Duality AI’s Offroad Semantic Scene Segmentation

Overview

Duality AI is excited to present the Offroad Autonomy Segmentation challenge, a challenge designed to explore cutting-edge AI training techniques. Participants will train a model using annotated images of a desert environment, then test that model on a novel, but still desert, environment. All the data is generated from Duality AI’s digital twin simulation platform, Falcon, using FalconCloud’s geospatial-based digital twin environments. Each team will work to achieve the most accurate and precise model by adjusting training parameters and processes. Along the way, participants will learn how Duality AI uses industry-proven techniques and tools and high-quality synthetic data to train AI models for context shifts, unseen environments, or difficult-to-access data, such as remote areas. Through this competition, participants will develop novel AI training skills, enhance their portfolio, build connections with Duality AI and other participants, and have the opportunity to win prizes and recognition!

Objectives

- Train a robust semantic segmentation model using the provided synthetic dataset to accurately segment an environment, which is crucial to off-road autonomy
- Evaluate model performance in novel (but similar) scenarios.
- Benchmark and optimize your model to improve accuracy, generalizability, and efficiency for real-world deployment scenarios.

Importance of Digital Twins for Segmentation and Off-road Autonomy

Unmanned Ground Vehicles \UGVs\ rely on robust computer vision models for effective path planning and decision-making in complex, dynamic environments. Among critical CV tasks, semantic segmentation plays a pivotal role in obstacle avoidance by providing fine-grained scene understanding. However, training high-performing segmentation models traditionally relies on supervised learning, which requires large, labeled datasets for effective training. Attaining this data in the real world, with necessary volume and variety for successful training, is generally costly and time-consuming, and often still proves insufficient.

Synthetic data presents a promising solution and has already been explored as a means to address data scarcity and improve model robustness in computer vision tasks. Creating the data costs much less and takes very little time compared to traditional data collection and annotation methods. Additionally, users can control variations and edge cases such as weather events, time of day, and specific environment characteristics, providing diverse data for robust training.

Hackathon Structure

Data Overview

- Participants will work with a dataset generated from FalconEditor of various desert environment twins.
- Classes:

ID	Class Name	Picture
100	Trees	
200	Lush Bushes	
300	Dry Grass	
500	Dry Bushes	
550	Ground Clutter	

600	Flowers	
700	Logs	
800	Rocks	
7100	Landscape	*all general ground that isn't another category*
10000	Sky	

Participants will process synthetic data, train an AI model to segment the data images, validate performance on unseen images from a separate desert environment, and optimize accuracy under realistic constraints.

i. Hackathon Tasks

To maximize efficiency, we recommend dividing responsibilities based on these roles. Proper delegation will help streamline the workflow and ensure high-quality results across the technical and presentation components.

1. **AI Engineering**
   - Train and fine-tune the model using the generated dataset.
   - Evaluate model performance.
   - Use optimization techniques to improve accuracy and reduce inference time.

2. **Documentation & Presenting Final Team Results**
   - Document the workflow, including:
     - Data augmentation/filtering strategies
     - model training
     - evaluation metrics such as loss graphs
   - Prepare a structured report and final presentation showcasing findings, results, and insights.
   - Create visualizations such as performance and loss graphs to highlight model behavior.

ii. Key Deliverables

At the end of the hackathon, teams must submit:

1. **A Trained Semantic Segmentation Model**
   a. Fully trained model that segments the testing images into categories.
   b. The package must include model weights, scripts, and config files.

2. **Performance Evaluation & Analysis Report,** including but is not limited to:
   a. **IoU Score** to evaluate model accuracy.
   b. **Loss graphs** to visualize performance.
   c. **Failure Case Analysis**, highlighting misclassifications and possible improvements.

iii. Judging Criteria

Teams will be evaluated on a 100-point scale, based on the following criteria:

a. Model Performance

IoU Score ……..……80 Points

- Measures the accuracy of pixel classification.

b. Performance Report Clarity

Structured Findings & Detailed Reporting……20 Points

- Well-organized documentation of the methodology, challenges, and solutions.
- Clearly outlined steps, including any dataset modifications, model training, and evaluation.

Teams must balance technical performance with clear and professional documentation to maximize their score.

iv. Important links

USE	LINK
Create a free Falcon account	https://falcon.duality.ai/auth/sign-up?utm_source=hackathon&utm_medium=instructions&utm_campaign=HackfestxDatathon
Download Dataset	
Discord Forum	https://discord.com/invite/dualityfalconcommunity
Submission link	https://forms.gle/t7YDc8VoWRCJYQaw7
Duality Registration	https://forms.gle/6BPyu8ciqXKhqDxJA

Task Instructions

i. AI Engineering:

This section will walk you through:

➔ Setting up your Falcon account

➔ Downloading and using the dataset

➔ Preparing your training environment

➔ Understanding the training workflow

1. **Create a Falcon Account:**
    a. Visit Falcon and sign up for an account if you don't have one
    b. Once registered, log in to access datasets, exercises, and tools.

2. **Download the Dataset**
    a. Download the dataset to your local machine (see the "Important Links" section).
        i. This drive has all the resources for all the Duality Hackathon tracks.
        ii. The dataset includes:
            - Pre-collected rgb color and segmented images
                - Separated into Train and Val folders
            - RGB color images in the testImages folder, to evaluate how well your model performs on unseen images. This includes 80% of the test data.
            - Sample train and test scripts
            - Environment setup scripts

- A script to visualize the segmentation using high-contrast colors

3. Set Up the Training Environment

a. Make sure you have Miniconda or Anaconda installed, and open an Anaconda Prompt window

b. Navigate to the ENV_SETUP sub folder

c. Run the setup_env.bat file in the Anaconda Prompt window

i. This will set up an environment called “EDU”, containing all the dependencies required to run the training and test scripts.

NOTE- Mac/Linux users, Create a setup_env.sh script with equivalent commands

4. Understand the Training Workflow

For these synthetic data competitions, the workflow varies from the traditional workflows:

a. train.py will train your model using the train and val images. The results of this step are important, but you will not know how robust your model is until you test it using unseen images.

b. test.py will test the model against images it HASN’T seen in training. The test images will be the same biome, but the actual location will be different.

5. Train the Model (of your choice) on the Sample Dataset

Once the EDU environment is ready and the dataset is in place:

a. Open an anaconda command prompt or a terminal

b. Navigate to the training and test scripts directory

c. Activate the environment by typing ‘*conda activate EDU*’ in the terminal.

d. Run the training command: ‘*python train.py*‘

This will begin training your model and save logs + checkpoints to the runs/ directory.

6. Establish Benchmark Results on the sample dataset

After training is completed, evaluate your model's performance by running the train script (train.py) in the same command prompt window. This tests its performance on real-world test images and gives you the following:

a. Predictions

b. Loss metrics

c. IoU score

Use the results as your benchmark, so that later, when you train with newer model settings, you can:

- Compare performance
- Track improvements
- Identify where the model struggled (e.g., specific classes, specific locations)

---

ii. Documentation & Presentation:

Ensure that your team's work is:

✔ Clear and understandable

✔ Reproducible by others

✔ Professional and impactful for judges

1. Keep it Structured & Organized using this General Structure

a. **Title & Summary:** Clearly state the purpose of the document.

b. **Step-by-Step Instructions:** Use numbered lists or bullet points.

c. **Diagrams & Visuals:** Use flowcharts, tables, and screenshots

d. **Graphs & Charts**: Show training loss, accuracy trends, and comparisons.

i. **Screenshots:** Use images from the runs/ folder after training
ii. **Before & After Images:** Show examples of correct vs. misclassified objects.

2. Document Everything, But Keep it Concise

a. Record major steps like dataset manipulation, training process, and evaluation.
b. Avoid overly technical language — aim for clarity.
c. Use clear, plain language—assume the reader is new to the project.

3. Example Entry:

a. **Task:** Model Training on Dataset
b. **Initial IoU Score:** 0.31
c. **Issue Faced:** Low recall for "Logs" class due to occlusion.
d. **Solution:** Augmented dataset with occlusion examples → **New, better score**

4. Documenting Failure Cases and Solutions

a. Include failure case images to illustrate what went wrong and how it was fixed.

5. Report Format \8 Pages Max)

Your Report should be concise, structured, and visually engaging. Use the following storytelling approach:

Problem → Fix→ Results → Challenges → Future Work

Page.No	Section	Description
1	Title	Team name, project name, brief tagline.

2	Methodology	Steps taken while training the model, and fine-tuned results.
3-4	Results & Performance Metrics	IoU score, confusion matrix, accuracy comparisons.
5-6	Challenges & Solutions	Key obstacles and how they were resolved.
7	Conclusion & Future Work	Final thoughts, and potential improvements.

---

3. Submission and Final Steps Instructions

i. Necessary Submission Files:

- A single, **Final Packaged Folder** that includes all necessary files:
  - Model file (.pkl file), training and inference scripts (train.py, test.py) and configuration files. This model will be used to test on the entire test dataset.
  - Any additional assets or scripts required to test your model

- A well-structured **Hackathon Report (PDF or DOCX)** that covers the following:
  - Methodology: Your training approach and setup
  - Challenges & Solutions: Issues faced and how you overcame them
  - Optimizations: Techniques used to improve model performance

Performance Evaluation: IoU score and Failure case analysis and observations.

- A **README.md or README.txt** that provides:
  - Step-by-step instructions to run and test your model
  - How to reproduce your final results
  - Any environment or dependency requirements
  - Notes on expected outputs and how to interpret them

Note: You are welcome to use your own models and custom training scripts or notebooks. However, you must train your model exclusively on the dataset provided for this challenge.

Important: Using any of the designated testing images for training purposes is strictly prohibited and will result in disqualification. Please ensure a clear separation between training, validation, and test sets throughout your workflow.

ii. Upload Instructions:

1. Ensure your submission folder contains all of the above
2. Compress everything into a .zip file.
3. Upload the zipped folder to a private GitHub repository of your own.
4. **Complete the submission form (see the "Important Links" section) :**
   a. Reporting your team's final score
   b. Providing the GitHub repository link
5. Finally, add the following reviewers as collaborators:
   a. Syed Muhammad Maaz
      i. Github Username – Maazyedm
   b. Rebekah Bogdanoff
      i. Github Username – rebekah-bogdanoff
   c. Evan Goldman
      i. Github Username – egold010
   d. Hamna Usman
      i. Github Username – hamnausman1
   e. Mahnoor Adeel
      i. Github Username – Mahnoor-Adeel

iii. After Submission:

1. Sharing Results & Feedback
    a. After submission, teams can showcase their models and insights.
    b. Feedback from judges will be provided after evaluation.

2. Future Opportunities & Improvements
    a. Continue exploring advanced topics such as:
        i. Self-supervised learning
        ii. Domain adaptation
        iii. Multi-view detection

3. Stay connected with us via Discord for:
    a. Future challenges
    b. Internship and apprenticeship opportunities c.
    Community events and AI workshops

NOTE: If you face any issues along the process join our discord channel linked above where we will be providing support to you for the hackathon.

---

4. Common Issues and Troubleshooting

FAQ's

1. **Will setup_env.bat Work on Both Windows & Mac?**
    - No, The setup_env.bat script is designed primarily for **Windows** environments.

- Alternatively, for Mac/Linux use create a setup_env.sh shell script equivalent that installs all required packages.

2. **My training process is too slow. What can I do?**
   - Here are a few tips to improve training speed:
     i. Reduce the batch size in your training configuration.
     ii. Close any unused applications or background processes to free up system resources.
     iii. If using GPU, monitor GPU usage with tools like nvidia-smi.
3. **How should I manage data transfer between team members or for submission?**
   - Why backup your data:
     i. Share results between teammates
     ii. Backup project files and checkpoints
   - We recommend using cloud storage platforms such as Google Drive, Dropbox, OneDrive, git

Support and Communication

1. **Join the official Duality Community Discord Server for:**
   a. Real-time help
   b. Announcements
   c. Live Q&A with organizers
   d. Invite Link Here

---

5. Glossary and Metric Benchmarks

Term	Definition
Digital Twin	A virtual replica of a real-world object or system.
Semantic Scene Segmentation	Every pixel in an image is labeled with a specific class.
IoU \Intersection over Union)	Measures how well a prediction from a model overlaps with the ground truth.
NMS \Non-Maximum Suppression)	A technique to remove duplicate detections of the same object.
Class Imbalance	When some object categories have significantly fewer samples than others.
Ground Truth	Manually labeled data specifying correct object locations and classes.
Training Loss	Lower loss = better training. If loss plateaus too high, the model may be underfitting. If the loss starts increasing, the model may be overfitting Expected Benchmark: Should steadily decrease
Inference Speed	Time taken to predict per image Expected Benchmark: < 50ms per image

We appreciate your hard work and look forward to reviewing your project! Feel free to share your work on LinkedIn and show the world your real-world problem solving skills in action!

Tag DualityAI to:

- Celebrate your team's efforts and creativity.
- Get noticed by industry experts and recruiters