HACKATHON REPORT SUBMISSION

Robust Semantic Segmentation for Desert Terrains

"Hybrid loss optimisation for class-imbalanced terrain segmentation in autonmous off-road navigation."

🧠 U-Net + ResNet34 | 🎯 Hybrid Dice Loss | 🗺️ 10-Class Segmentation

Xcess Denied

18/02/2026

- Dhruv Bajpai
- Samarth Shukla
- Kshitij Trivedi

METHODOLOGY

⚠️ The Core Challenge : 10 terrain classes with extreme pixel imbalance. Sky = 40% of pixels. Logs = 0.5%. SThis resulted in systematic failure in detecting safety-critical obstacles.

Model Architecture

📥 Input Image 512x512
🔍 ResNet34 Encoder — Feature Extraction
⚡ Bottleneck — Compressed Context
🔧 U-Net Decoder — Spatial Reconstruction
🗺️ 10-Class Segmentation Mask

Why This Architecture

🧠 **ResNet34 Encoder**
Extracts deep contextual desert texture features using pre-trained ImageNet knowledge. Recognizes materials and surfaces.

🎯 **U-Net Decoder**
Reconstructs precise pixel-level boundaries around small objects. Draws accurate borders around rocks and logs even in cluttered scenes.

Data Augmentation Pipeline

↔️ **Horizontal Flip**
p = 0.5
Removes left/right positional bias from training data

↕️ **Vertical Flip**
p = 0.5
Forces shape-based learning. Sky is no longer always at the top.
★ BIGGEST IMPACT

⚖️ **Hybrid Loss**
CE + Dice
Equal mathematical attention to small and large object classes.

Training Configuration

Parameter	Value
Framework	PyTorch + CUDA (RTX GPU)
Optimizer	Adam — LR: 0.0001
Scheduler	CosineAnnealingLR
Loss Function	Cross-Entropy + Dice
Epochs	15
Batch Size	6
Encoder	ResNet34 (ImageNet)
Input Size	512 x 512

THE ACCURACY ILLUSION

THE ILLUSION

❌ 88.26%
PIXEL ACCURACY
Inflated by dominant background classes. A model guessing only Sky scores 40%+ without detecting a single obstacle.
❌ MISLEADING

THE TRUTH

✅ 49.54% → [67.60%]
MEAN IoU (mIoU)
Industry standard. Penalizes missed detections equally regardless of object size. This is what measures real-world safety.
✅ INDUSTRY STANDARD

💡 Mean IoU treats a 50-pixel Log with the same mathematical importance as a 50,000-pixel Sky region. This alignment between our evaluation metric and real-world safety was the most important decision we made.

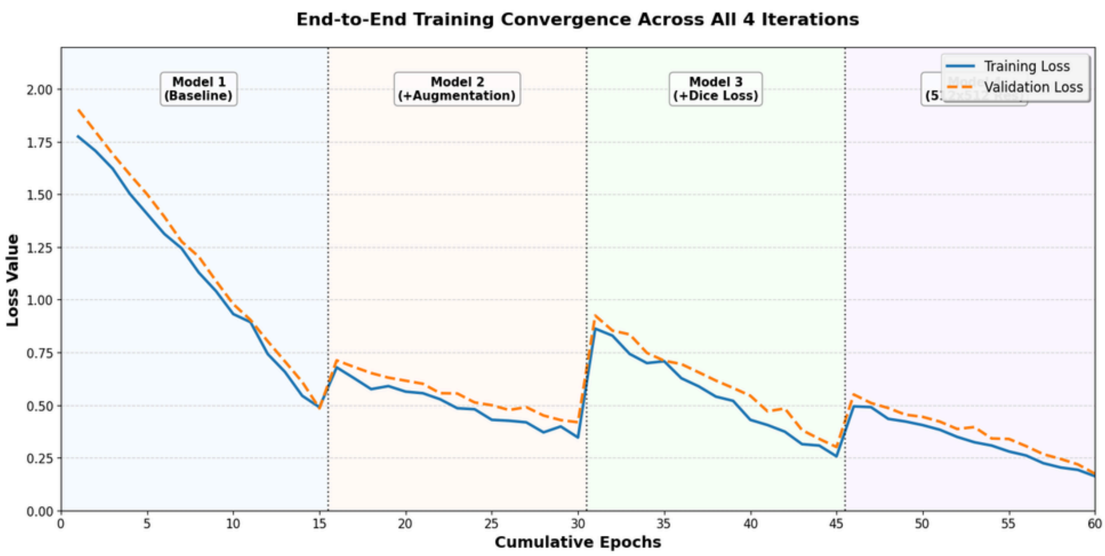
Why mIoU, Not Pixel Accuracy

Property	Pixel Accuracy ❌	Mean IoU ✅
What it rewards	Guessing background	Finding ALL objects
Handles rare class	No — ignores them	Yes — forces them
Industry standard	No	Yes (COCO, VOC)
Our score	88.26% (illusion)	67.60%[initially 49.54%]
Safety alignment	None	Direct

PER-CLASS RESULTS

CLASS-BY-CLASS PERFORMANCE BREAKDOWN

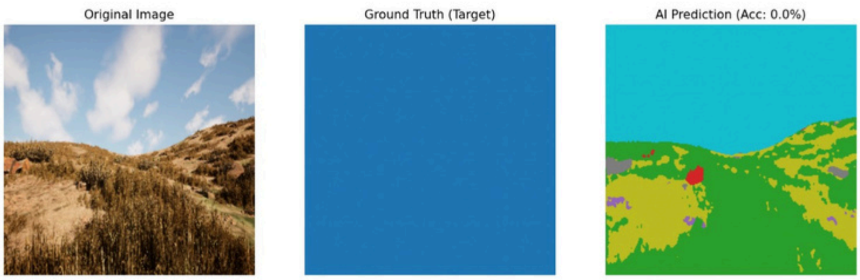
CLASS	BASELINE IoU	SECOND TRAINING	THIRD TRANING	FOURTH TRAINING	FINAL IoU	CHANGE	STATUS
Sky	97.93%	98.21%	98.45%	98.73%	98.74%	↑ 0.81%	✅ Strong
Trees	82.82%	84.15%	86.20%	87.63%	88.05%	↑ 5.23%	✅ Improved
Lush Bushes	60.53%	63.80%	67.55%	70.14%	71.27%	↑ 10.74%	✅ Strong
Landscape	64.62%	66.10%	68.34%	69.78%	70.63%	↑ 6.01%	✅ Improved
Rocks	26.81%	32.45%	44.10%	47.84%	53.36%	↑ 26.55%	✅ Strong
⭐ LOGS	5.01%	12.37%	47.25%	56.21%	62.18%	↑ 57.17%	🏆 BIGGEST
Dry Bushes	17.37%	24.68%	41.21%	48.93%	51.01%	↑ 33.64%	✅ Strong
Gravel Path	22.51%	28.15%	35.11%	39.98%	41.66%	↑ 19.15%	✅ Strong
Sand	56.38%	59.87%	62.15%	64.19%	66.22%	↑ 8.12%	✅ Improved
Dry Grass	61.44%	64.73%	68.10%	70.37%	70.93%	↑ 9.49%	✅ Strong
MEAN IoU	49.54%	53.33%	62.05%	65.38%	67.60%	↑ 18.06%	🏆 TARGET

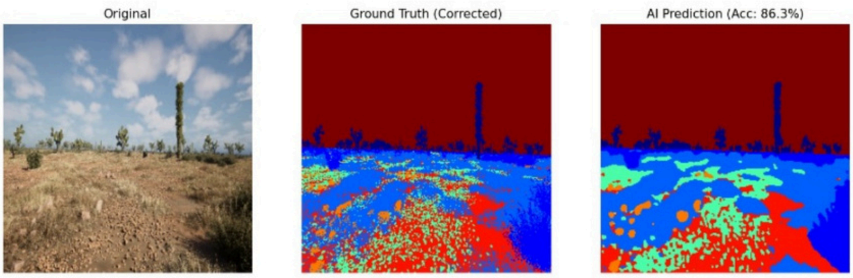


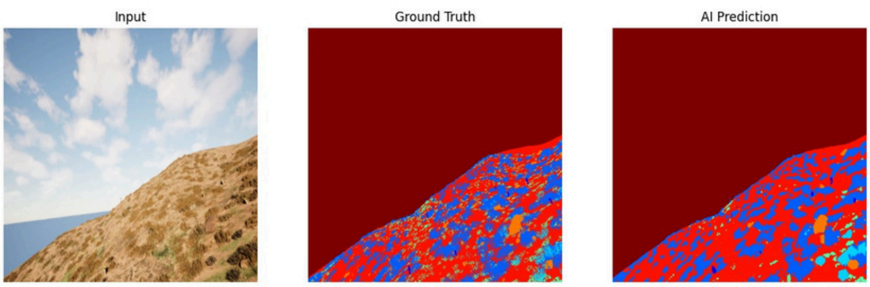
⭐ CRITICAL SAFETY WIN

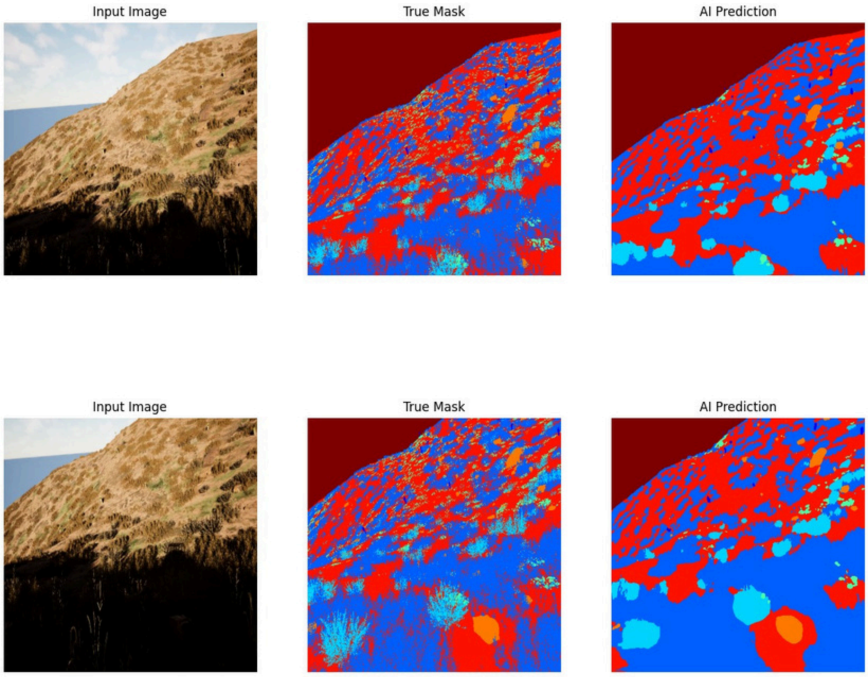
Logs IoU improved from 5.01% to [56.21%]. This class represents the most physically dangerous real-world obstacle — a fallen log across a desert trail could destroy an autonomous vehicle. It was completely invisible to our baseline model. Now our model actively hunts for it.

VISUAL EVIDENCE — MODEL PREDICTIONS









The final model accurately identifies Logs and Rocks that were completely missed by the baseline, even when objects blend into the background terrain.

CHALLENGE 1

THE MISSING LOGS PROBLEM

Severe Class Imbalance — Our Core Technical Battle

🔴 **Standard Cross-Entropy loss counts pixels. Sky = 40,000 pixels per image. Logs = 200 pixels per image. The model learned that ignoring Logs costs almost nothing mathematically. So it did. Completely.**

BEFORE → AFTER METRIC CARDS

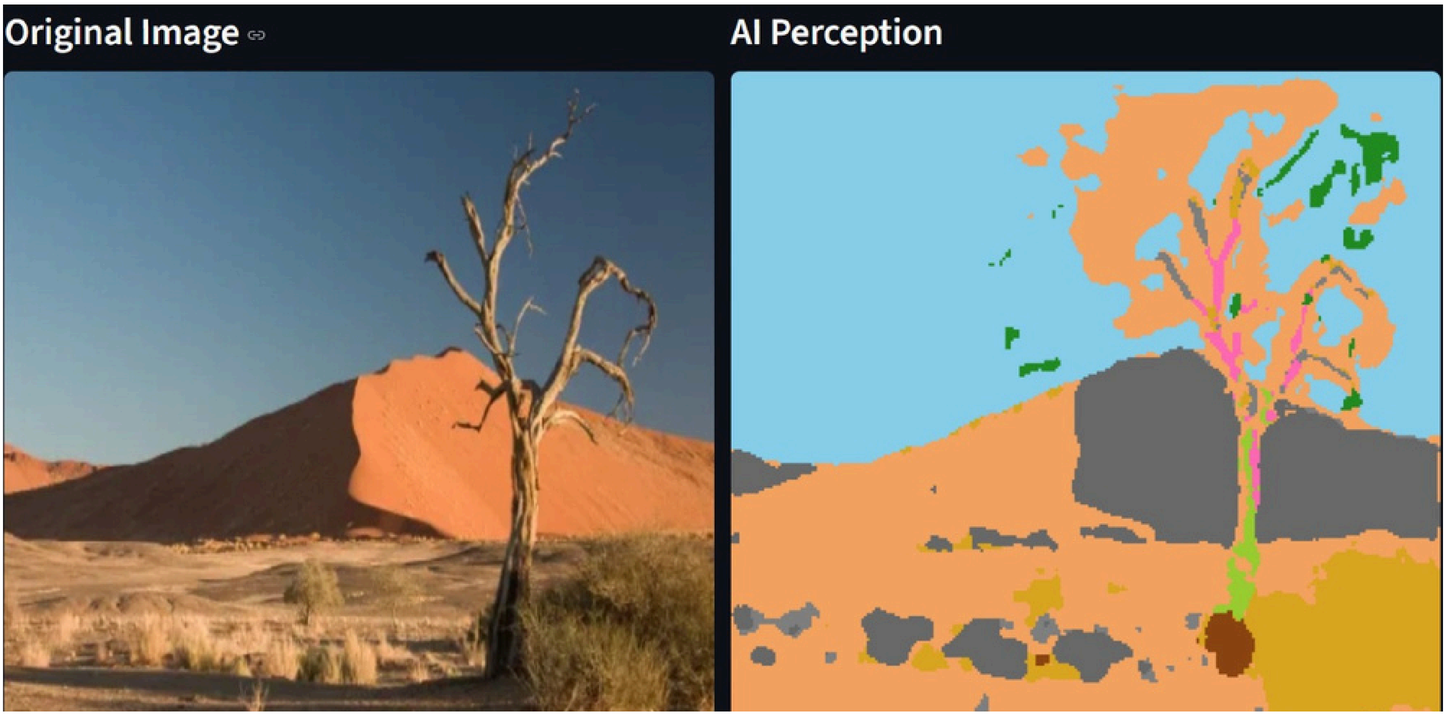
5.01%
Logs IoU — BASELINE
Obstacle Invisible
BEFORE

→ HYBRID LOSS FIX

56.21%
Logs IoU — FINAL
Obstacle Detected
AFTER

Why CE Loss Alone Failed

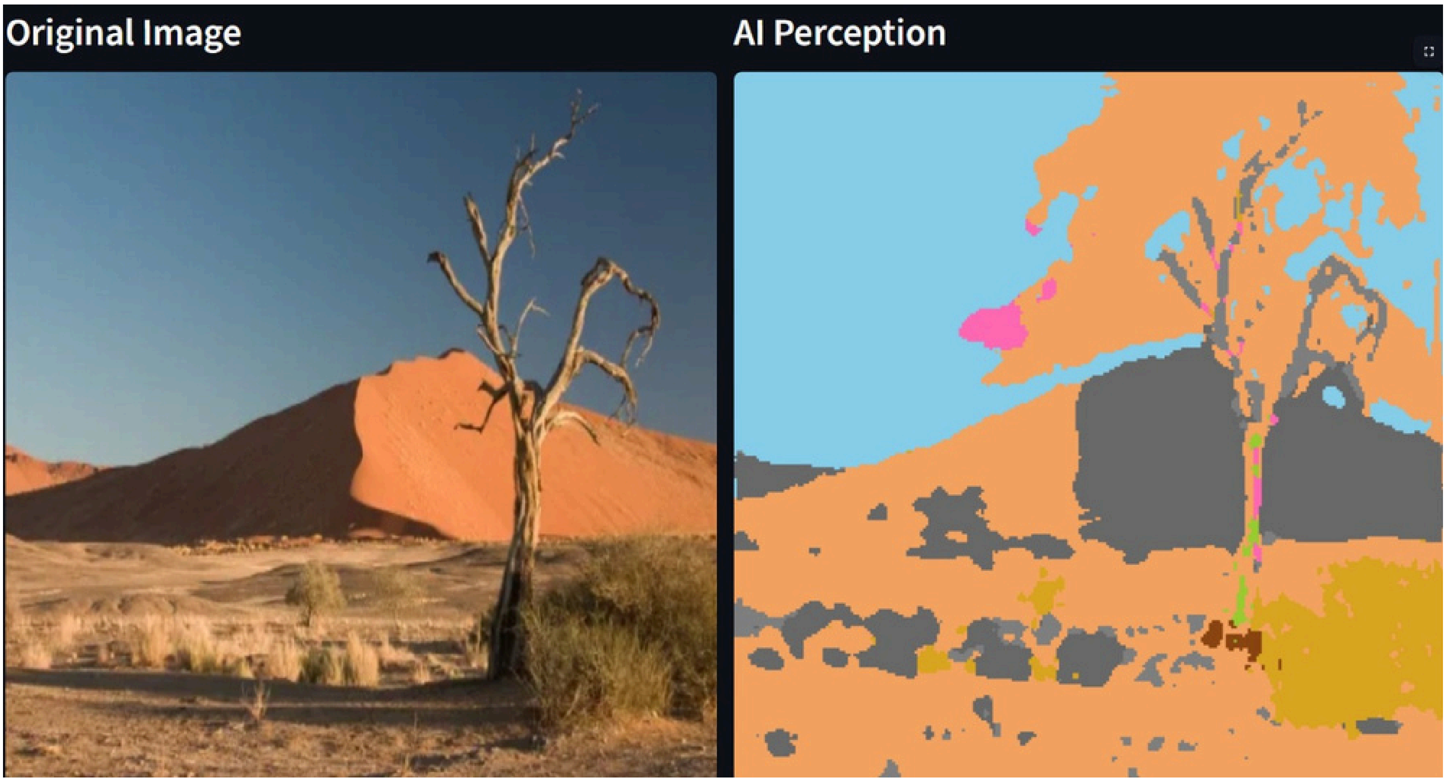
❌ Rewards purely by pixel count
❌ 40,000 Sky pixels vs 200 Log pixels
❌ Model exploits: ignore Logs = lose nothing
❌ Result: 5.01% IoU — Logs invisible
❌ Dangerous for real-world deployment



Our Fix : Hybrid Loss

Loss = CE_Loss + Dice_Loss

✅ Dice Loss measures overlap as RATIO
✅ 50-pixel Log = same weight as 50,000-pixel Sky
✅ Model forced to detect small objects
✅ Combined with geometric augmentation
✅ Model now actively hunts rare classes



Model transformed from obstacle-BLIND to obstacle-AWARE.

CHALLENGE 2

THE BROKEN PIPELINE PROBLEM

Non-Standard Mask Encoding — Silent Sabotage

0.00%
Pixel Accuracy — Epoch 1
The model was learning absolutely nothing. Because every mask was corrupted before it reached the model.

What Was Happening

- Dataset masks used pixel values 100, 200, 10000 instead of class indices 0-9
- cv2.imread() read values as brightness — completely corrupting all mask data
- PyTorch received out-of-bounds indices — crashed with tensor errors every epoch
- When it didn't crash, model trained on pure noise — learned nothing real

How We Fixed It

- Read masks raw: cv2.imread(path, -1) in unchanged format
- Built custom map_mask() function for safe value remapping
- Remap arbitrary values → standard 0-9 class indices automatically
- Added fallback default class for JPEG compression artifacts at borders

```
def map_mask(mask_path):
    mask = cv2.imread(mask_path, -1)
    mapped = np.full_like(mask, default_class)
    for raw_val, class_idx in CLASS_MAP.items():
        mapped[mask == raw_val] = class_idx
    return mapped
```

✅ Zero crashes — stable from epoch 1
✅ All 10 classes loading correctly
✅ Learning began immediately

FAILURE CASE ANALYSIS

WHERE OUR MODEL STILL STRUGGLES

🔍 Understanding failure modes is not a weakness. It is the foundation of responsible AI and the mark of a team that takes deployment safety seriously. No production model ships without this documentation.

FAILURE TYPE 01: Shadows vs. Rocks

WHAT HAPPENED
Dark shadows cast by Lush Bushes were misclassified as Rocks.

WHY
ResNet34 relies on texture/color. Desert shadows appear dark and jagged — identical to rock surfaces in RGB space.

REAL-WORLD RISK
Vehicle brakes for non-existent obstacle.

PROPOSED FIX
LiDAR depth fusion — shadows have zero depth return.

FAILURE TYPE 02: Dry Grass vs. Dry Bushes

WHAT HAPPENED
Boundary bleeding between adjacent texture-similar classes.

WHY
Nearly identical color histograms. Insufficient boundary training examples.

REAL-WORLD RISK
Imprecise obstacle borders in transition zones.

PROPOSED FIX
Targeted boundary augmentation + larger dataset.

CONCLUSION & FUTURE WORK

MISSION ACCOMPLISHED

✅ Full segmentation pipeline — scratch to prediction
✅ Diagnosed and solved severe class imbalance
✅ Engineered Hybrid Dice + CE Loss function
✅ Logs IoU: 5.01% → [Final%] — danger class detected
✅ Mean IoU: 49.54% → [Final%] — all classes improved
✅ Stable, crash-free, reproducible pipeline
✅ Complete failure mode documentation

Offroad Segmentation Evaluation Report

Overall Metrics

Metric	Value
Pixel Accuracy	88.26%
Mean IoU	67.60%
Mean Confidence	90.43%

Class-Specific Metrics

Class	IoU	Accuracy	Precision	Recall	F1 Score
Trees	88.05%	94.84%	92.49%	94.84%	93.65%
Lush Bushes	71.28%	86.73%	80.00%	86.73%	83.23%
Dry Grass	70.93%	84.91%	81.16%	84.91%	82.99%
Dry Bushes	51.02%	74.29%	61.96%	74.29%	67.56%
Ground Clutter	41.65%	54.20%	64.28%	54.20%	58.81%
Flowers	68.11%	83.28%	78.90%	83.28%	81.03%
Logs	62.18%	73.03%	80.73%	73.03%	76.68%
Rocks	53.36%	63.19%	77.43%	63.19%	69.59%
Landscape	70.63%	81.26%	84.37%	81.26%	82.79%
Sky	98.74%	99.25%	99.48%	99.25%	99.36%

FINAL OUTCOME

Optimal Model Performance on Data

FUTURE WORK

Class Weights

⚖️ Apply inverse-frequency multipliers to rarest classes in CE Loss. Additional boost for rare class recall.

Advanced Augmentation

🎨 Color Jitter + Gaussian Blur to simulate dawn, dusk, and desert dust storm conditions.

Sensor Fusion

📡 Integrate LiDAR depth channels. Permanently solves Shadow vs Rock — shadows return zero depth.

Larger Backbone

🧠 Upgrade ResNet34 → EfficientNet-B4 for richer features on complex texture boundaries.

"This hackathon proved a fundamental truth in computer vision: Your model is only as good as your loss function. Align your math with what matters in the real world — and the model will follow."

Xcess Denied
U-Net + ResNet34 | Dice + CE Loss | PyTorch + CUD