Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 5: 3D Occluded Object Detection System

Bi-Weekly Update 2

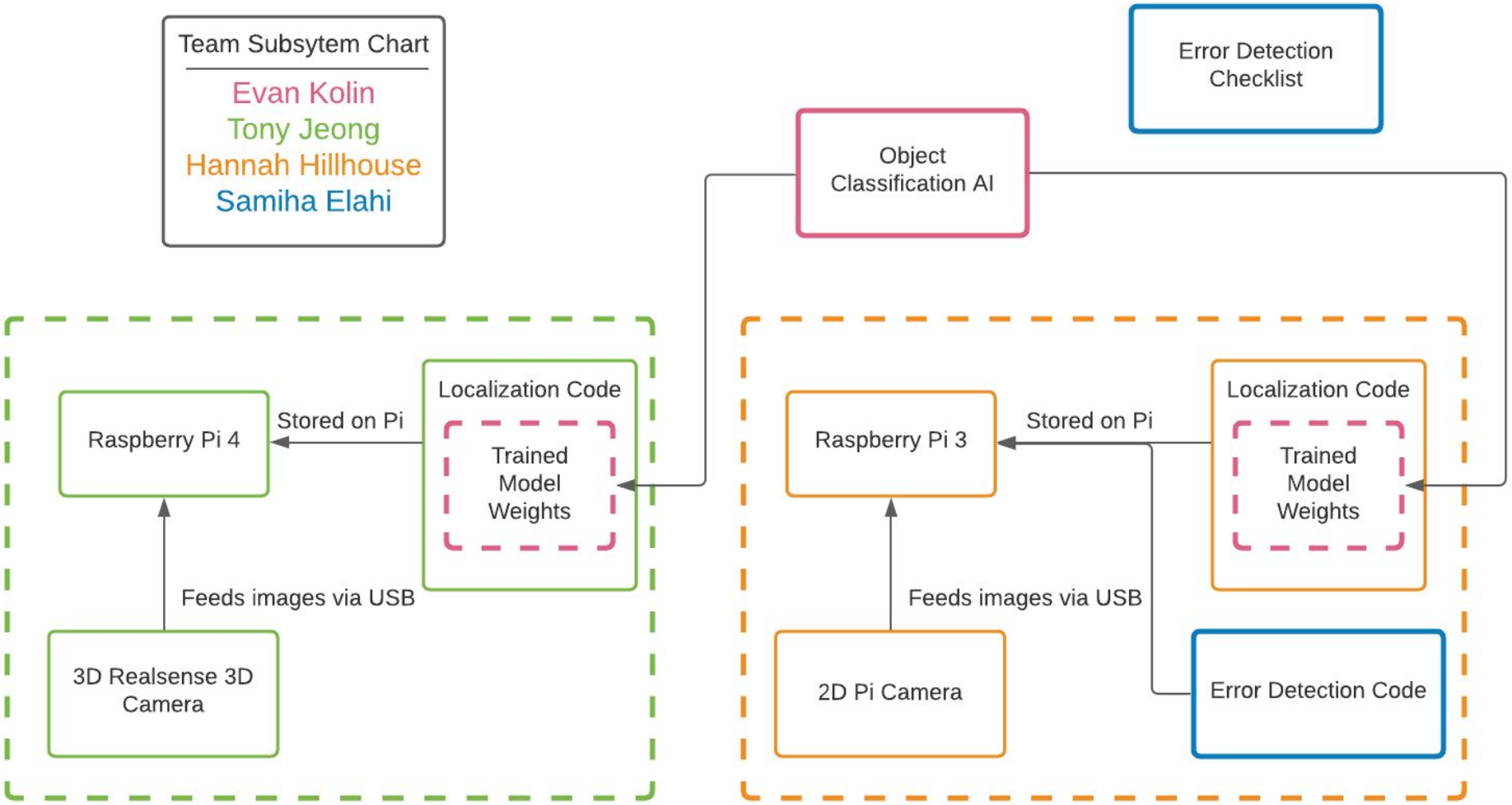
Team members:
Evan Kolin
Tony Jeong
Hannah Hillhouse
Samiha Elahi

Sponsor:
Kevin Nowka

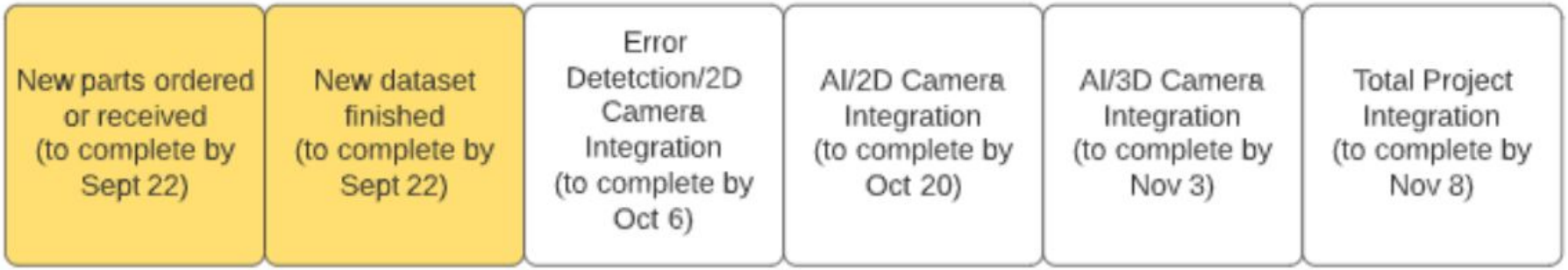
Project Summary

Our team has the task of classifying and localizing objects within a bin of multiple parts and moving preselected parts to a new empty bin. We have two cameras, one to look into the bin with all the parts, and one to look at the bin with already picked parts. Both cameras are attached to a raspberry pi that holds the object classification model and the object localization code.

Project/Subsystem Overview



Project Timeline

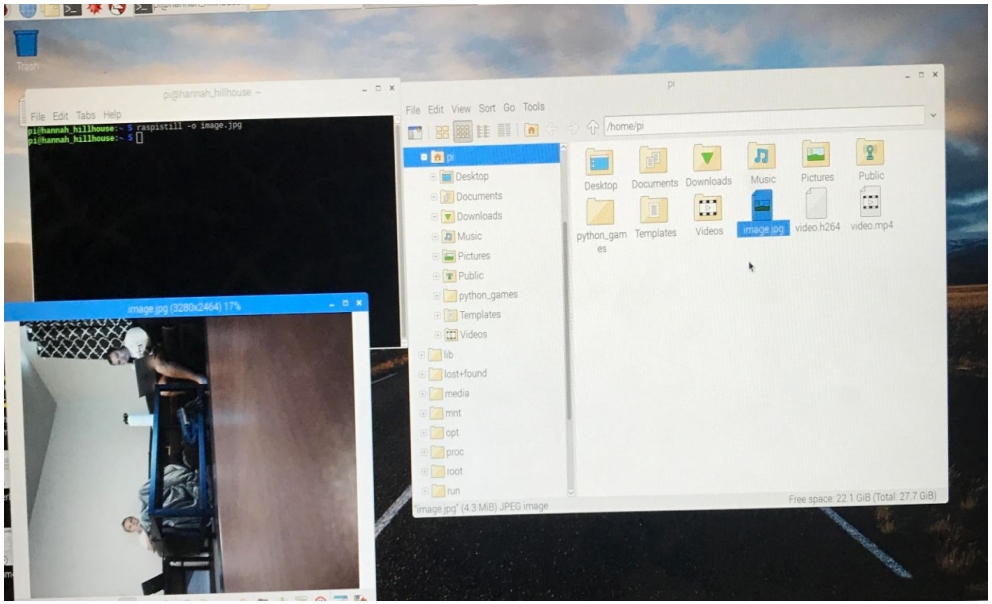


2D Camera Subsystem

Accomplished since last update 12 hrs	Ongoing progress/problems
- Raspberry Pi and Camera Setup and working - Python code moved to google collab	- correcting a few bugs with google collab - moving python code to raspberry pi and integrating error detection subsystem and AI Subsystem - start validating

2D Camera Subsystem

- I have to transfer code from google collab onto raspberry pi
- I have to train a new data set with new parts
- Validate new trained dataset
- Start integrating with AI trained dataset and integrate with Error detection during lab this week



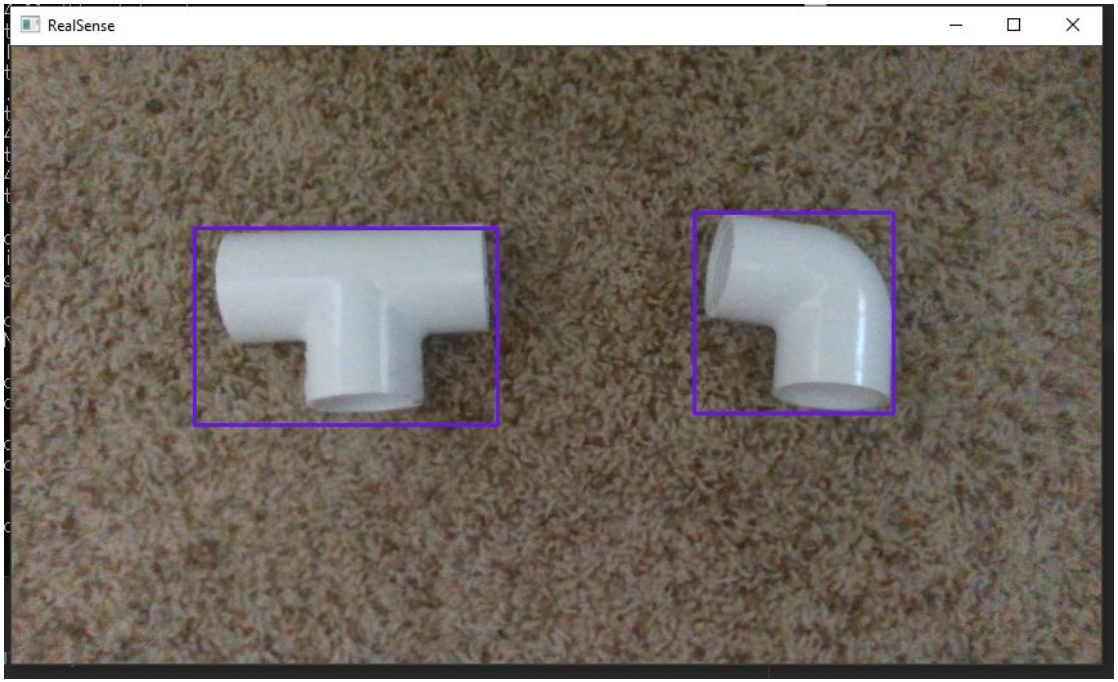

```
[ ]  #import cap as cap
     import cv2
     from google.colab.patches import cv2_imshow

     thres = 0.45 #threshold to detect object
     #reading in the coco file for object names
     classNames = []
     classFile = 'coco.names'

     #extracting names of the file using loop

     with open(classFile, 'rt') as f:
       classNames = f.read().rstrip('\n').split('\n')
```

3D Camera Subsystem

Accomplished since last update 7 hrs	Ongoing progress/problems
- Tested on new material.	- working on Pi. - test with new algorithm model that Evan worked on.

3D Camera Subsystem

- It works, but not perfect yet because I am using pre-trained model by intel. I have to see how it works with Evan's trained model.



Object Classification Subsystem

Accomplished since last update 20 hrs	Ongoing progress/problems
• export model into ".pb" to allow integration and testing with camera subsystems • Adjusted model for new dataset	• Test functionality with 2D subsystem in following week • Rebuild image database to pull labels

Machine Learning Subsystem

- Previously thought I had a predictions error, but I actually had incorrect image labels. Exported a trained model to be used with 2D subsystem and I expect that it will work.
- Redesign image database so that it is easier to pull images and split image data efficiently between training, test, and validation.

```
configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weightsPath = 'frozen_inference_graph.pb'
```

```
Actual:
[0 1 0 1 0 1 1 1 0 1 1 1 1 0 0 0 0 1 0 0 1 1 1 1 0 0 1 1 0 1 0 0 1 1 1 1 1
 1 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 1 1
 1 0 0 1 1 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1]
Predictions:
[0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 0 1 0 1 1 1 1 0 1
 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 0
 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 1 0 0]
```

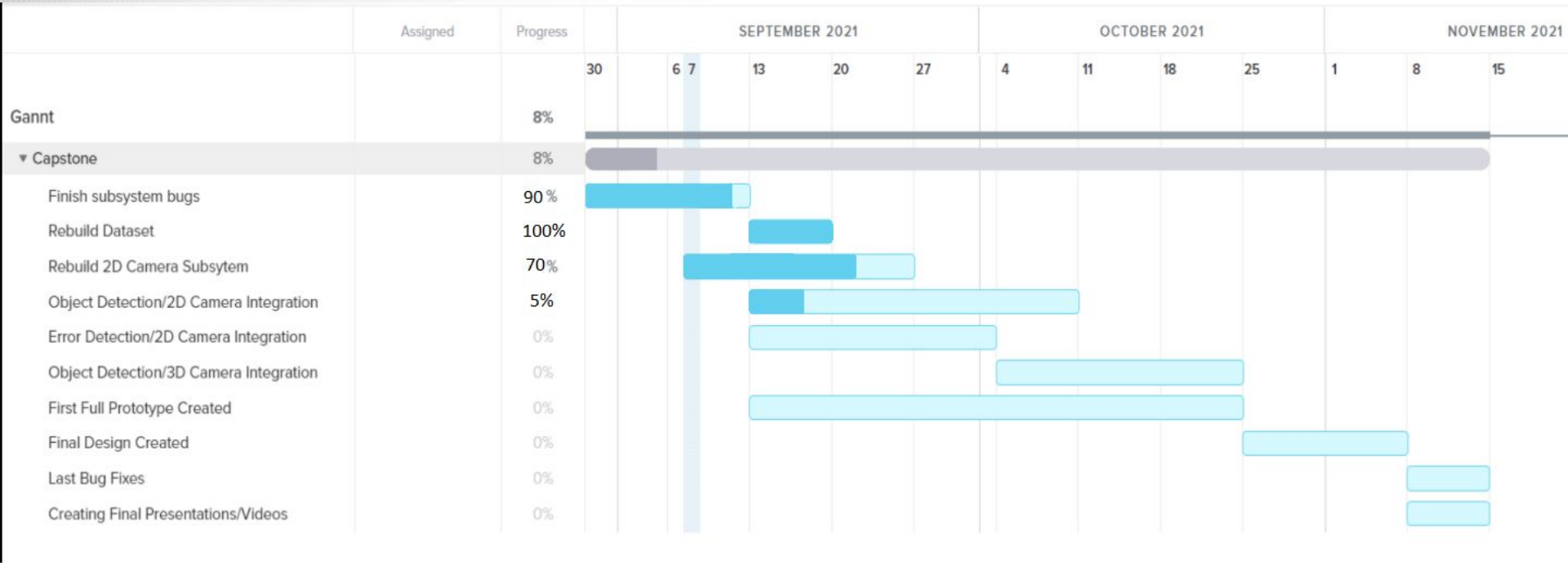
Error Detection & Handling Subsystem

Accomplished since last update 10 hrs	Ongoing progress/problems
- Implemented a way to handle three different type of errors - Implemented a txt file to log the errors	- Implement a method to handle errors when the system thinks the order requirement has been met but actually it has not. - Integrate this subsystem with the 2D camera subsystem asap.

Error Detection & Handling Subsystem

- I have to come up with a way to better handle the parts/objects that are occluded
- Run the code with the new dataset to check if we run into a new type of error
- Start integrating with the 2D camera subsystem

Execution Plan



Thank You

Questions?