

Blockers:

- Waiting for CPU?
- 100 images took 15m for preprocessing

Requirements:

- N/A

Last week:

- Dataset:
 - Developed system to track and visualize dataset coverage
 - Building dataset pipeline in a batch processing maner
 - Finalize on API research and complete the pre-processing program and put together the dataset building pipeline
- Jetson Orin:
 - Deployed inference for different network on Orin - need to check power usage
- Pipeline:
 - Add visualization demo

Next steps:

- Finalize the API tool for dataset building pipeline
- Run Inference on Jetson Orin

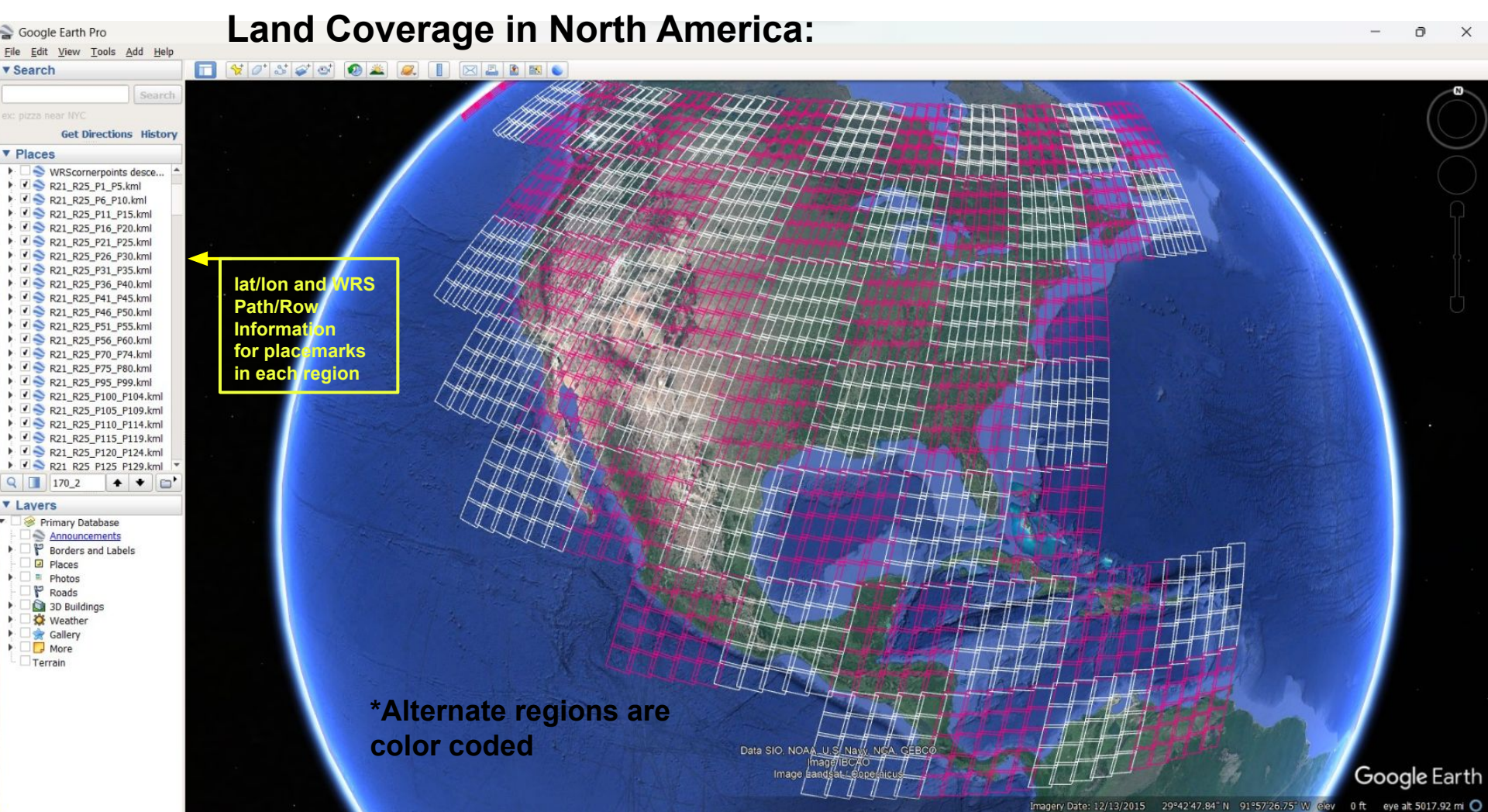
Interfaces

Avionics:

- Determine power usage of Orin, coordinate this usage with power budget

GNC:

- Discuss about partitioning Orin CPU/GPU usage
- Determine GNC estimation software that needs to be run on Orin/how that interacts with duty cycling the Orin



Batch Processing Dataset Pipeline

Get Region Info -> Search Data by Region and Dates via [USGS M2M API](#) -> Filter by Cloud Cover, Day/Night -> Download via M2M API -> Pre-processing & Augmenting Data

Dataset Intermediate Size*

$313,414 \text{ KB} * 14,600 * 6 \approx 27.5 \text{ TB}$

- > Split into 200 batches of 30 Regions
- > Can start running before workstation setup complete
- > Batches can be accessed through different API if needed

Dataset Final Size*

$1,643 \text{ KB} * 14,600 * 2 * 6 * 4 \approx 1 \text{ TB}$

**Tentative Intermediate Size: Size of single LandSat RGB image constructed from raw bands (KB) * Number of unique images * 6 variations (Could Cover, Day/Night)*

**Tentative Final Size: Size of single pre-processed LandSat RGB image (KB) * Number of unique images * ~2 camera frames per LandSat image * 6 variations (Could Cover, Day/Night) * ~4 augmentations (Brightness/Contrast/Rotation)*

Pipeline:

Stage 1 - Region Classifier

Stage 2 -

1. Landmark Detection
 - a. Detect landmarks -> Given landmarks' positions -> Calculate our position



Region "A"

YOLOv8 A

Landmark

Position



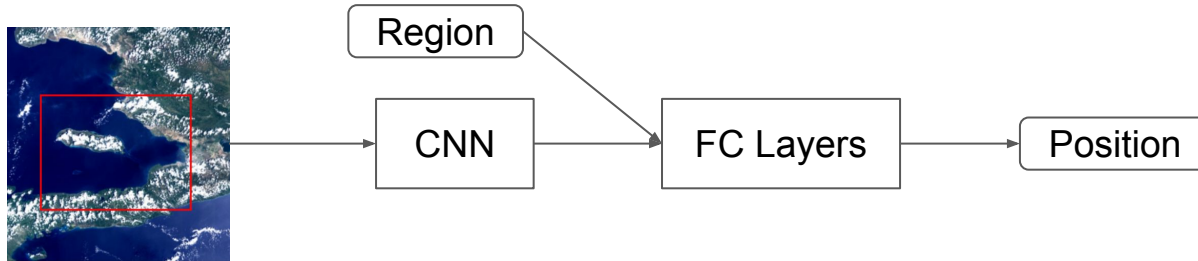
```
[{"class": "island A", "confidence": 0.98,  
  "bounding_box": {"x": 120, "y": 150,  
    "width": 50, "height": 75}  }]
```

Pipeline:

Stage 1 - Region Classifier

Stage 2 -

1. Landmark Detection
 - a. Detect landmarks -> Given landmarks' positions -> Calculate our position
2. End-to-end:
 - a. Image + Region -> Neural Network -> Our position



Pros & Cons for Landmark Detection:

1. more accurate if we can precisely locate the landmarks
2. more obvious and makes more "human" sense
3. may not be able to find landmarks in certain pictures
4. need lots of manually-labelled pictures & landmark data (position etc.)
5. more overhead in terms of post-processing network output