Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 5: 3D Occluded Object Detection System

Bi-Weekly Update 5

Team members:
Evan Kolin
Tony Jeong
Hannah Hillhouse
Samiha Elahi

Sponsor:
Kevin Nowka

Project Summary

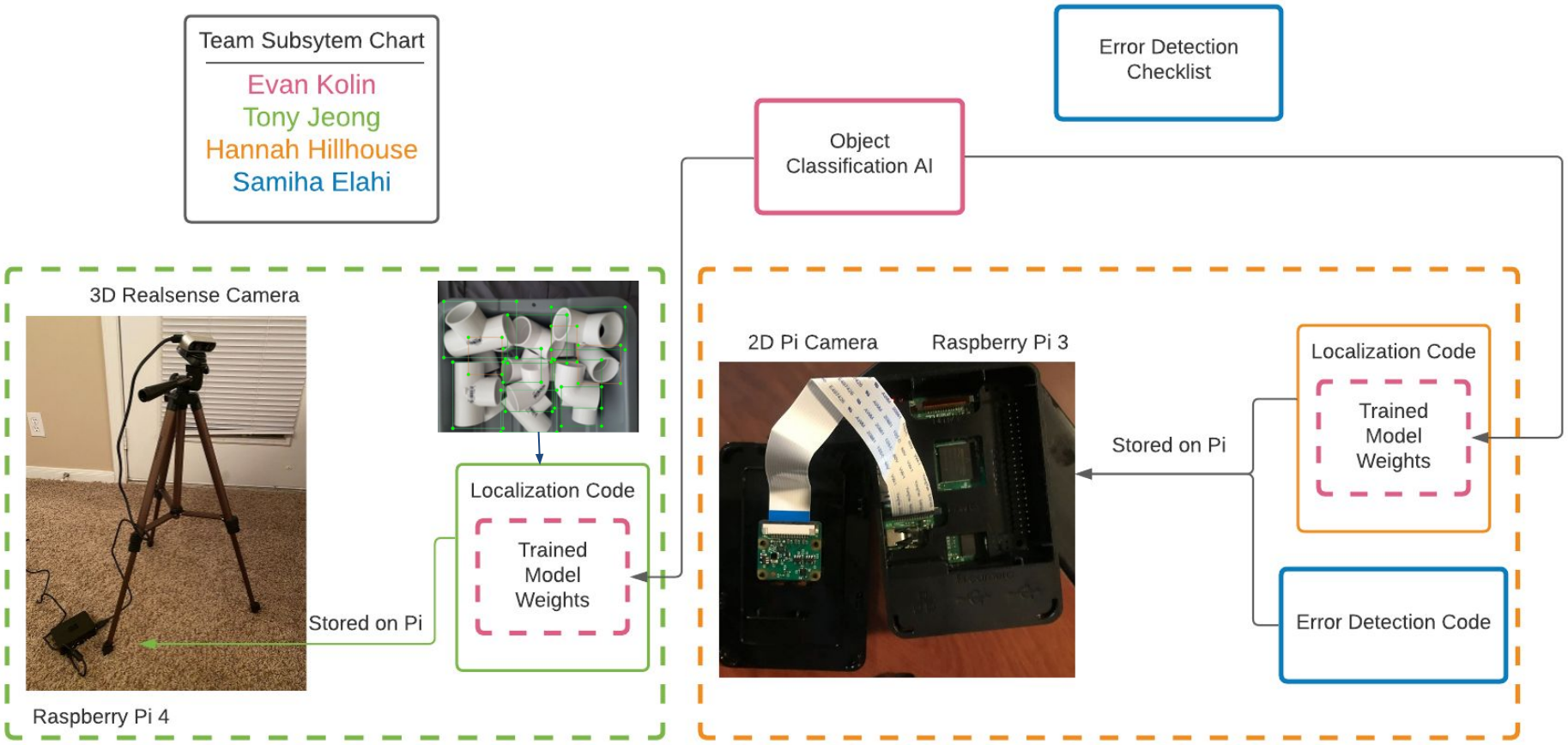
Problem:

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

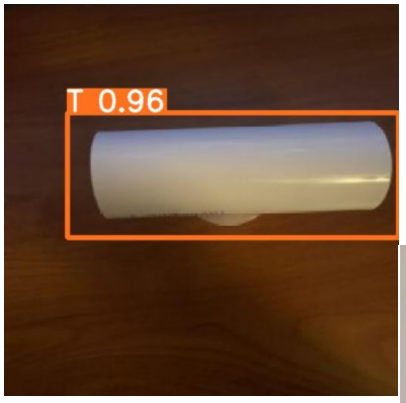
Hannah Hillhouse

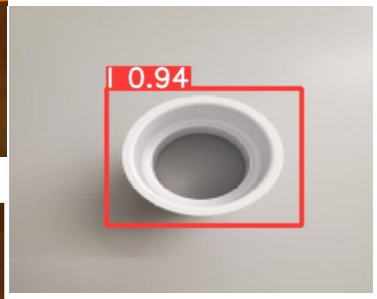
Accomplished since last update 20+ hrs	Ongoing progress/problems
- have trained dataset running on new data, tested, and validated that it work - started integration with error detection	- running into errors with integration with error detection such as util function calls - finish integration and start validating system

2D Camera Subsystem

Hannah Hillhouse

- I have trained tested and validated the new data set using yolov5 and it is completely working
- Integration with error detection is currently having errors using certain function calls. error detection used tensorflow and I used pytorch so once we figure out what calls are actually doing in error detection we can complete integration.





```
image 31/36 /content/yolov5/SeniorDesign-1/test/images/IMG_2987_jpeg.rf.0ff077ceb549c77f446596790314f6df.jpg: 416x416 1 T, Done. (0.025s)
```

3D Camera Subsystem

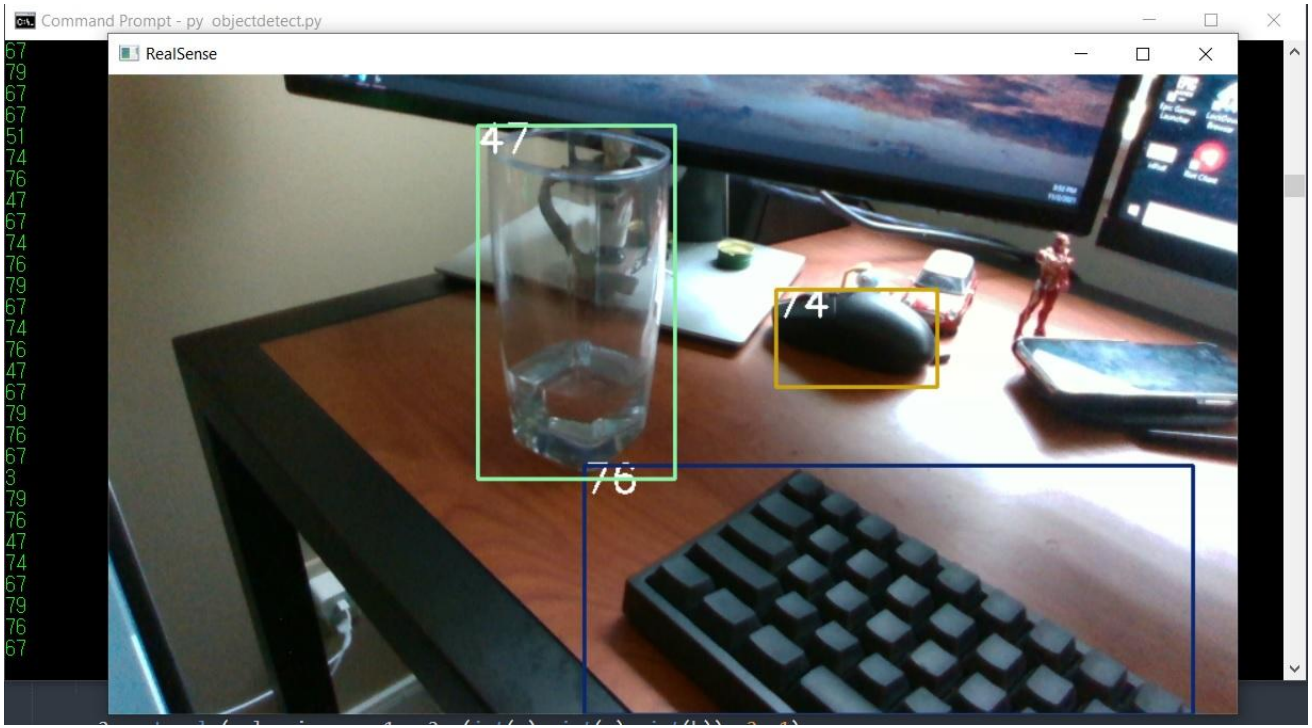
Tony Jeong

Accomplished since last update 8 hrs	Ongoing progress/problems
-Fully finished my code.	-Working on validation of trained model. -Helping with integrating error detection of 2D camera.

3D Camera Subsystem

Tony Jeong

The numbers on each box are the labels for detected objects. As Evan's trained model will be integrated, those numbers will represent I, L, T shape of the PVC pipes.



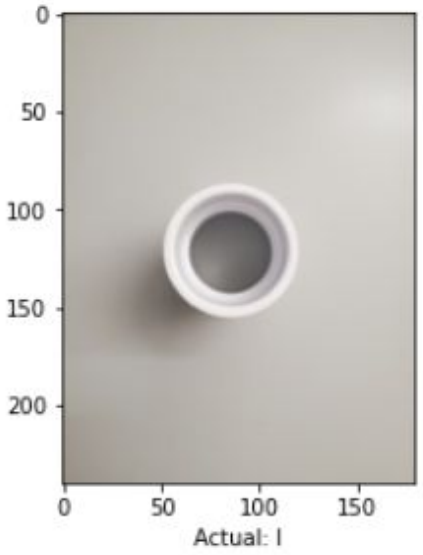
Object Classification Subsystem

Evan Kolin

Accomplished since last update 30+ hrs	Ongoing progress/problems
• Fixed txt to csv error • split model into two • setup labelIMG to create more complex occluded images • Fixed multiple errors with localization data being split from classification data	• Trained model will output all localization data as [1,1,1,1]

Object Classification Subsystem

van Kolin

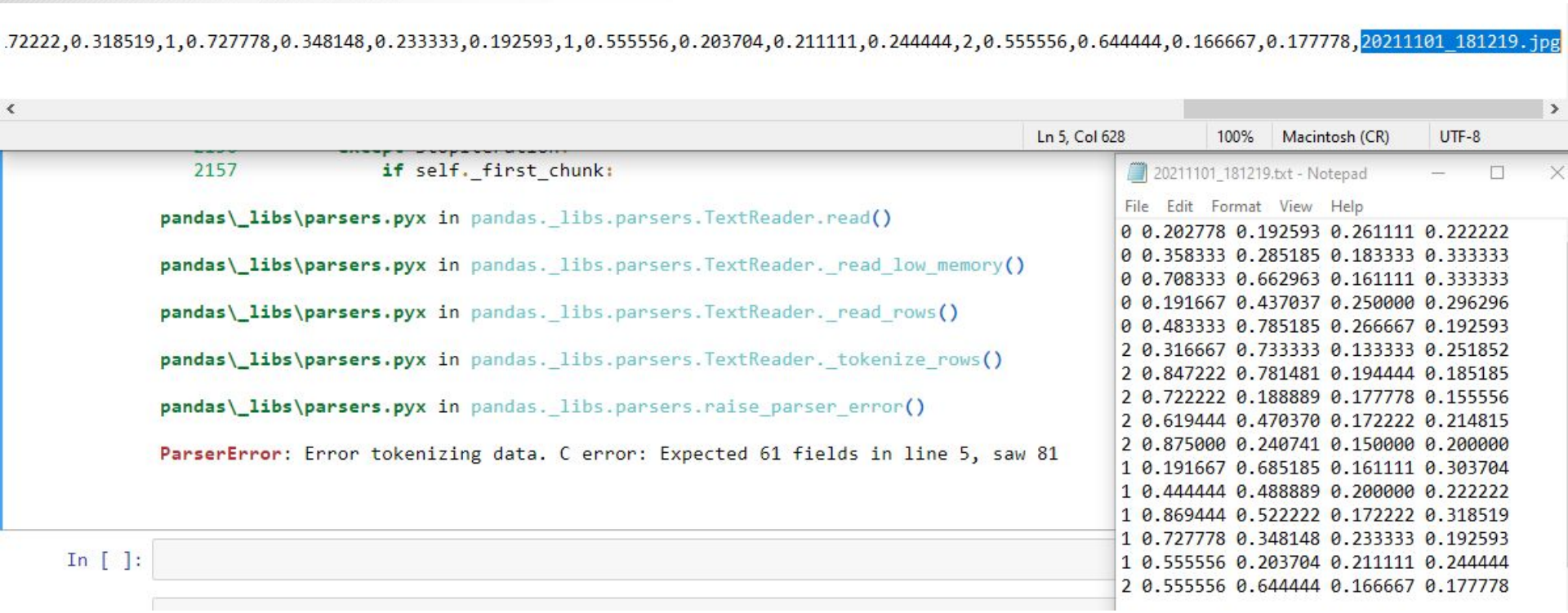



```
classification:  [ 24.416311 -22.281399 -23.06865 ]
localization:  [1. 1. 1. 1.]
```

Example crowded image

Localization data is always [1,1,1,1]
possible error is translating decimal data (0.00 - 1.00)
into picture coordinates (X: 0px-180px, Y:0px-240px)

Object Classification Subsystem Cont.



In trying to fix the localization error, I came across a new error where file input isn't being read correctly.

Object Classification Subsystem Cont.

How Object Classification and Localization Subsystem integrated with 3D Camera Subsystem

1.) Tony loads my model into his code

```
# load tensorflow
print("[INFO] Loading model...")
PATH_TO_CKPT = "frozen_inference_graph.pb"
```

2.) Pulls classification and localization data out of model

```
# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
```

3.) Draws bounding boxes onto image

```
boxes = np.squeeze(boxes)
classes = np.squeeze(classes).astype(np.int32)
scores = np.squeeze(scores)

for idx in range(int(num)):
    class_ = classes[idx]
    score = scores[idx]
    box = boxes[idx]

    if class_ not in colors_hash:
        colors_hash[class_] = tuple(np.random.choice(range(256), size=3))

    if score > 0.6:
        left = int(box[1] * color_frame.width)
        top = int(box[0] * color_frame.height)
        right = int(box[3] * color_frame.width)
        bottom = int(box[2] * color_frame.height)

        p1 = (left, top)
        p2 = (right, bottom)
        # draw box
        r, g, b = colors_hash[class_]
        cv2.rectangle(color_image, p1, p2, (int(r), int(g), int(b)), 2, 1)

cv2.namedWindow('RealSense', cv2.WINDOW_AUTOSIZE)
cv2.imshow('RealSense', color_image)
cv2.waitKey(1)
```

Error Detection & Handling Subsystem

Samiha Elahi

Accomplished since last update - 8 hrs	Ongoing progress/problems
- Helped Hannah fix bugs in her code to train, test and validate the new data set - Continued integrating the error detection with the 2D camera subsystem	- Figure out the issues with certain function calls - Complete integration with the 2D subsystem and then start validating asap with various test cases

Error Detection & Handling Subsystem

- Take various before and after pictures to see the effectiveness of solutions on solving their given problems by shaking the bin, changing the camera angle or rearranging the parts. .
- Once the error detection and the 2D subsystem are integrated, I will again double check the solutions work using the integrated subsystems.

Problems	Solution
Object outside camera frame	Change the camera angle
AI thinks T shaped is an I shaped pvc	Rearrange the parts
Object occluded/hidden	Shake the bin
AI not detecting a part at all	Rearrange the parts
AI detecting the wrong object	Move the position of the objects

Urgent matters

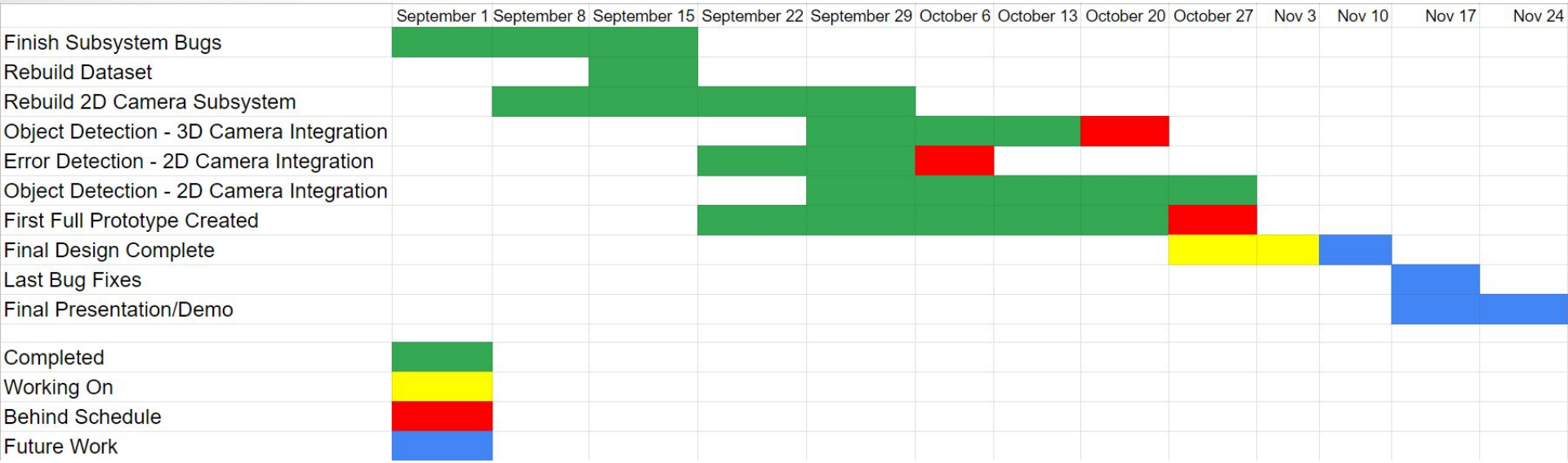
-Validate code runs live on raspberry pi's

-Create simple user interface for creating "orders" for parts

-Error detection image handling checklist, for things like measuring effectiveness of shaking a bucket in resolving the classification of an occluded object.

-Prove localization works on occluded objects!

Execution Plan



Thank You

Questions?