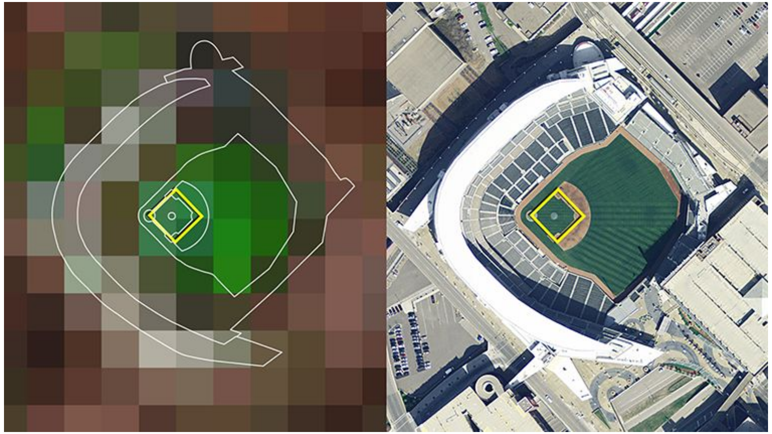


Blockers:
- **N/A**

Requirements:
- **N/A**

Last week:
- **Dataset:**
  - Pipeline
  - X image (GSD - 30 meters, 7000 x 7000 pixels)
  - Investigated classical CV, preprocessing
- **Pipeline:**
  - Add visualization demo
  - Improve models

Next step:
  - Finalize dataset requirement and construct dataset
  - Setup 1 camera and software
  - Documentation
  - Add visualized demo
  - Improve models
  - Run “hello world” 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

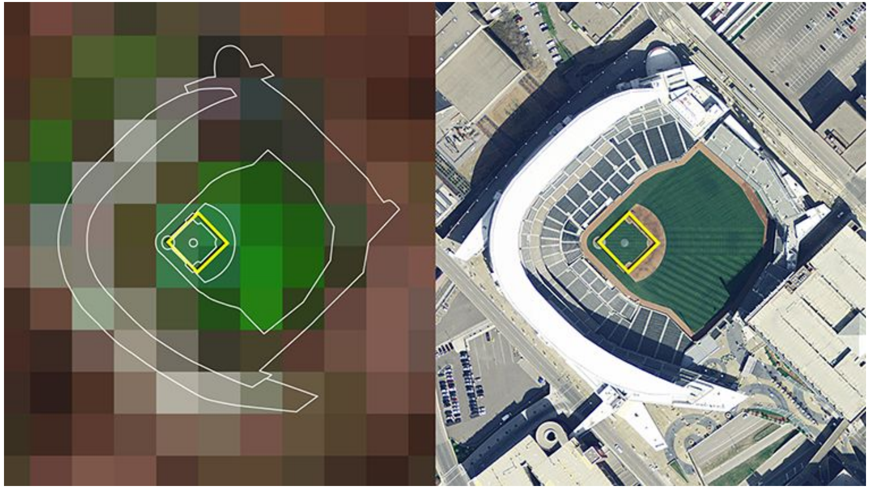
Camera Choice: TEVI-AR1335-C-S85-IR-EVK

Camera & Lenses:
Active Pixels: **4208 (H) x 3120 (V)**
Pixel Size: 1.1 µm x 1.1 µm
Telephoto 16mm M12 Lens

Known:
Altitude: 450km
Focal Length: **16**, 20, 22 mm

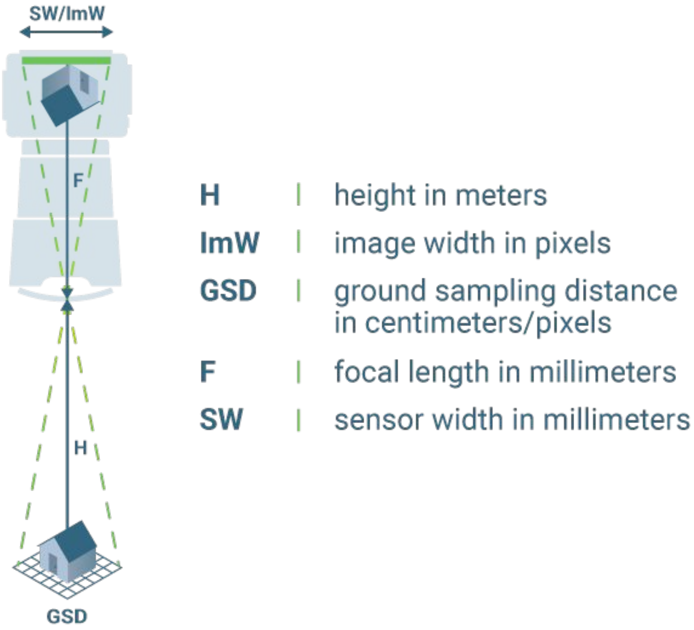
Focal Length (mm)	GSD (m/px)	Image Size (H)	Image Size (V)	swath (H) (km)	swath (V) (km)
16	**30.9375**	**4208**	**3120**	**130.185**	**96.525**
20	**24.75**	4208	3120	104.148	77.22
22	**22.5**	4208	3120	94.68	70.2

LandSat GSD: 30m



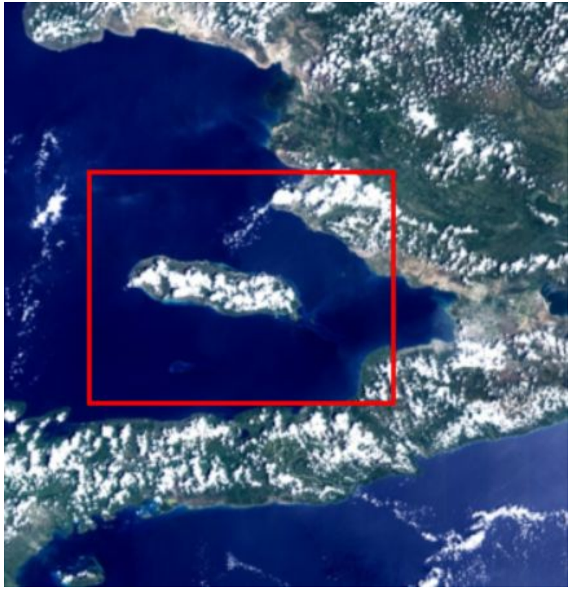
Target Field, Minneapolis, Minnesota, Home of 2014 MLB All-Star Game, Landsat image (left), aerial photograph (right). ***A single Landsat pixel is roughly the size of a baseball diamond.***

https://landsat.gsfc.nasa.gov/article/picturing-a-pixel/

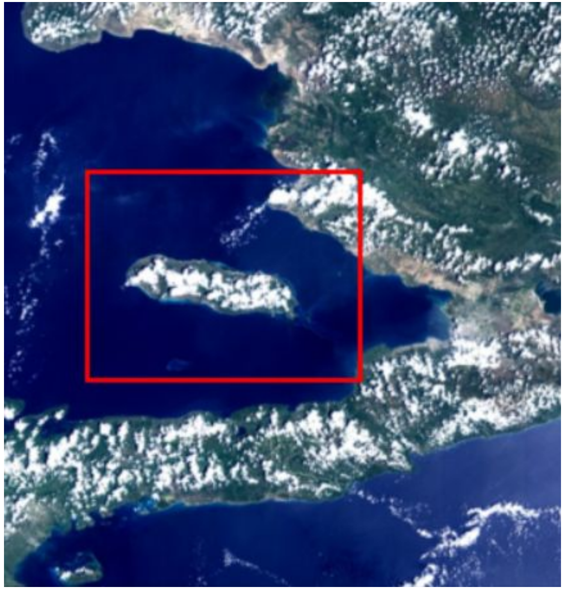


LandSat frame vs Camera frame

16 mm focal length



20 mm focal length



22 mm focal length

LandSat Image

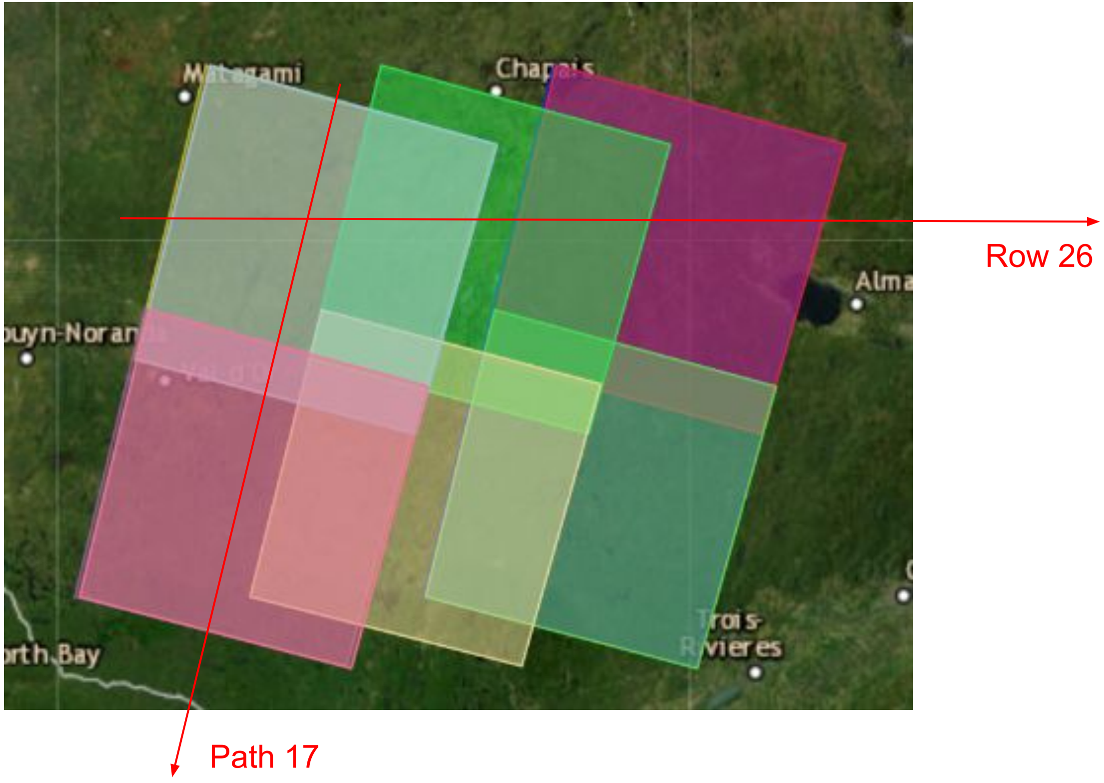
~ 6510 x 6656

Training data that matches the camera frame

~ 4128 x 3147

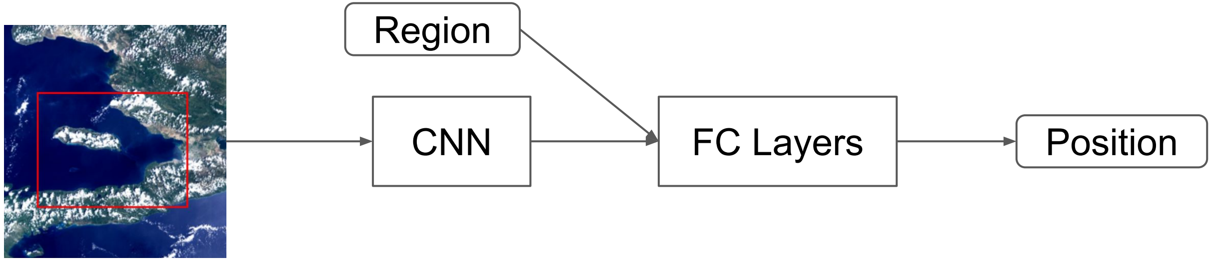
- **MaxPooling** Landsat image to reduce the size without losing landmark features
- **Cropping** to maintain the aspect ratio

Reason why Cropping works: LandSat images overlap



Pipeline:
Stage 1 - Region Classifier
Stage 2 -

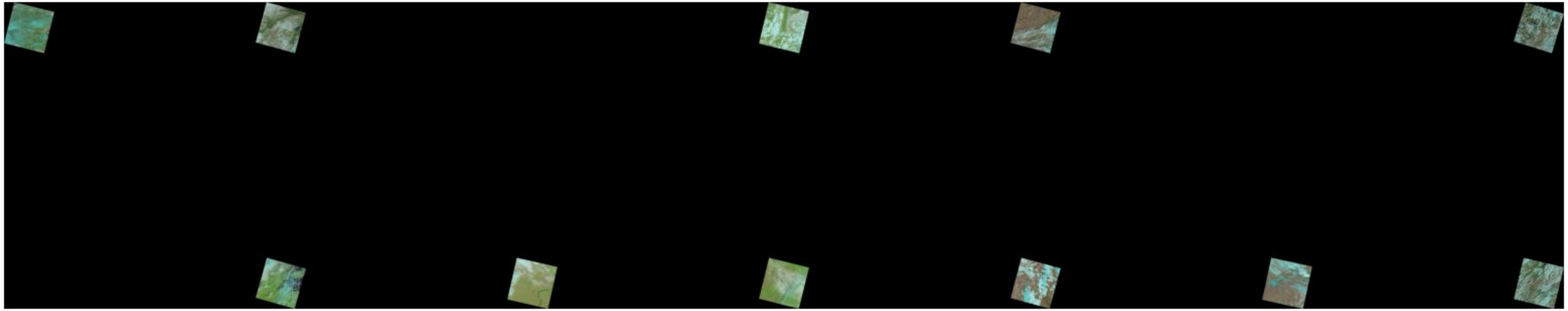
1. Landmark Detection
   a. Detect landmarks -> Given landmarks' positions -> Calculate our position
2. End-to-end:
   a. Image + Region -> Neural Network -> Our position

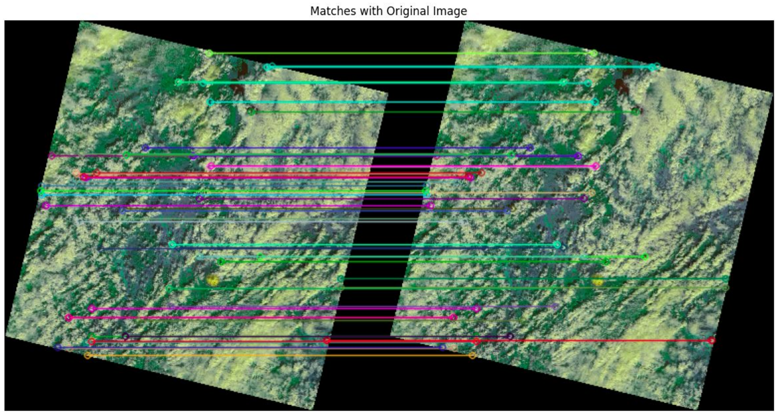


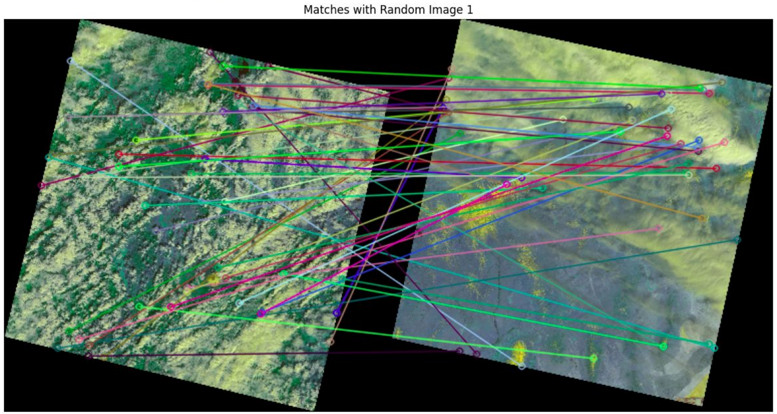
Pros & Cons for Landmark Detection:

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

Pipeline:
Stage 1 - Region Classifier







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}   }]