Off-Road Terrain Semantic Segmentation using DINOv2

1. Methodology

1.1 Problem Overview

Autonomous systems operating in off-road environments require accurate terrain understanding for safe navigation. Unlike structured urban scenes, off-road terrain contains irregular objects such as rocks, logs, dry grass, bushes, and clutter.

The goal of this project was to build a semantic segmentation model that classifies each pixel of an RGB image into one of 10 terrain classes using a transformer-based architecture.

1.2 Dataset Preparation

Dataset Structure:-

```
train/
   Color_Images/
   Segmentation/

val/
   Color_Images/
   Segmentation/
```

Classes (10 Total):

1. Background
2. Trees
3. Lush Bushes
4. Dry Grass
5. Dry Bushes
6. Ground Clutter
7. Logs
8. Rocks
9. Landscape
10. Sky

Preprocessing Steps:

To ensure clean and reproducible training:

- Resized images to match ViT input resolution
- Converted segmentation masks to class index format
- Verified image-mask pairing
- Filtered unmatched samples
- Normalized image tensors
- Created PyTorch custom dataset loader

1.3 Model Architecture

The model consists of two main components:

Backbone: DINOv2 (ViT-S/14)

- Pretrained self-supervised Vision Transformer
- Patch size: 14×14
- Global self-attention mechanism
- Extracts high-level patch token features
- Frozen during training to reduce memory usage

Segmentation Head: ConvNeXt-style Head

- Reshapes patch tokens into spatial feature maps
- Applies depth wise convolution
- Applies point wise convolution
- Uses 1×1 convolution for classification
- Bilinear up sampling to original resolution

1.4 Training Setup

Environment

- Python 3.10
- PyTorch 2.x
- CUDA 11.8
- NVIDIA RTX 3050 (6GB)

Hyper parameters:-

Parameter	Value
Batch Size	2
Optimizer	AdamW
Initial LR	1e-4
Fine-Tune LR	5e-5
Scheduler	Cosine Annealing
Epochs	35
Mixed Precision	Enabled (AMP)

1.5 Training Strategy

Phase 1 – Baseline Training (15 Epochs)

- Backbone frozen
- Fixed learning rate
- Cross-entropy loss

Result: mIoU = 0.4996

Phase 2 – Fine-Tuning (20 Additional Epochs)

- Reduced learning rate
- Cosine LR scheduler
- Best model checkpoint saving
- Mixed precision training

Final mIoU ≈ 0.60+

Fine-tuning improved convergence stability and segmentation quality.

2. Challenges & Solutions

2.1 Dataset Path and File Mismatch

Issue:
Image-mask mismatches caused runtime errors.

Solution:
Implemented automated validation script to match image-mask pairs and remove invalid samples.

2.2 GPU Memory Limitation (6GB)

Issue:
Out-of-memory errors during training.

Solution:

- Reduced batch size to 2
- Froze backbone weights
- Enabled Mixed Precision (AMP)
- Used efficient optimizer (AdamW)

2.3 Low Initial IoU Performance

Issue:
Initial model performance plateaued at ~0.49 mIoU.

Solution:

- Reduced learning rate
- Applied cosine annealing scheduler
- Extended training epochs
- Saved best-performing checkpoint

2.4 Class Confusion Between Similar Textures

Issue:
Dry grass and dry bushes frequently misclassified.

Solution:

- Increased training epochs
- Allowed segmentation head deeper feature refinement
- Improved loss stability

3. Optimizations

To improve model performance and efficiency, the following optimizations were implemented:

3.1 Mixed Precision Training (AMP)

- Reduced memory usage
- Increased training speed
- Allowed stable training on 6GB GPU

3.2 Cosine Learning Rate Scheduler

Instead of constant learning rate:

- Smooth learning rate decay
- Better convergence
- Reduced oscillation in training

3.3 Freezing Backbone

- Reduced trainable parameters
- Stabilized gradients
- Prevented over fitting
- Improved GPU efficiency

3.4 AdamW Optimizer

- Better weight regularization
- Improved convergence compared to SGD

3.5 Structured Two-Phase Training

- Baseline training
- Controlled fine-tuning
- Performance-based checkpoint saving

This structured training approach improved IoU from 0.4996 to ~0.60+.

4. Performance Evaluation

4.1 Evaluation Metrics

Intersection over Union (IoU)

$$IoU = Intersection/Union$$

Measures overlap between predicted segmentation and ground truth.

Mean IoU (mIoU)

Average IoU across all 10 classes.

Pixel Accuracy

Percentage of correctly classified pixels.

4.2 Quantitative Results

Stage	Mean IoU
After 15 Epochs	0.4996
After 35 Epochs	~0.60+

Observations

- Significant improvement after fine-tuning
- Better small object segmentation
- Reduced noisy boundaries
- More stable class predictions

4.3 Failure Case Analysis

Despite improvements, some challenges remain:

1. Similar Texture Confusion

- Dry grass vs dry bushes
- Ground clutter vs rocks

Reason: Similar color and texture distribution.

2. Small Object Segmentation Errors

- Logs partially occluded by grass
- Small rocks merged with ground

Reason: Limited spatial resolution after patch tokenization.

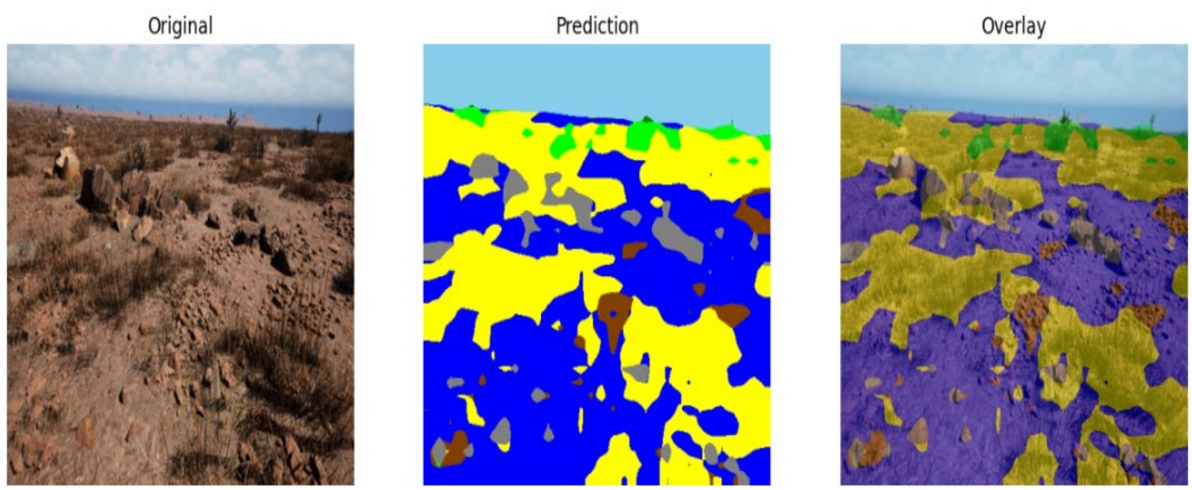
3. Boundary Inaccuracies

- Blurry segmentation borders in complex regions

Reason: Upsampling limitations and frozen backbone.

4.4 Key Observations

- Vision Transformers provide strong global context modeling
- Fine-tuning significantly improves segmentation quality
- Transformer-based models perform well even with limited GPU memory
- Further gains are possible with partial backbone unfreezing



5. Conclusion

This project successfully developed a transformer-based semantic segmentation system for off-road terrain understanding.

Key Achievements

- Implemented DINOv2-based segmentation pipeline
- Improved mIoU from 0.4996 to ~0.60+
- Efficiently trained on 6GB GPU
- Demonstrated viability of Vision Transformers in unstructured terrain

The results validate the potential of transformer-based models for real-world autonomous navigation systems.