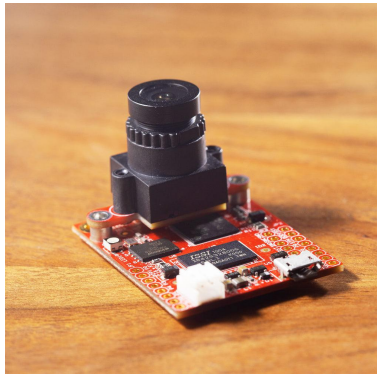
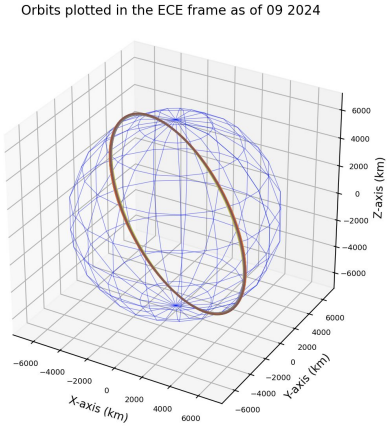




Blockers

- Lack of concrete power and compute usage numbers: Due to the model/pipeline not being finalized (Lower the better.)
- Access to GPU workstation and start training/testing
- For sim: best tool to use? Build from scratch or use Orekit/Skyfield/GMAT to simulate observations?

Week's Results

- Updated level two requirements.
- Started researching different models and pipelines, including YOLO v8, YOLO v10, and contrastive learning models.
- Initiated research into camera modules, overall architecture, and preprocessing.
- Started working on orbit/imaging sim

By Sunday:

Dataset loading, Earth sim, choose cameras, camera model

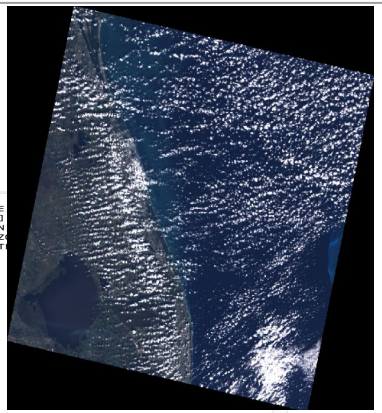
Interfaces

- **Compute requirements**: Obtain the required frame rate and other data (gnc), then provide the compute requirements to the avionics team.
- **Cameras**: Collaborate with both the mechanical and avionics teams to finalize the number of cameras and camera architecture
- Both of these tasks will be completed by this weekend.

	Camera Module v1	Camera Module v2	Camera Module 3	Camera Module 3 Wide	OpenMV Cam RT1062
Pixel Width (µm)	1.4	1.12	1.4	1.4	1.4
Pixel Size (µm^2)	1.9600	1.2544	1.9600	1.9600	1.4
Pixel Size (m^2)					
Focal length (mm)	3.6	3.04	4.74	2.75	2.8
Focal length (m)	0.0036	0.00304	0.00474	0.00275	0.0028
Price ($)	25	25	25	35	120
Sensor resolution	2592 × 1944 pixels	3280 × 2464 pixels	4608 x 2592 pixels	4608 x 2592 pixels	2952x1944 pixels
GSD at altitude (km)					
400	0.0002177777778	0.0001650526316	0.0001654008439	0.0002850909091	0.0002
450	0.000245	0.0001856842105	0.0001860759494	0.0003207272727	0.000225
500	0.0002722222222	0.0002063157895	0.0002067510549	0.0003563636364	0.00025
550	0.0002994444444	0.0002269473684	0.0002274261603	0.000392	0.000275
600	0.0003266666667	0.0002475789474	0.0002481012658	0.0004276363636	0.0003
650	0.0003538888889	0.0002682105263	0.0002687763713	0.0004632727273	0.000325
700	0.0003811111111	0.0002888421053	0.0002894514768	0.0004989090909	0.00035
750	0.0004083333333	0.0003094736842	0.0003101265823	0.0005345454545	0.000375

```
(eedl) stoneyey@StonedeMBP eedl-main % python eedl.py -h
usage: eedl.py [-h] [-b BOUNDS [BOUNDS ...]] [-g GRID_KEY] [-i IDATE
               [-m MAXIMS] [-se {18,19,s2}] [-o OUTPATH] [-r REGION]
               [-c CRS] [-cc CLOUD_COVER_MAX] [-ccgt CLOUD_COVER_MIN
               [-cm CUSTOM_MOSAICS] [-vb VERTICAL_BUFFER] [-hb HORIZ
               [-np NPROCS] [-rm REGION_MOSAIC] [-rc REGION_COMPOSIT

options:
  -h, --help            show this help message and exit
  -b BOUNDS [BOUNDS ...], --bounds BOUNDS [BOUNDS ...]
  -g GRID_KEY, --grid_key GRID_KEY
  -i IDATE, --idate IDATE
  -f FDATE, --fdate FDATE
  -s SCALE, --scale SCALE
  -m MAXIMS, --maxims MAXIMS
  -se {18,19,s2}, --sensor {18,19,s2}
  -o OUTPATH, --outpath OUTPATH
  -r REGION, --region REGION
  -e {GEOTiff}, --format {GEOTiff}
  -sd SEED, --seed SEED
  -c CRS, --crs CRS
  -cc CLOUD_COVER_MAX, --cloud_cover_max CLOUD_COVER_MAX
  -ccgt CLOUD_COVER_MIN, --cloud_cover_min CLOUD_COVER_MIN
  -ba BANDS [BANDS ...], --bands BANDS [BANDS ...]
  -cm CUSTOM_MOSAICS, --custom_mosaics CUSTOM_MOSAICS
  -vb VERTICAL_BUFFER, --vertical_buffer VERTICAL_BUFFER
  -hb HORIZONTAL_BUFFER, --horizontal_buffer HORIZONTAL_BUFFER
  -gd GDRIVE, --gdrive GDRIVE
  -np NPROCS, --nprocs NPROCS
  -rm REGION_MOSAIC, --region_mosaic REGION_MOSAIC
  -rc REGION_COMPOSITE, --region_composite REGION_COMPOSITE
```



Blockers

- The estimation of power usages for ML + image processing + feature matching + relative pose of cameras rpt the world frame
- Access to GPU workstation to start training/testing

Week's Results

- Comparison of each camera model + GSD and price
- Dataset: github.com/CMUAbstract/eedl
- 700KB - 1MB per image in average
- Hardware survey:
  - Jetson Orin NX: 16G VRAM, max 25W
  - Jetson Orin Nano: 8G VRAM, max 15W or 4G VRAM, max 10W
- Preprocessing: Variance of Laplacian（Blur）, Specular Highlight Detection (glare), OpenCV Out-of-focus/motion Deblur Filter (recovery), ML classifier (filter)

Interfaces

- GNC: discussed interface for the simulation. Working on determining the camera calibration models for the sim.
- **Jetson Orin NX** is better suited for high-performance tasks that require more memory and power, making it ideal for more demanding AI/ML workloads.
- **Jetson Orin Nano** targets users looking for lower power consumption, with the option to scale down to 4GB VRAM and 10W for even more constrained environments.
- **Orin NX** is the more powerful and flexible option, while the **Orin Nano** is designed for smaller-scale, low-power

Blockers

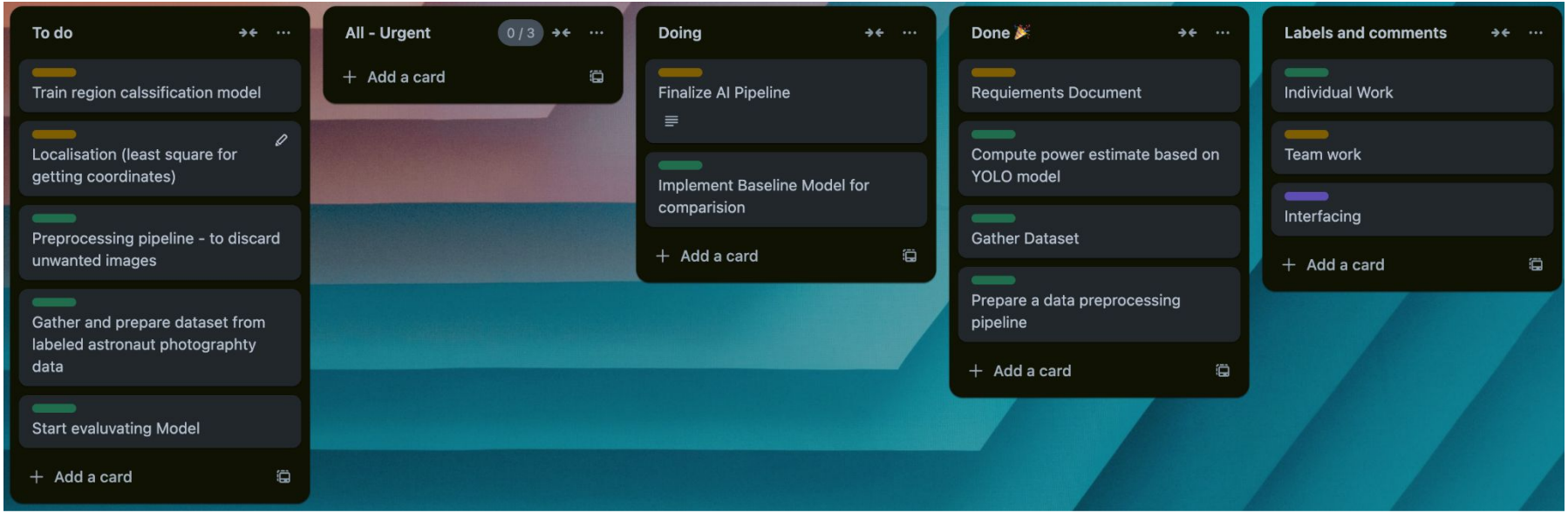
- Need method for labelling raw data
- Need preliminary framework for evaluating different models

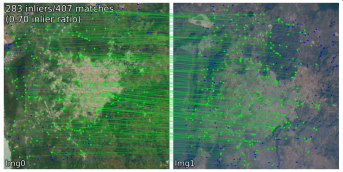
Week's Results

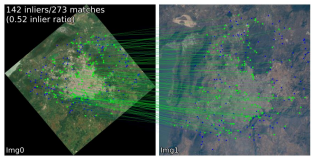
- Completed code for dataset loading from EEDL
- Added data augmentation techniques for simulating different situations in orbital movement (off-nadir, motion blur, etc.)
- Secured computational resources for training model.
- Started Preparing the VINSAT YOLO pipeline for baseline implementation.
- Exploring alternative options simultaneously.

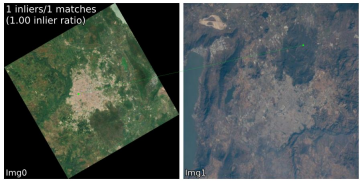
Interfaces

- GNC: Set max off-nadir degrees to 30
- Avionics: Settle on Camera number and configuration


To do
Train region calssification model
Localisation (least square for getting coordinates)
Preprocessing pipeline - to discard unwanted images
Gather and prepare dataset from labeled astronaut photographty data
Start evaluvating Model
Add a card
All - Urgent
0 / 3
Add a card
Doing
Finalize AI Pipeline
Implement Baseline Model for comparision
Add a card
Done
Requiements Document
Compute power estimate based on YOLO model
Gather Dataset
Prepare a data preprocessing pipeline
Add a card
Labels and comments
Individual Work
Team work
Interfacing
Add a card







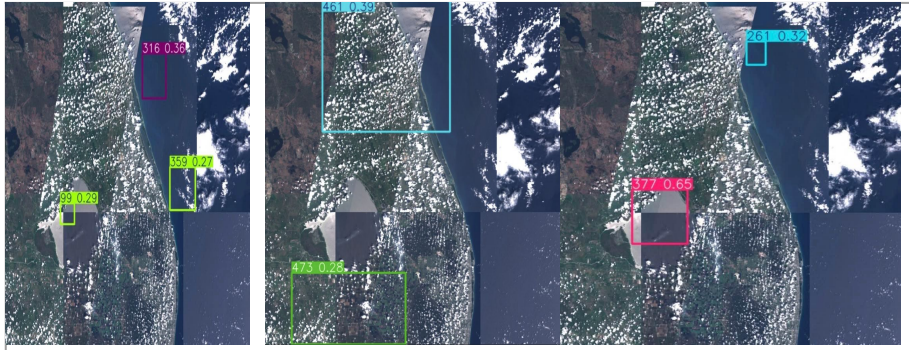
Blockers

Week's Results

- Testing PoC of matching-based methods (vary scenes + rotation and scale) → rot variant ← (plan) SfM
- Initialize pipelines + try to improve acc.
- Survey methods for computing uncertainty of landmarks ← gaussian (gnc preference)

Interfaces

- GNC: list of camera models



Blockers

Check performance of the set up off nadir, determine the degree based on the real performance

Set up evaluation matrix for baseline and current model

Lab's GPU

Week's Results

- Set up the argus 1 as baseline model on GPU
- Adjust the model using test code and analyze the output
- For the region classification, locate it on local (previous worked only on GPU)

Interfaces

Clarify the scenario of images detection that needed (under discussion)

Daytime scenarios provide high-resolution images with natural lighting, while at night time, low light conditions introduce noise, requiring advanced image enhancement techniques