Semantic Segmentation of Drone Imagery Using SegFormer: Challenges and Improvements

Sushant Haluwai

April 28, 2025

Abstract

This report presents a semantic segmentation project aimed at classifying every pixel in drone imagery into meaningful categories such as roads, trees, and vehicles. Using the SegFormer model, a transformer-based architecture, we initially faced challenges including class imbalance, low resolution in predictions, and unstable training. Through targeted improvements—such as class weighting, upsampling, and a learning rate scheduler—we significantly enhanced the model's performance. The final model achieved a mean accuracy of 0.888, IoU of 0.256, and Dice score of 0.307, with notable gains in detecting minority classes like bicycles and dogs.

1 Introduction

Semantic segmentation is a critical task in computer vision, enabling pixel-level classification of images into meaningful categories. This project focuses on applying semantic segmentation to drone imagery, where the goal is to label each pixel as part of objects like roads, trees, buildings, or vehicles. Such segmentation is invaluable for applications like urban planning, environmental monitoring, and search-and-rescue operations.

We employed **SegFormer** [**?**], a state-of-the-art transformer-based model, due to its efficiency and ability to handle large images. However, initial experiments revealed challenges, including poor performance on minority classes (e.g., bicycles, dogs), low-resolution predictions, and unstable training. This report details how we addressed these issues through strategic improvements, resulting in a more robust and accurate segmentation model.

2 Related Work

Semantic segmentation has seen significant advancements with models like U-Net [**?**] and DeepLab [**?**]. Recently, transformer-based models like SegFormer [**?**] have emerged, offering improved efficiency and accuracy. SegFormer, in particular, has shown strong performance on benchmarks like Cityscapes [**?**]. However, applying it to drone imagery presents unique challenges, such as class imbalance and large image sizes, which are addressed in this project.

3 Methodology

3.1 SegFormer Architecture

SegFormer [**?**] is a transformer-based model that processes images in patches, using a hierarchical transformer encoder to capture both local and global context. The decoder then upsamples the features to produce pixel-wise predictions. Its lightweight design makes it suitable for large images, such as those from drones.

3.2 Improvements to the Initial Code

The initial implementation of SegFormer faced several drawbacks, which were addressed through the following improvements:

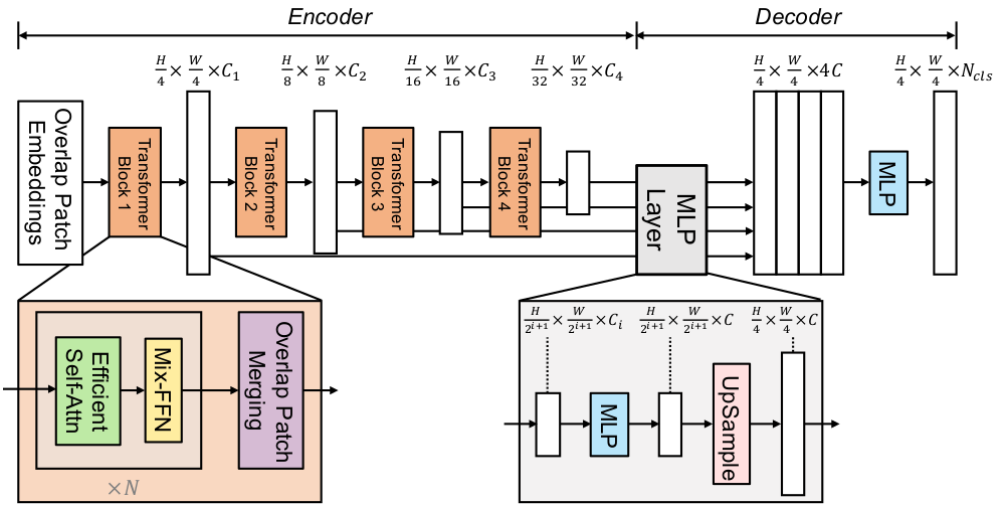

Figure 1: SegFormer Architecture [**?**]

3.2.1 Class Weighting

To combat class imbalance, we introduced class weights into the cross-entropy loss function. The weight w_c for each class c was calculated as:

$$w_c = \frac{1}{frequencyofclassc + \epsilon}$$

where ϵ is a small constant to avoid division by zero. This adjustment amplified the importance of minority classes like 'bicycle' and 'dog', improving their detection.

3.2.2 Upsampling

The initial predictions were at a lower resolution than the input images, leading to blurry segmentation masks. We applied bilinear interpolation to upsample the logits:

$$upsampled\_logits = interpolate(logits, size = (H, W), mode = "bilinear")$$

This ensured the predictions matched the original image resolution.

3.2.3 Learning Rate Scheduler

To stabilize training, we implemented a polynomial learning rate scheduler:

$$lr_t = lr_0 \cdot \left(1 - \frac{t}{T}\right)^{0.9}$$

where lr_0 is the initial learning rate, t is the current epoch, and T is the total number of epochs. This gradually reduced the learning rate, promoting smoother convergence.

3.2.4 Patch Inference

To handle large drone images, we used patch inference with a sliding window approach. The overlap between patches was set to 25% of the patch size:

$$overlap = 0.25 \times patch\_size$$

Predictions from overlapping patches were averaged to produce seamless results.

4 Experiments

4.1 Dataset

We used the `semantic_drone_dataset`, which contains high-resolution drone images and corresponding semantic label maps with 24 classes, including ‘paved-area‘, ‘grass‘, ‘bicycle‘, and ‘dog‘.

4.2 Training Setup

- **Data Split**: 80% training, 20% validation. - **Augmentations**: Random resizing, cropping, and horizontal flipping. - **Batch Size**: 4 - **Epochs**: 50 - **Optimizer**: AdamW with initial learning rate 0.001

4.3 Evaluation Metrics

- **Accuracy**: Overall pixel-wise accuracy. - **Intersection over Union (IoU)**: Measures the overlap between predicted and true regions. - **Dice Score**: Evaluates the similarity between predicted and true segmentations.

5 Results

The initial code produced suboptimal results, with minority classes like ‘bicycle‘ and ‘dog‘ achieving Dice and IoU scores of 0.000. After improvements, the model achieved:

- **Mean Accuracy**: 0.888 - **Mean IoU**: 0.256 - **Mean Dice**: 0.307

Notably, minority classes saw significant gains:

Class	Dice (Initial)	Dice (Improved)
paved-area	0.892	0.894
bicycle	0.000	0.238
dog	0.000	0.035

Table 1: Comparison of Dice Scores for Selected Classes

6 Discussion

The improvements—particularly class weighting and upsampling—were crucial in addressing the initial code’s limitations. Class weighting mitigated the class imbalance, allowing the model to detect minority classes effectively. Upsampling resolved the resolution mismatch, producing sharper and more accurate segmentation masks. However, some classes, such as ‘conflicting‘, still pose challenges and may require additional techniques like focal loss or more training data.

7 Conclusion

This project demonstrates the successful application of SegFormer to semantic segmentation of drone imagery, enhanced through targeted improvements. By addressing class imbalance, resolution issues, and training stability, we achieved significant performance gains, particularly for minority classes. Future work could explore advanced loss functions or additional data augmentation to further improve results.