Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 5: 3D Occluded Object Detection System

Bi-Weekly Update 3

Team members:
Evan Kolin
Tony Jeong
Hannah Hillhouse
Samiha Elahi

Sponsor:
Kevin Nowka

Project Summary

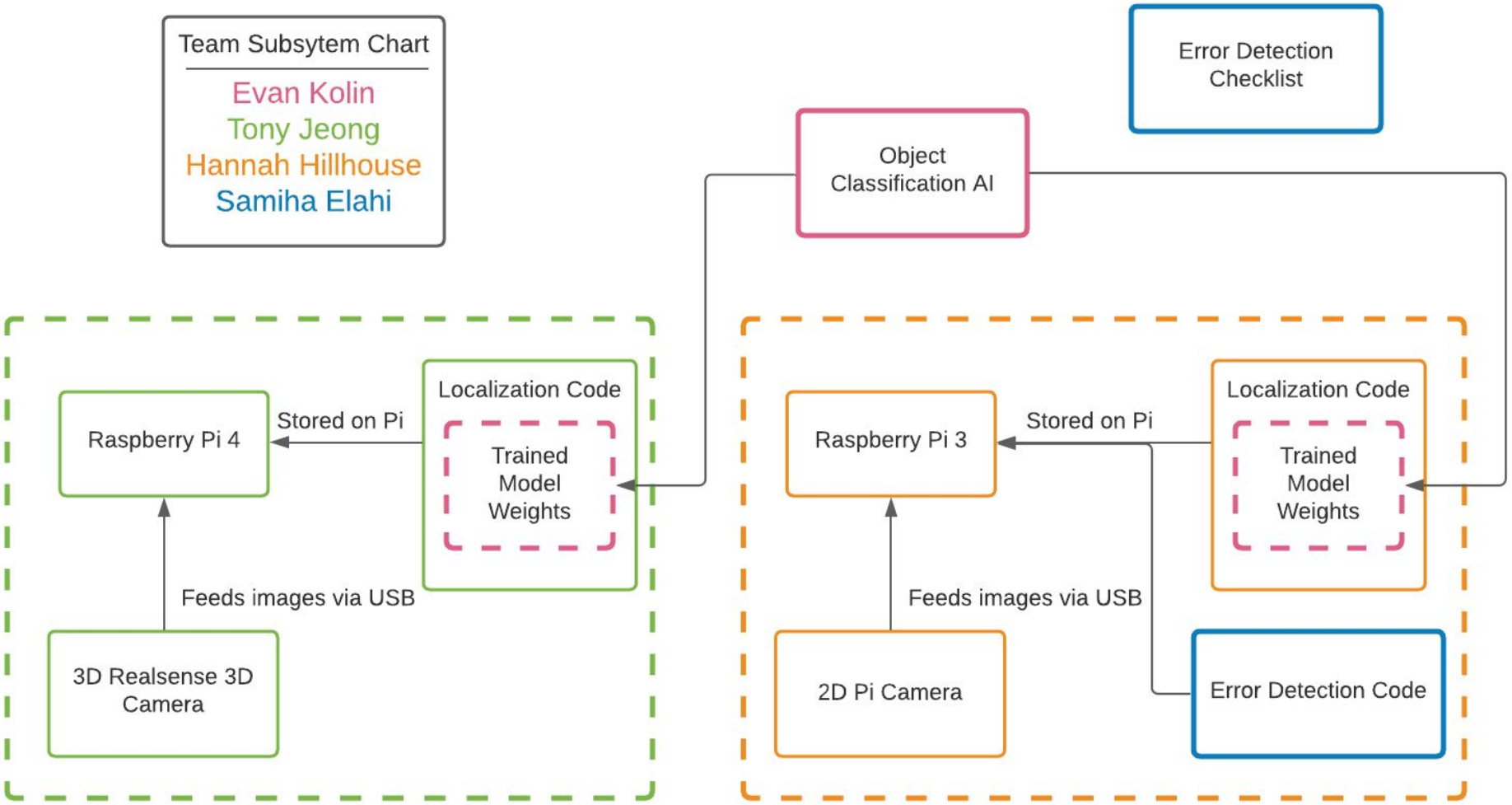
Problem:

Manufacturers are using people to pick and place parts into bins for further use. This is not only costly to the company but also not time efficient and results in a very cluttered workspace.

Solution:

We will create an object detection system in which will localize and classify objects within the parts bins and pick and place them into their corresponding bins for further use. 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

New parts ordered or received (completed Sept 22)	New dataset finished (completed Sept 22)	Error Detection/2D Camera Integration (to complete by Oct 18)	AI/2D Camera Integration (to complete by Oct 25)	AI/3D Camera Integration (to complete by Nov 5)	Total Project Integration (to complete by Nov 10)

2D Camera Subsystem

Accomplished since last update 30+ hrs	Ongoing progress/problems
- discovered google collab will not work with the opencv functions needed so I transferred code to anaconda and worked on debugging - started integration with error detection	- Get python code working and debug the opencv(cv2) errors and complete subsystem. - integrate fully with the error detection

2D Camera Subsystem

- Started integration on google collab with Error Detection and realized that opencv functions were not included on google collab that is needed for 2D object detection to run.
- Currently debugging the opencv issue and working on a solution to get it up and running so I can finish integration with error detection

```
In [1]: runfile('C:/Users/hanna/OneDrive/Desktop/ECEN-404/SeniorDesign.py', wdir='C:/Users/hanna/
OneDrive/Desktop/ECEN-404')
Traceback (most recent call last):

  File "C:\Users\hanna\OneDrive\Desktop\ECEN-404\SeniorDesign.py", line 11, in <module>
    import cv2

ModuleNotFoundError: No module named 'cv2'
```

3D Camera Subsystem

Accomplished since last update 8 hrs	Ongoing progress/problems
-Integrated my part to the raspberry pi 4 that fully functions my code on the intel 3D camera.	- Integrating Evan's trained model on my code.

3D Camera Subsystem

```
Traceback (most recent call last):
  File "objectdetect.py", line 58, in <module>
    feed_dict={image_tensor: image_expanded})
  File "C:₩Users₩ehddn₩AppData₩Roaming₩Python₩Python36₩site-packages₩tensorflow₩python₩client₩session.py", line 968, in run
    run_metadata_ptr)
  File "C:₩Users₩ehddn₩AppData₩Roaming₩Python₩Python36₩site-packages₩tensorflow₩python₩client₩session.py", line 1134, in _run
    e.args[0])
TypeError: Cannot interpret feed_dict key as Tensor: The name 'image_tensor:0' refers to a Tensor which does not exist. The operation, 'image_tensor', does
not exist in the graph.
```

We tried to integrate Evan’s trained model into my code, but it is making error.

Object Classification Subsystem

Accomplished since last update 12 hrs	Ongoing progress/problems
• Exported new model • Transferred trained model to a frozen inference graph • Attached new frozen inference graph to 3D subsystem	• Blocked with 2D integration until it's functional • Almost done with 3D integration, need to solve input error

Object Classification Subsystem

- Spent majority of my time looking at other subsystems code to learn how to best integrate.
- Can’t integrate with 2D subsystem until 2D subsystem works
- 3D integration subsystem expects an input tensor variable with name “image_tensor”, but my models input doesn’t have one.

```
----------------------------------------------------------------
Frozen model inputs:
[<tf.Tensor 'x:0' shape=(None, 28, 28, 1) dtype=float32>]
Frozen model outputs:
[<tf.Tensor 'Identity:0' shape=(None, 3) dtype=float32>]
```

Error Detection & Handling Subsystem

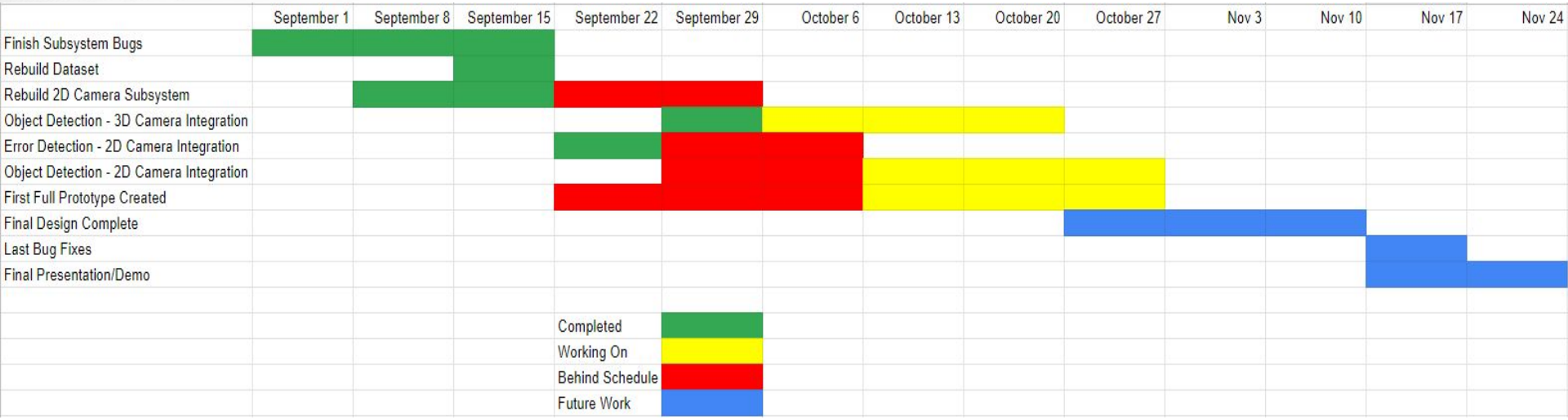
Accomplished since last update - 6 hrs	Ongoing progress/problems
- Improved the code so now the user no longer needs to manually input the number of parts needed every time - Tested the code on a set of images that are bad to check if they are easily fixable	- Error detection code has been done and we started integration with the 2D subsystem but we can just test it until we solve the opencv issue

Error Detection & Handling Subsystem

- Try helping Hannah to solve the opencv issue so that we can finish integration asap.

- Run the error detection code with the new set of data to check if we run into any new type of errors.

Execution Plan



Thank You

Questions?