

01 Ignitia Hackathon – Offroad Semantic Scene Segmentation

1. Overview – What is this hackathon about? (Pages 1–2)

Duality AI created this challenge to let participants practice advanced AI training using **synthetic (computer-generated) data** instead of real photos.

- You get **annotated images** (RGB photos + pixel-level segmentation masks) of **desert environments**.
- You train a **semantic segmentation model** on this data.
- Then you test how well it works on **new, unseen desert locations** (same general biome = desert, but different specific place → this tests generalization / domain shift handling).
- All data comes from **Falcon** (their digital twin platform using geospatial-based 3D simulated worlds).
- Goal = build the **most accurate model** possible by tuning training (architecture, hyperparameters, augmentations, etc.).
- You learn how synthetic data helps when real data is expensive, rare, dangerous, or hard to label (e.g. remote deserts, bad weather, night, etc.).

Benefits for participants:

- Build new skills in training robust models for **context shifts / unseen environments**.
- Improve your CV/portfolio.
- Network with Duality AI people and others.
- Win prizes + recognition.

Objectives (very clear):

- Train robust segmentation model on provided synthetic desert data.
- Test performance on novel (but still desert) scenes.
- Optimize for **accuracy**, **generalization**, and **efficiency** (important for real robots/vehicles).

Why synthetic data + digital twins matter:

- Real-world labeled data for off-road is very expensive and slow to collect.
- UGVs (unmanned ground vehicles) need good perception for path planning / obstacle avoidance.
- Semantic segmentation = every pixel gets a class label → critical for understanding "where can I drive?" vs. "obstacle / bush / tree / rock".

- Synthetic data: cheap, fast, controllable (add rain, different times of day, specific objects), huge variety → great for training robust models.

2. Data Overview – What exactly are you segmenting? (Pages 3–4)

Dataset = synthetic desert scenes from FalconEditor.

Classes (pixel labels you must predict) — these are the things the model has to distinguish:

ID	Class Name	Description / typical look
100	Trees	Cacti / desert trees
200	Lush Bushes	Greener bushes
300	Dry Grass	Dry/brown grass patches
500	Dry Bushes	Drier, sparser bushes
550	Ground Clutter	Small debris, rocks, trash on ground
600	Flowers	Small desert flowers
700	Logs	Fallen wood / branches
800	Rocks	Stones / boulders
7100	Landscape	General traversable ground (not in other classes)
10000	Sky	Sky / background

(Note: Some classes like "Landscape" act as "everything else that's ground but not special". Sky is often included in segmentation tasks.)

You train on **Train + Val** folders (RGB + mask pairs).

You get **testimages** folder (only RGB, no masks — these are unseen locations for final evaluation).

3. Hackathon Tasks & Roles (Pages 4–5)

They suggest dividing team work:

AI Engineering role:

- Train & fine-tune segmentation model.
- Evaluate performance.
- Optimize (higher IoU + faster inference if possible).

Documentation & Presentation role:

- Document everything: augmentations, filtering, training setup, loss curves, etc.

- Make report + presentation with visuals (graphs, before/after images, failure cases).
- Create loss/performance graphs.

Key deliverables (must submit these):

1. Trained model package → weights + train/test scripts + config files.
2. Performance report → IoU, loss graphs, failure case analysis + suggestions.

4. Judging Criteria (100 points total) (Page 5–6)

- **Model Performance** → **IoU score** = 80 points (main thing — how well pixels are correctly classified vs ground truth).
- **Report Clarity** → 20 points (well-organized, clear methodology, challenges/solutions, professional).

Balance good model + good documentation.

5. Important Links (Page 6)

- Sign up for free Falcon account: <https://falcon.duality.ai/auth/sign-up?...> (with utm tags for tracking hackathon participants)
- Download dataset: <https://falcon.duality.ai/secure/documentation/hackathon-segmentation-desert?...>
- Discord community/support: <https://discord.com/invite/dualityfalconcommunity>

6. Step-by-Step Task Instructions – AI Engineering (Pages 7–9)

1. Create Falcon account (free).
2. Download dataset (go to segmentation track section).
 - Contains: Train/Val (RGB + masks), testImages (RGB only), sample train.py / test.py, visualization script, env setup.
3. Setup environment:
 - Use Miniconda/Anaconda.
 - Go to ENV_SETUP folder.
 - Run `setup_env.bat` (Windows) → creates "EDU" conda env with PyTorch, segmentation libs, etc.
 - Mac/Linux → write your own `setup_env.sh` with same pip/conda installs.
4. Training workflow (different from normal because of synthetic → unseen test):
 - `train.py` → trains on train/val → saves checkpoints/logs in `runs/`.
 - `test.py` → runs on unseen testImages → gives predictions, loss, **IoU score**.
5. Run training: `conda activate EDU` → `python train.py`

6. After training → run test to get benchmark IoU on unseen data.

- Use this as baseline → try changes (model, augmentations, learning rate, etc.) → compare new IoU.

7. Documentation & Presentation Guidelines (Pages 9–11)

- Make work **clear, reproducible, professional**.
- Structure report (max 8 pages):
 1. Title + team/project summary
 2. Methodology (steps, fine-tuning)
 - 3–4. Results (IoU, confusion matrix, graphs, comparisons)
 - 5–6. Challenges & solutions (with images!)
- 3. Conclusion + future work

Storytelling: Problem → Fix → Results → Challenges → Future

Include:

- Loss curves, before/after segmentation images.
- Failure cases (e.g. confuses dry grass vs dry bushes? Logs hidden?).
- Screenshots from `runs/` folder.

8. Submission Instructions (Pages 11–13)

Must submit:

- One zipped folder with:
 - train.py, test.py (or your custom scripts/notebooks)
 - config files
 - model weights
 - Any extra scripts needed to run
- PDF/DOCX report (methodology, challenges, optimizations, IoU + failure analysis)
- README.md / .txt → exact steps to run/reproduce, dependencies, expected output

Rules:

- Train **only** on provided dataset (no external data).
- **Never** use test images for training/validation — instant disqualification.
- Upload zip to **private GitHub repo**.
- Fill submission form (link in important links) → report your final IoU + GitHub link.
- Add these as collaborators:
 - Maazsyedm (Syed Muhammad Maaz)
 - rebekah-bogdanoff (Rebekah Bogdanoff)
 - egold010 (Evan Goldman)

9. After Submission (Page 13)

- Share your work (LinkedIn, tag DualityAI).
- Get judge feedback later.
- Future: self-supervised learning, domain adaptation, multi-view, etc.
- Stay in Discord for more events, internships, workshops.

10. Common Issues & FAQs (Pages 13–15)

- `setup_env.bat` = Windows only → Mac/Linux users must make `.sh` equivalent.
- Training slow? → smaller batch size, close apps, check GPU usage (`nvidia-smi`).
- Backup/share data → use Google Drive / Git / cloud.

Glossary / expected behavior:

- IoU = main metric (higher = better overlap).
- Loss should steadily decrease.
- Inference < 50 ms/image is good benchmark.
- Watch for overfitting (loss increases), underfitting (loss high & flat).

This is a classic synthetic-to-(semi-)real generalization challenge for off-road perception — very useful skill for autonomous vehicles / robotics.