GUAVA

Grab UAV Accurately

CONTENTS

01 1. Team members
Introduction

ROS
Inbae Kang
Chung-Ang University

Radar
Youngjin Kim
Sogang University
Haeeun Lee
Seoul Women's University

Rail, Radar
Kyungyeon Park
SungKyunKwan University

Drone, Web
Kyeongnam Kim
Sejong University

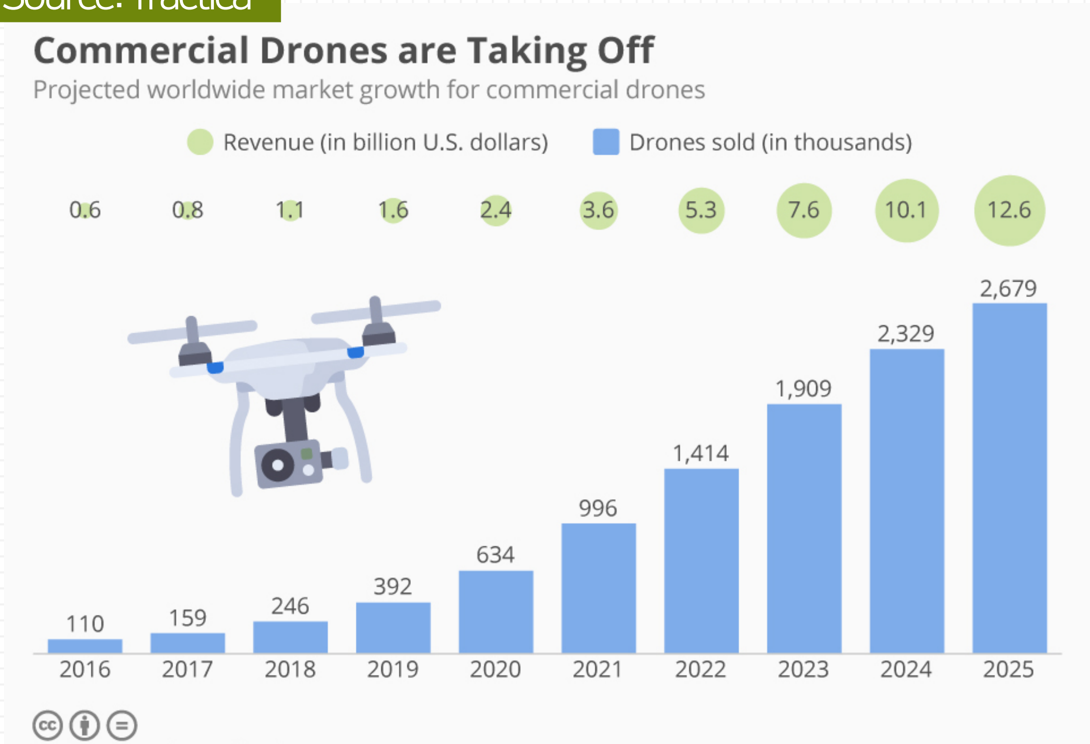
Camera
Dojin Kim
Handong University

Thanks to **Seongha Park**
Yongho Kim
Jieun You

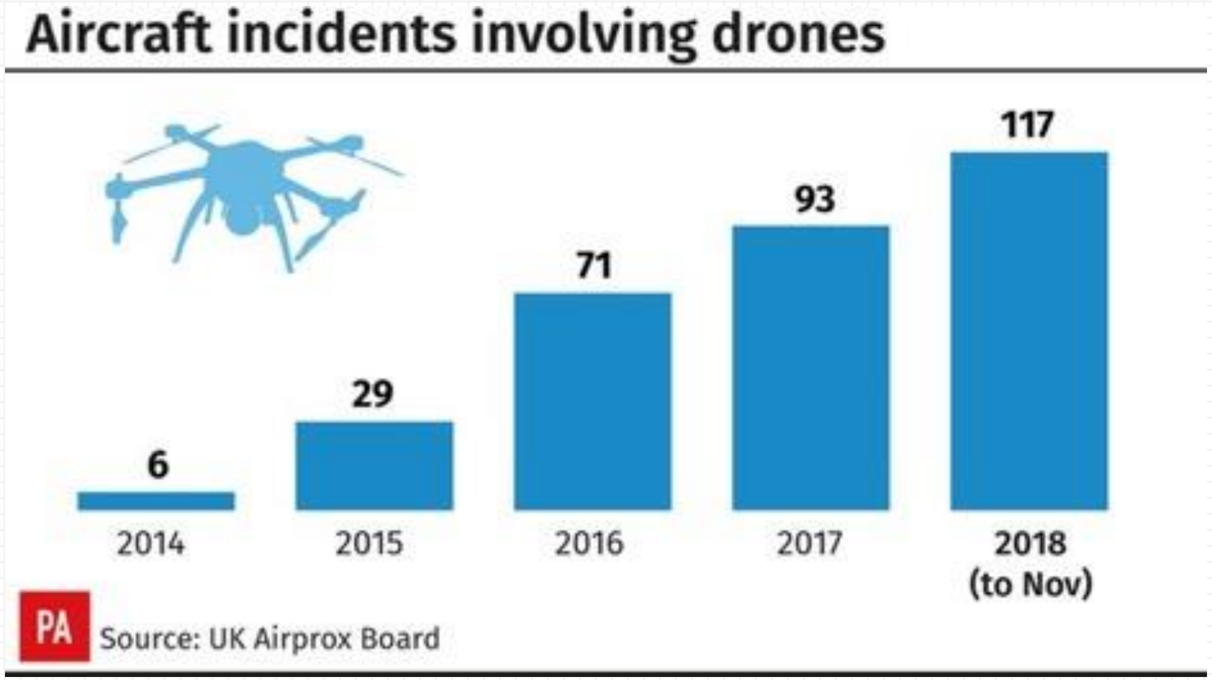
Introduction

- As the number of drones is increasing, the number of incidents involving drones is also increasing.
- There is a demand for drone surveillance system to prevent these incidents in advance.

Source: Tractica



Source: UK Airprox Board



01 2. Background
Introduction

- There are technologies that detect drones, but too expensive.

  → Our goal is making detecting UAV system with high accuracy and low cost.

76 GHz drone detection radar

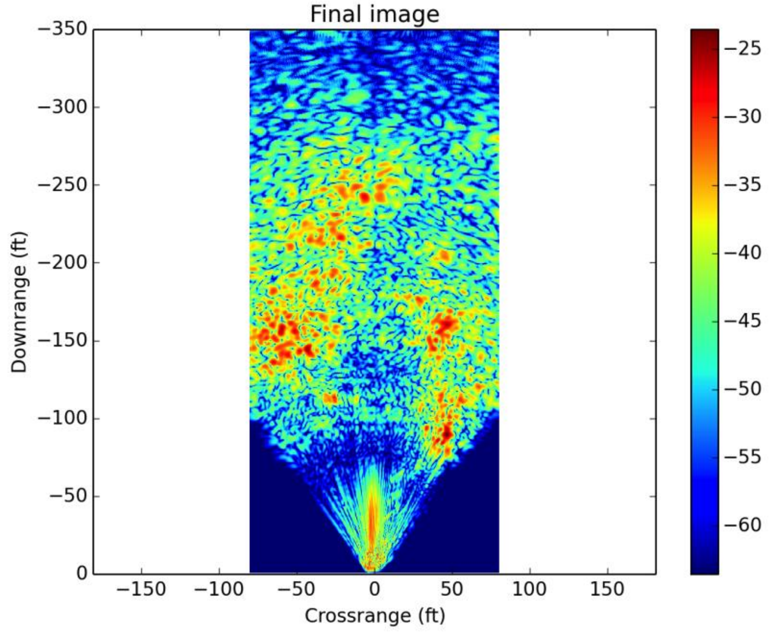
aaronia drone detection system

dmt drone detection

Introduction

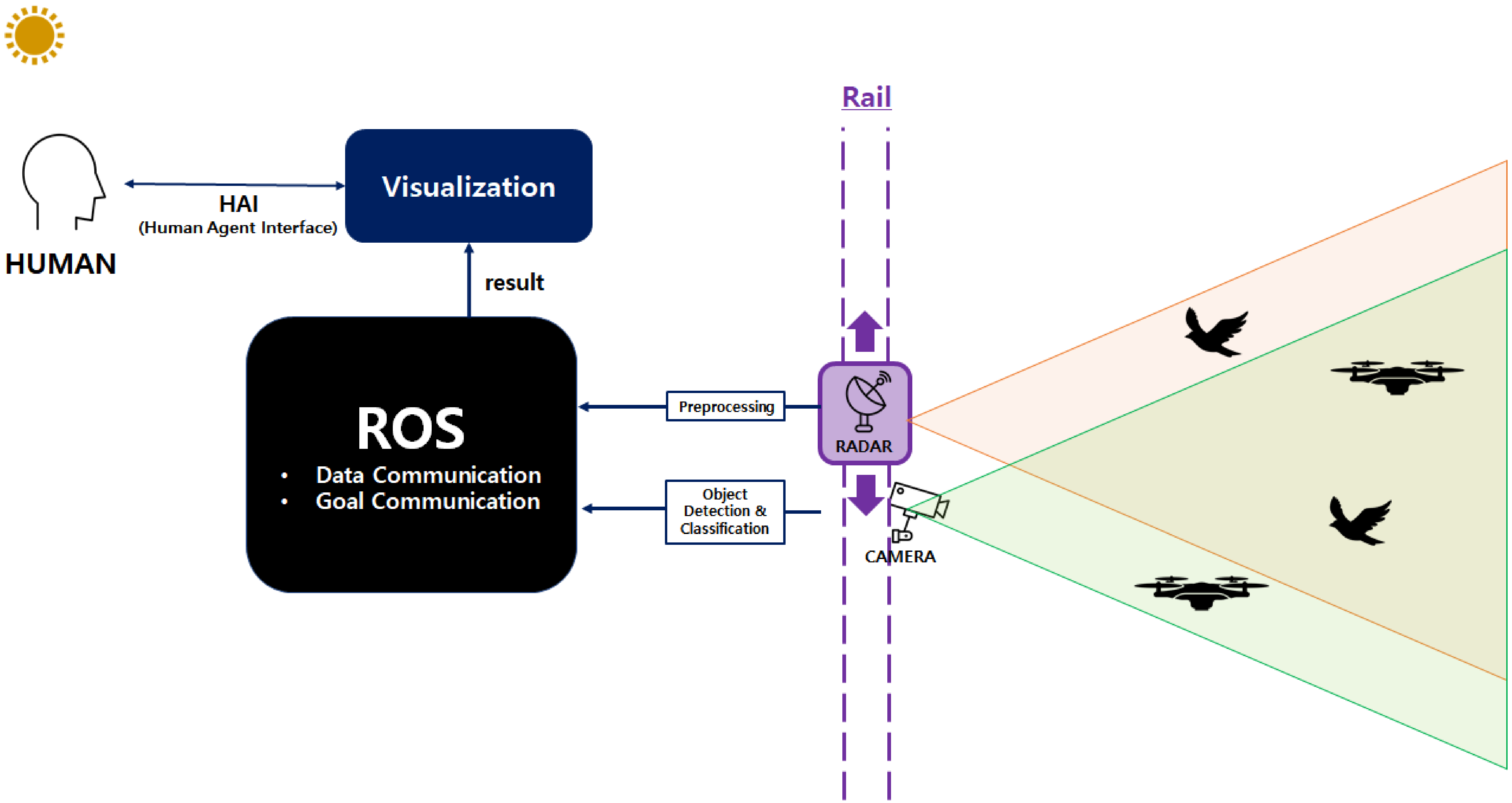
- Our project focuses on implementing **low cost UAV surveillance system.**
- We utilize **raspberry pi, rail, pi camera, and radar** that are affordable in price.
- We provide information of **high reliability** using camera and radar.
- And we also provide not only detect UAV but is able to provide distance information and direction of UAV's movement.

SAR Image



Drone Detection

Introduction



ROS

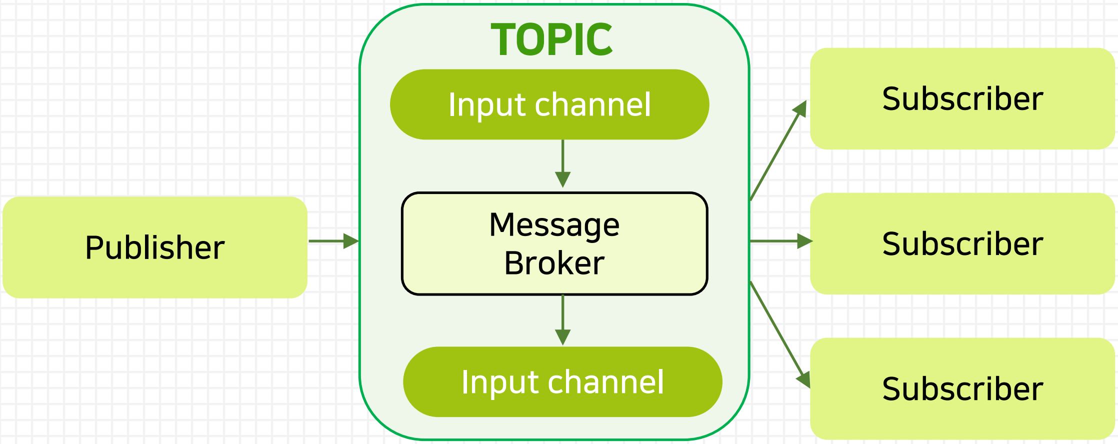
- What is the ROS
  - open-source, meta-operating system for your robot.
  - It provides the services, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management.
  - It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

- In our project, We uses this framework to create packages for each feature and implement them as ros nodes and topics.

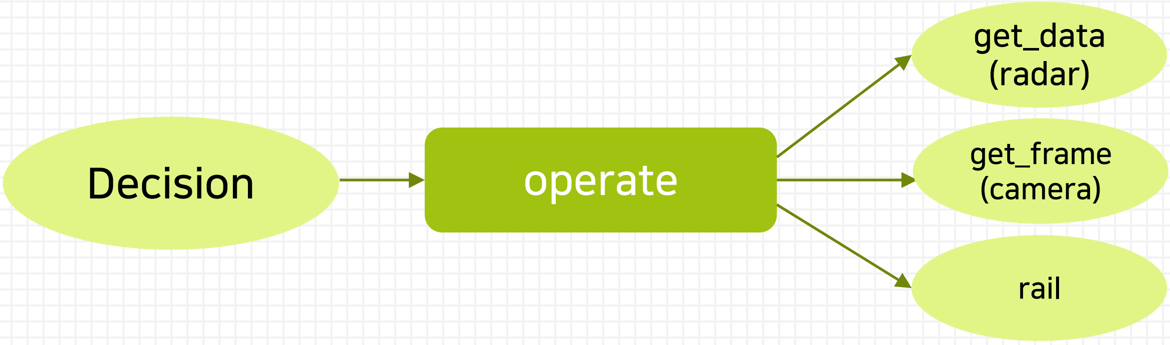
ROS

- **Publisher/Subscriber Model** is used in **ROS**
  - Enable an application to announce events to **multiple interested consumers asynchronously**, without coupling the senders to the receivers.
  - For example, Decision node give an order to multiple nodes in each package with one topic.

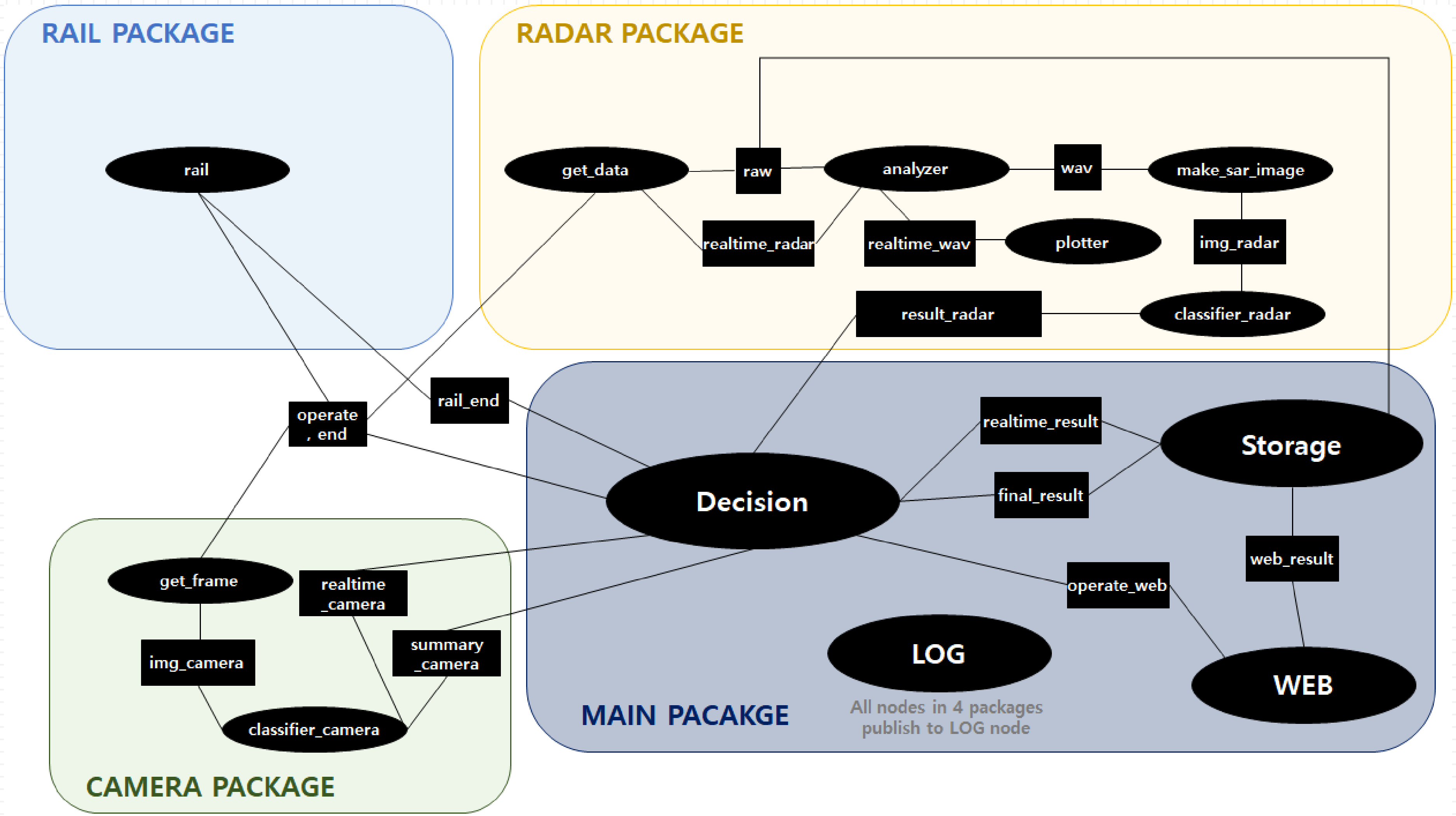
Publisher/Subscriber Model



Example in GUAVA Project



Architecture of our system

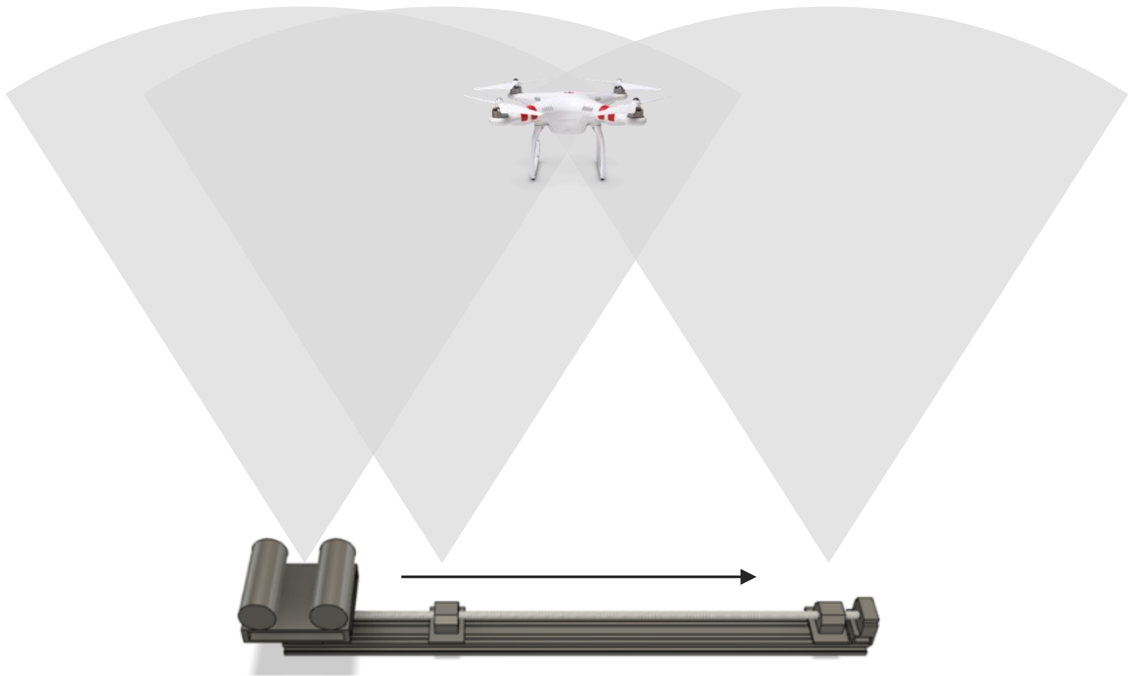


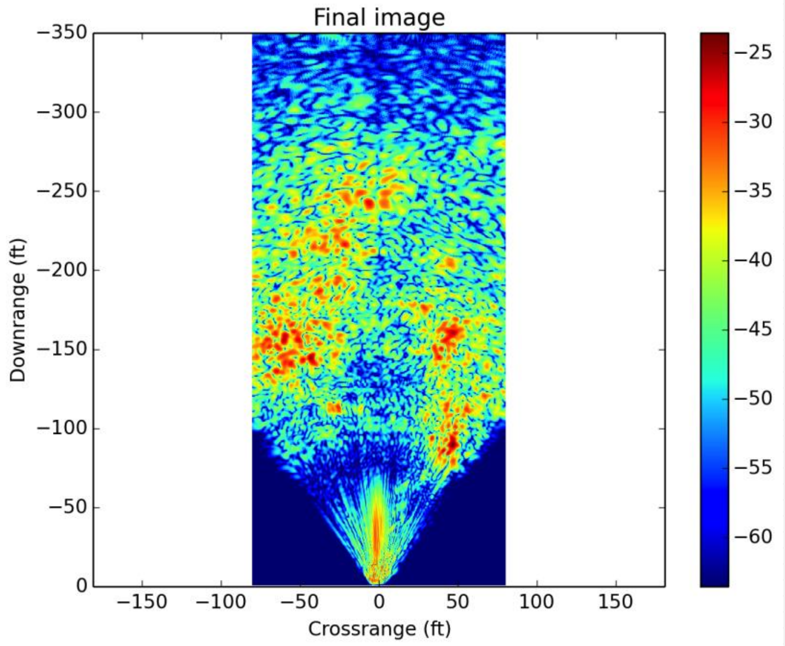
Rail

- For making SAR image, radar antennas need to move at a constant speed.

  → Our rail moves **4/7 inches/sec** during **70 seconds.** (moves 40 inches)

How SAR image is created



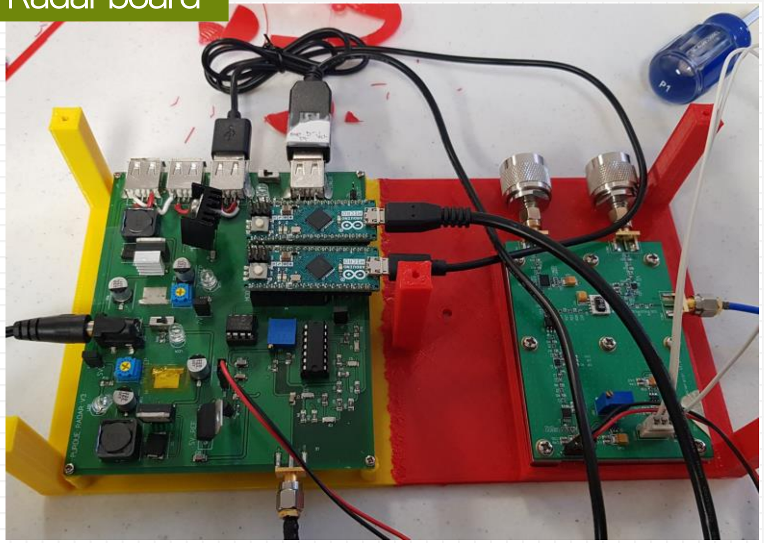


Goal of making rail is **low-cost and high-efficiency!**

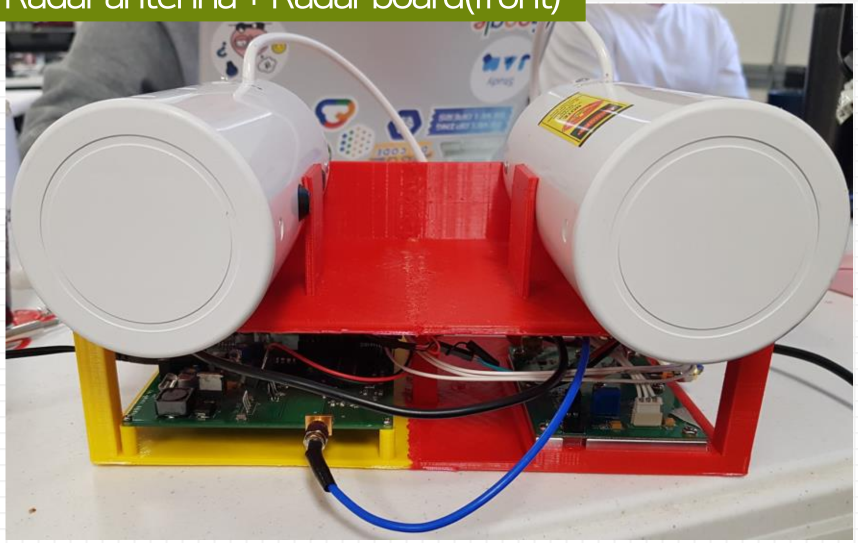
Rail

- Since the wire connecting the radar board and the radar antenna is short, the two must be placed on the rail together.
- So we need a plate to put them together. → made them using 3D printer
- If radar board and antenna are close or if two antennas are close, interference might occurs.
  → there should be enough distance between each of them.

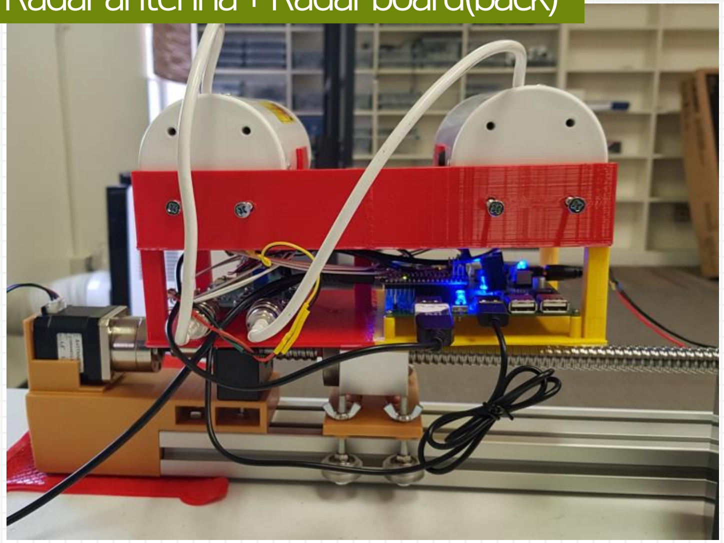
Radar board



Radar antenna + Radar board(front)



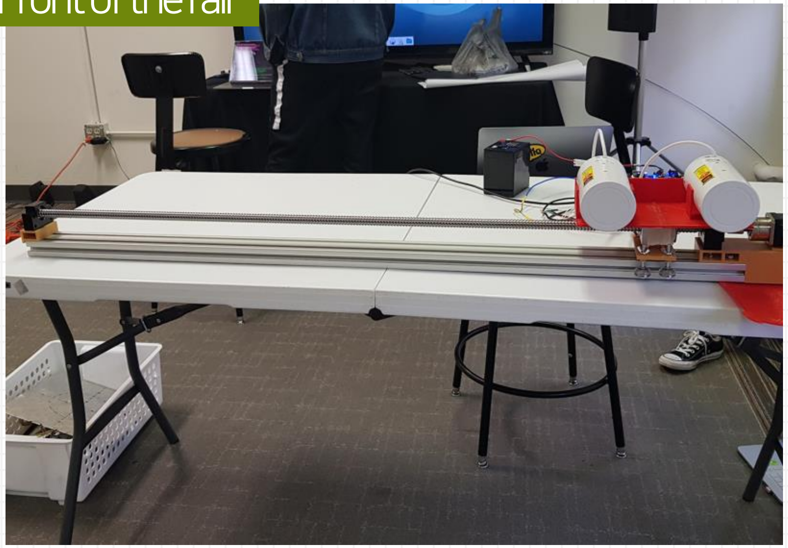
Radar antenna + Radar board(back)



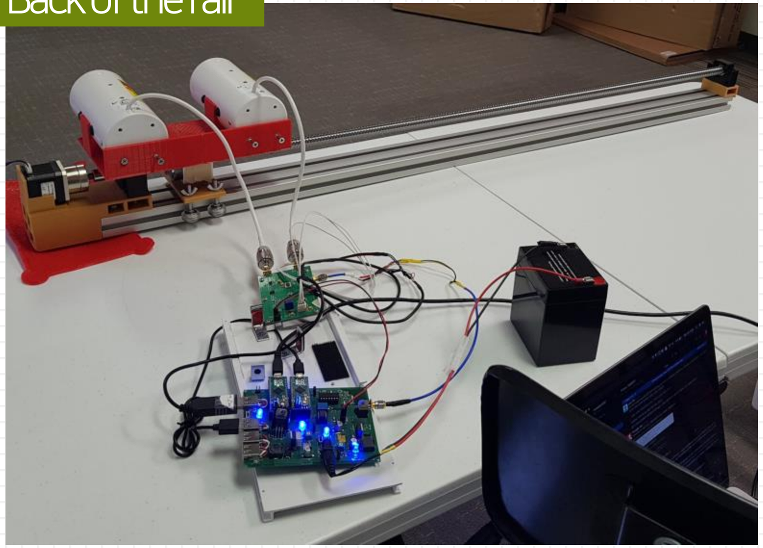
Rail

- Use stepper **motor, stepper motor driver, and arduino** to control the rail.
- Made connector using 3D printer to connecting each component.
- Connect the bearings and the ball screw using the supporter to make the ball screw **move linearly** without rotating.

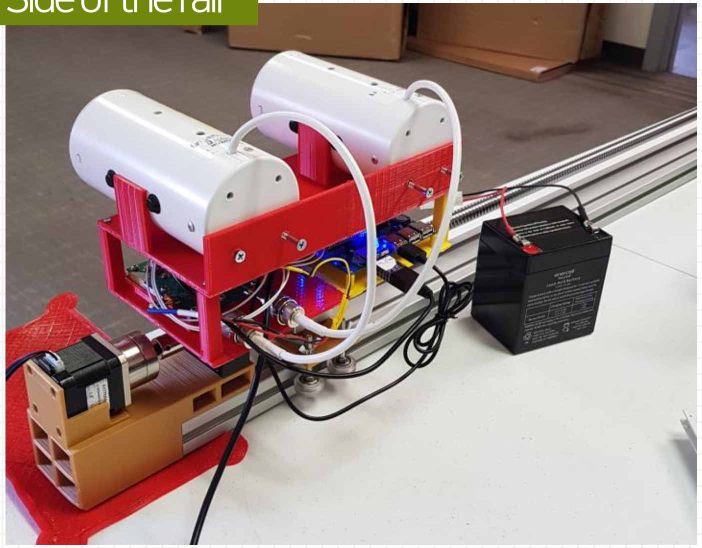
Front of the rail



Back of the rail



Side of the rail

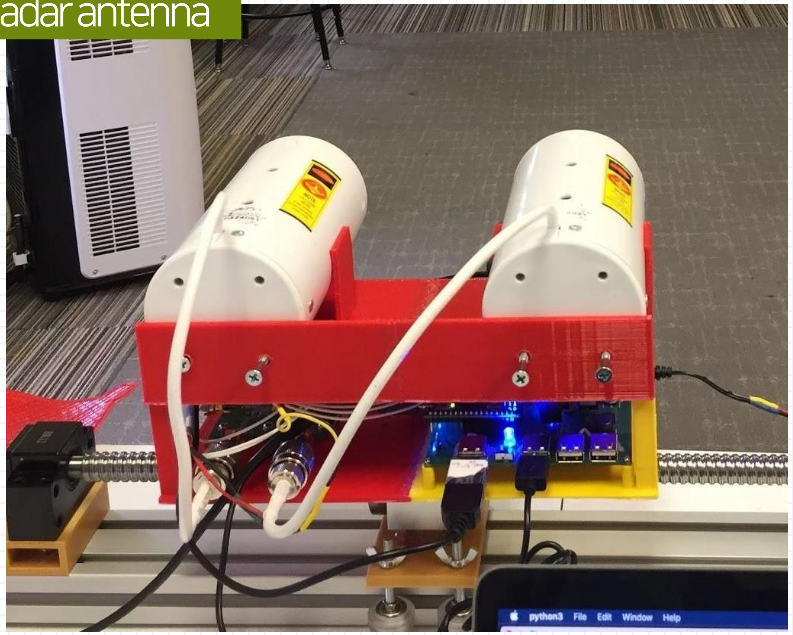


04 1. Why Radar?

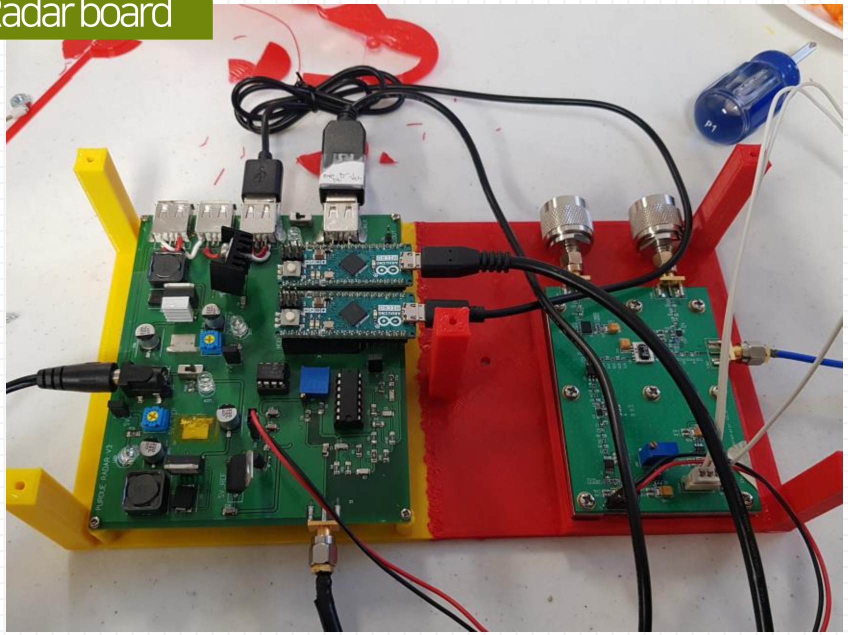
Radar

- Unlike camera, radar is capable during the night where light is insufficient.
- Radar has the advantage of obtaining distance between object and radar.
- Thus, rather than using a drone detecting system with camera, using radar will allow users to collect more information.
- In fact, if we use both camera and radar instead of one, it is expected to have enhanced detector reliability.

Radar antenna



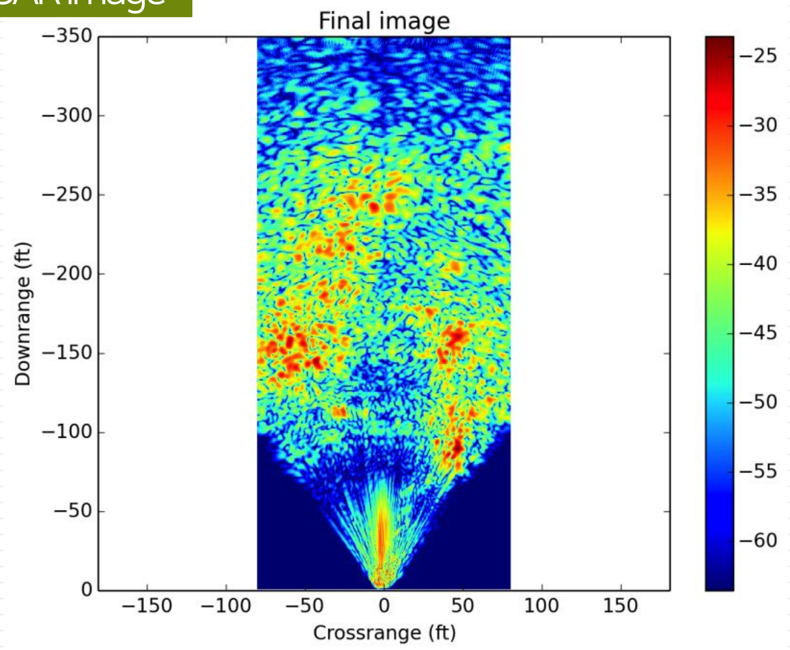
Radar board



Radar

- Synthetic-Aperture Radar is a form of radar that is used to create two-dimensional images or three-dimensional reconstructions of objects, such as landscapes.
- It needs signal processing to forms the SAR, and the process allows the creation of higher-resolution images than would otherwise be possible with a given physical antenna.
- SAR is typically mounted on a moving platform, so we built a rail to move radar.

SAR image



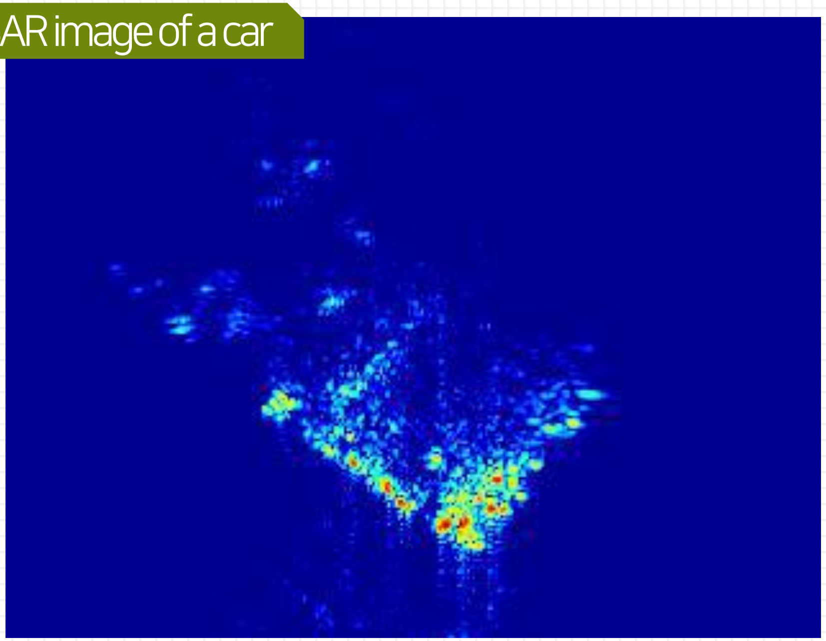
04 3. Desired system outcome

Radar

- Using a radar moving on the rail, collect SAR image.
- Detect drone using trained machine learning model with SAR image.

Camera image of a car

SAR image of a car



Radar

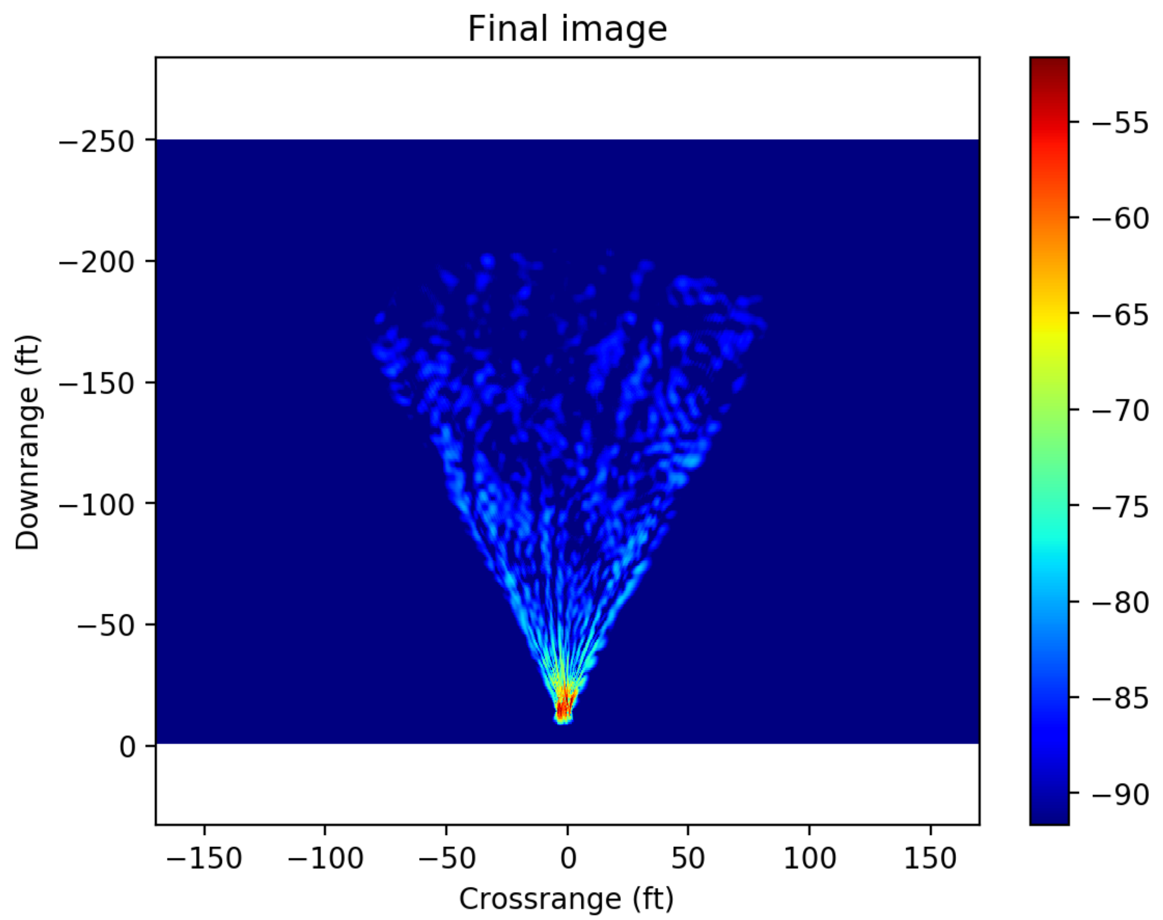
- Succeed at making SAR image, but image is not clear to determine whether it is drone or not.

- Lack of time to produce sufficient SAR image to feed machine learning model.

- So we can't decide what kind of neural network to use, and can't build model to classify.

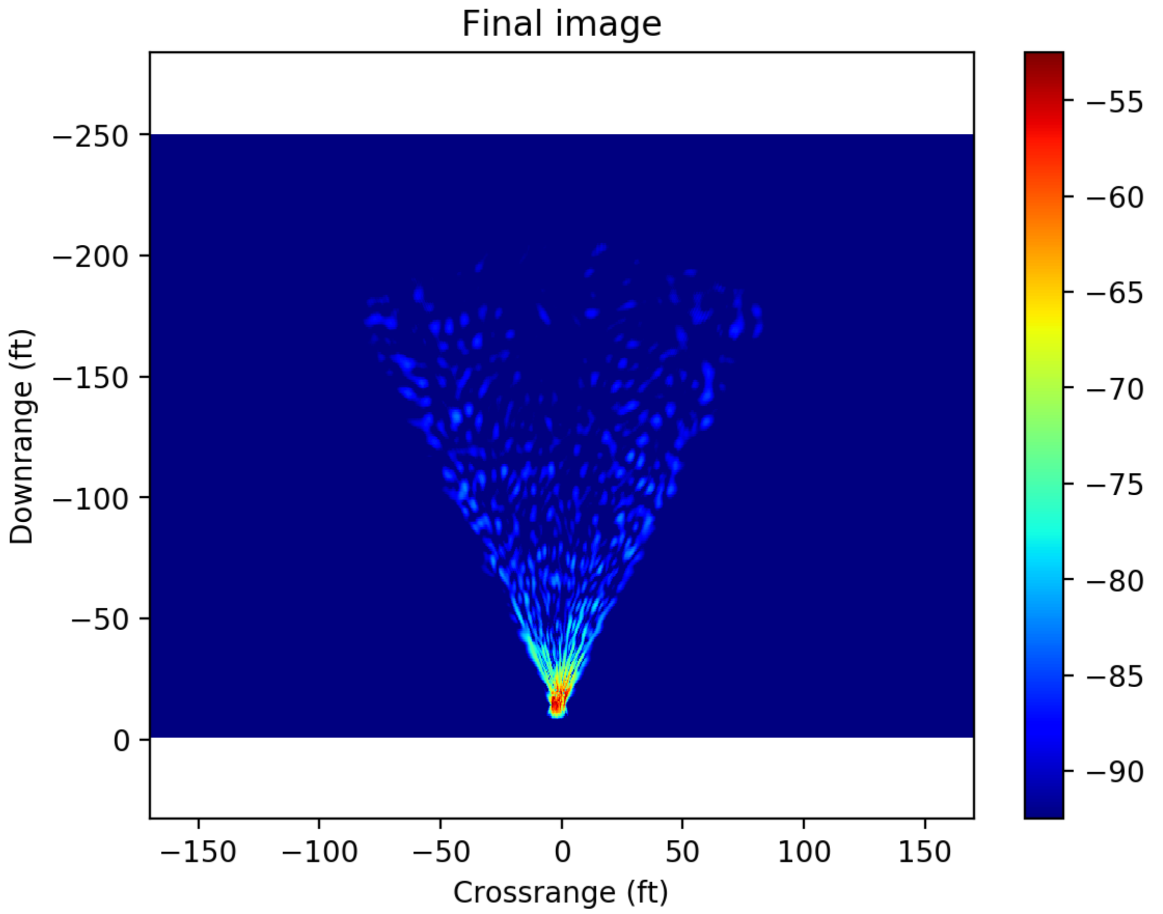
Radar

When a person is in front of the radar in 20ft



Radar

When a drone on a chair is in front of the radar in 20ft



Camera

- Used yolov3-tiny as object detection model since raspberry pi has limited computational resource.
- Trained the model using transfer learning method.
- Trained custom class and additional 3208 images on ImageNet pre-trained yolov3-tiny.conv15 model.

Raspberry Pi Camera

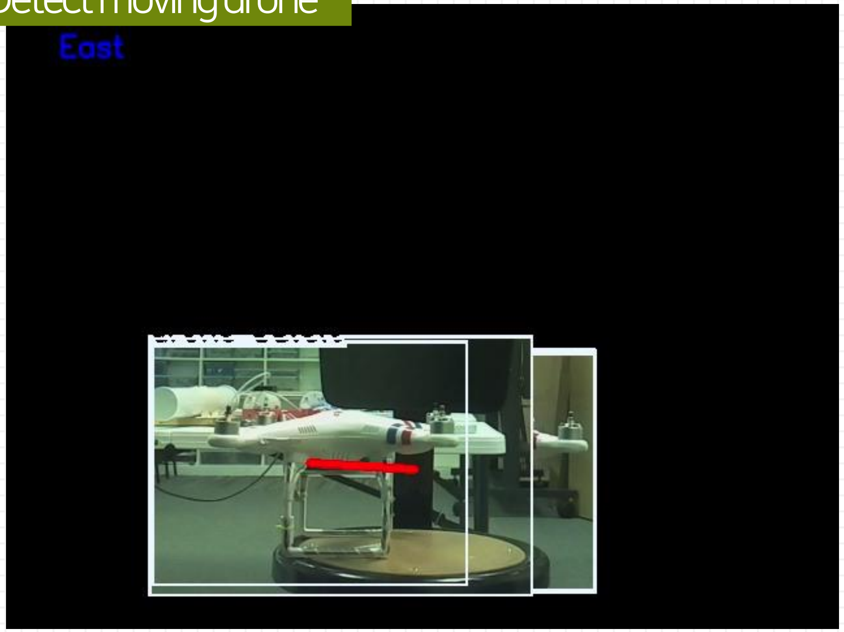
Example of detecting drone

Camera

1. Detect UAV from camera image and draw bounding boxes.
2. Generate summary image for one rail cycle.
   A. Collect images while rail is operating.
   B. Once rail finishes cycle, iterates over frames and merges images into one that have detected UAV.
   C. By comparing first and last frame, figure out to which direction UAV moved.

Detect not moving drone

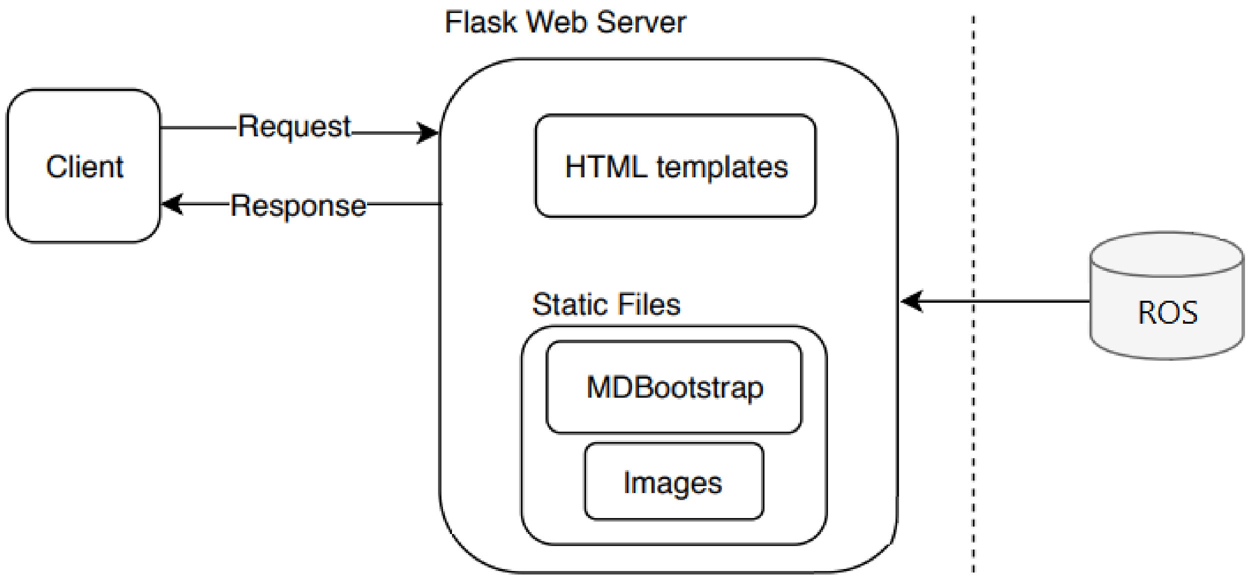
Detect moving drone



Web

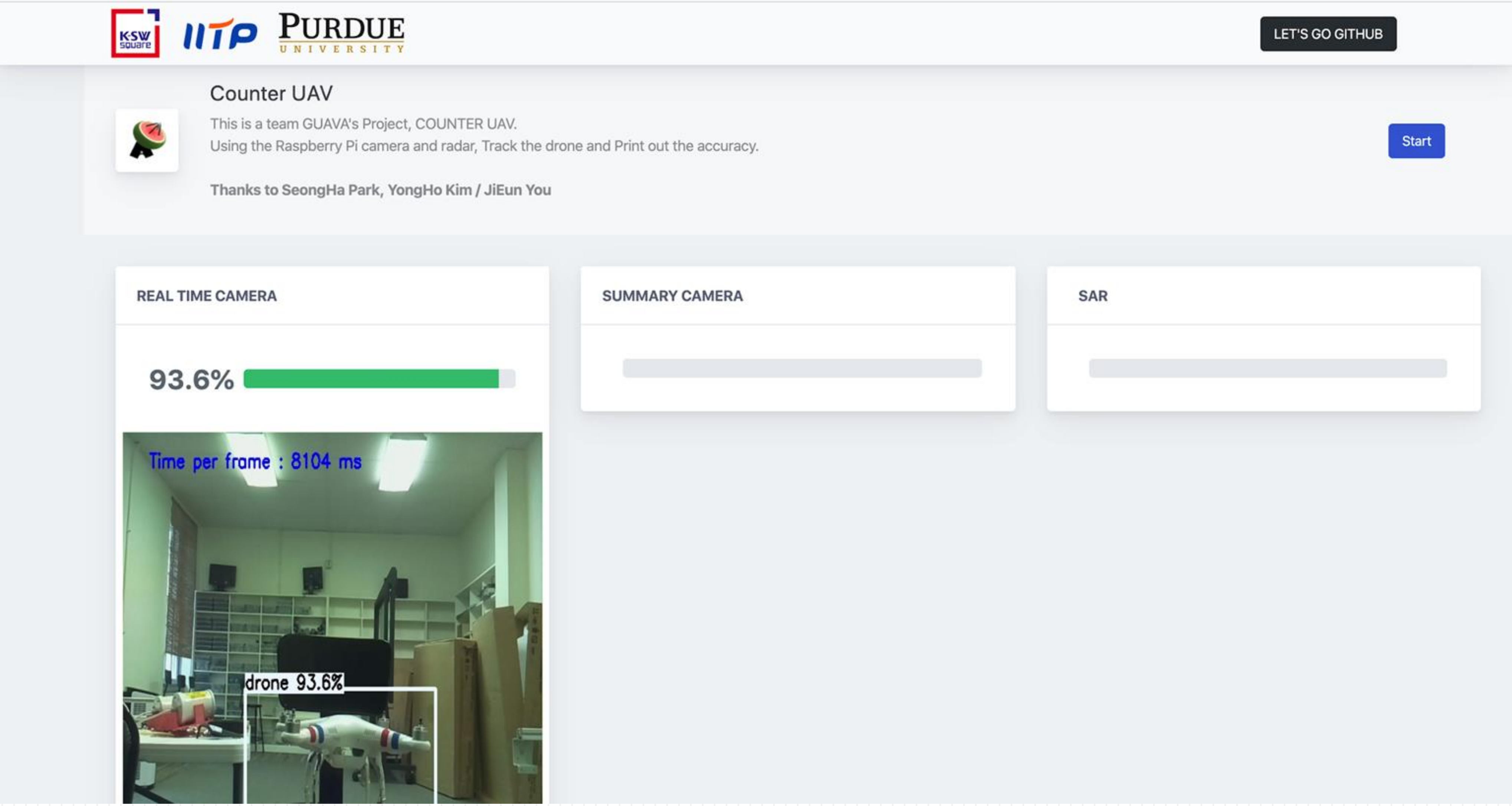
- The web was created to visualize all data outcomes.
- (SAR, Real-time Camera, Summary Camera) within the ROS.

File structure diagram



Result

Our Flask web server



Result

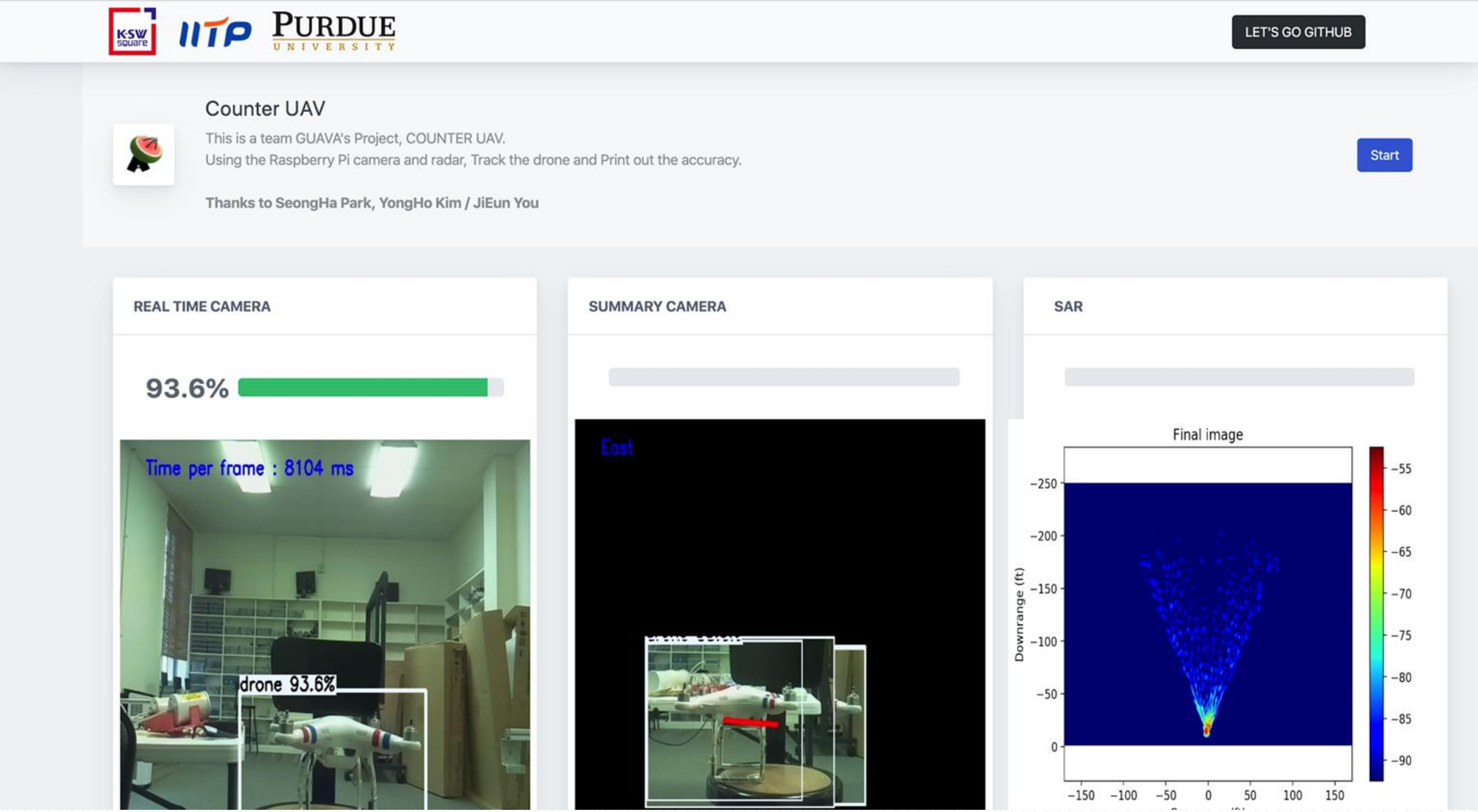
- ROS
  - **Design** overall system structure and **connect** our whole system using **ROS**.
- Rail
  - Success to **make rail**, move radar with **constant speed**.
- Radar
  - Check radar's frequency range and found out radar's **sampling rate is not static**.
  - Receive raw data from antennas and **make SAR image**.
- Camera
  - Detect drones with **YOLO-v3 tiny** and send detected result to decision node.
- Drone/Web
  - Connecting to the path that we wanted via a received message and **displaying the image**.

Result

- ROS
  - It was **unfamiliar framework** and there are few references in Korean.
- Rail
  - Hard to control stepper motor with **fast speed** → need another stepper motor.
- Radar
  - We changed antenna **to longer one to shorter one** → **gain** of radar signal was significantly **lower.**
  - We didn't succeed at making perfect SAR image, result SAR image until now is **unidentifiable even in human's eye.**
  - Our SAR imaging code was referenced from MIT. Since **lack of information of parameters** used in code, We spend lots of time to figure out role of each parameters and there were many trials and errors.
- Camera
  - Hard to **set the environment** and it takes lots of time.
- Drone/Web
  - Hard to find place to **practice flying drones**.

08 1. Expected final result
Future Works

Our Flask web server



Future Works

- Use another stepper motor to make radar moves faster.
- Connect arduino and raspberry pi, and import with ROS.

START

Rail

Radar

- Make better SAR image.
- Produce enough data for training machine learning model.
- Find appropriate neural network to classify SAR image.

Future Works

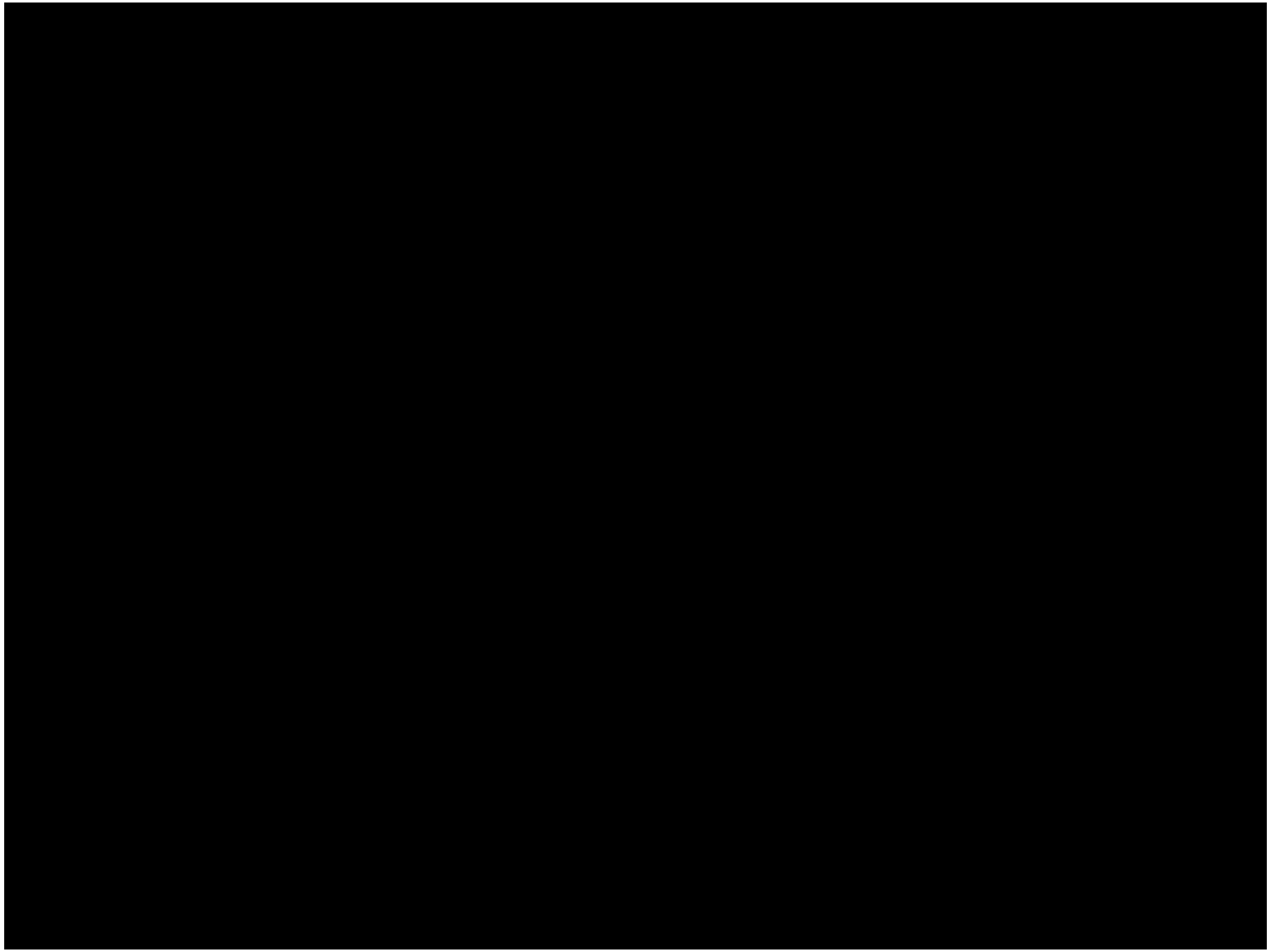
- Enhance performance of detection by training the model with data augmentation and images extracted directly from raspberry pi.

Camera

Drone/web

- Start the ROS program when the start button is pressed.
- Automatically display result image.

Demo



Thank you.

any question?