12/02/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

80 days before May 1st

Updates

- Vision:
  - New datasets
  - YOLO training result
  - LD net inference updates
- Estimation integrated testing:
  - Testing and validating coordinate transformations and batch optimiser modules with Landsat images
- Testbed:
  - Preliminary research on V2 darkroom design
  - Research about multi camera calibration and color/brightness correction

Blockers

- Camera finalization (avionics?)
- Differences between training pictures and test photos → need to test with the new camera

Weekly Plan

- Vision
  - Develop YOLO training plan with region/parameter priority
  - Get high-performing (generalizable) LD model for one region for orbital pass test in V2
  - Complete integration of RC + LD inference on Jetson
  - Improve RCnet accuracy with new dataset
- Estimation
  - Test the coordinate transformations using real world points seen from the reference camera to verify the camera unit vectors
  - Test the batch optimizer with simulated world coordinates and camera matrices.
- MCM FSW testing
  - Test and validate Control software on PyCubed with simulated sensor readings

Interface dependencies

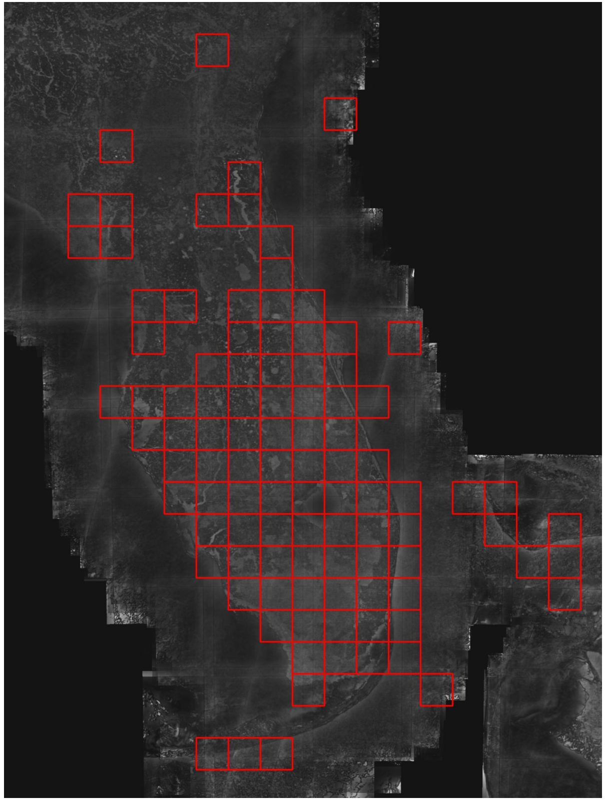
-

Vision

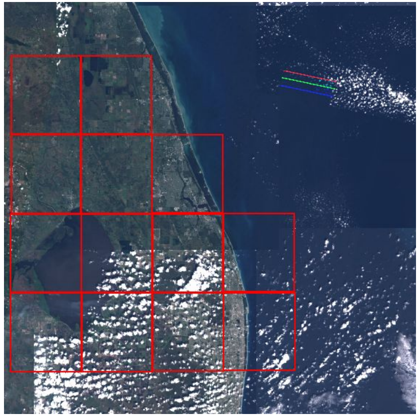
- **Dataset**
  - Integrated new downloader to download sentinel 2 mosaics to current pipeline
  - Generated new YOLO datasets
    - 2000 sentinel 2 train images + 500 rotated landsat images
    - Different landmark sizes: 25x25km, 30x30km, 35x35km
- **Training**
  - **RC:**
    - Implemented inference pipeline to work on jetson
    - Trained Efficientnet-b0 on old data but hit ceiling at ~40% mAP. Training with new data to start today - experimenting to increase mAP to ~70% this week
  - **LD:**
    - Trained YOLO models of different sizes (S, M, L) on Sentinel train/Landsat val set
      - Diverse augmentation params needed to generalize
      - mAP-50 above **0.9**, mAP50-95 above **0.77** for all three model sizes after 300 epochs
    - Evaluation script to track MSE, missed detections, extraneous detections across val set
    - Next step: prune landmarks and tune landmark size and number based on centroid pixel error
- **Jetson**
  - LD inference class updated to provide diverse output, testing script updated to provide evaluation of mean square error (MSE) for centroid pixel coordinates and average MSE for one inference run

LD Datasets

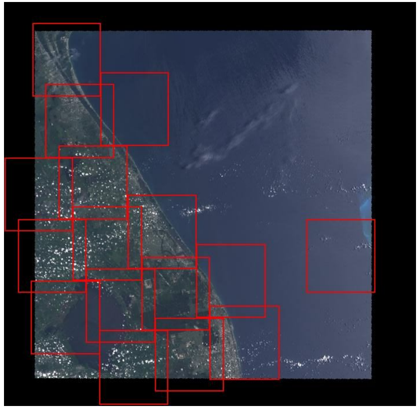
Landmark size: 25km x 25km



Train: Sentinel-2 mosaic



Validation: Rotated Landsat 8



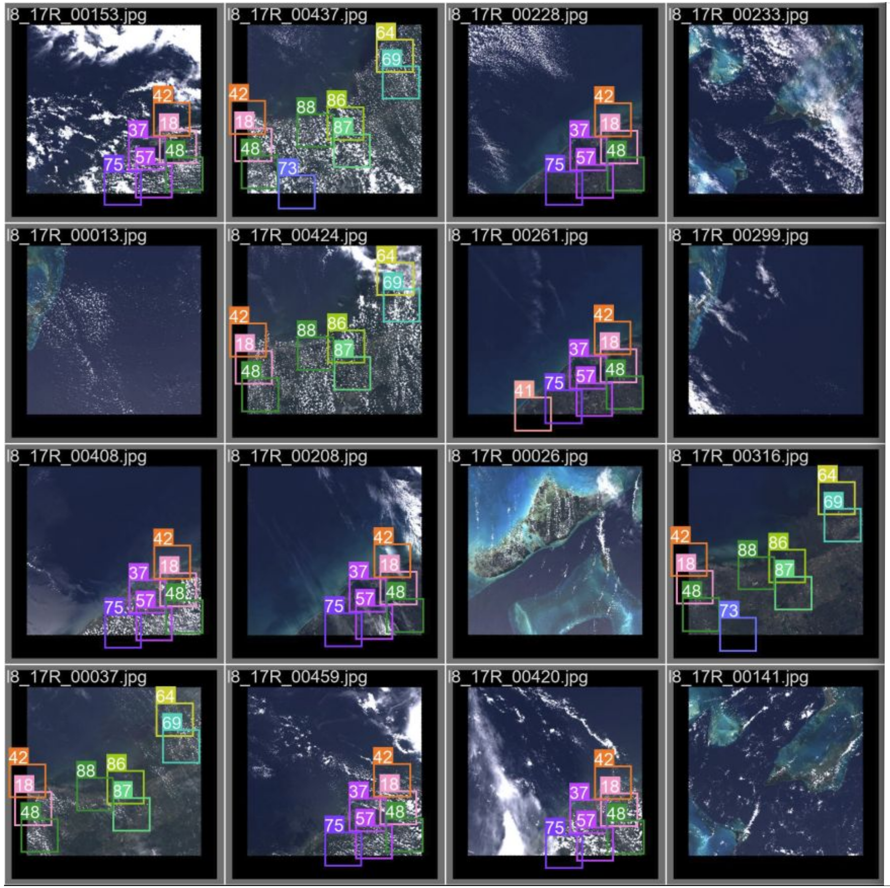
Training Result - medium vs. large model

Augmentations:

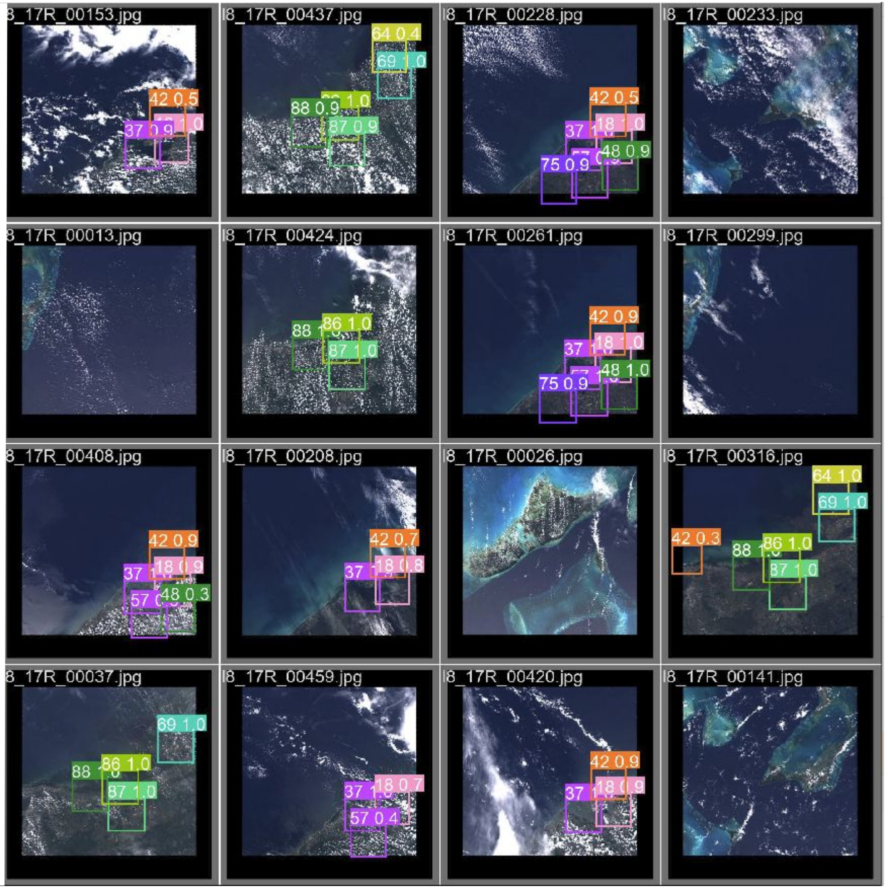
- Rotation
- Scale
- Perspective
- Translation
- Hsv color
- erasing

Training Result - Validation on Landsat

GT



Detected



Integrated testing

- Tested batch optimiser and coordinate transformation using landsat images displayed on the screen and image captured by testbed camera. - **Incorrect approach**
  - Mapped the centroids of the detected bounding boxes in the original Landsat ground truth to the image captured by the testbed camera.
  - Used these centroids to generate camera vectors pointing to landmark points
  - Ran the batch optimizer to determine attitude

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D Image Coordinates

Intrinsic properties (Optical Centre, scaling)

Extrinsic properties (Camera Rotation and translation)

3D World Coordinates

Next steps

- Test the coordinate transformations using real world points seen from the reference camera to verify the camera unit vectors
- Test the batch optimizer with simulated world coordinates and camera matrices.