

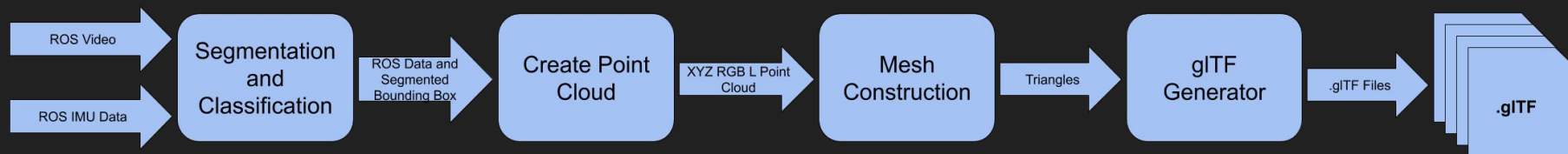
# DroneMOM

*Drone Model Output Machine*

Taylor Nelms, John Marcao, Eric Micallef

# Overview

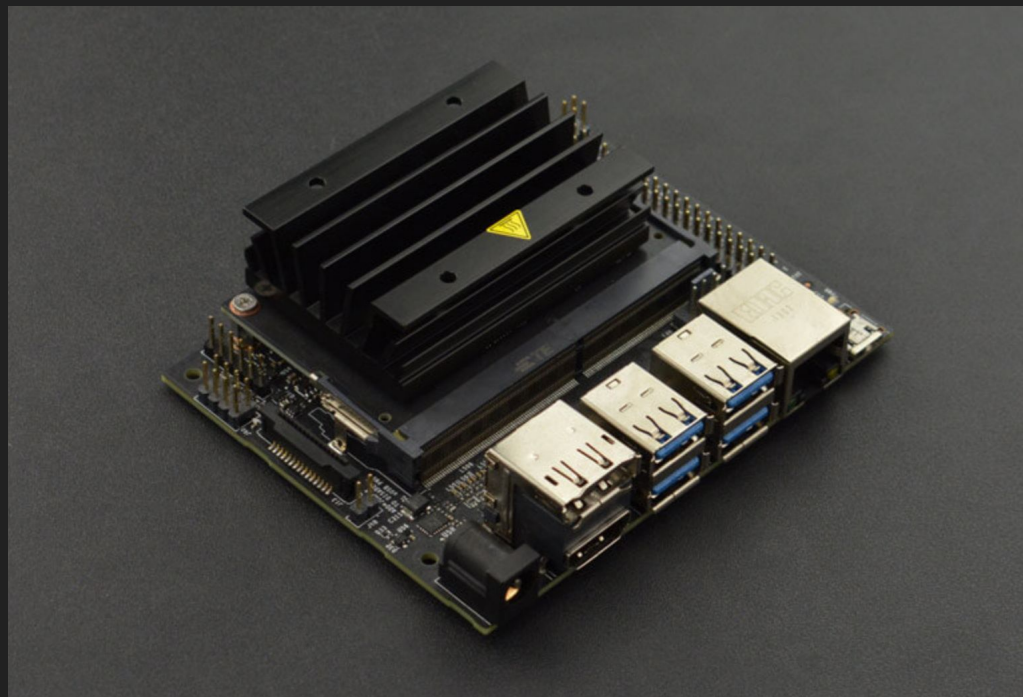
Create a tool that converts incoming visual and inertial measurement unit data from an Unmanned Aerial Vehicle and use it to generate glTF models of classified objects in the scene.



# Hardware

Nvidia Jetson Nano

Powerful chip but has its  
limitations!



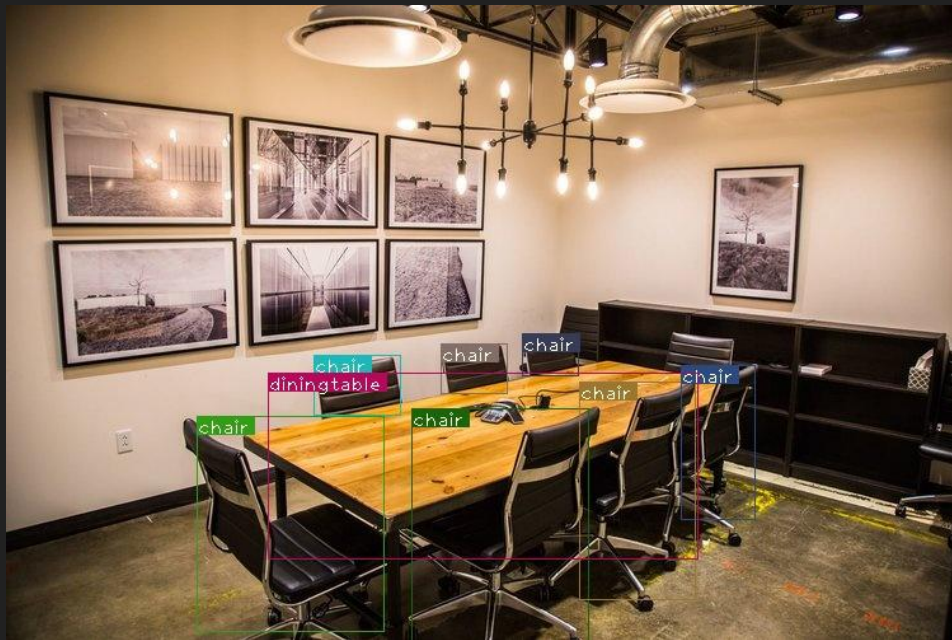
# What is ROS

- ROS stands for Robot Operating System
- Publisher Subscriber message passing middleware
- Rosbags allow easy playback for refining algorithms
- Rosbags allow us all to work independently without dependencies
- Unified interfaces
- Open source

# Classification and Segmentation

- Accomplished
  - ROS pipeline architecture implemented
  - Integrated testing in ROS for images and video
  - Some object detection and segmentation.
- Challenges
  - Have seen “This is a known issue and will be fixed in our next release.” too many times...
  - Have seen “This issue has been fixed in current release.” (Current release not supported on jetson nano .... )
  - Getting about 5 FPS without tensorRT
- Next Week
  - Look into Mask R-CNN ( Thanks shadow team! )
  - Need to resize incoming images. ( should make it more accurate and faster )
  - Convert pb to UFF format
  - Figure out how to load a UFF file with tensorRT

# Classification + Detection



```
Objects Detected:      bird
-----
person.jpg             predicted in  0.522 seconds
Objects Detected:      person dog horse
-----
messi.jpg              predicted in  0.516 seconds
Objects Detected:      person person person sports ball
-----
dog.jpg                predicted in  0.512 seconds
Objects Detected:      bicycle truck dog
-----
herd_of_horses.jpg     predicted in  0.512 seconds
Objects Detected:      horse horse horse horse
-----
office.jpg             predicted in  0.528 seconds
Objects Detected:      chair chair chair chair chair chair diningtable
-----
img3.jpg               predicted in  0.544 seconds
Objects Detected:      car car car car car car car car truck traffic light
-----
```

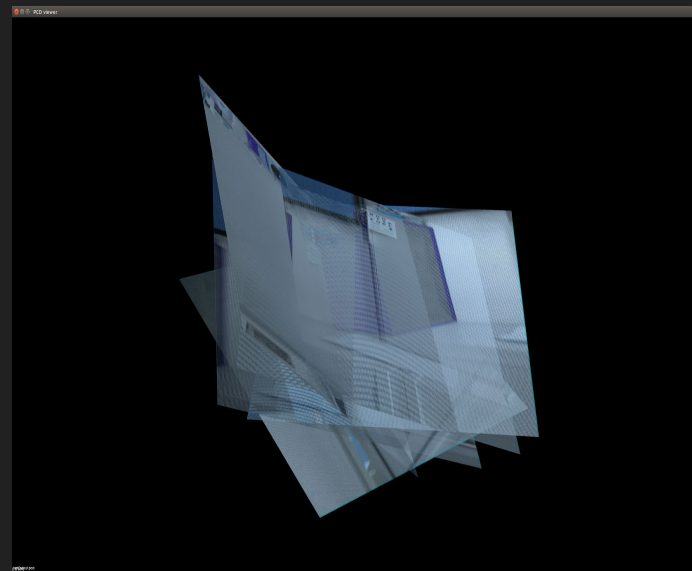
## SUMMARY

```
-----
Task                  : Time Taken (in seconds)
```

```
Reading addresses     : 0.001
Loading batch         : 3.607
Detection (8 images)  : 4.296
Output Processing     : 0.000
Drawing Boxes         : 0.172
Average time_per_img  : 1.009
-----
```

# Point Cloud Generation

- Accomplished
  - Camera transformations for generating image position
  - Integrated point cloud generation node into ROS pipeline
  - Found and integrated relevant libraries for storing and manipulating data
- Challenges
  - Generating accurate pose data for camera from IMU sensor data
  - Estimating depth data from multiple images
- Next Week
  - Generate depth data points for FLANN-matched feature points
  - Allow for building of point cloud incrementally (avoiding duplicate points)



# GLTF Mesh Construction

- Accomplished
  - Mesh construction from Test ROS data
  - Integration with PCL libraries
  - Output to VTK File Format
- Challenges
  - Too many data points
  - Too slow (2 FPS)
- Next Week
  - Output to GLTF File Format
  - Separate data reading and file construction
  - Downsample incoming data





End