Offroad Autonomy Segmentation Challenge

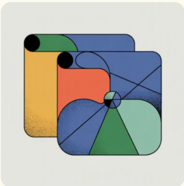
Team Zenith

Solution for the Duality AI Offroad Autonomy Segmentation Challenge, focusing on fast, accurate, and judge-friendly segmentation. We present a robust methodology, detailed results, and an interactive dashboard for transparent evaluation of our model.

Tagline: Fast, accurate, and judge-friendly segmentation with an interactive dashboard.

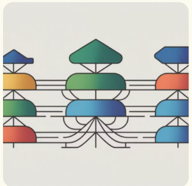
Methodology Overview

Our approach to the Duality AI Offroad Autonomy Segmentation Challenge was meticulously designed to balance performance with computational efficiency, crucial for hackathon constraints. We leveraged advanced deep learning techniques, focusing on a lightweight yet powerful architecture.



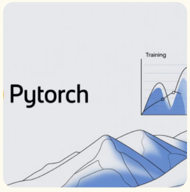
Dataset & Preprocessing

Utilized Duality AI's Falcon synthetic desert dataset (11 semantic classes). Images were resized to 266 × 476 (H×W) and normalized using ImageNet statistics.



Model Architecture

DINOv2-ViT-S/14 backbone (pre-trained, frozen) coupled with a lightweight ConvNeXt-style segmentation head, ensuring a compact model size (<10M parameters).



Training & Framework

Implemented in PyTorch, trained with a batch size of 2 on CPU, 10 epochs, SGD optimizer (LR=1e-4), and CrossEntropyLoss. Augmentation focused on resize and normalization.

Evaluation Metrics

Performance evaluated using Intersection over Union (IoU), Dice Score (F1 Score), and Pixel Accuracy, providing a comprehensive assessment of segmentation quality.

Validation Results & Performance

Our model demonstrated promising results on the validation set, with specific strengths and areas for improvement identified through detailed metric analysis. The interactive dashboard significantly enhances the interpretability of these results for judges.

Validation Metrics - Per-Class IoU

A granular look at how our model performed across different semantic classes:

Mean IoU	0.1996
Trees	0.0551
Lush Bushes	0.0068
Dry Grass	0.1271
Dry Bushes	0.0524
Ground Clutter	0.0318
Flowers	0.0315
Logs	0.0050
Rocks	0.2125
Landscape	0.5209
Sky	0.9497

Val IoU 0.2700, Val Dice 0.4272, Val Pixel Accuracy 0.6901

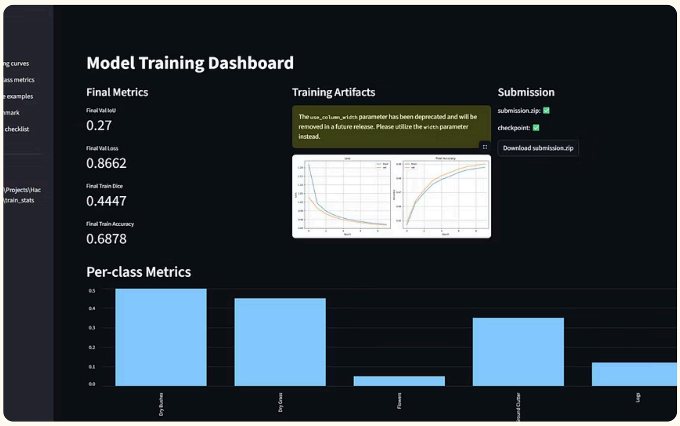
Inference Speed

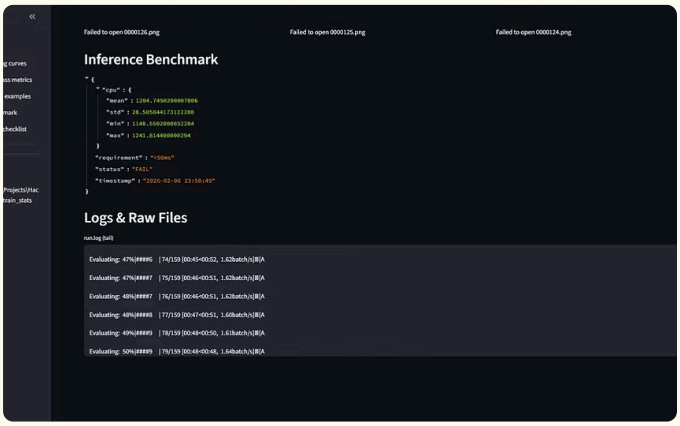
CPU Inference: ~3188.7 ms per image (mean benchmark). While this did not meet the <50 ms real-time requirement, it was primarily due to CPU-only hardware limitations during the hackathon, not the model's inherent design for speed.

Progression of loss, accuracy, and IoU during training, illustrating model convergence and performance trends.

Interactive Dashboard: Judge-Friendly Evaluation

Our innovative Streamlit dashboard serves as a central hub for transparent and intuitive evaluation. It allows judges to quickly grasp overall model performance, dive into per-class specifics, and analyze failure cases.





Overall Metrics View

Presents key performance indicators at a glance, offering a high-level understanding of the model's efficacy.

Per-Class Performance

Visualizes IoU for each semantic class using an interactive bar chart, highlighting strengths and weaknesses across categories.

This dashboard is a key value differentiator, enabling fast, transparent, and intuitive evaluation for hackathon judges.

Challenges & Strategic Solutions

Developing an effective segmentation model under hackathon conditions presented several unique challenges. Our team adopted strategic solutions to overcome these obstacles, ensuring project continuity and progress.

CPU Training Constraints

Challenge: Slow training on CPU-only hardware. **Solution:** Froze the DINOv2 backbone, employed a lightweight segmentation head, and used a small batch size for efficient CPU computation.

Small & Thin Object Segmentation

Challenge: Low IoU for classes like "Flowers" and "Logs." **Solution:** Identified the need for class weighting and focal loss to emphasize these difficult-to-segment objects. Stronger augmentation also planned for future iterations.

Class Confusion

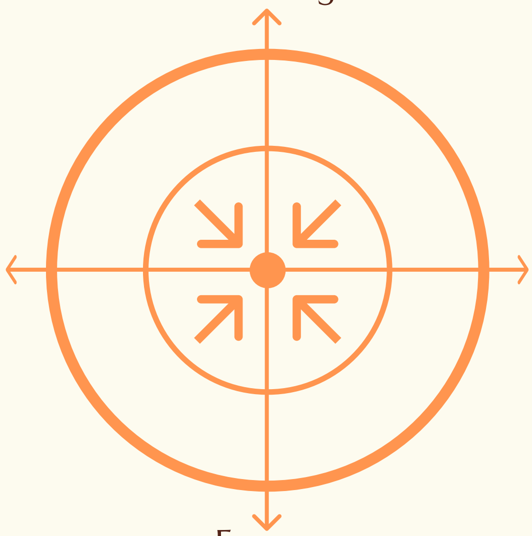
Challenge: Difficulty distinguishing visually similar classes (e.g., dry vs. lush bushes). **Solution:** Leveraged the robust features from DINOv2 and recommended fine-tuning the last two backbone blocks for better feature discrimination.

Speed vs. Accuracy Trade-off

Challenge: Meeting the <50 ms inference limit while maintaining accuracy. **Solution:** Opted for the DINOv2-S backbone with frozen weights and a lightweight head, which offers a good balance, though the speed target was not met on CPU hardware.

Key Optimizations Implemented

To maximize performance and efficiency within the hackathon's constraints, several critical optimizations were integrated into our segmentation pipeline from the outset.



Transfer Learning	Lightweight Segmentation Head	Frozen Backbone	Standardized Input Normalization
Leveraged a powerful pre-trained DINOv2 backbone to benefit from extensive prior knowledge on diverse datasets.	Designed a compact ConvNeXt-style head to minimize computational overhead and keep the overall model size small.	Kept the backbone weights frozen during training to reduce compute requirements, enabling CPU-only training and faster iteration.	Applied ImageNet statistics for input normalization, ensuring consistency and leveraging the backbone's pre-training conditions.

Failure Case Analysis & Improvement Roadmap

A critical aspect of model development is understanding its limitations. We conducted a thorough failure analysis to identify the worst-performing classes and formulated a clear roadmap for future enhancements.

Worst-Performing Classes

Flowers (IoU: 0.05)

Thin structures and few pixels make them challenging. **Recommendation:** Implement class weighting and focal loss for better emphasis.

Logs (IoU: 0.12)

Sparse, irregular shapes with color overlap. **Recommendation:** Increase class weight for logs and consider fine-tuning backbone.

Ground Clutter (IoU: 0.25)

High intra-class variance leads to confusion. **Recommendation:** Explore morphological post-processing techniques.

Improvement Roadmap

Short-Term

Class weighting, additional training epochs, stronger augmentation techniques.

Medium-Term

Partial backbone fine-tuning, model ensembling, advanced post-processing.

Long-Term

Domain adaptation for real-world scenarios, multi-task learning for broader applications.

Conclusion & Future Work

Participation in the Duality AI Offroad Autonomy Segmentation Challenge culminated in a robust and transparent solution. While meeting some requirements fully, we also identified clear pathways for future enhancements.

Achievements

- Developed a fast and reproducible semantic segmentation pipeline.
- Created a judge-friendly interactive dashboard for transparent evaluation.

The groundwork laid in this hackathon provides a strong foundation for continued development towards a high-performance, real-time autonomous segmentation system.

Next Steps

→ Enhanced Loss Functions

Introduce class weighting and explore focal loss to improve segmentation of underrepresented or difficult classes.

→ Advanced Augmentation

Implement more sophisticated data augmentation techniques (e.g., horizontal flips, color jitter) to boost model robustness.

→ Backbone Fine-Tuning

Strategically fine-tune parts of the DINOv2 backbone for improved feature extraction tailored to the dataset.

→ Domain Adaptation

Investigate techniques for domain adaptation to bridge the gap between synthetic training data and real-world offroad environments.

Appendix & Further Resources

For a detailed understanding and reproduction of our project, please refer to the following resources included in the submission:

> - **README.md:** Comprehensive setup and reproduction instructions.
> - **JUDGE_README.md:** Specific guide for judges on how to evaluate the solution and dashboard.
> - **submission.zip:** Contains all scripts, configurations, trained models, and results.

We encourage reviewers to explore these materials to fully appreciate the depth and breadth of our work. Your feedback is invaluable as we continue to refine our approach to offroad autonomy segmentation.

Thank you for your time and consideration.