Project 3: 3D Perception

Writeup / README

1. Provide a Writeup / README that includes all the rubric points and how you addressed each one.

You're reading it!

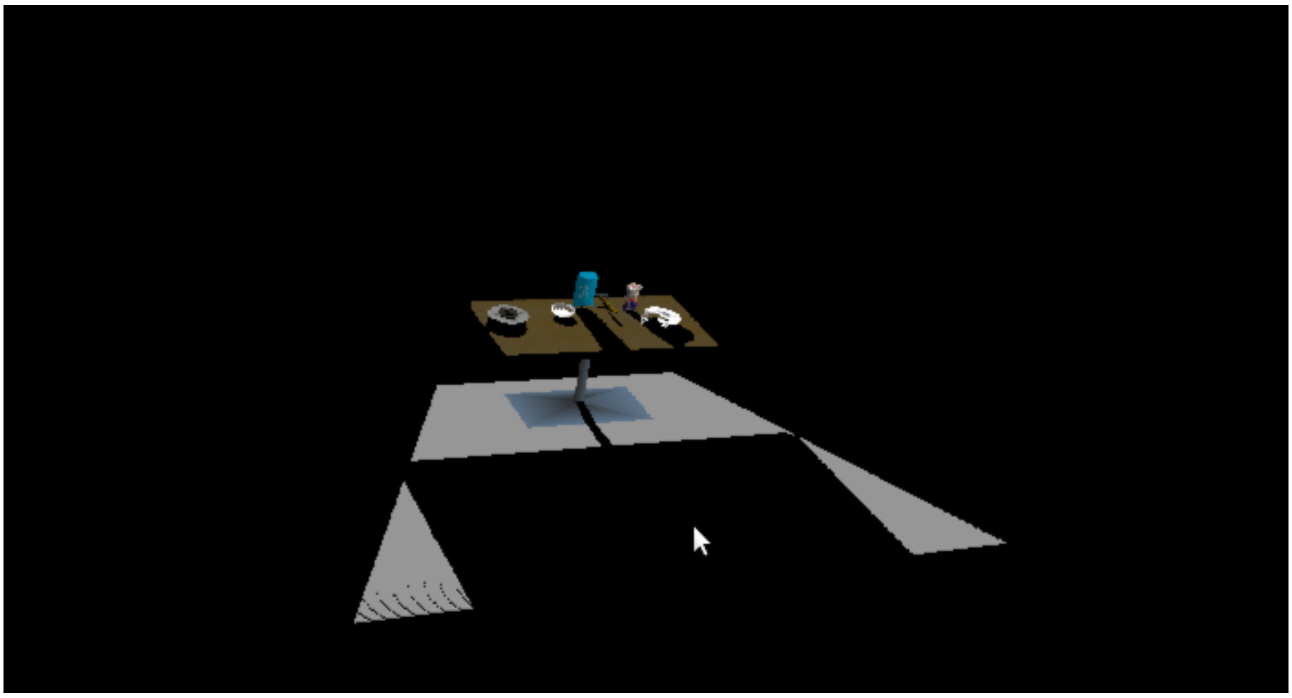
Exercise 1, 2 and 3 Pipeline Implemented

Complete Exercise 1 steps. Pipeline for filtering and RANSAC plane fitting implemented.

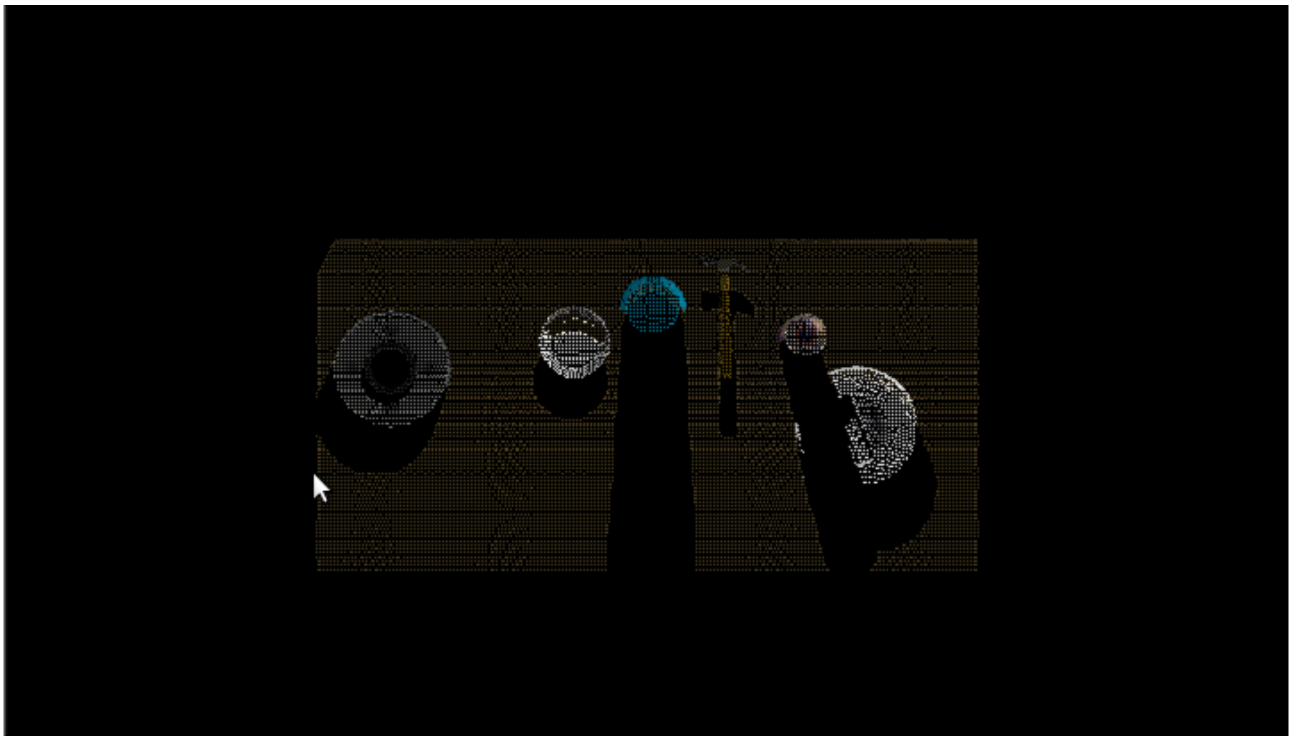
I have created a pipeline to recognize the object on the table.

You can find the steps below.

First; Calibration, filtering and segmentation
1- The image should be calibrated.
2- Apply voxel grid down sampling to decrease the size of the point for each image. But the information loss should be minimum.

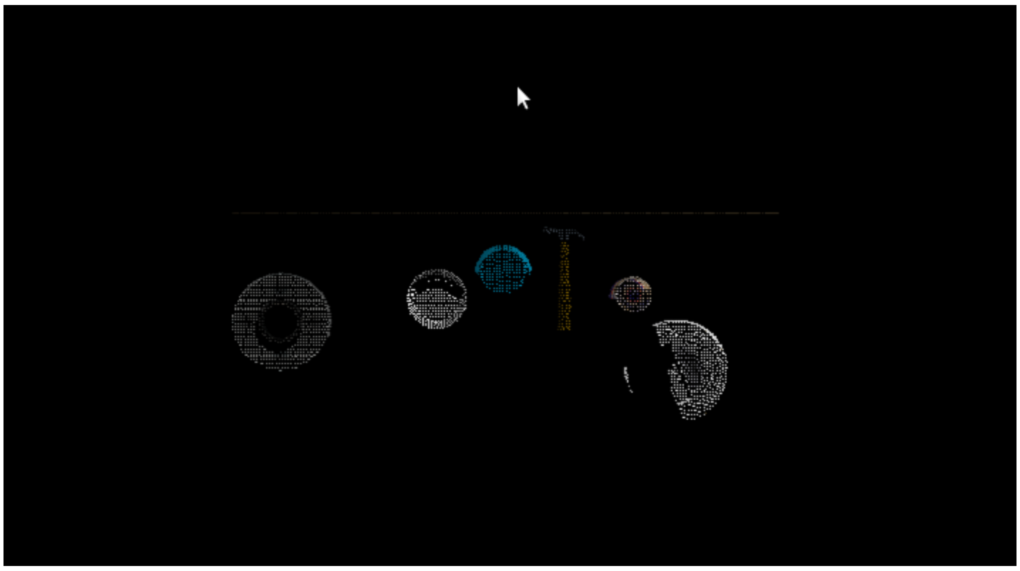


3- Apply pass through filtering to focus on area of interest and discard rest of the image.



4- Apply Ransac segmentation to detect the objects to be recognized.

Ransac use segmentation to find inliers and outliers. In that scenario, inliers are background such as table. And outliers are object on the table. You can find a sample image below.



5- Remove the outliers.

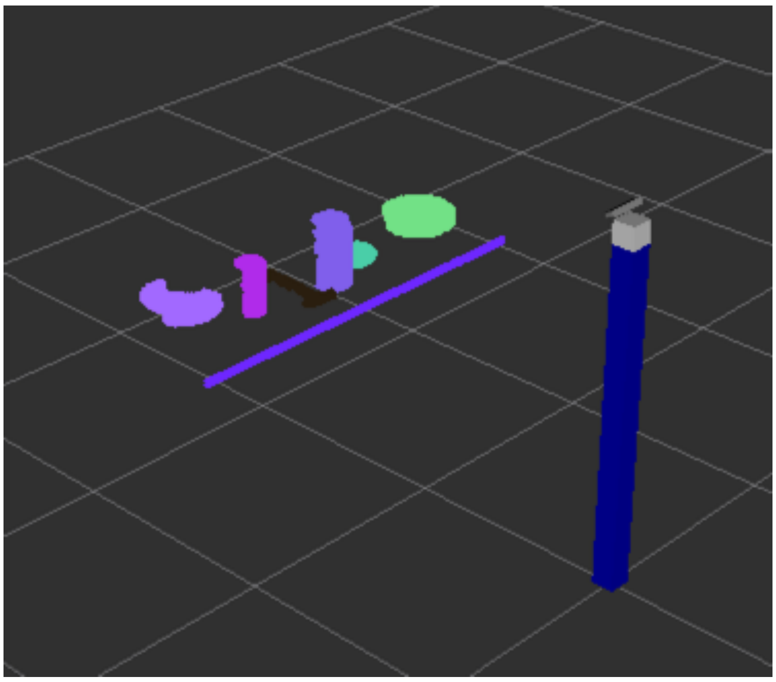
You can find all implementation here:
https://github.com/clockworks/nd-robond-perception-project/blob/master/pr2_robot/scripts/project_template.py

Complete Exercise 2 steps: Pipeline including clustering for segmentation implemented.

There are two main approached for clustering, k-means and DBScan.

kmeans is better when you know how many cluster will be
DBScans is also known as euclidian clustering, since it is based on euclidian distance. It is better when you don't know how many cluster to expect.

You can find the output of the clustering below.



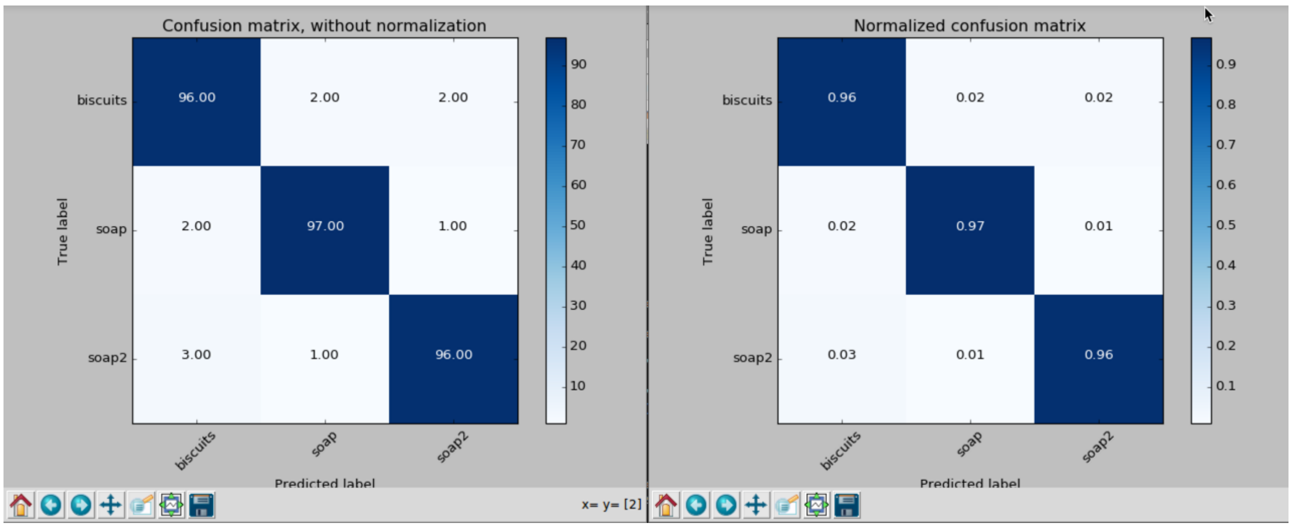
Steps for clustering on Ros

Take the camera data as a point cloud.
Filter data point cloud
Segment the individual objects using euclidian clustering.
publish result point cloud as a message to the topic.

Complete Exercise 3 Steps. Features extracted and SVM trained. Object recognition implemented.

You can find the confusion matrix for each test worlds.

test world 1:


Confusion matrix, without normalization
Normalized confusion matrix
True label
Predicted label
x= y= [2]

Confusion matrix, without normalization

True label \ Predicted label	biscuits	soap	soap2
biscuits	96.00	2.00	2.00
soap	2.00	97.00	1.00
soap2	3.00	1.00	96.00

Normalized confusion matrix

True label \ Predicted label	biscuits	soap	soap2
biscuits	0.96	0.02	0.02
soap	0.02	0.97	0.01
soap2	0.03	0.01	0.96

test world 2:

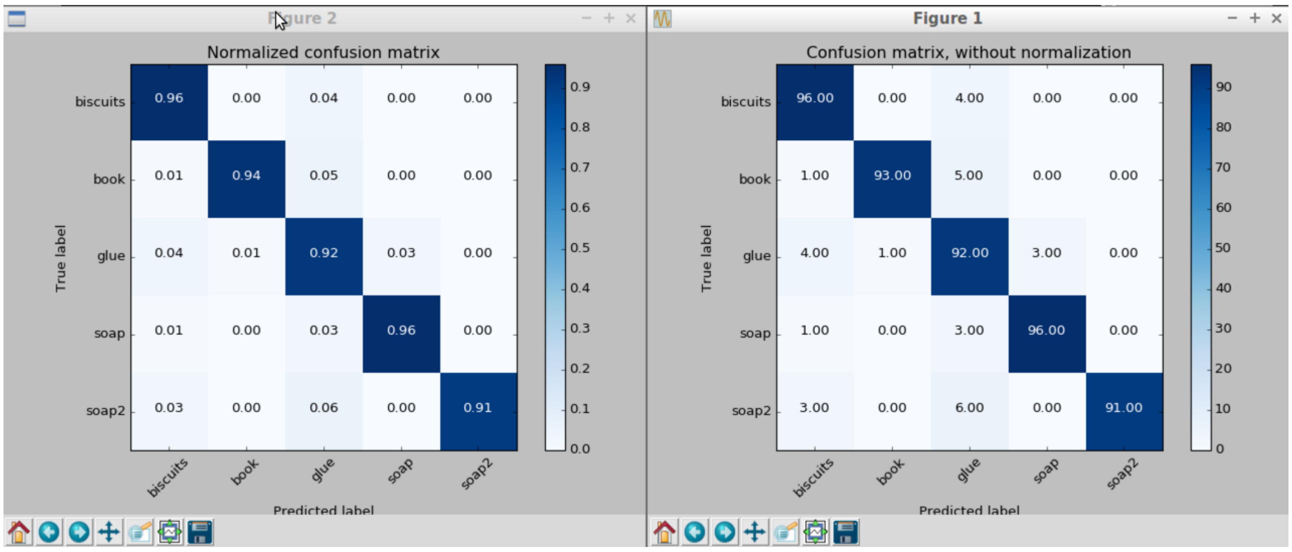


Figure 2
Figure 1
Normalized confusion matrix
Confusion matrix, without normalization
True label
Predicted label

Normalized confusion matrix

True label \ Predicted label	biscuits	book	glue	soap	soap2
biscuits	0.96	0.00	0.04	0.00	0.00
book	0.01	0.94	0.05	0.00	0.00
glue	0.04	0.01	0.92	0.03	0.00
soap	0.01	0.00	0.03	0.96	0.00
soap2	0.03	0.00	0.06	0.00	0.91

Confusion matrix, without normalization

True label \ Predicted label	biscuits	book	glue	soap	soap2
biscuits	96.00	0.00	4.00	0.00	0.00
book	1.00	93.00	5.00	0.00	0.00
glue	4.00	1.00	92.00	3.00	0.00
soap	1.00	0.00	3.00	96.00	0.00
soap2	3.00	0.00	6.00	0.00	91.00

Pick and Place Setup

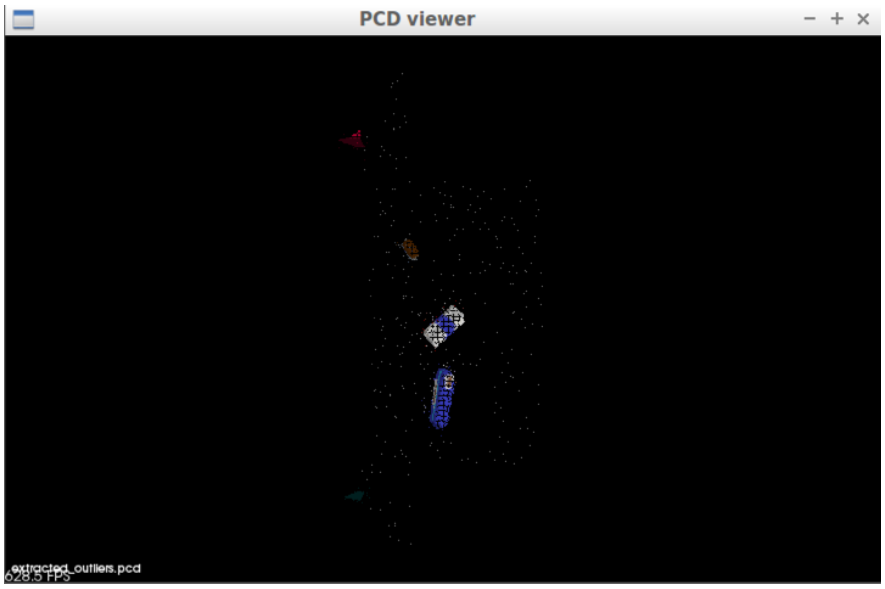
For all three tabletop setups (`test*.world`), perform object recognition, then read in respective pick list (`pick_list_*.yaml`). Next construct the messages that would comprise a valid `PickPlace` request output them to `.yaml` format.

Although confusion matrix and accuracy are expected. Robot could not predict items accurately. You can find the results below. Any feedback and comment about code would be very valuable.

Output file contains item than the image. Because I only write unique items to the output file. If robot predicts accurately, file will contains all items.

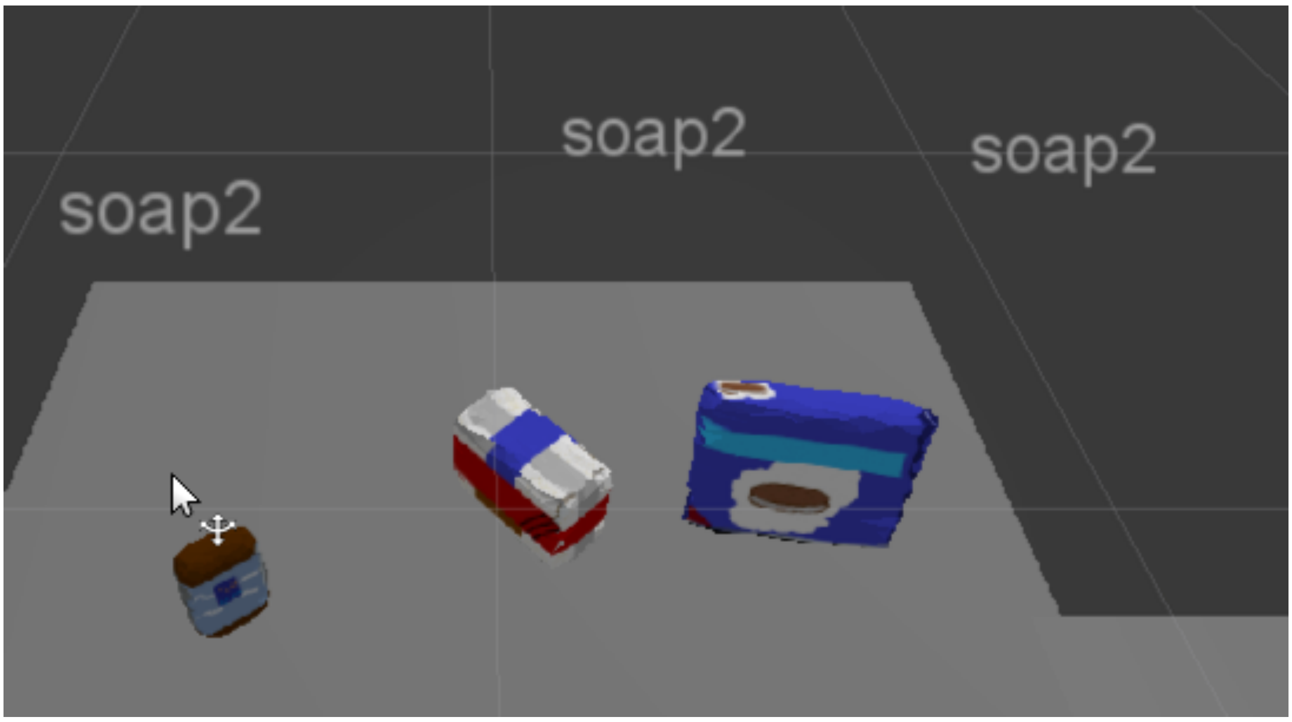
test world 1:

Extracted outliers:



confusion matrix

items

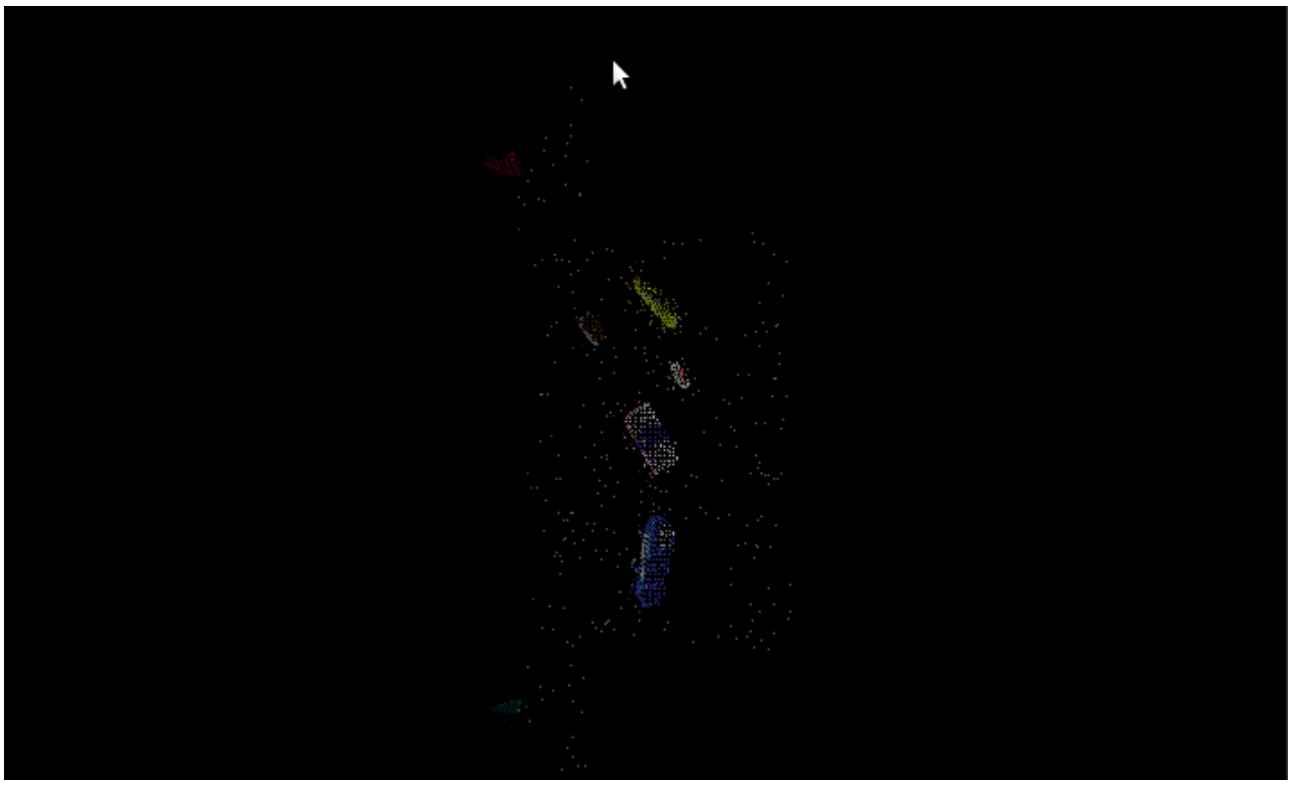


output1 yaml file content:

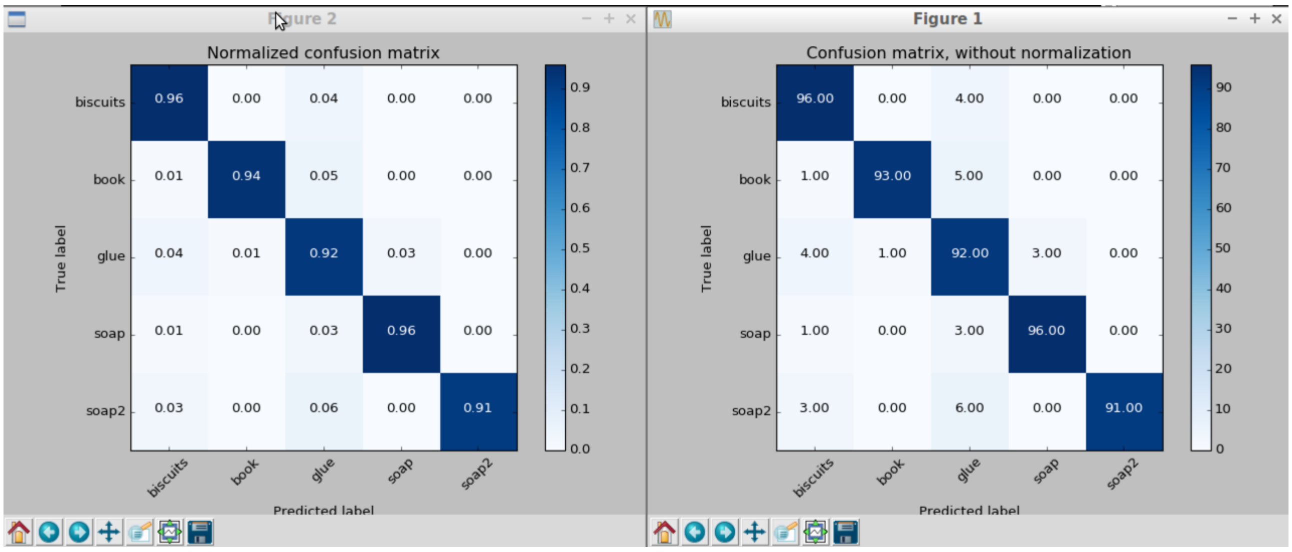
```
soap2:
- 0.39537882804870605
- 0.07007074356079102
- 0.3619803190231323
```

test world 2:

Extracted outliers:



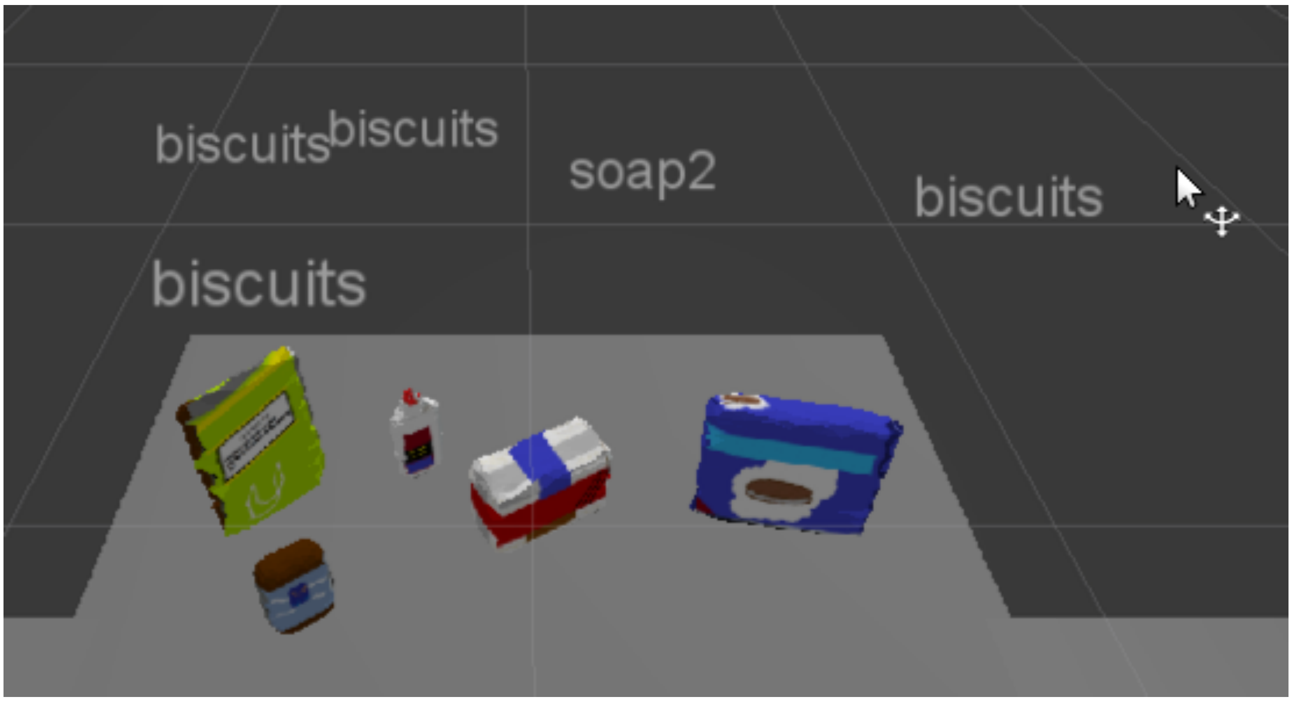
confusion matrix:



	biscuits	book	glue	soap	soap2
biscuits	0.96	0.00	0.04	0.00	0.00
book	0.01	0.94	0.05	0.00	0.00
glue	0.04	0.01	0.92	0.03	0.00
soap	0.01	0.00	0.03	0.96	0.00
soap2	0.03	0.00	0.06	0.00	0.91

	biscuits	book	glue	soap	soap2
biscuits	96.00	0.00	4.00	0.00	0.00
book	1.00	93.00	5.00	0.00	0.00
glue	4.00	1.00	92.00	3.00	0.00
soap	1.00	0.00	3.00	96.00	0.00
soap2	3.00	0.00	6.00	0.00	91.00

items:

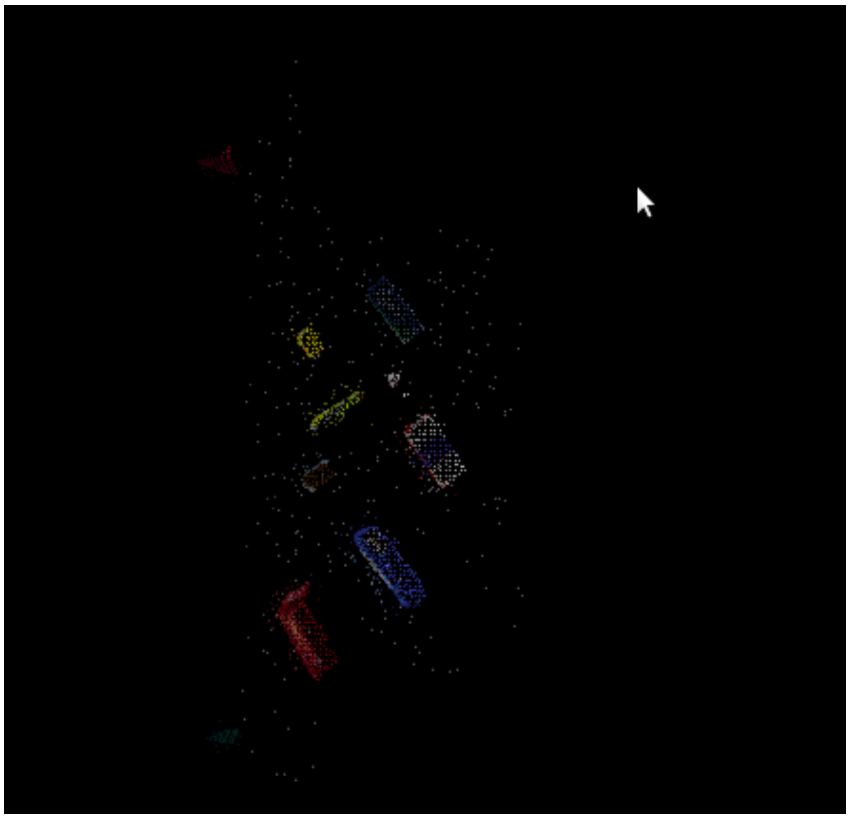


output2 yaml file content:

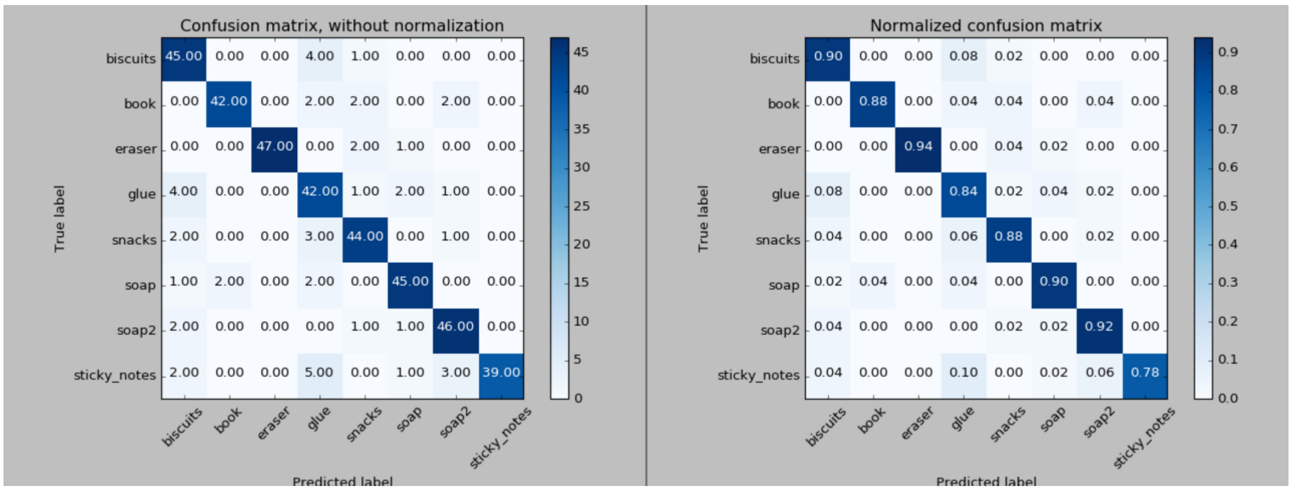
```
biscuits:
- 0.6760640740394592
- 0.14034660160541534
- 0.4637441635131836
soap2:
- 0.5710335373878479
- 0.04242696985602379
- 0.6973682641983032
```

test world 3:

Extracted outliers:



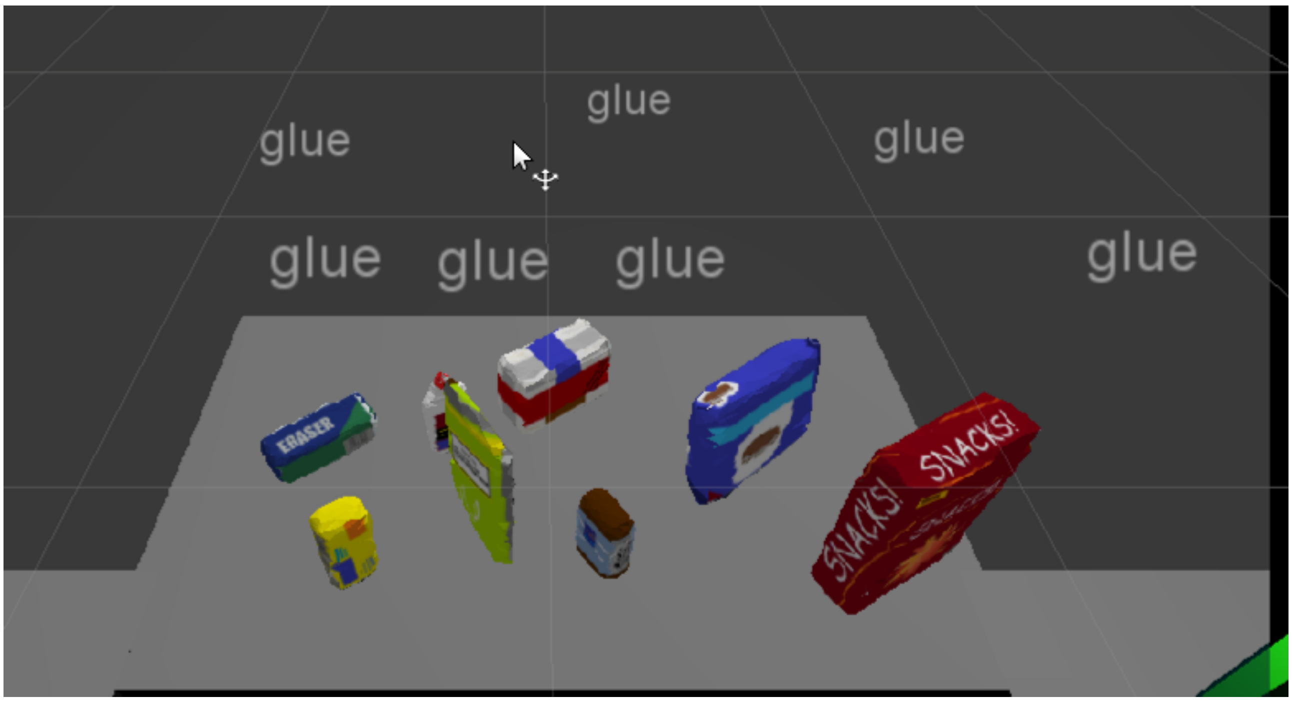
confusion matrix:



	biscuits	book	eraser	glue	snacks	soap	soap2	sticky_notes
biscuits	45.00	0.00	0.00	4.00	1.00	0.00	0.00	0.00
book	0.00	42.00	0.00	2.00	2.00	0.00	2.00	0.00
eraser	0.00	0.00	47.00	0.00	2.00	1.00	0.00	0.00
glue	4.00	0.00	0.00	42.00	1.00	2.00	1.00	0.00
snacks	2.00	0.00	0.00	3.00	44.00	0.00	1.00	0.00
soap	1.00	2.00	0.00	2.00	0.00	45.00	0.00	0.00
soap2	2.00	0.00	0.00	0.00	1.00	1.00	46.00	0.00
sticky_notes	2.00	0.00	0.00	5.00	0.00	1.00	3.00	39.00

	biscuits	book	eraser	glue	snacks	soap	soap2	sticky_notes
biscuits	0.90	0.00	0.00	0.08	0.02	0.00	0.00	0.00
book	0.00	0.88	0.00	0.04	0.04	0.00	0.04	0.00
eraser	0.00	0.00	0.94	0.00	0.04	0.02	0.00	0.00
glue	0.08	0.00	0.00	0.84	0.02	0.04	0.02	0.00
snacks	0.04	0.00	0.00	0.06	0.88	0.00	0.02	0.00
soap	0.02	0.04	0.00	0.04	0.00	0.90	0.00	0.00
soap2	0.04	0.00	0.00	0.00	0.02	0.02	0.92	0.00
sticky_notes	0.04	0.00	0.00	0.10	0.00	0.02	0.06	0.78

All items are tagged as glue



Output3 yaml file

glue:
- 0.5595681667327881
- 0.049852184951305396
-   0.6189662218093872