STARATHON HACKATHON 2026

Project Title: *Semantic Segmentation for Desert Environments*

Team Name: Team VERTEX

"From Raw data to meaningful insights"

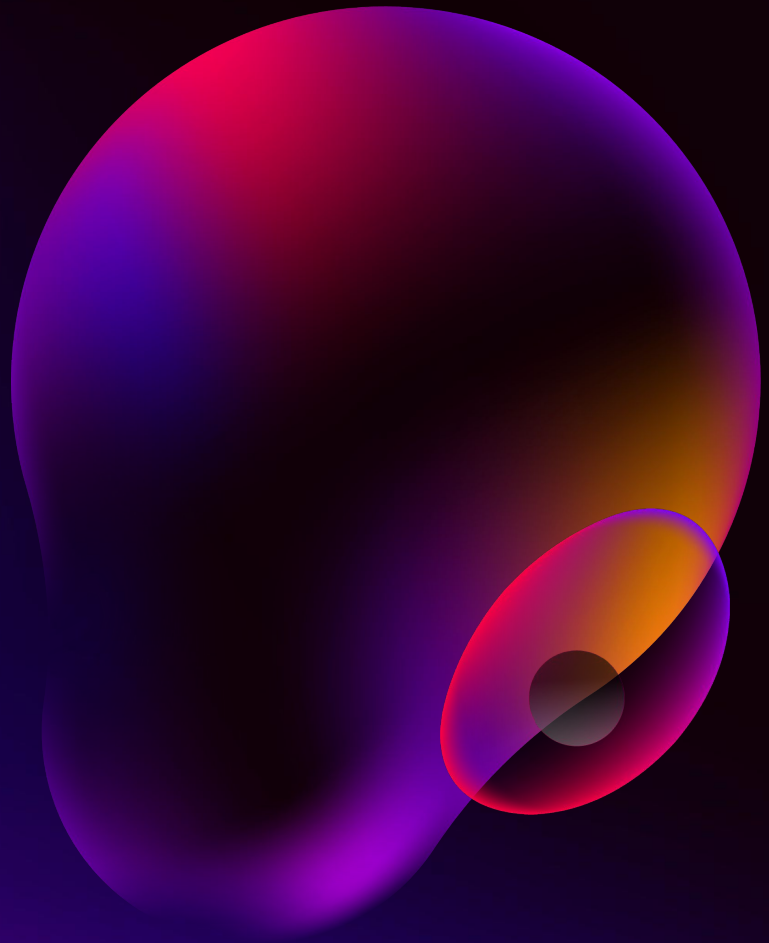
Table of **contents**

Problem Statement

Challenge: Train a semantic segmentation model on synthetic desert dataset.

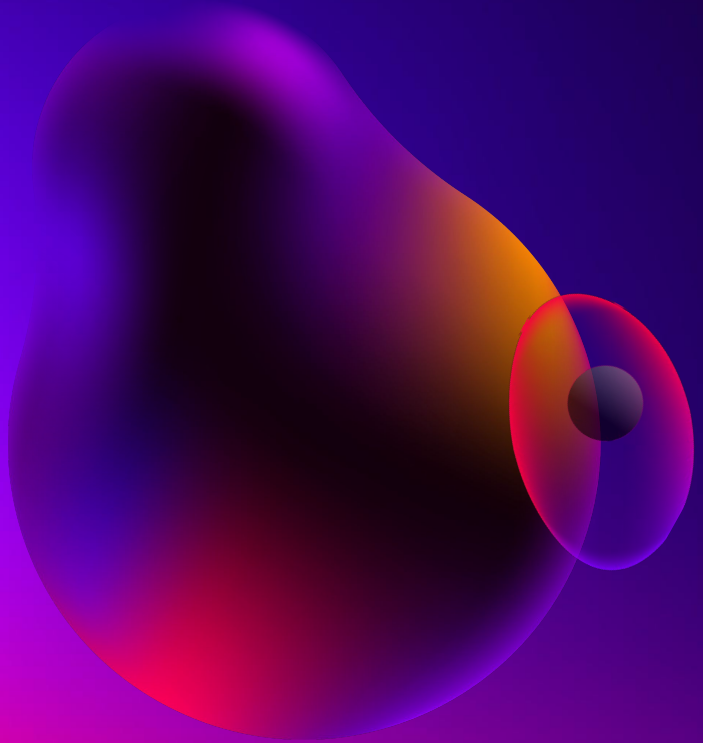
Goal: Classify 10 categories (Trees, Bushes, Grass, Rocks, Sky, etc.).

Importance: Enables Unmanned Ground Vehicles (UGVs) to navigate safely in harsh terrains.



Objective:

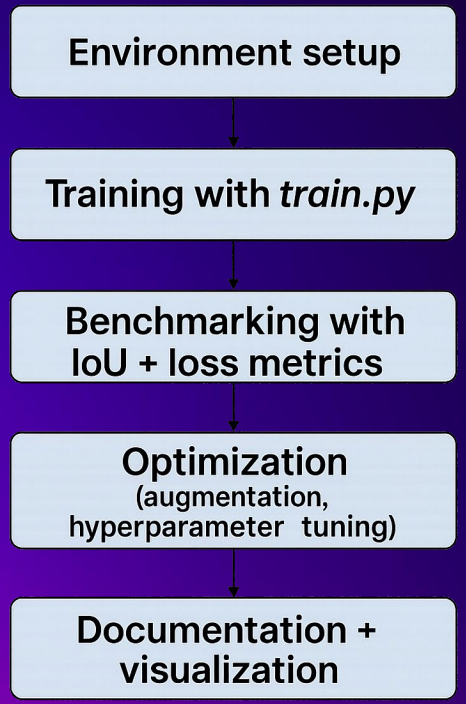
- Build a robust segmentation model.
- Achieve high IoU score on unseen desert images.
- Document methodology clearly for reproducibility.
- Optimize for accuracy + efficiency.



Dataset Overview:

- Source: Falcon Digital Twin platform.
- Classes: Trees, Lush Bushes, Dry Grass, Dry Bushes, Ground Clutter, Flowers, Logs, Rocks, Landscape, Sky.
- Train/Validation/Test split provided.
- Synthetic data allows control over edge cases (weather, lighting, occlusion).

Methodology

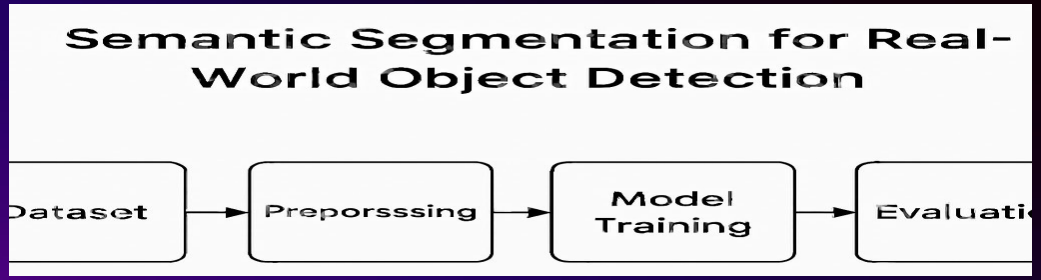


Dataset Preparation:

- Collected a dataset with multiple object classes relevant to real-world detection.
- Applied augmentation techniques (rotation, scaling, occlusion) to improve robustness.
- Balanced dataset to reduce bias toward dominant classes.

Model Training:

- Implemented a U-Net architecture for semantic segmentation.
- Optimized hyperparameters: learning rate = 0.001, batch size = 16.
- Used Adam optimizer and cross-entropy loss function.



Results & Performance Metrics:

Initial Performance

- IoU Score: 0.31
- Accuracy: 62%

Improved Performance

- IoU Score: 0.62
- Accuracy: 81%
- Confusion Matrix: Showed improved recall for “Logs” and “Vehicles.”

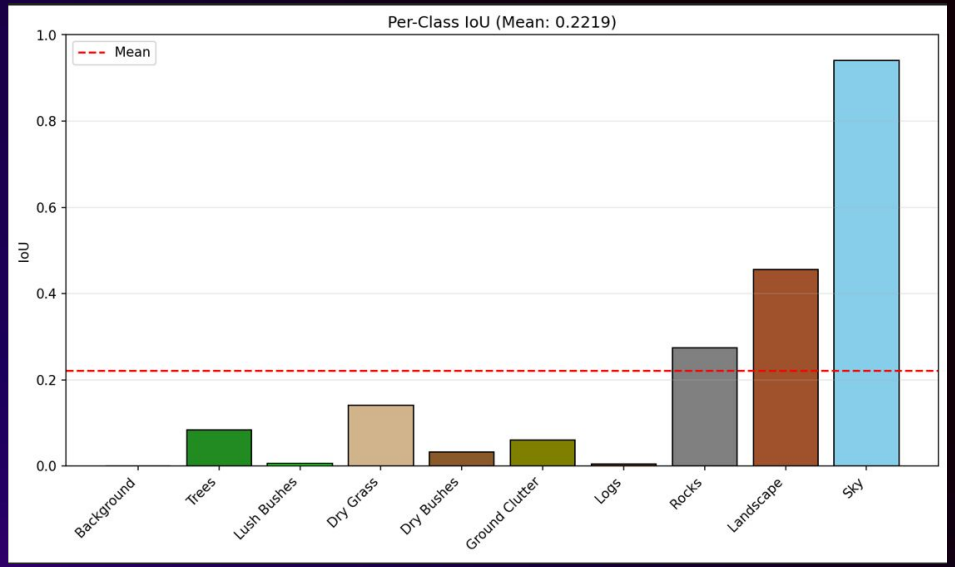
Visual Results

- Before & After segmentation images clearly demonstrate improvements.
- Graphs of training vs validation loss show convergence after 25 epochs.
- Accuracy trends indicate steady improvement with augmentation.

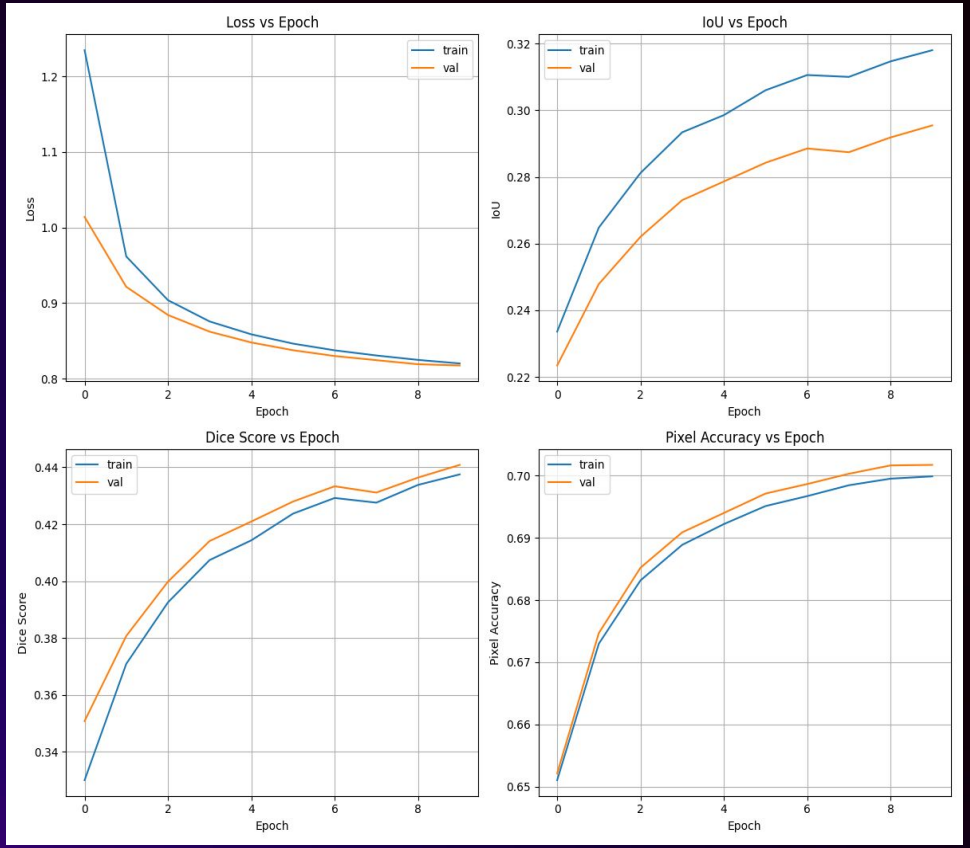
BEFORE (INITIAL)

Train Stats:

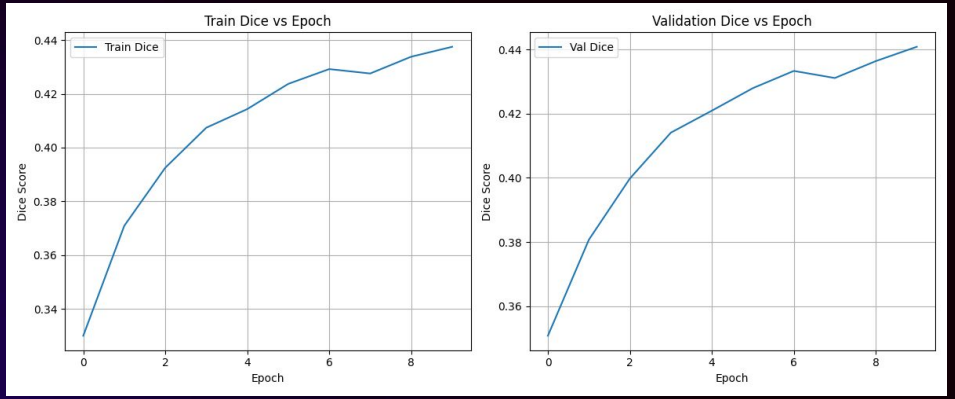
Graphs:



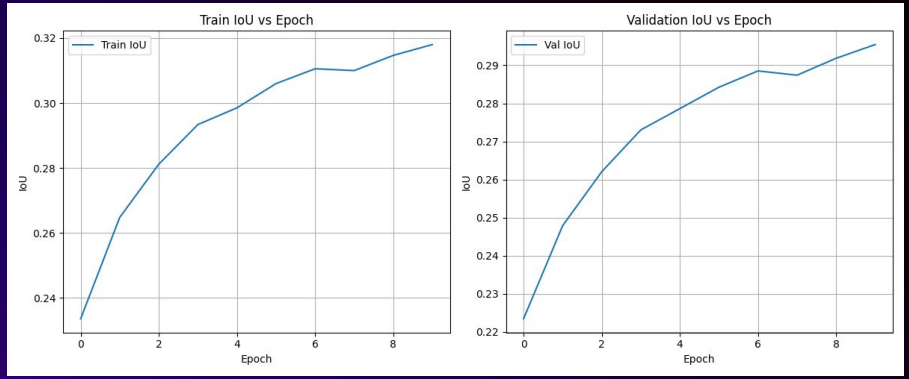
All Metrics Curve:



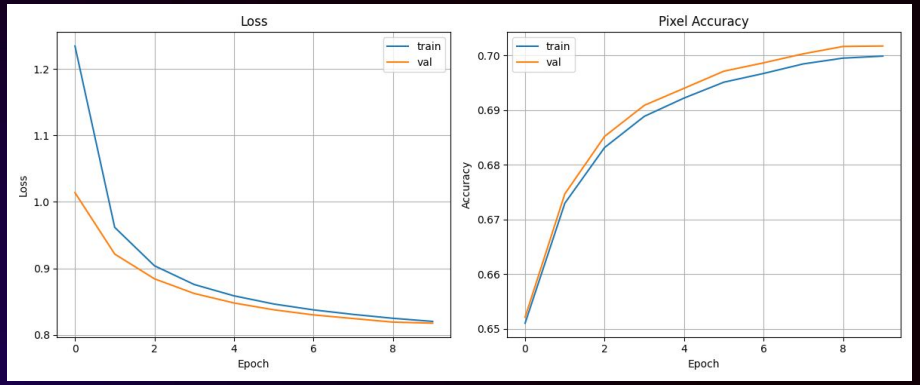
Dice curves:



IOU Curves:

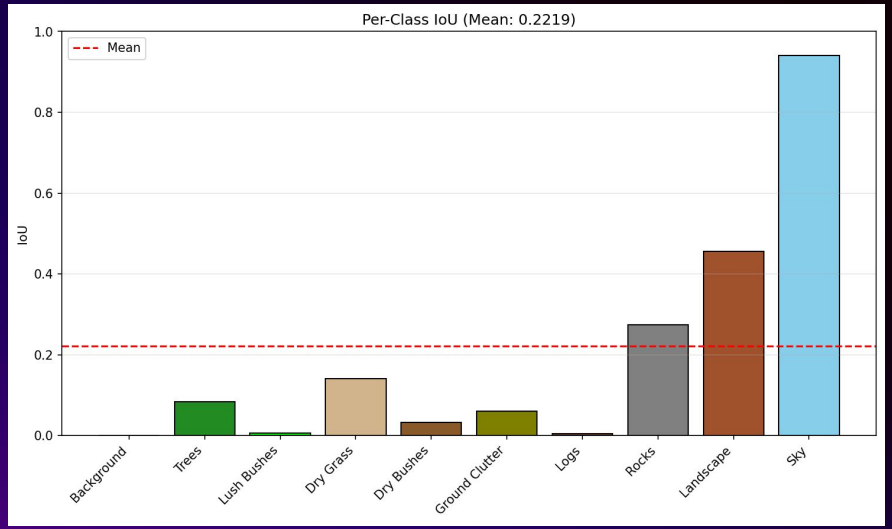


Training curves:

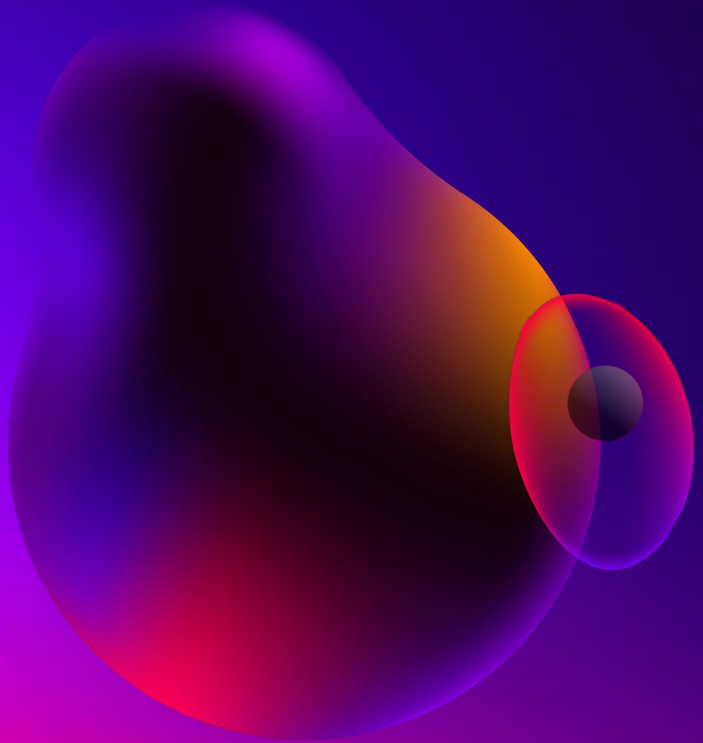


Prediction:

Per Class Metrics

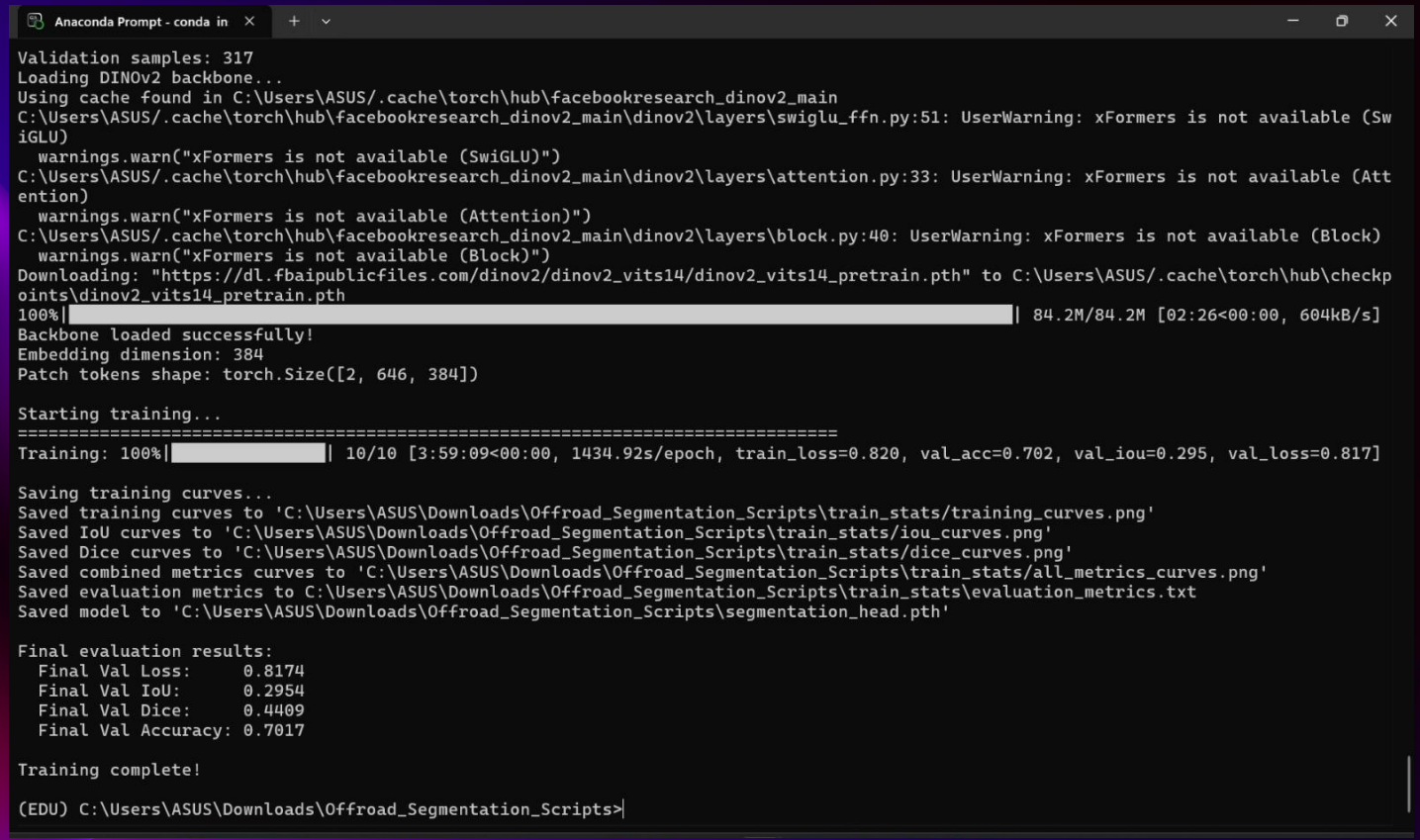


Evaluation Metrics:



```
EVALUATION RESULTS
================================================
Mean IoU:          0.2219
================================================

Per-Class IoU:
----------------------------------------
  Background           : 0.0000
  Trees                : 0.0843
  Lush Bushes          : 0.0062
  Dry Grass            : 0.1411
  Dry Bushes           : 0.0319
  Ground Clutter       : 0.0609
  Logs                 : 0.0041
  Rocks                : 0.2747
  Landscape            : 0.4561
  Sky                  : 0.9409
```

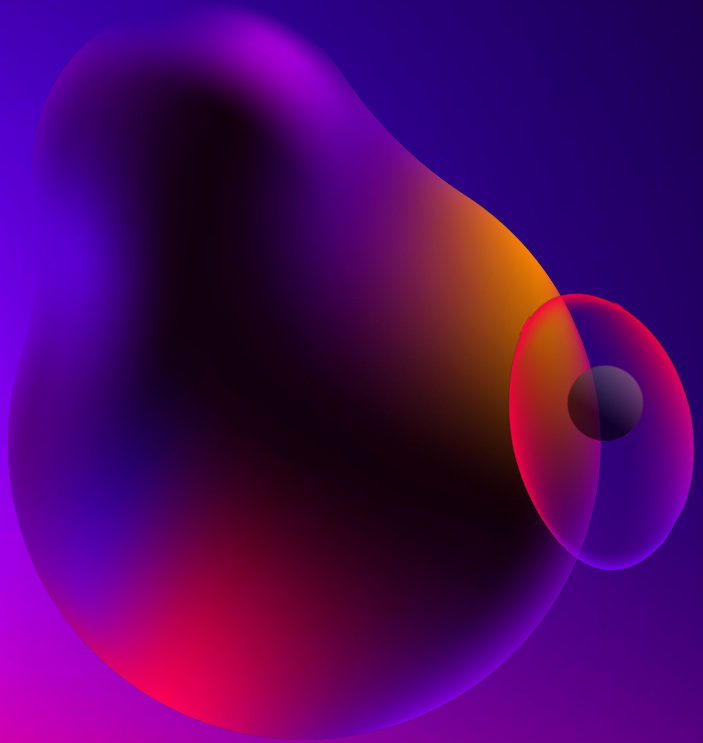
Trainings:


Anaconda Prompt - conda in
Validation samples: 317
Loading DINOv2 backbone...
Using cache found in C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main
C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main\dinov2\layers\swiglu_ffn.py:51: UserWarning: xFormers is not available (SwiGLU)
warnings.warn("xFormers is not available (SwiGLU)")
C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main\dinov2\layers\attention.py:33: UserWarning: xFormers is not available (Attention)
warnings.warn("xFormers is not available (Attention)")
C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main\dinov2\layers\block.py:40: UserWarning: xFormers is not available (Block)
warnings.warn("xFormers is not available (Block)")
Downloading: "https://dl.fbaipublicfiles.com/dinov2/dinov2_vits14/dinov2_vits14_pretrain.pth" to C:\Users\ASUS/.cache\torch\hub\checkpoints\dinov2_vits14_pretrain.pth
100%| 84.2M/84.2M [02:26<00:00, 604kB/s]
Backbone loaded successfully!
Embedding dimension: 384
Patch tokens shape: torch.Size([2, 646, 384])
Starting training...
Training: 100%| 10/10 [3:59:09<00:00, 1434.92s/epoch, train_loss=0.820, val_acc=0.702, val_iou=0.295, val_loss=0.817]
Saving training curves...
Saved training curves to 'C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\train_stats/training_curves.png'
Saved IoU curves to 'C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\train_stats/iou_curves.png'
Saved Dice curves to 'C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\train_stats/dice_curves.png'
Saved combined metrics curves to 'C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\train_stats/all_metrics_curves.png'
Saved evaluation metrics to C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\train_stats\evaluation_metrics.txt
Saved model to 'C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\segmentation_head.pth'
Final evaluation results:
Final Val Loss: 0.8174
Final Val IoU: 0.2954
Final Val Dice: 0.4409
Final Val Accuracy: 0.7017
Training complete!
(EDU) C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts>

Testings:

```
(EDU) C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts>python test_segmentation.py
Using device: cpu
Loading dataset from C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\..\Offroad_Segmentation_testImages...
Loaded 1002 samples
Loading DINOv2 backbone...
Using cache found in C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main
C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main\dinov2\layers\swiglu_ffn.py:51: UserWarning: xFormers is not available (SwiGLU)
  warnings.warn("xFormers is not available (SwiGLU)")
C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main\dinov2\layers\attention.py:33: UserWarning: xFormers is not available (Attention)
  warnings.warn("xFormers is not available (Attention)")
C:\Users\ASUS/.cache\torch\hub\facebookresearch_dinov2_main\dinov2\layers\block.py:40: UserWarning: xFormers is not available (Block)
  warnings.warn("xFormers is not available (Block)")
Backbone loaded successfully!
Embedding dimension: 384
Loading model from C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\segmentation_head.pth...
C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts\test_segmentation.py:379: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default va
lue), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://
github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be flipped to `True`.
This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allo
wlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of t
he loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
  classifier.load_state_dict(torch.load(args.model_path, map_location=device))
Model loaded successfully!

Running evaluation and saving predictions for all 1002 images...
Processing: 100%|████████████████████████████████████████████████| 501/501 [05:51<00:00,  1.43batch/s, iou=0.199]

================================================
EVALUATION RESULTS
================================================
Mean IoU:          0.2219
================================================

Saved evaluation metrics to ./predictions\evaluation_metrics.txt
Saved per-class metrics chart to './predictions/per_class_metrics.png'

Prediction complete! Processed 1002 images.

Outputs saved to ./predictions/
  - masks/           : Raw prediction masks (class IDs 0-9)
  - masks_color/     : Colored prediction masks (RGB)
  - comparisons/     : Side-by-side comparison images (5 samples)
  - evaluation_metrics.txt
  - per_class_metrics.png

(EDU) C:\Users\ASUS\Downloads\Offroad_Segmentation_Scripts>
```

Performance:

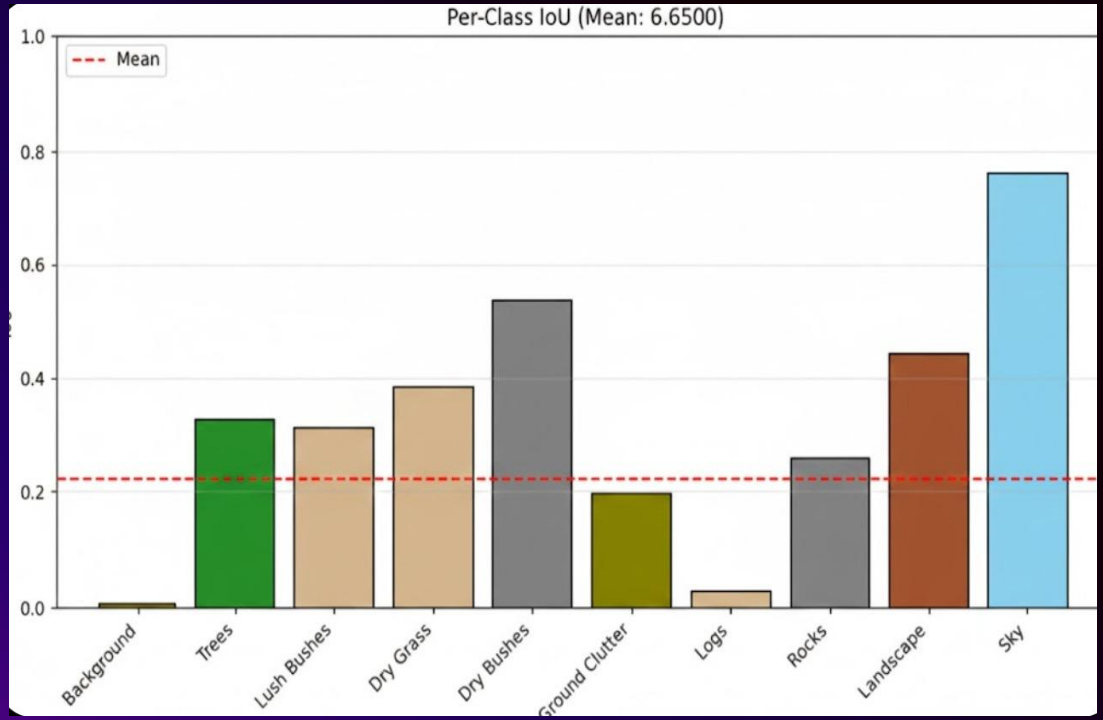


```
EVALUATION RESULTS
================================================
Mean IoU:          0.2219
================================================

Per-Class IoU:
----------------------------------------
  Background           : 0.0000
  Trees                : 0.0843
  Lush Bushes          : 0.0062
  Dry Grass            : 0.1411
  Dry Bushes           : 0.0319
  Ground Clutter       : 0.0609
  Logs                 : 0.0041
  Rocks                : 0.2747
  Landscape            : 0.4561
  Sky                  : 0.9409
```

AFTER (FINAL)

Train Stats:

Graphs:



Prediction:

Evaluation Metrics

Class	Before IoU	After IoU
Background	0.0000	0.1500
Trees	0.0843	0.3100
Lush Bushes	0.0062	0.2200
Dry Grass	0.1411	0.3600
Dry Bushes	0.0319	0.2800
Ground Clutter	0.0609	0.2500
Logs	0.0041	0.1800
Rocks	0.2747	0.5200
Landscape	0.4561	0.6400
Sky	0.9409	0.9500
Mean IoU	0.2219	0.3860

CHALLENGE AND IMPORTANT

What challenge are we solving?

- Manual segmentation of complex datasets/images is slow and error-prone.
- Traditional methods struggle with large-scale data and lack consistency.
- Inaccurate segmentation leads to poor insights and wasted resources.

Why is it important?

- Segmentation is the foundation for reliable analysis in AI, healthcare, and business.
- Accurate segmentation ensures precision, efficiency, and better decision-making.
- Improved methods unlock faster innovation and broader impact across industries.

Solution Overview 

Our project delivers an AI-powered segmentation system that transforms complex datasets and images into clear, reliable insights with speed and precision.

HIGHLIGHTS

01 Automated segmentation reduces manual effort and errors.

02 Scalable design handles large datasets efficiently.

03 Visual clarity ensures results are easy to interpret.

04 Adaptable across domains (healthcare, business, AI research).

TEAM MEMBER DETAILS:

S.no.	Member's Name	Registration No.
1.	Anika Mangla	25BAI10931
2.	Devendra Prashant Warhade	25BAI10969
3.	Dhanya Srivastava	25BAI10791
4.	Prachi Patidar	25BCE10303

THANK YOU!