Vessel Re-Identification

Training a vessel ReID model on pre-cropped satellite images and deploying a FAISS gallery for similarity search.

Quick start

```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
pip install -e .
```

Data layout

See `data/README.md` for the expected CSV format and folder structure.

Typical workflow

1. Prepare ID-disjoint splits (train/val/gallery/query).
2. Train the embedding model with triplet loss.
3. Build the FAISS gallery from known boats.
4. Run inference with new images and a similarity threshold.

Generate splits:

```
python -m vessel_reid.cli.split_ids --csv data/all_labels.csv --out-dir
data
```

Or generate splits directly from image filenames:

```
python -m vessel_reid.cli.split_ids --image-dir data/images --out-dir
data
```

Commands

Train:

```
python -m vessel_reid.cli.train --config configs/train.yaml
```

Build gallery:

```
python -m vessel_reid.cli.build_gallery --config configs/gallery.yaml
```

Query:

```
python -m vessel_reid.cli.infer --config configs/inference.yaml --image
path/to/query.jpg --length-m 42.7
```

Notes

- Assumes each image contains one boat (no detection stage).
- For evaluation, keep boat IDs disjoint across train/val/test.
- In deployment, it is expected to keep multiple images per boat in the gallery.
  538006648_bd6fdad1039641bca89994c2689f6bc5.jpg

Changelog - Branch: `mica`

`src/vessel_reid/cli/train.py`

- **Change**: Added mixed precision training with `torch.cuda.amp`
- **Why**: Gets 2-3x speedup on A100 GPU by using half-precision where possible
- **Change**: Added `pin_memory=True` and `persistent_workers=True` to DataLoader
- **Why**: Makes data loading faster by keeping workers alive and pinning memory

`configs/train.yaml`

- **Change**: Increased `batch_size` from 32 to 128
- **Why**: A100 has 40GB of VRAM, was barely using any of it with batch size 32
- **Change**: Increased `num_workers` from 4 to 8
- **Why**: Load data faster with more parallel workers

`src/vessel_reid/utils/seed.py`

- **Change**: Set `cudnn.benchmark=True` and `cudnn.deterministic=False`
- **Why**: Lets cuDNN find the fastest algorithms for this hardware

`src/vessel_reid/api/api_helper.py`

- **Change**: Increased API limit from 100 to 1000 events
- **Why**: Grab way more vessel detections in one API call

- **Change**: Added `offset` parameter for pagination support
- **Why**: Allow fetching results beyond the first 1000 events

`src/vessel_reid/api/populate_data.py`

- **Change**: Increased time window from 1 day to 30 days
- **Why**: Need more historical data to find multiple images of each boat
- **Change**: Added vessel grouping and filtering (minimum 3 images per vessel)
- **Why**: Triplet loss needs multiple images of the same vessel to learn properly
- **Change**: Added pagination loop to fetch all available results
- **Why**: Don't miss any data if there are more than 1000 events in the time window
- **Change**: Save fetched event IDs to `data/fetched_event_ids.txt`
- **Why**: Track which events were fetched for future reproducibility