Off-Road Autonomy Semantic Segmentation Using Synthetic Data

TEAM: THE 404 SOCIETY

1. ISHAN SHRIVASTAVA
2. DIVYANSH MEHTA
3. SURBHI KUMARI
4. TANISHA DESHPANDE

Training an AI model to understand desert environments for autonomous navigation.

This report explains how we trained and evaluated a semantic segmentation model to identify terrain elements in off-road desert environments. The model IS trained using synthetic data and tested on new environments to evaluate generalization and real-world readiness.

PROBLEM AND OBJECTIVE

Autonomous vehicles must understand terrain to move safely.

Off-road environments are complex and unpredictable.

Goal: train an AI model to label every pixel in desert images.

Test model performance on new unseen environments.

Focus: accuracy, robustness, and real-world applicability.

METHODOLOGY

We used:

DeepLabV3-ResNet50 (pretrained)

Final classifier layer modified for 10 classes

```
model = torchvision.models.segmentation.deeplabv3_resnet50(weights="DEFAULT")
model.classifier[4] = nn.Conv2d(256, NUM_CLASSES, kernel_size=1)
```

Training Setup

Parameter	Value
Optimizer	Adam
Learning Rate	1e-4
Epochs	40
Loss Function	CrossEntropyLoss
Device	CUDA (RTX 4050)
Image Size	256x256

Data Processing

Images resized to 256×256

Masks resized using nearest interpolation

Pixel encoding applied using class mapping

Converted to tensors before training

- ## INITIAL RESULTS

First Training Attempt

CPU training

No augmentation

Plateau observed

Final IoU: ~0.50

Problem:

Model stopped improving after 20 epochs.

- ## OPTIMIZATION STRATEGY

Challenge 1: Performance Plateau

Observation: IoU stuck at ~0.50 despite increasing epochs.

Root Cause: Limited data variation → model overfitting to patterns.

Solution: Data Augmentation

Added: Random Horizontal Flip

Random Rotation (±10°)

Code snippet:

```
if random.random() > 0.5:
    image = cv2.flip(image, 1)
    mask = cv2.flip(mask, 1)
```

- ## DATA SOURCE-SYNTHETIC DATA

Real off-road data is expensive and difficult to collect.

Synthetic environments provide perfectly labeled training data.

Enables large-scale and controlled experimentation.

Data used in this project is generated by Duality AI using **Falcon** and **Falcon Cloud**

- ## METHODOLOGY AND WORKFLOW

1.Dataset Preparation

Loaded synthetic desert images and segmentation masks

Organized images and labels

2.Data Preprocessing

Resized images to fixed dimensions

Normalized pixel values

Converted masks to class labels

3.Data Augmentation

Random flipping

Rotation

Brightness variation

Noise injection

4.Model Training

Used DeepLabV3 segmentation architecture

Compared predictions with ground truth

Optimized model using loss minimization

5.Evaluation

Measured IoU score

Visual inspection of predictions

Tested on unseen environment

- ## SYSTEM FLOW DIAGRAM

Synthetic Dataset

↓

Preprocessing & Augmentation

↓

Model Training

↓

Prediction Generation

↓

Loss Calculation

↓

Weight Optimization

↓

Performance Evaluation (IoU)

↓

Testing on New Environment

DATA PIPELINE TABLE

Stage	Input	Process	Output
Data Loading	Images + masks	Organize dataset	Structured data
Preprocessing	Raw images	Resize + normalize	Clean data
Training	Processed data	Neural network learning	Model
Evaluation	Predictions	IoU calculation	Accuracy score
Testing	New environment	Model inference	Generalization

• RESULTS & PERFORMANCE METRICS

Training loss decreased steadily across epochs

IoU increased as model learned terrain features

Model successfully generalized to new environment

Epoch 5 → 0.538

Epoch 10 → 0.550

Epoch 30 → 0.563

Final → 0.561

Improvement: +0.06 IoU

- ## METRICS

Training Loss vs Epoch

IoU vs Epoch

Confusion Matrix

Accuracy Comparison (baseline vs improved model

- ## CHALLENGES AND SOLUTIONS

Challenge Slow Training

CPU Training Time: 23 minutes per epoch

Solution: Installed CUDA-enabled PyTorch.

After GPU setup: 1.2 minutes per epoch

Training time **reduced by ~18x.**

This allowed rapid experimentation and tuning.

Result

Improved class detection

Higher IoU score

Conclusion

Model successfully learned pixel-level terrain classification

Synthetic data enabled scalable training

IoU improved through augmentation and optimization

Model shows potential for real-world off-road autonomy

- **Future Improvements**
  1. Train with mixed real + synthetic data
  2. Improve small object detection
  3. Reduce inference time
  4. Test in more diverse terrain environments
  5. Deploy in real-time navigation system

GIT repository: https://github.com/ishaann24/the404society