16.86
Segmenter	53.99	**62.07**	0.05	0.0	**74.36**	3.49	0.5	5.75	3.61	3.24	0.0	0.0	7.78	45.19	18.57
Swin-UPerNet	48.23	61.19	7.22	0.0	73.75	13.29	0.16	0.16	3.09	8.96	**0.01**	0.01	4.8	49.97	19.35
ProGress	**62.13**	51.08	**11.93**	0.0	61.75	**20.27**	0.74	**4.79**	1.26	**11.73**	0.0	0.15	**13.09**	55.28	**21.01**

RELLIS-3D and RUGD differ largely in visual appearance, terrain structure, and object distribution, making direct transfer challenging. Consequently, all methods exhibit relatively low mIoU under this strict zero-shot setting (Table 9). Despite this, ProGRess outperforms all baselines, demonstrating stronger robustness under domain shift. Particularly, it achieves the best IoU on several key classes, including **grass** (62.13), **pole** (11.93), **vehicle** (20.27), **asphalt** (4.79), **log** (11.73), and **bush** (13.09). Overall, our model attains the highest mIoU of 21.01%, surpassing all baselines in this challenging cross-domain scenario. These results highlight the ability of ProGRess to generalize across visually and structurally distinct domains without target-domain supervision.

5. Limitations, Future Directions and Ethical Considerations

Limitations and Future Directions. While our proposed model remains competitive in computational efficiency as compared to the baselines, the parameter-count and FLOPS are higher as compared to the real-time models such as PIDNet [39], BiseNetV1 [40], STDC [10] and DDRNet [17] (discussed in supplementary material, Section 10). Furthermore, we breakdown the FLOPS for each block in our model and also propose the measures to reduce the computations while preserving the accuracy for future reference in supplementary material (Section 9). Future work includes model evaluation on unstructured categories within structured scenes, motivated by the decent performance on Cityscapes (see supplementary material, Section 13).

Ethical considerations. This work presents semantic segmentation models for unstructured environments with applications in areas such as autonomous navigation, disaster response, agriculture, and ecological monitoring. We emphasize ethical deployment and discourage any use that may pose risks or violate human rights. Although our method enhances situational awareness in complex terrains, it should be applied only within transparent, accountable, and supervised frameworks. All models are trained on public datasets containing no sensitive content, and we will release all code, configurations, and pretrained weights to support reproducibility and responsible use.

6. Conclusion

This paper presents RMAE-ProGRess, a lightweight encoder-decoder framework for semantic segmentation in unstructured environments, a setting underexplored in prior work. We introduce a reduced MAE (RMAE) encoder by pruning layers from a MAE pretrained ViT-Base model, and a novel ProGRess decoder composed of PLF, LCAR, and BFF modules. Our method achieves state-of-the-art performance on RELLIS-3D, RELLIS-3DC, and RUGD datasets, outperforming strong CNN and transformer-based baselines while maintaining competitive parameter count. We also demonstrate strong performance under zero-shot domain transfer, showing the robustness of the proposed design. Overall, our results show that careful architectural design can yield accurate solutions for semantic segmentation in unstructured settings. With ProGRess, we seek to advance model design and promote benchmarking in unstructured settings, contributing to **progressing the field** and inspiring further research beyond structured benchmarks.

Acknowledgment

This work was supported by the Center of Excellence in AI/ML (CoE-AIML) at Howard University under Contract W911NF-20-2-0277 with the U.S. Army Research Laboratory. However, any opinions, findings, conclusions, or recommendations expressed in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agency.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 2

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 3, 1

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 2, 6

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5, 6, 2

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2, 6, 3

[6] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in neural information processing systems*, 34:9355–9366, 2021. 6, 2

[7] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 5, 1

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 1, 2, 5

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 6, 1

[10] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9716–9725, 2021. 8, 3, 4

[11] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 6

[12] Biao Gao, Xijun Zhao, and Huijing Zhao. An active and contrastive learning framework for fine-grained off-road semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 24(1):564–579, 2022. 2

[13] Jin Gao, Shubo Lin, Shaoru Wang, Yutong Kou, Zeming Li, Liang Li, Congxuan Zhang, Xiaoqin Zhang, Yizheng Wang, and Weiming Hu. An experimental study on exploring strong lightweight vision transformers via masked image modeling pre-training. *International Journal of Computer Vision*, pages 1–33, 2025. 2

[14] Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments. *IEEE Robotics and Automation Letters*, 7(3):8138–8145, 2022. 1, 2, 5, 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 4

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 3, 1

[17] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021. 8, 4

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4

[19] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv preprint arXiv:1907.12273*, 2019. 6, 2

[20] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 1110–1116. IEEE, 2021. 1, 2, 5

[21] Youngsaeng Jin, David Han, and Hanseok Ko. Memory-based semantic segmentation for off-road unstructured natural environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 24–31. IEEE, 2021. 2, 5

[22] Youngsaeng Jin, David Han, and Hanseok Ko. Trseg: Transformer for semantic segmentation. *Pattern recognition letters*, 148:29–35, 2021. 5

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2, 4

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 6, 2, 3

[25] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 2

[26] Peizhou Ni, Xu Li, Dong Kong, and Xiaoqing Yin. Scene-adaptive 3d semantic segmentation based on multi-level boundary-semantic-enhancement for intelligent vehicles. *IEEE Transactions on Intelligent Vehicles*, 9(1):1722–1732, 2023. 2

[27] Rudra PK Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019. 6

[28] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 6, 2

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[30] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017. 6, 2, 5

[31] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021. 6, 2

[32] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5229–5238, 2019. 5, 2

[33] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 3

[34] Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007. IEEE, 2019. 1, 2, 5

[35] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang, and Yizhou Yu. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv preprint arXiv:1903.11816*, 2019. 6

[36] Tianyi Wu, Sheng Tang, Rui Zhang, Juan Cao, and Yongdong Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Transactions on Image Processing*, 30:1169–1179, 2020. 6

[37] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 2, 4, 5

[38] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. 3, 5, 6, 2

[39] Jiacong Xu, Zixiang Xiong, and Shankar P Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19529–19539, 2023. 8, 4

[40] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 8, 4

[41] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International journal of computer vision*, 129:3051–3068, 2021. 6

[42] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 173–190. Springer, 2020. 2, 6

[43] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 6, 2

[44] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 5, 6

[45] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European conference on computer vision (ECCV)*, pages 267–283, 2018. 6, 2, 5

[46] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 6, 2, 5

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 1, 2

[48] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 593–602, 2019. 6, 2