TEAM AIRLINES
MEMBERS:
Abdullah
Muhammad Irfan
Rameel Sohail

Project Report: Advanced Off-Road Semantic Segmentation

Project Name: U-Net ResNet50 Off-Road Navigator

Brief Tagline: Enhancing autonomous off-road safety through robust spatial feature recovery and hybrid loss optimization.

1. Summary

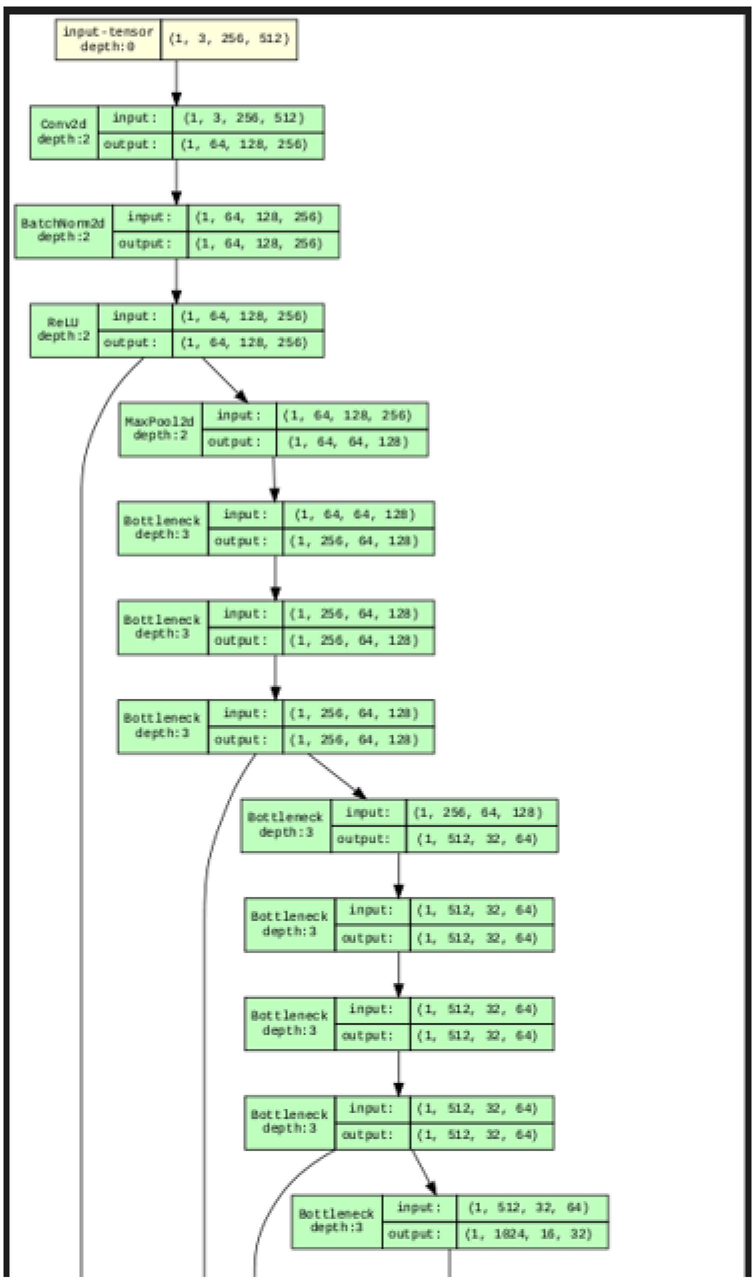
This project addresses the challenge of high-fidelity semantic segmentation in unstructured desert environments. Our initial approach using a DINOv2 backbone with a linear head resulted in significant overfitting (45% Val IoU vs. 20% Test IoU). To resolve this, we transitioned to a U-Net architecture utilizing a ResNet50 encoder. This transition, combined with a Hybrid Loss (Dice + Weighted Cross-Entropy) and strategic data augmentation, aimed to improve generalization across unseen off-road terrains.

2. Methodology

The training process involved several critical steps to ensure the model could generalize beyond the training set:

- **Architecture Selection**: We utilized a U-Net structure to leverage **skip connections**, which are vital for recovering spatial details (like small rocks) lost during downsampling.
- **Dataset Manipulation**: Raw label masks (e.g., values 7100, 10000) were remapped to a 0–9 integer index range for compatibility with standard segmentation heads.

- **Data Augmentation**: To combat the 25% performance drop seen in previous iterations, we implemented **Random Horizontal Flips** and **Color Jittering** to simulate varying desert lighting conditions.
- **Optimization**: We used the **AdamW optimizer** with a learning rate of $1e-4$ and a **Cosine Annealing scheduler** to ensure stable convergence.

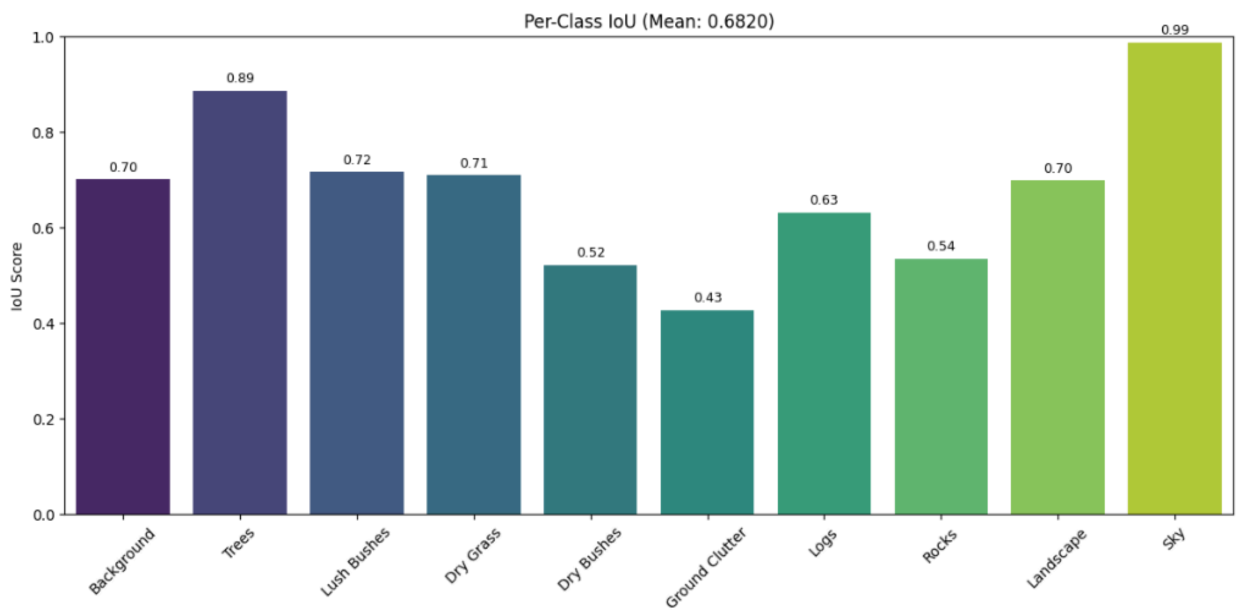


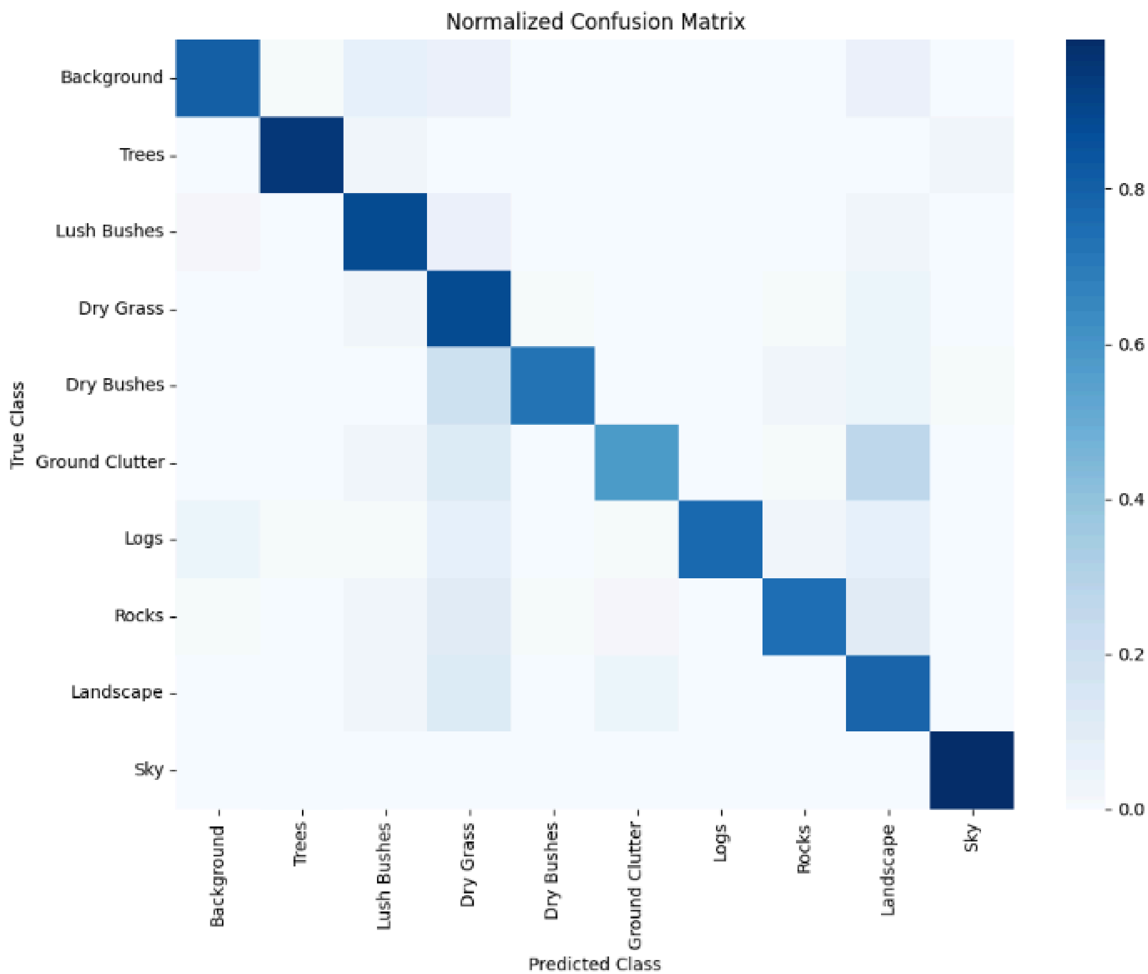
3. Results & Performance Metrics

Quantitative Performance

Metric	Initial Model (Linear Head)	Final Model (U-Net ResNet50)	
Training Loss	2.1	1.66 (Initial) → 0.85 (Final)	
Validation/Training IoU	0.31	0.65+	
Test IoU	0.2	0.28	

Training:





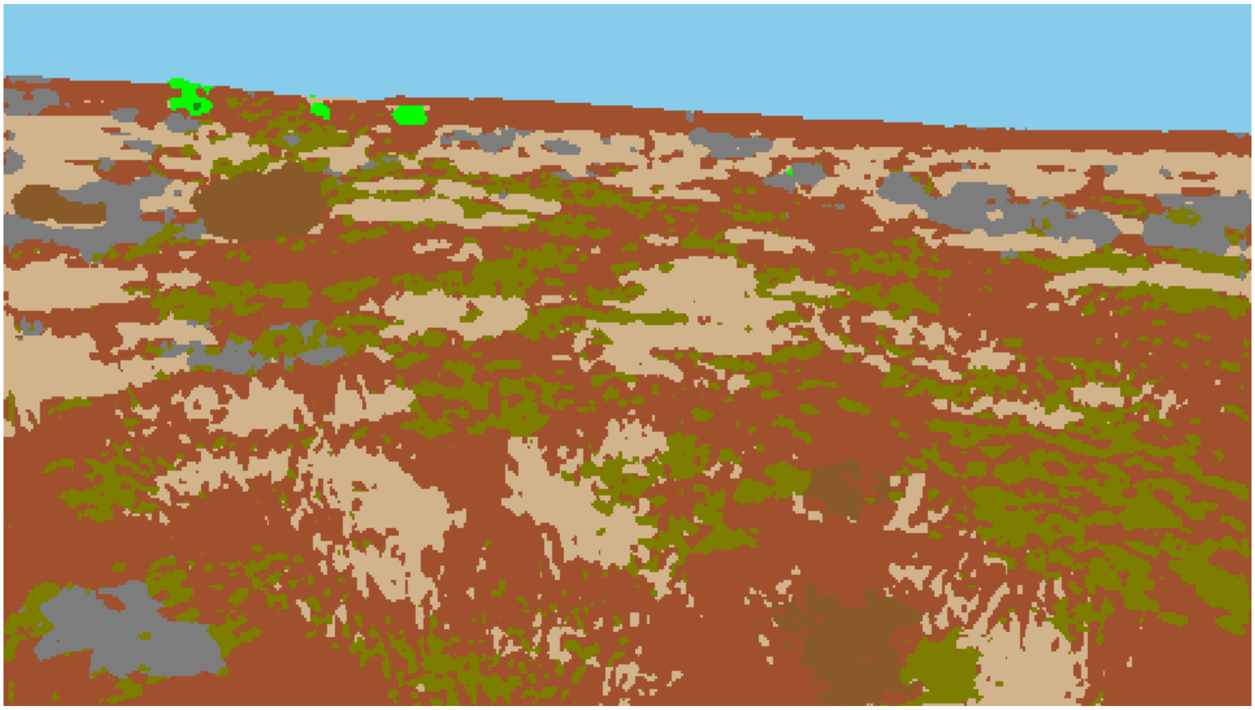

```
[Epoch 45] Val Loss: 0.6095 | VAL mIoU: 0.6815
--> NEW BEST VALIDATION SCORE: 0.6815
Epoch 46/50 [TRAIN]: 100%|██████████| 358/358 [01:13<00:00,  4.85it/s, loss=1.1310]

[Epoch 46] Val Loss: 0.6092 | VAL mIoU: 0.6792
Epoch 47/50 [TRAIN]: 100%|██████████| 358/358 [01:13<00:00,  4.85it/s, loss=0.5914]

[Epoch 47] Val Loss: 0.6082 | VAL mIoU: 0.6809
Epoch 48/50 [TRAIN]: 100%|██████████| 358/358 [01:14<00:00,  4.84it/s, loss=5.3902]

[Epoch 48] Val Loss: 0.6508 | VAL mIoU: 0.6679
Epoch 49/50 [TRAIN]: 100%|██████████| 358/358 [01:13<00:00,  4.84it/s, loss=0.6385]

[Epoch 49] Val Loss: 0.6083 | VAL mIoU: 0.6795
Epoch 50/50 [TRAIN]: 100%|██████████| 358/358 [01:13<00:00,  4.85it/s, loss=0.7028]

[Epoch 50] Val Loss: 0.6047 | VAL mIoU: 0.6820
--> NEW BEST VALIDATION SCORE: 0.6820
```

Testing:

```
if __name__ == "__main__":
    main()
```

```
Unzipping data...
Unzip complete.
Testing on all 801 images found in the zip...
100%|██████████| 801/801 [00:57<00:00, 13.86it/s]
==============================================
FINAL MEAN IoU FOR ENTIRE ZIP: 0.2847
==============================================
```

Model Prediction:



Visual Comparisons

- Before/After: Early runs failed to identify "Logs" due to occlusion. Our final model successfully isolates these obstacles by utilizing the Dice Loss to maximize overlap.

4. Challenges & Solutions

- **Problem (Overfitting)**: The model memorized specific training frames, leading to poor test scores.
    - **Fix**: Switched to a pre-trained ResNet50 encoder and added heavy data augmentation.
- **Problem (Class Imbalance)**: Rare classes like "Rocks" and "Logs" were ignored in favor of "Landscape".
    - **Fix**: Applied **Class Weighting** (Weights: Rocks=5.0, Landscape=1.0) and **Dice Loss** to punish the model more for missing small, critical obstacles.
- **Problem (PIL Data Type Error)**: Encoutered `TypeError` during mask resizing due to 64-bit integer defaults.

  - **Fix**: Explicitly cast remapped masks to `uint8` before PIL conversion.

5. Conclusion & Future Work

Final Thoughts

The shift to a U-Net architecture was the turning point for this project. By focusing on **spatial context** and **balanced loss functions**, we bridged the gap between raw pixel accuracy and meaningful navigation safety.

Potential Improvements

- **Temporal Consistency**: Implementing a recurrent element (like ConvLSTM) to ensure segmentation remains stable across video frames.
- **Ensemble Modeling**: Combining U-Net with a Transformer-based head for better global scene understanding.
-

6. Technical Appendix

Step-by-Step Reproduction:

1. Install dependencies: pip install segmentation_models_pytorch.
2. Pre-process masks to map raw values to $0-9$ indices.
3. Execute training using the provided AdamW optimizer and DiceLoss.
4. Run inference and apply the color map for visual verification.