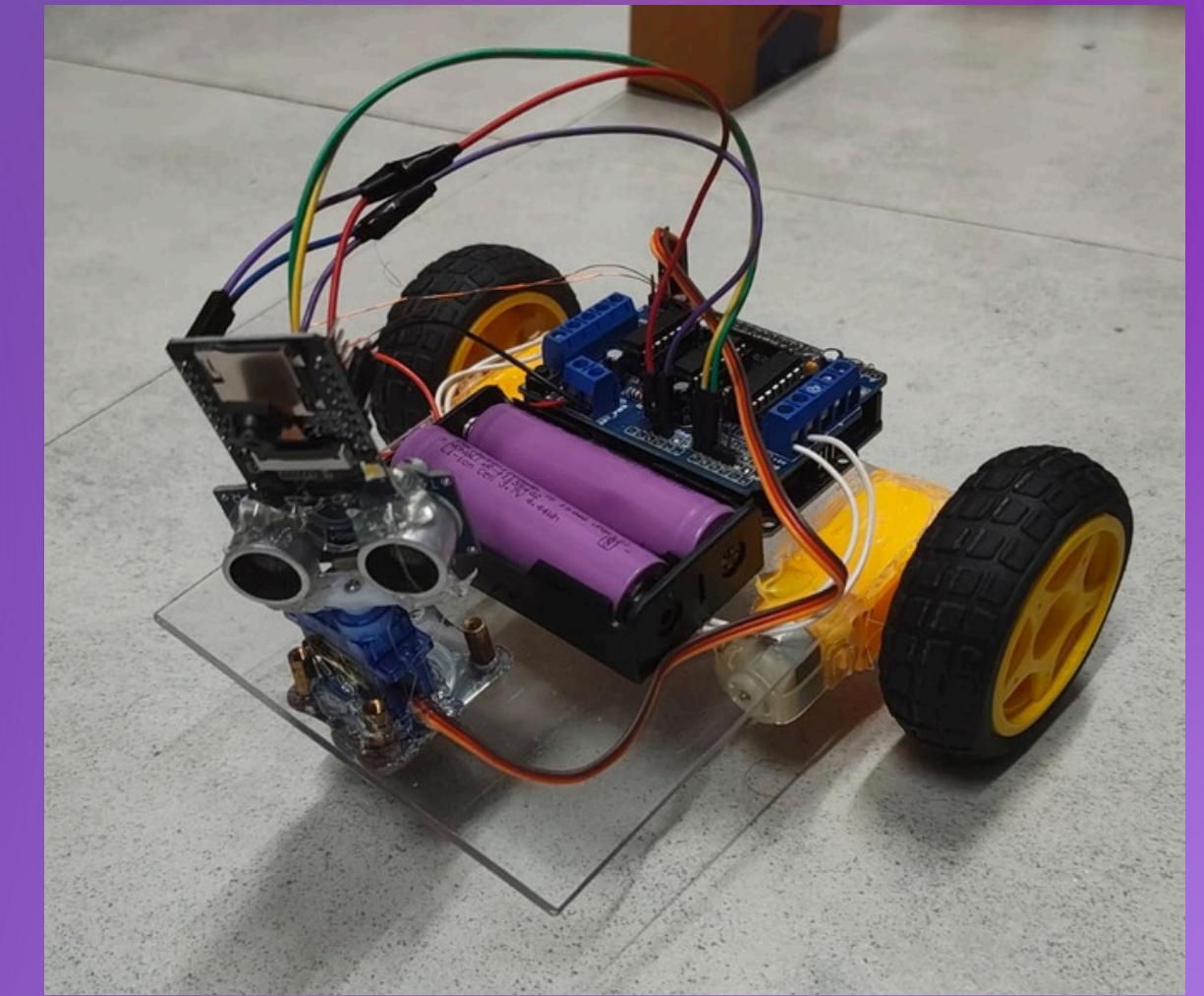


# STARTATHON 2026

- MODEL: AI-BASED AUTONOMOUS ROVER USING SEMATIC SCENE SEGMENTATION
- TEAM NAME: METHXAI
- TEAM MEMBERS: ANIKET KUMAR  
ROHAN JARIA  
AARAV CHOUKSEY  
PARTH PANCHOLI
- ORGANIZED BY: HACK WITH INDIA & E-CELL VIT  
BHOPAL UNIVERSITY



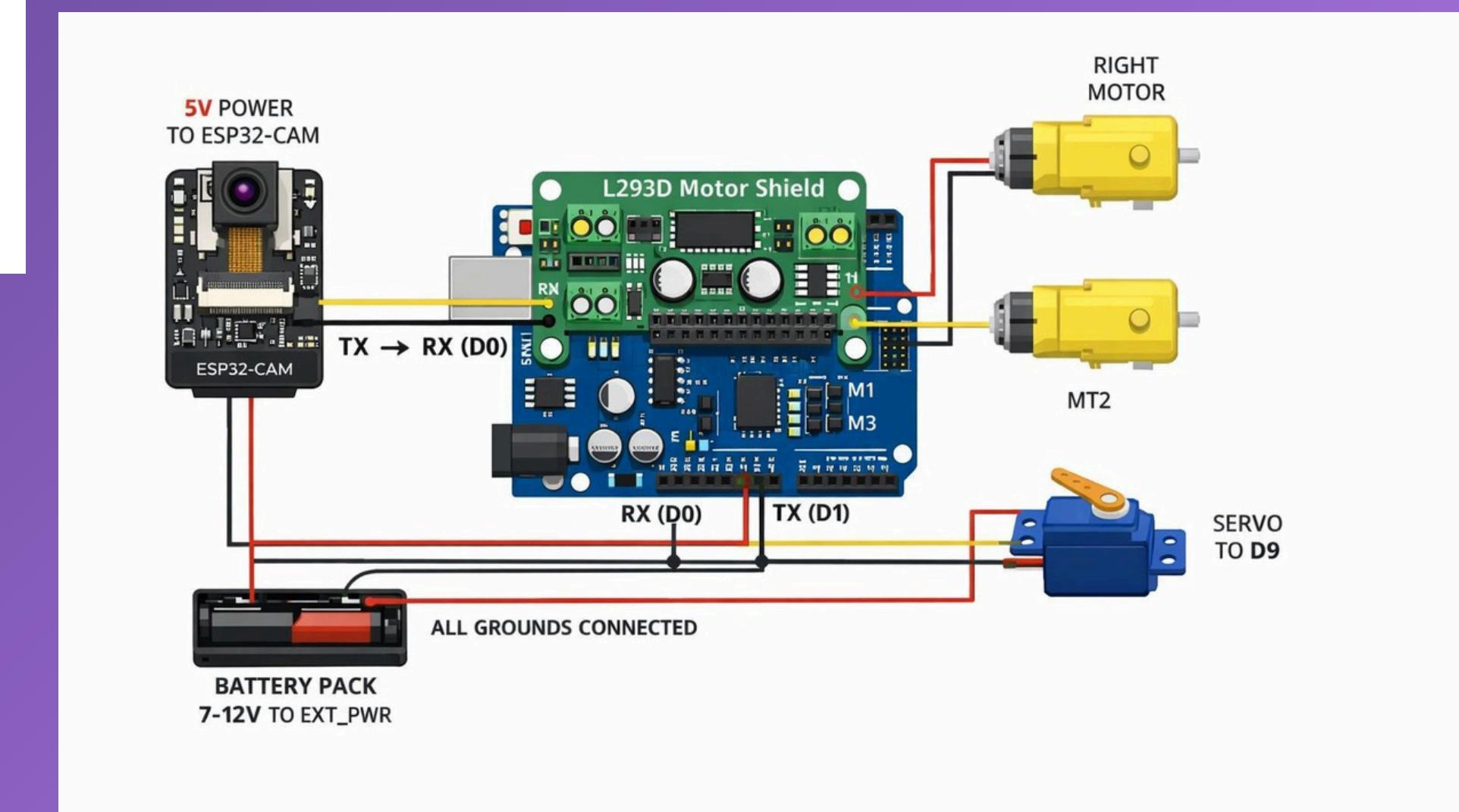
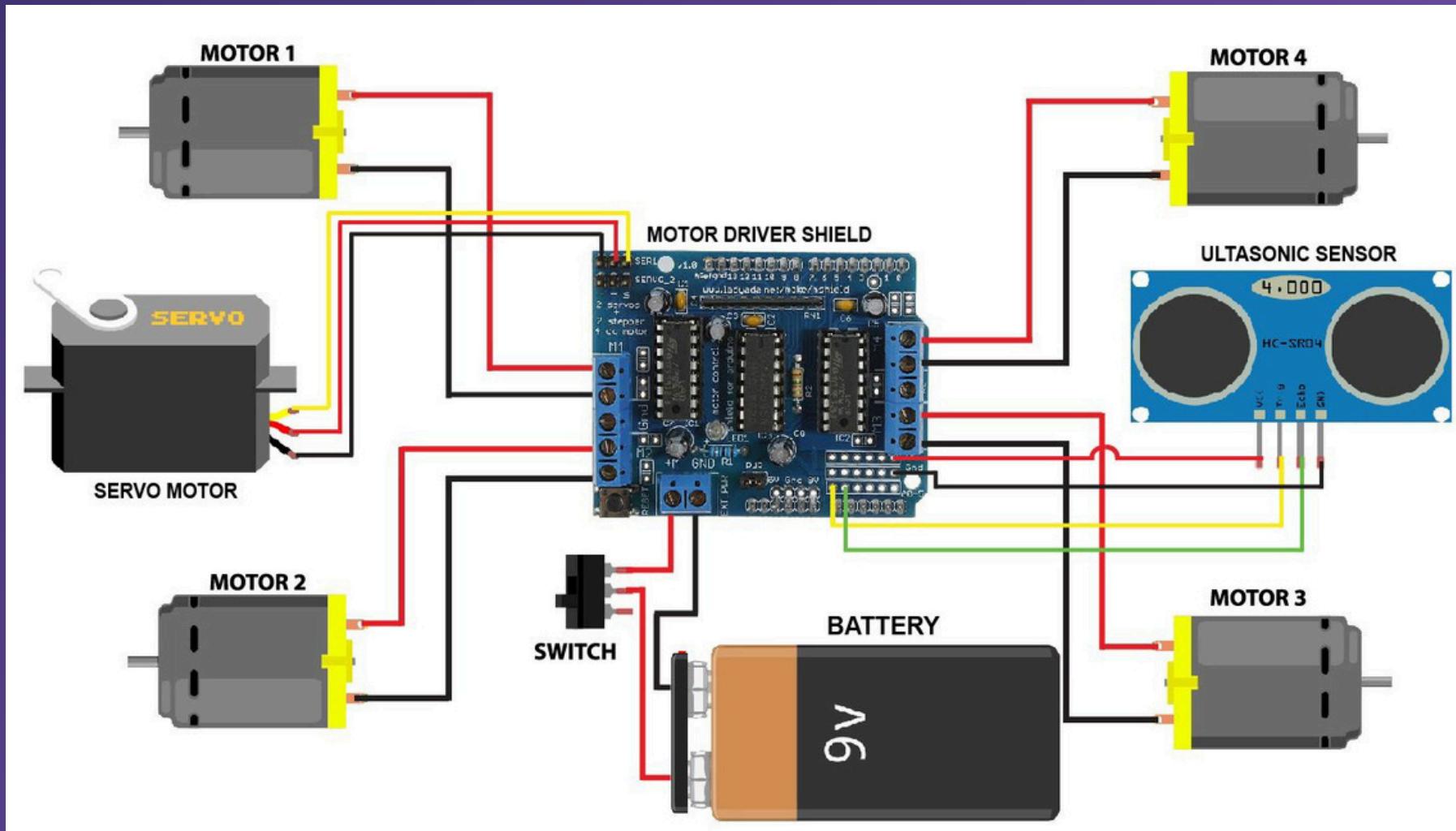
# PROBLEM STATEMENT

- Autonomous systems must navigate diverse and unpredictable environments such as urban areas, indoor spaces, and natural terrains. Variations in obstacles, lighting, and surface types make real-time navigation challenging and require continuous scene understanding.
- Training these systems – especially for pixel-level tasks like semantic segmentation – demands large amounts of accurately labeled data, which is expensive and time-consuming to collect and annotate.
- To ensure safe and adaptive navigation, autonomous platforms need robust scene understanding that identifies obstacles, surface types, and navigable regions beyond simple object detection.

# MODEL OVERVIEW

- Our model is a deep learning-based semantic segmentation system that processes live camera input and classifies each pixel as obstacle or navigable area for accurate scene understanding.
- Trained on synthetic digital twin data, it performs real-time inference and converts segmentation results into steering and movement commands for autonomous navigation.

# CIRCUIT



# WORKING

- The camera continuously captures real-time images of the surrounding environment.
- Each captured frame is sent to the trained semantic segmentation model for pixel-level scene analysis.
- The AI model classifies every pixel into categories such as obstacles, terrain, and safe navigation areas.
- The segmentation output is converted into a terrain map highlighting obstacles and drivable regions.
- A decision-making algorithm analyzes the terrain map to determine the safest movement direction.
- Control commands such as forward motion or turning are generated based on obstacle position.
- These commands are transmitted to the Arduino controller.
- The Arduino drives the motors and servo to move the rover accordingly.
- This process runs continuously, enabling real-time autonomous navigation.

# KEY FEATURES

## 1. AI-Based Semantic Segmentation

- Uses deep learning to perform pixel-level scene understanding instead of simple object detection.

## 2. Terrain-Agnostic Navigation

- Designed to work across multiple environments (indoor, urban, natural terrain), not limited to one specific setting.

## 3. Real-Time Obstacle Detection

- Processes live camera input and identifies obstacles instantly for safe navigation.

---

EVALUATION RESULTS [TTA=ON]

Mean IoU: 0.5023

Pixel Accuracy: 0.8127

---

Background	: 0.1211
Trees	: 0.6876
Lush Bushes	: 0.6598
Dry Grass	: 0.6014
Dry Bushes	: 0.6337
Ground Clutter	: 0.1871
Logs	: 0.1319
Rocks	: 0.2452
Landscapes	: 0.7891
Sky	: 0.9667

Metrics saved to ./predictions/  
All output saved to ./predictions/

## 4. Intelligent Decision-Making

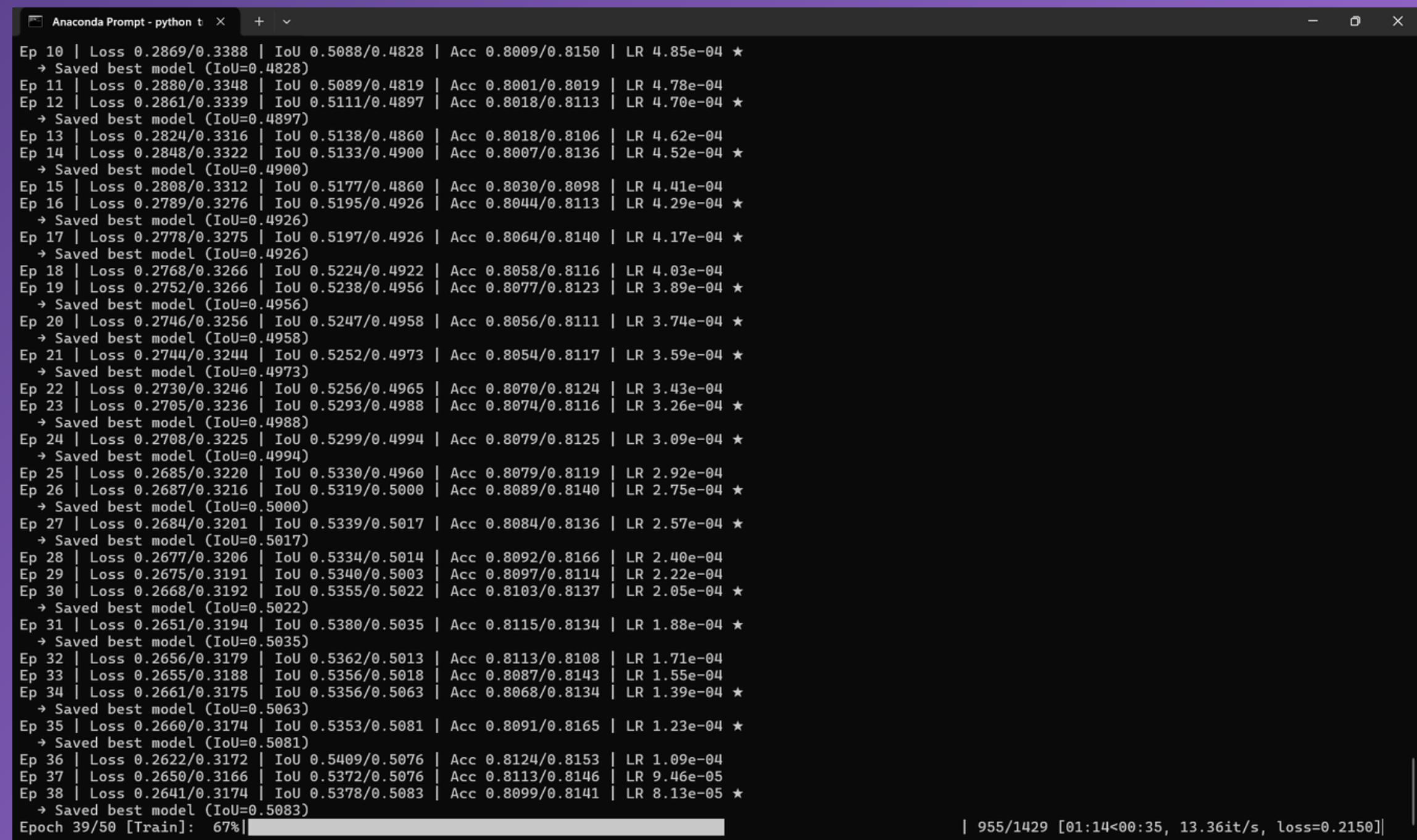
- Converts segmentation output into steering and movement commands automatically.

## 5. Hardware Integration

- Deploys AI model onto an Arduino-based rover for practical, real-world demonstration.

## 6. Scalable & Extensible Design

- System can be expanded to include multi-axis stabilization, advanced path planning, or domain adaptation.



```
Anaconda Prompt - python t  x + v
Ep 10 | Loss 0.2869/0.3388 | IoU 0.5088/0.4828 | Acc 0.8009/0.8150 | LR 4.85e-04 ★
→ Saved best model (IoU=0.4828)
Ep 11 | Loss 0.2880/0.3348 | IoU 0.5089/0.4819 | Acc 0.8001/0.8019 | LR 4.78e-04
Ep 12 | Loss 0.2861/0.3339 | IoU 0.5111/0.4897 | Acc 0.8018/0.8113 | LR 4.70e-04 ★
→ Saved best model (IoU=0.4897)
Ep 13 | Loss 0.2824/0.3316 | IoU 0.5138/0.4860 | Acc 0.8018/0.8106 | LR 4.62e-04
Ep 14 | Loss 0.2848/0.3322 | IoU 0.5133/0.4900 | Acc 0.8007/0.8136 | LR 4.52e-04 ★
→ Saved best model (IoU=0.4900)
Ep 15 | Loss 0.2808/0.3312 | IoU 0.5177/0.4860 | Acc 0.8030/0.8098 | LR 4.41e-04
Ep 16 | Loss 0.2789/0.3276 | IoU 0.5195/0.4926 | Acc 0.8044/0.8113 | LR 4.29e-04 ★
→ Saved best model (IoU=0.4926)
Ep 17 | Loss 0.2778/0.3275 | IoU 0.5197/0.4926 | Acc 0.8064/0.8140 | LR 4.17e-04 ★
→ Saved best model (IoU=0.4926)
Ep 18 | Loss 0.2768/0.3266 | IoU 0.5224/0.4922 | Acc 0.8058/0.8116 | LR 4.03e-04
Ep 19 | Loss 0.2752/0.3266 | IoU 0.5238/0.4956 | Acc 0.8077/0.8123 | LR 3.89e-04 ★
→ Saved best model (IoU=0.4956)
Ep 20 | Loss 0.2746/0.3256 | IoU 0.5247/0.4958 | Acc 0.8056/0.8111 | LR 3.74e-04 ★
→ Saved best model (IoU=0.4958)
Ep 21 | Loss 0.2744/0.3244 | IoU 0.5252/0.4973 | Acc 0.8054/0.8117 | LR 3.59e-04 ★
→ Saved best model (IoU=0.4973)
Ep 22 | Loss 0.2730/0.3246 | IoU 0.5256/0.4965 | Acc 0.8070/0.8124 | LR 3.43e-04
Ep 23 | Loss 0.2705/0.3236 | IoU 0.5293/0.4988 | Acc 0.8074/0.8116 | LR 3.26e-04 ★
→ Saved best model (IoU=0.4988)
Ep 24 | Loss 0.2708/0.3225 | IoU 0.5299/0.4994 | Acc 0.8079/0.8125 | LR 3.09e-04 ★
→ Saved best model (IoU=0.4994)
Ep 25 | Loss 0.2685/0.3220 | IoU 0.5330/0.4960 | Acc 0.8079/0.8119 | LR 2.92e-04
Ep 26 | Loss 0.2687/0.3216 | IoU 0.5319/0.5000 | Acc 0.8089/0.8140 | LR 2.75e-04 ★
→ Saved best model (IoU=0.5000)
Ep 27 | Loss 0.2684/0.3201 | IoU 0.5339/0.5017 | Acc 0.8084/0.8136 | LR 2.57e-04 ★
→ Saved best model (IoU=0.5017)
Ep 28 | Loss 0.2677/0.3206 | IoU 0.5334/0.5014 | Acc 0.8092/0.8166 | LR 2.40e-04
Ep 29 | Loss 0.2675/0.3191 | IoU 0.5340/0.5003 | Acc 0.8097/0.8114 | LR 2.22e-04
Ep 30 | Loss 0.2668/0.3192 | IoU 0.5355/0.5022 | Acc 0.8103/0.8137 | LR 2.05e-04 ★
→ Saved best model (IoU=0.5022)
Ep 31 | Loss 0.2651/0.3194 | IoU 0.5380/0.5035 | Acc 0.8115/0.8134 | LR 1.88e-04 ★
→ Saved best model (IoU=0.5035)
Ep 32 | Loss 0.2656/0.3179 | IoU 0.5362/0.5013 | Acc 0.8113/0.8108 | LR 1.71e-04
Ep 33 | Loss 0.2655/0.3188 | IoU 0.5356/0.5018 | Acc 0.8087/0.8143 | LR 1.55e-04
Ep 34 | Loss 0.2661/0.3175 | IoU 0.5356/0.5063 | Acc 0.8068/0.8134 | LR 1.39e-04 ★
→ Saved best model (IoU=0.5063)
Ep 35 | Loss 0.2660/0.3174 | IoU 0.5353/0.5081 | Acc 0.8091/0.8165 | LR 1.23e-04 ★
→ Saved best model (IoU=0.5081)
Ep 36 | Loss 0.2622/0.3172 | IoU 0.5409/0.5076 | Acc 0.8124/0.8153 | LR 1.09e-04
Ep 37 | Loss 0.2650/0.3166 | IoU 0.5372/0.5076 | Acc 0.8113/0.8146 | LR 9.46e-05
Ep 38 | Loss 0.2641/0.3174 | IoU 0.5378/0.5083 | Acc 0.8099/0.8141 | LR 8.13e-05 ★
→ Saved best model (IoU=0.5083)
Epoch 39/50 [Train]: 67%
```

# SYSTEM ARCHITECTURE

## Software / AI

- Semantic segmentation deep learning model
- Synthetic dataset for training and testing
- Data augmentation and model optimization
- Real-time inference pipeline

## Perception

- Camera module for live environment capture

## Hardware

- Arduino-based robotic car chassis
- Servo motors and DC motors for movement control
- Motor driver module

# POSSIBLE IMPACT

- **Enhanced Safety in Autonomous Navigation**

Real-time semantic segmentation enables accurate obstacle detection and terrain understanding, reducing collision risks and improving navigation reliability.

- **Cost-Effective Development**

Training with synthetic digital twin data minimizes the need for expensive real-world data collection and manual annotation.

- **Scalable Across Environments**

The system can adapt to urban roads, indoor spaces, and natural terrains, making it suitable for diverse real-world applications.

- **Improved Decision-Making**

Pixel-level scene understanding allows smarter movement decisions compared to basic object detection systems.

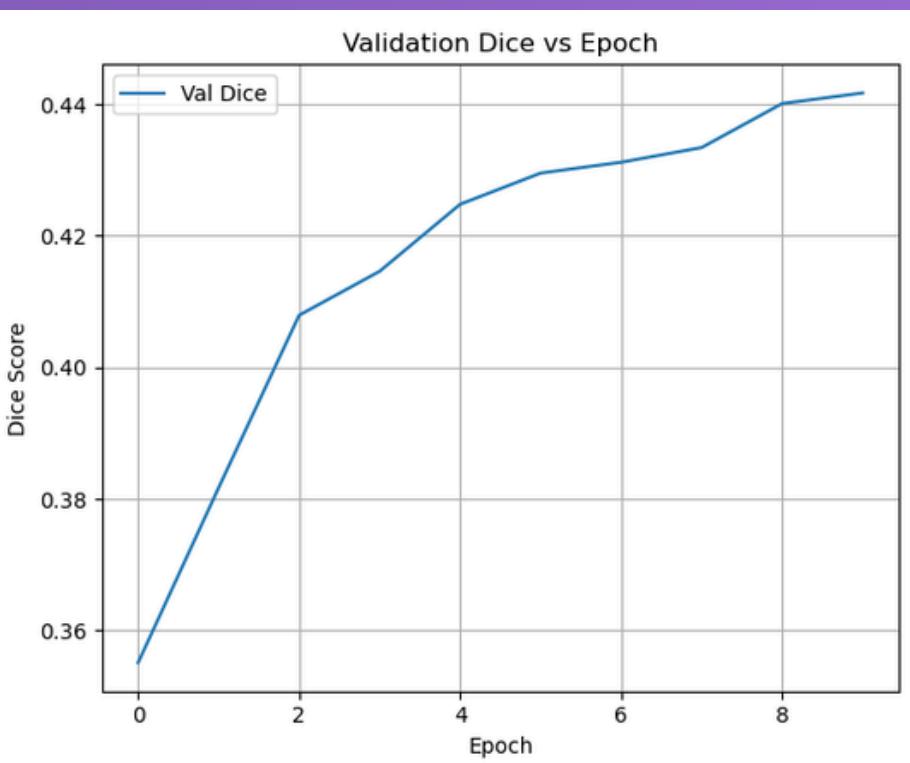
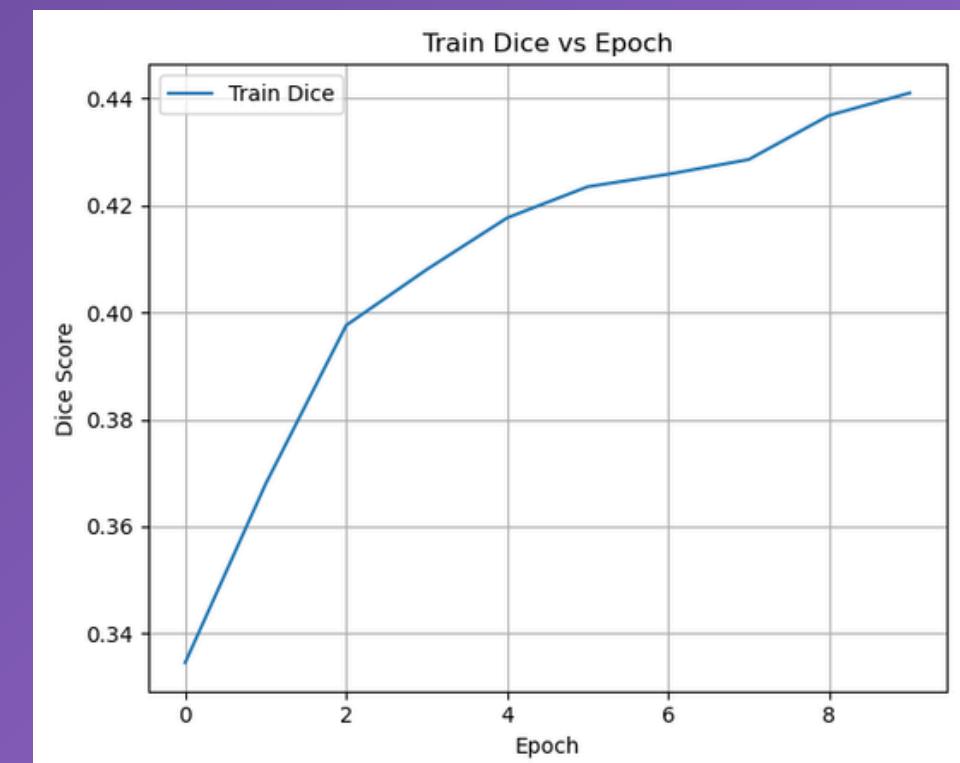
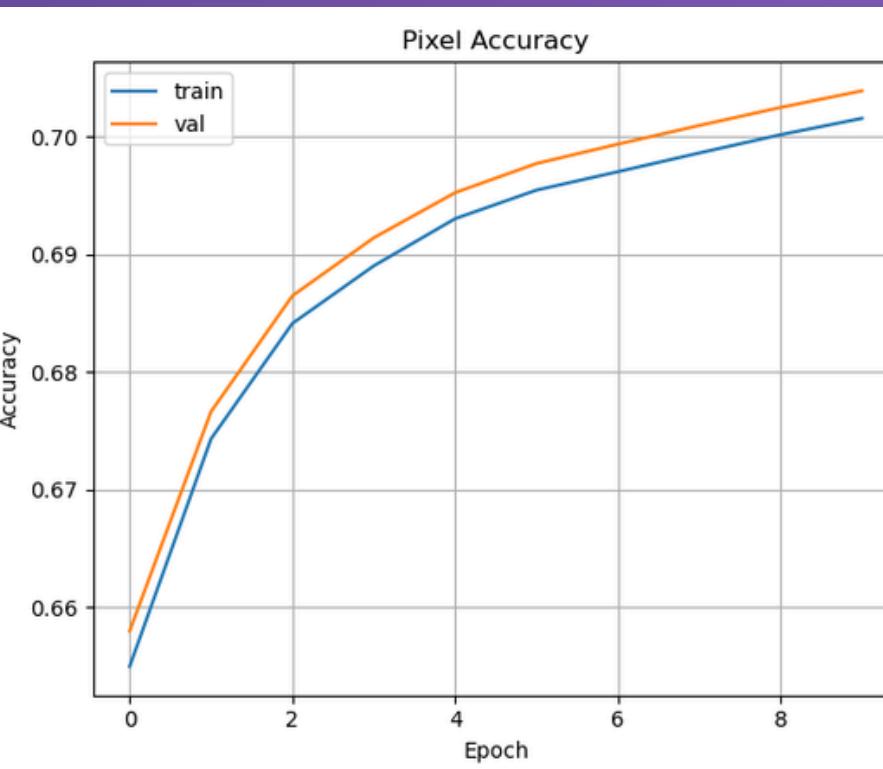
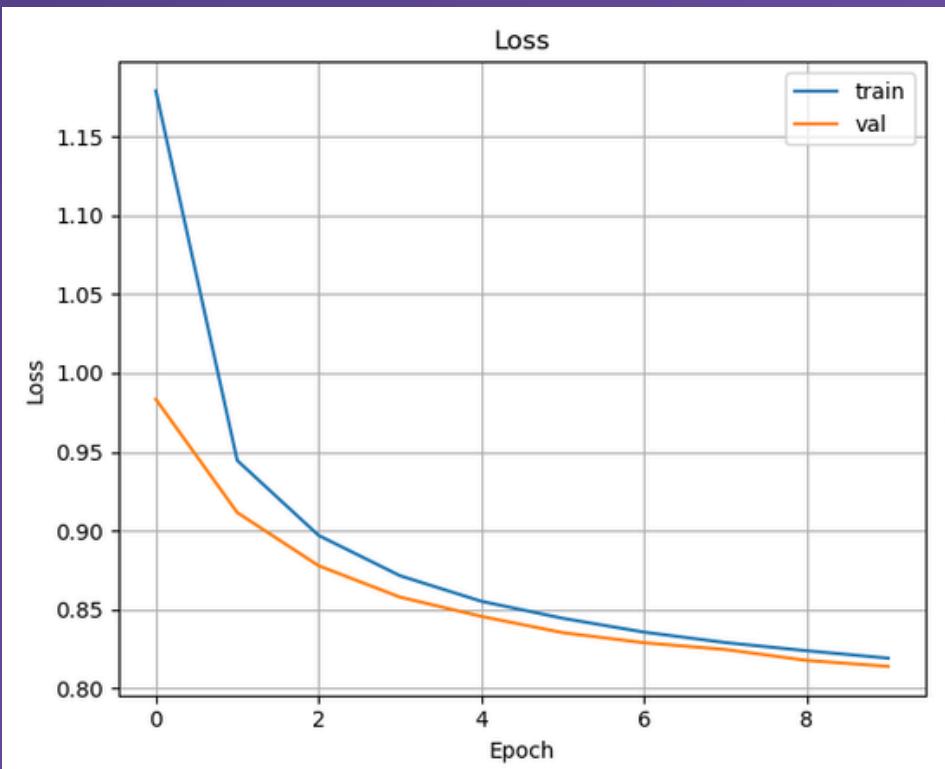
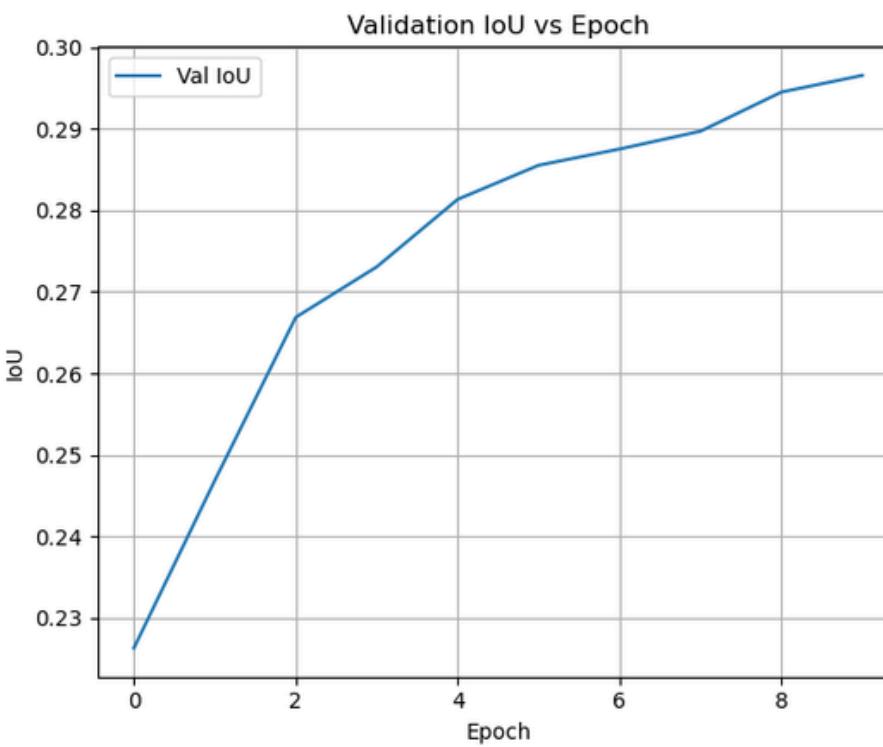
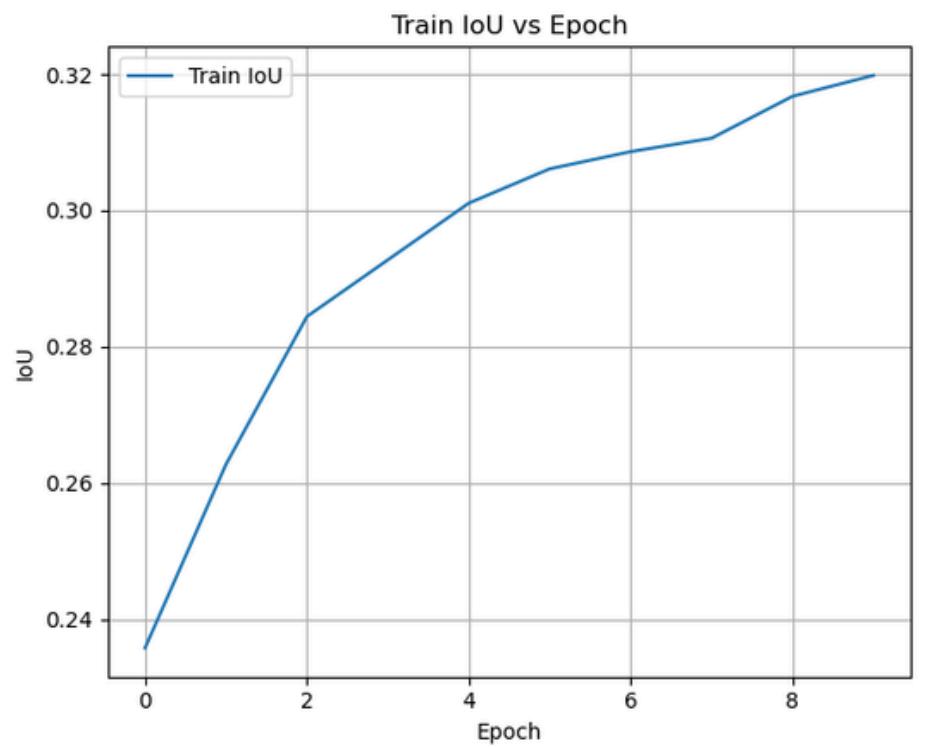
- **Foundation for Advanced Robotics**

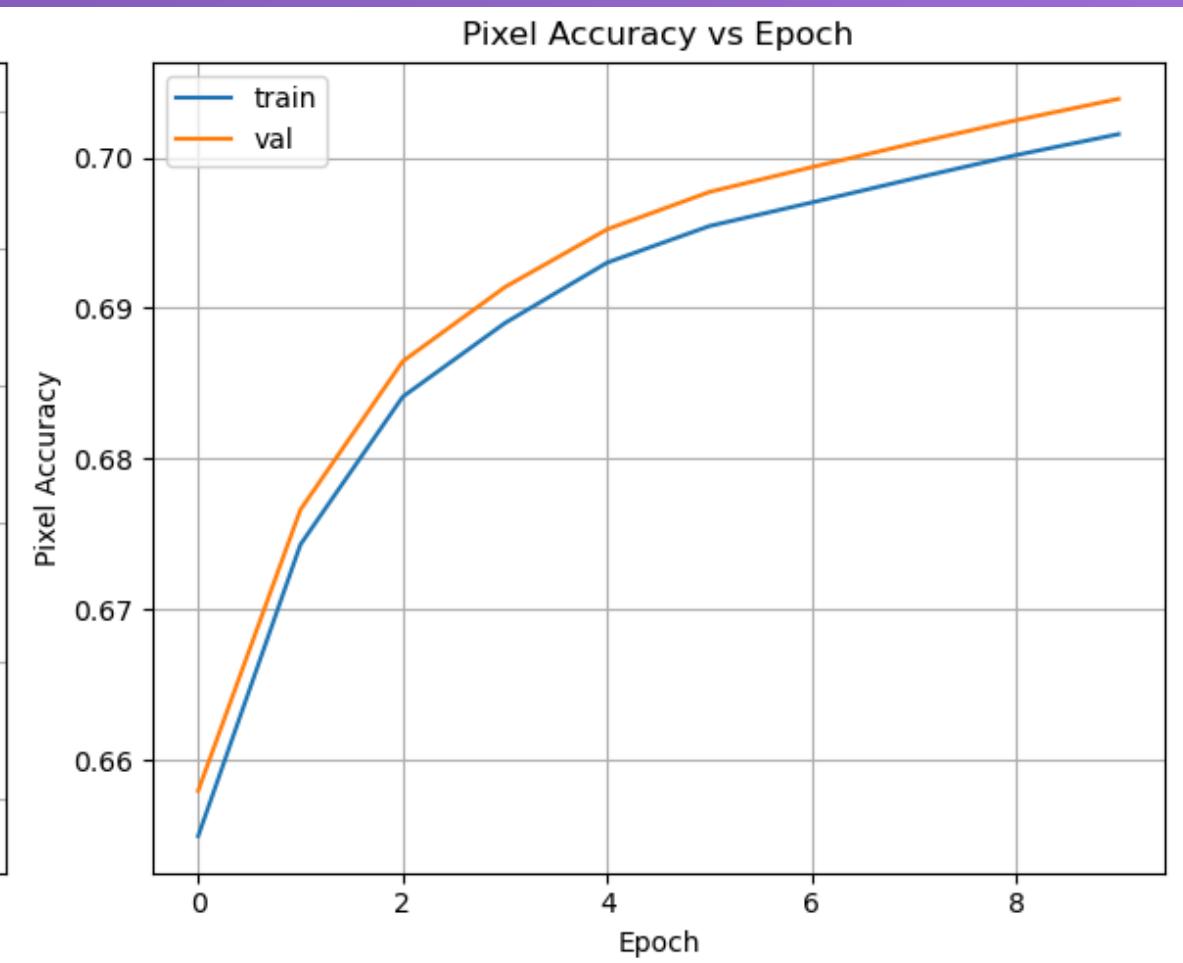
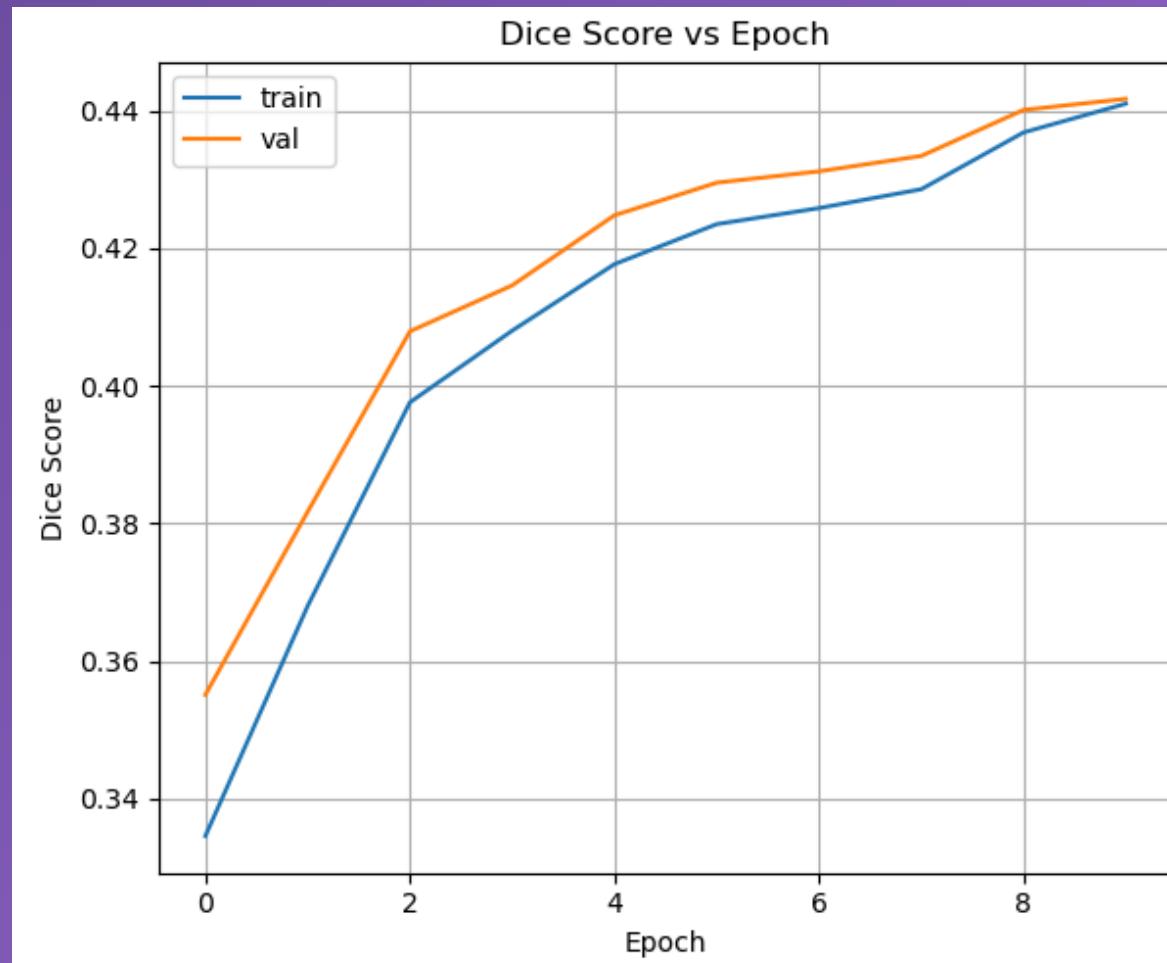
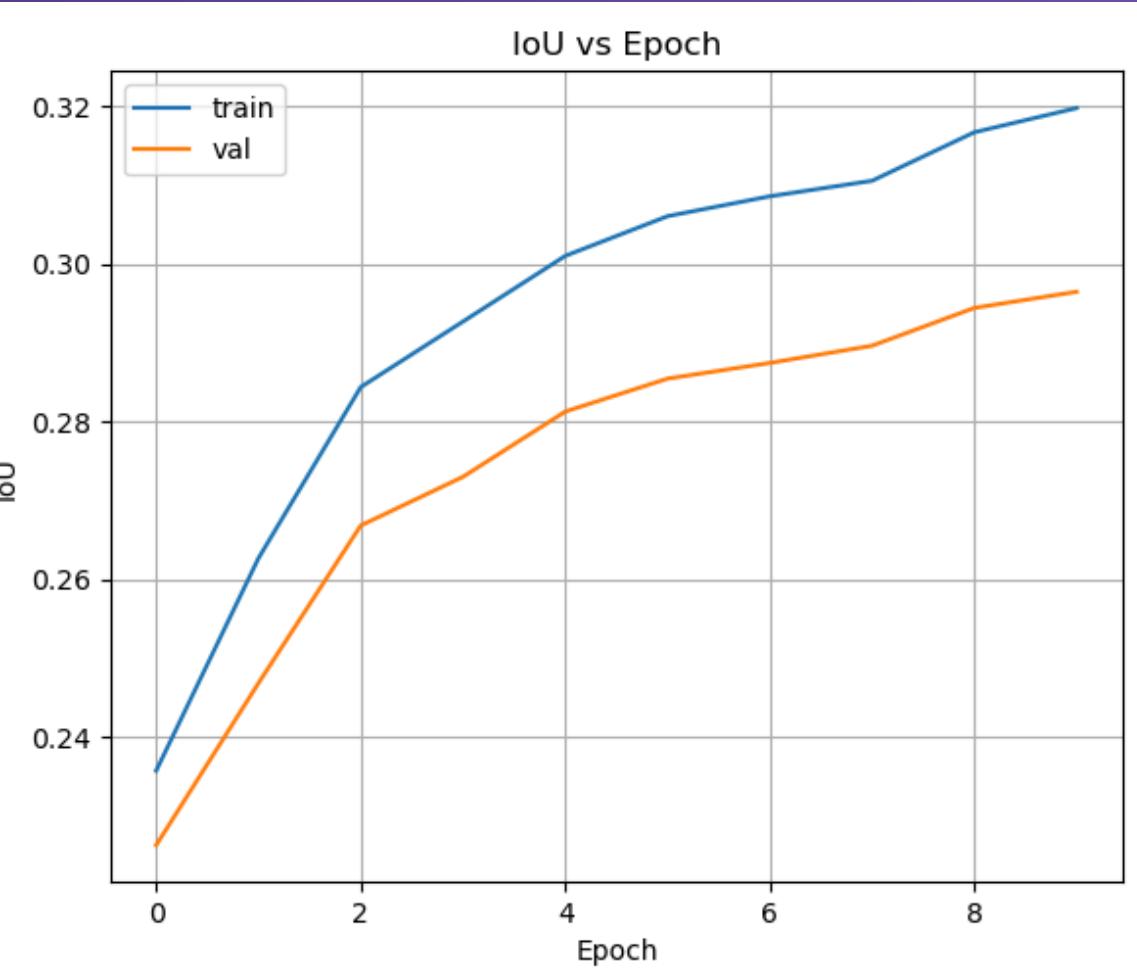
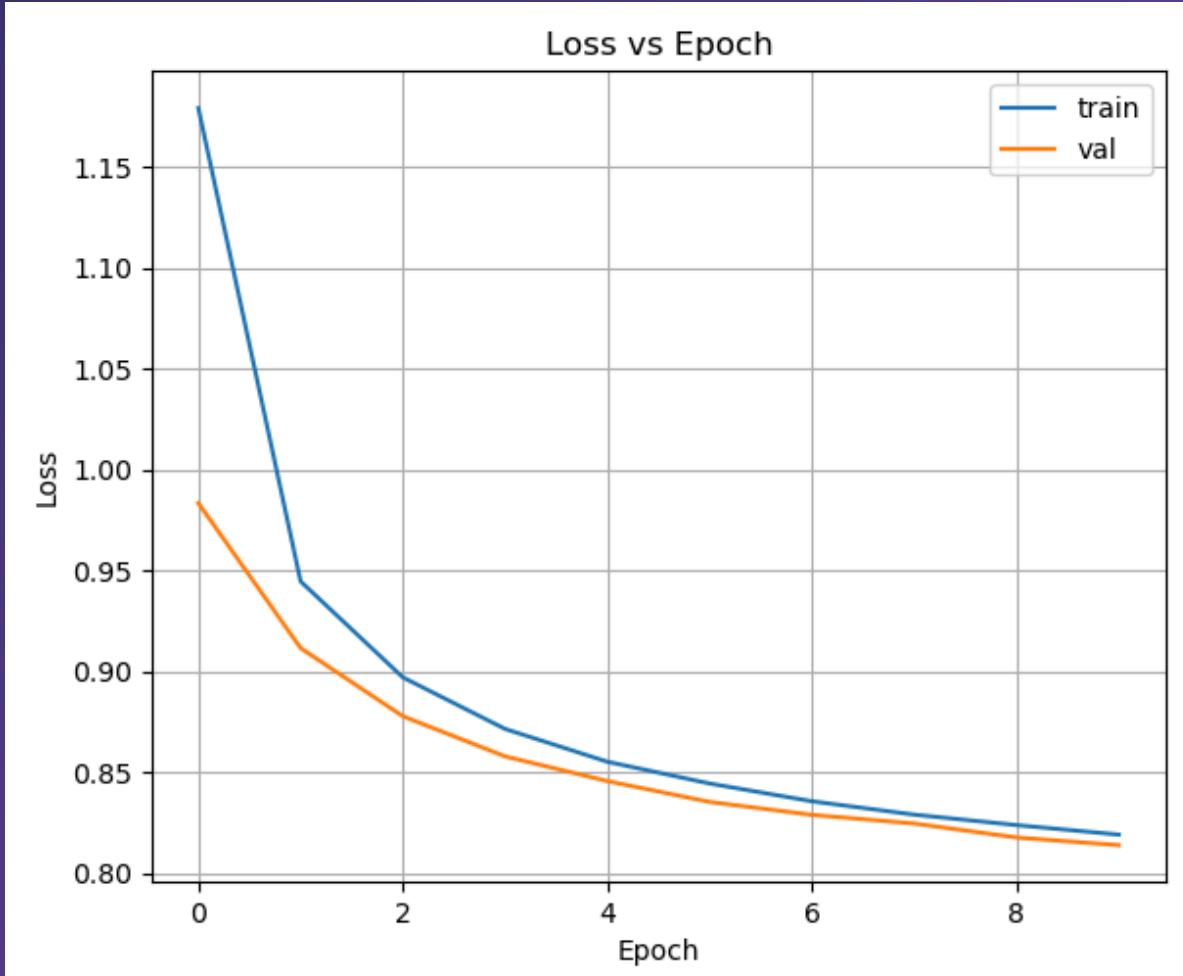
The project can be extended to applications such as delivery robots, agricultural monitoring, warehouse automation, surveillance rovers, and disaster-response systems.

- **Educational and Research Value**

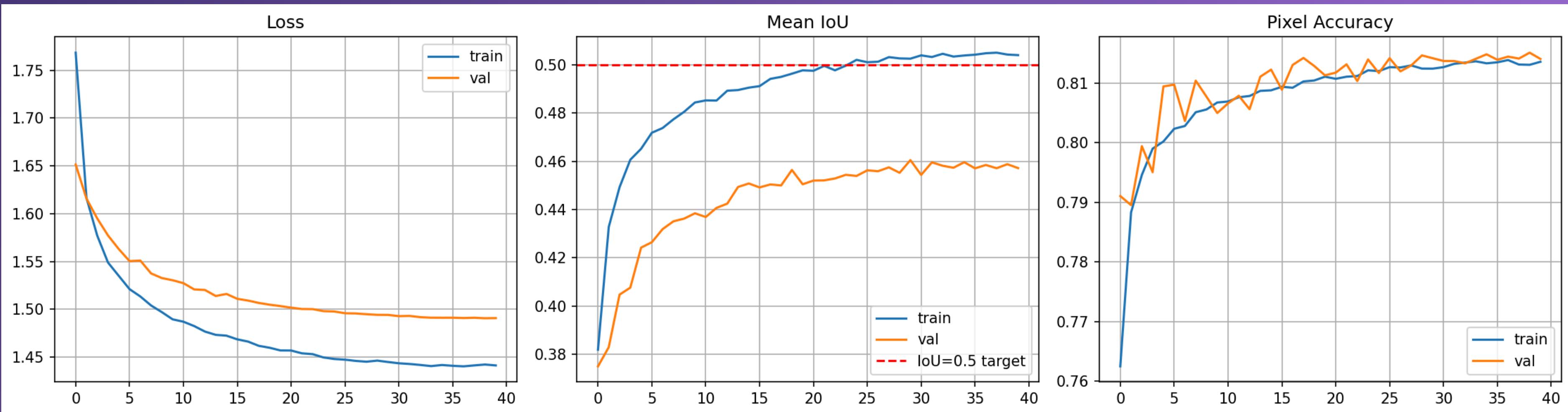
Provides a practical framework combining AI perception with embedded hardware (Arduino-based control), useful for robotics and autonomous systems research.

# INITIAL GRAPHS

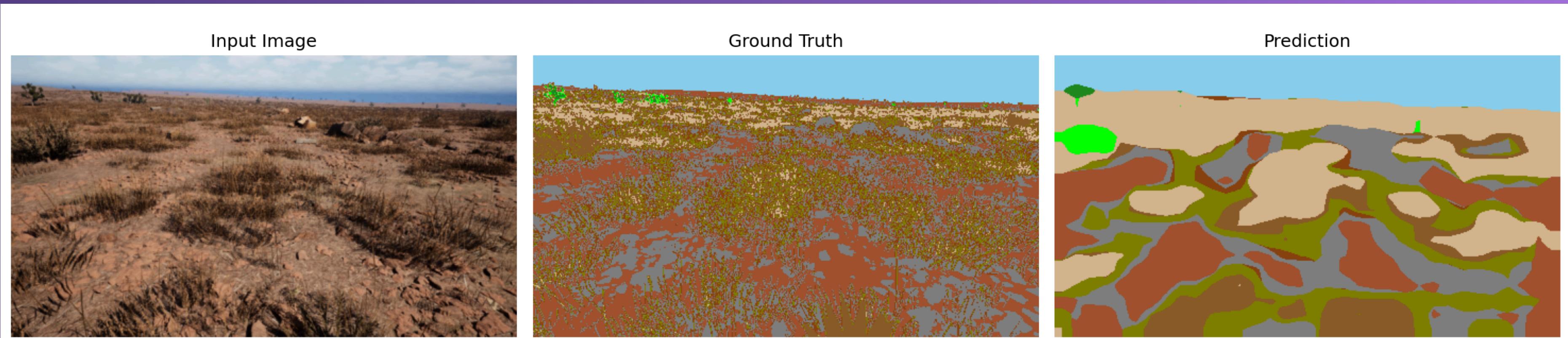
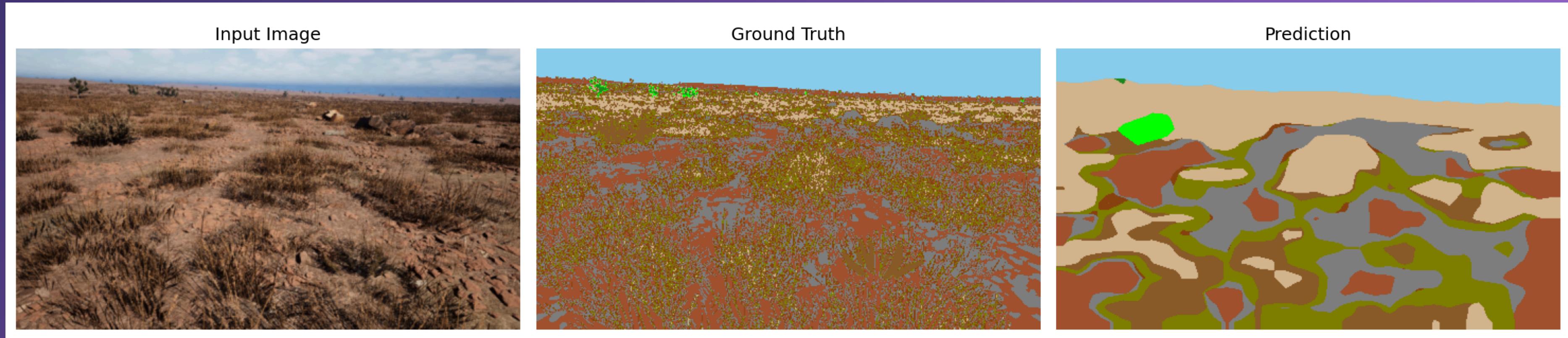




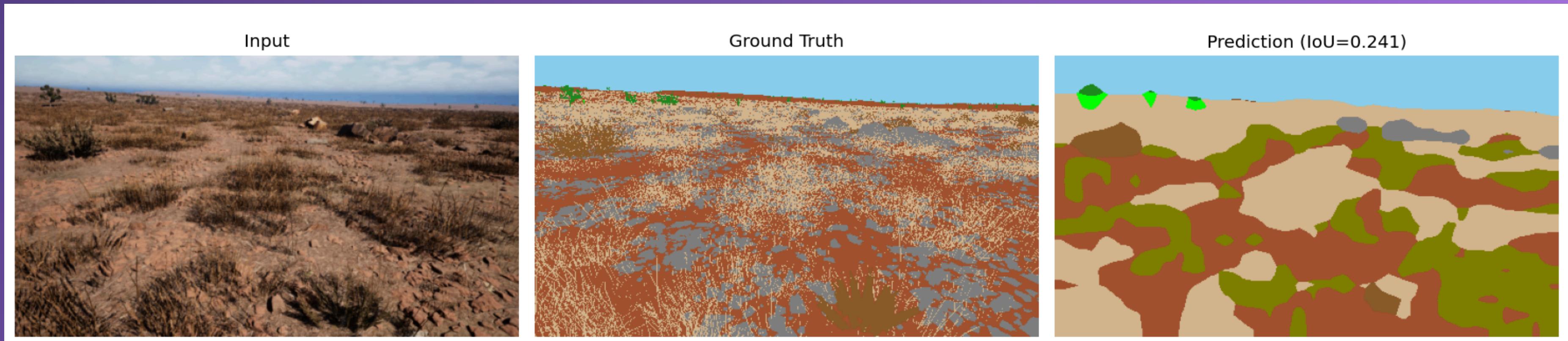
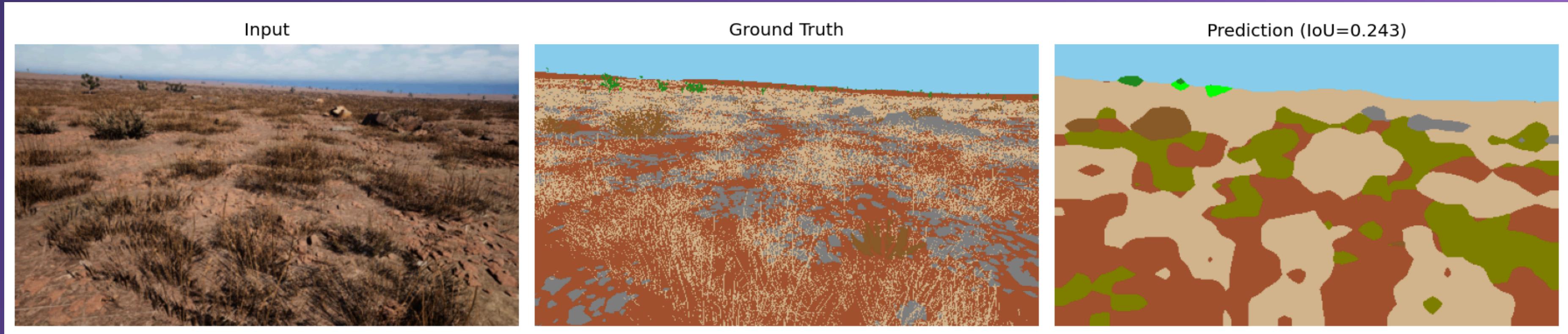
# FINAL GRAPHS



# INITIAL SEMANTIC SEGMENTATION



# FINAL SEMATIC SEGMENTATION



THANK  
YOU