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!

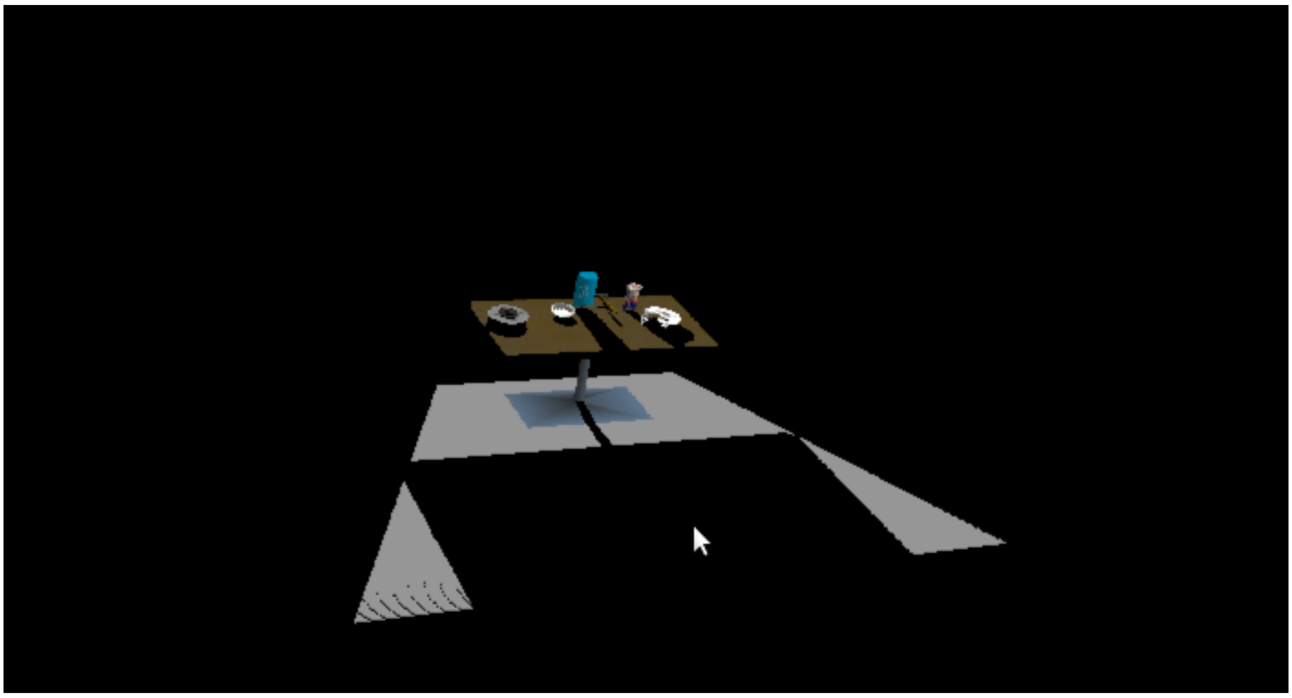
Exercise 1, 2 and 3 Pipeline Implemented

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

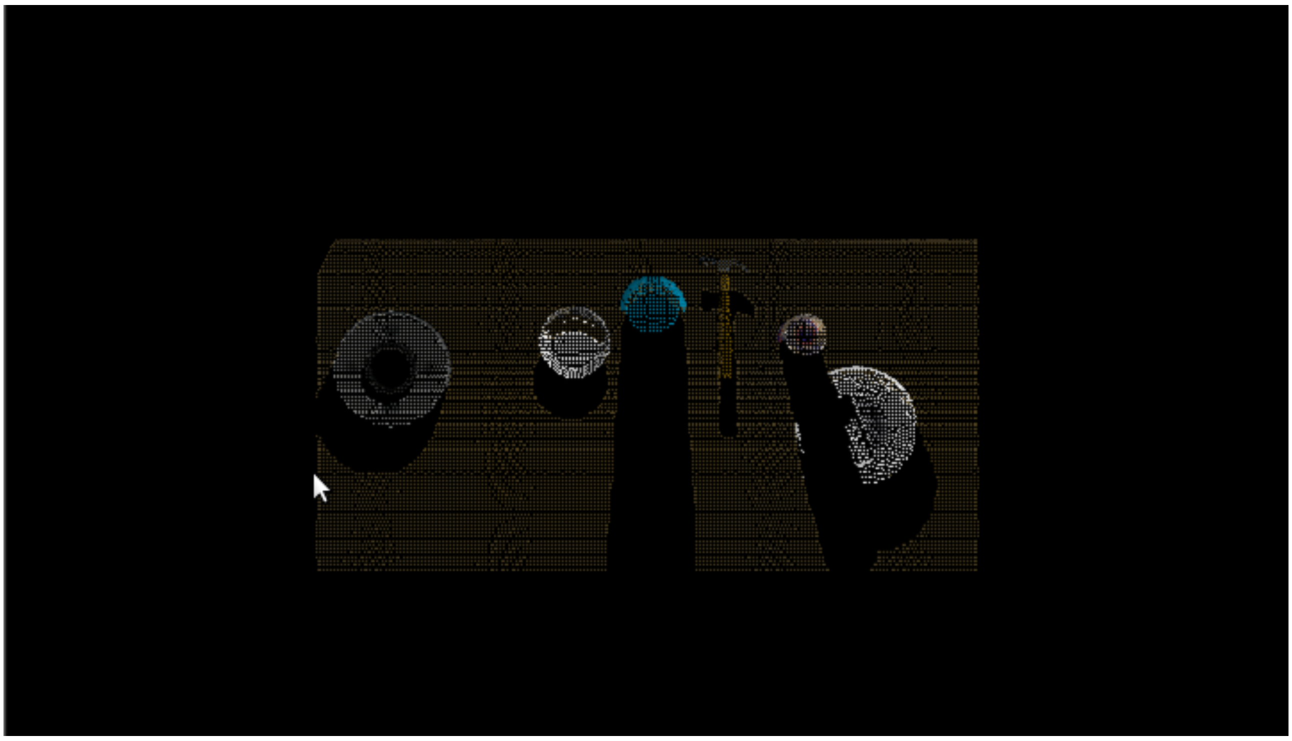
I have created a pipeline to recognize the object on the table.

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.

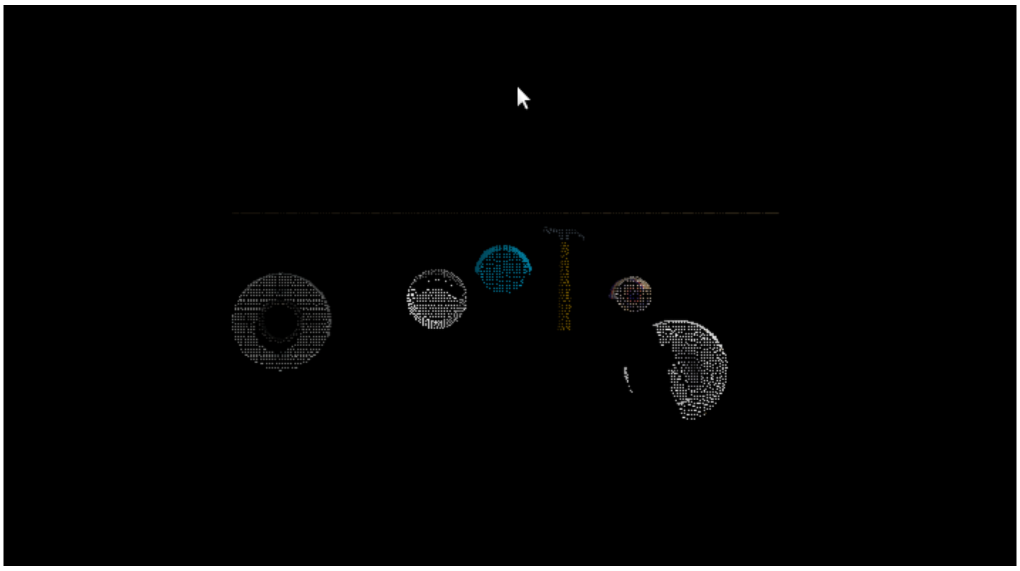


3- Apply pass through filtering to focus on area of interest and discard rest of the image.



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.

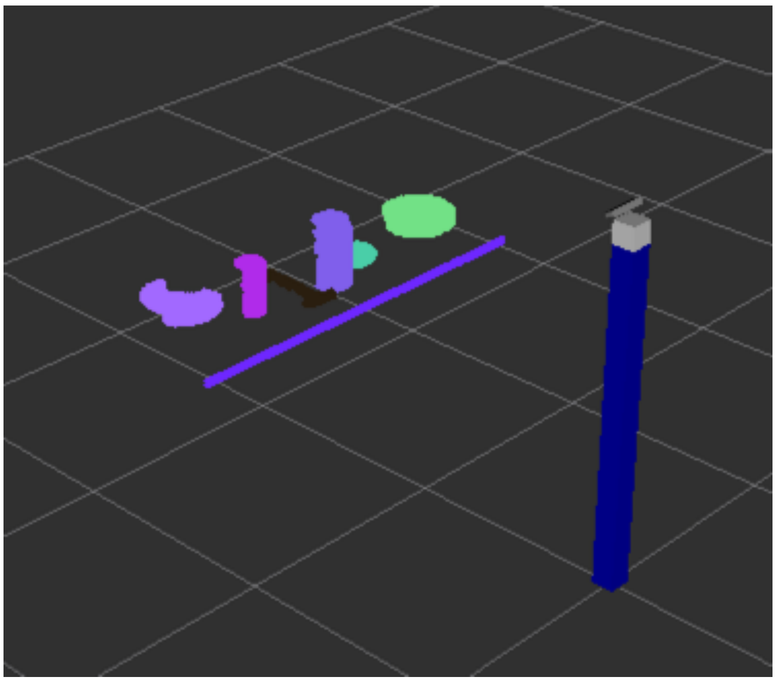
You can find all implementation here:
https://github.com/clockworks/nd-robond-perception-project/blob/master/pr2_robot/scripts/project_template.py

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.



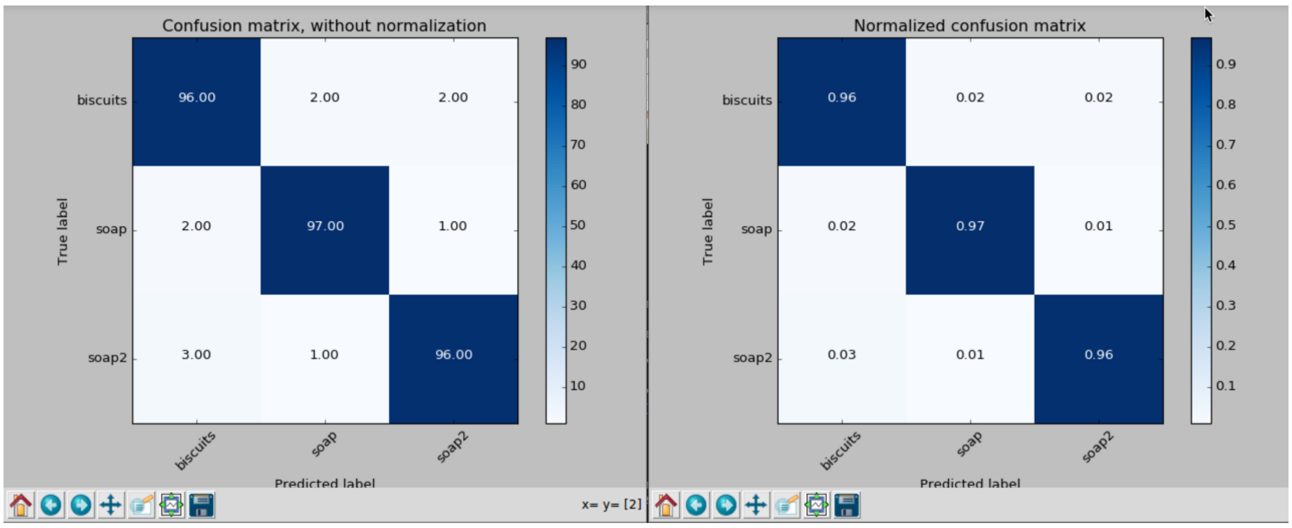
Steps for clustering on Ros

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:



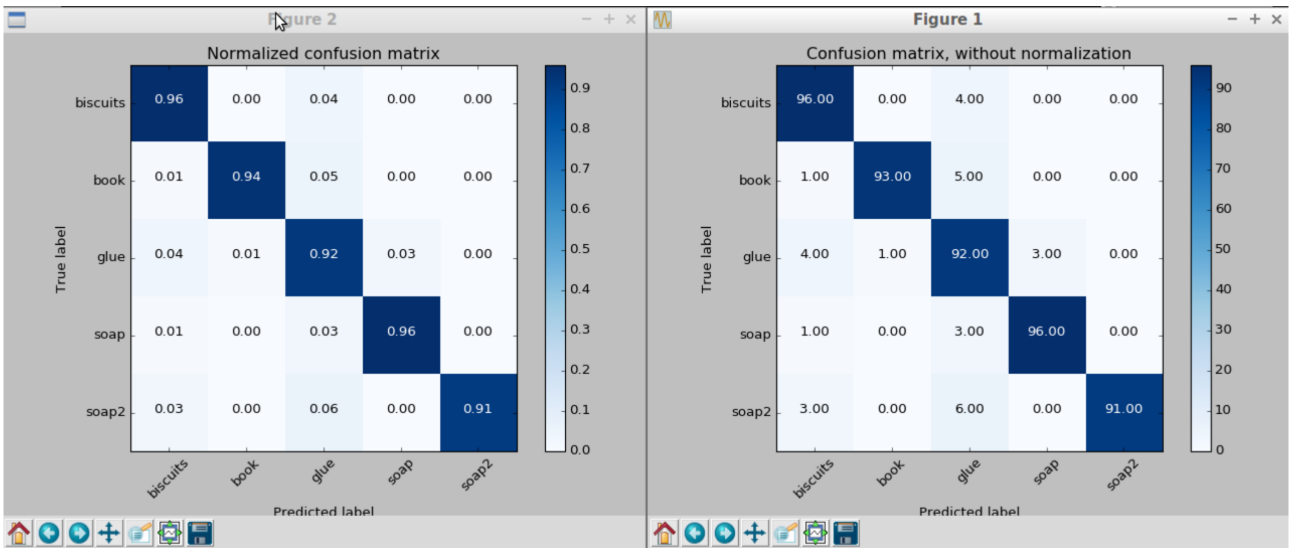
Confusion matrix, without normalization

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:



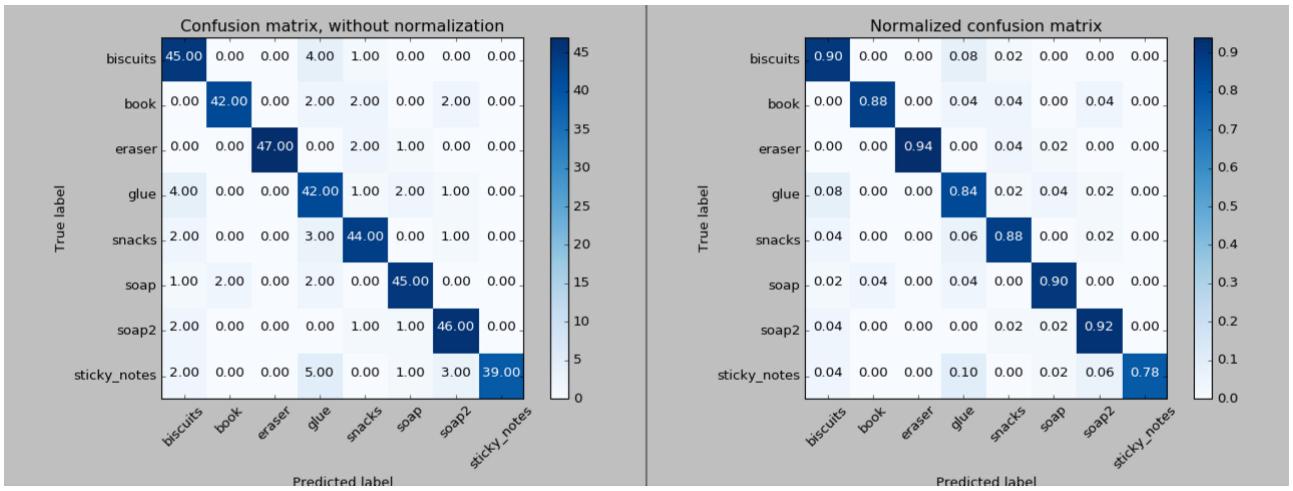
Normalized confusion matrix

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

test world 3:



	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

I used sensor_stick in exercise 3 for feature capture and training. I did not move training code to project folder. I keep it as a seperate repository. You can find the repository and related code below:

https://github.com/clockworks/nd-robond-perception-exercise/tree/master/Exercise-3

capture_features.py
https://github.com/clockworks/nd-robond-perception-exercise/blob/master/Exercise-3/sensor_stick/scripts/capture_features.py

features.py:
https://github.com/clockworks/nd-robond-perception-exercise/blob/master/Exercise-3/sensor_stick/src/sensor_stick/features.py

train_svm.py
https://github.com/clockworks/nd-robond-perception-exercise/blob/master/Exercise-3/sensor_stick/scripts/train_svm.py

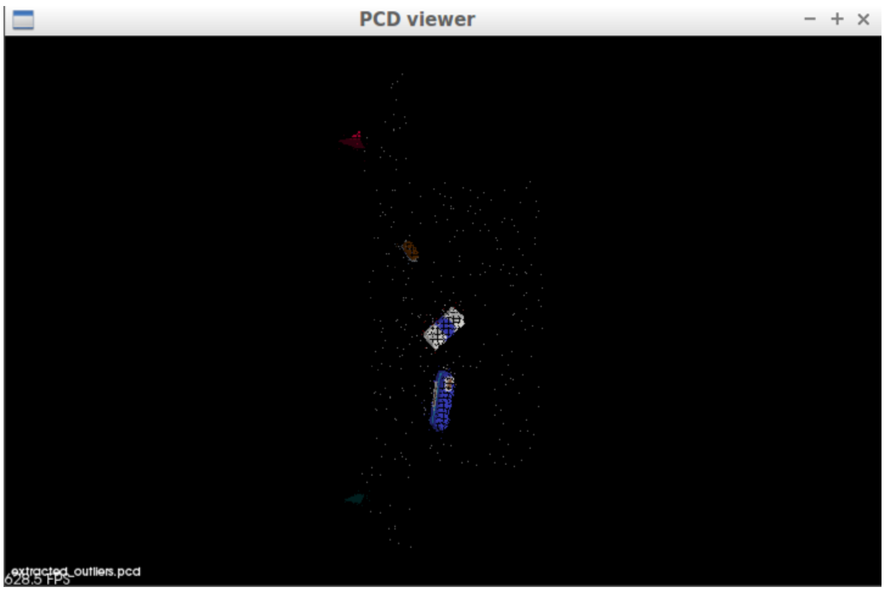
Pick and Place Setup

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.

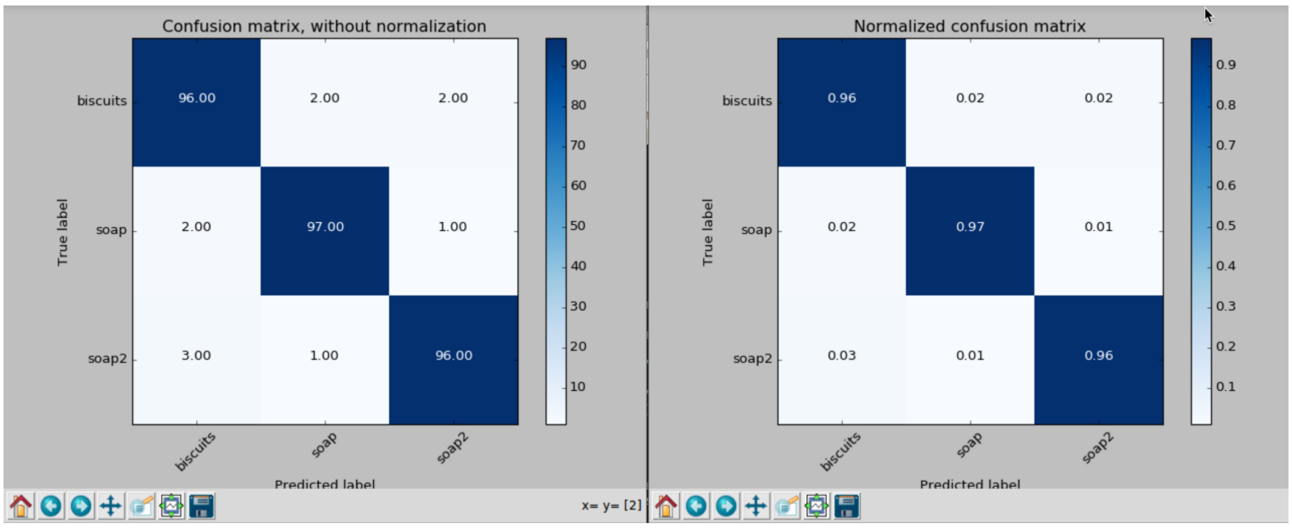
The pipeline correctly identfy the object. You can see the details below.

test world 1:

Extracted outliers:



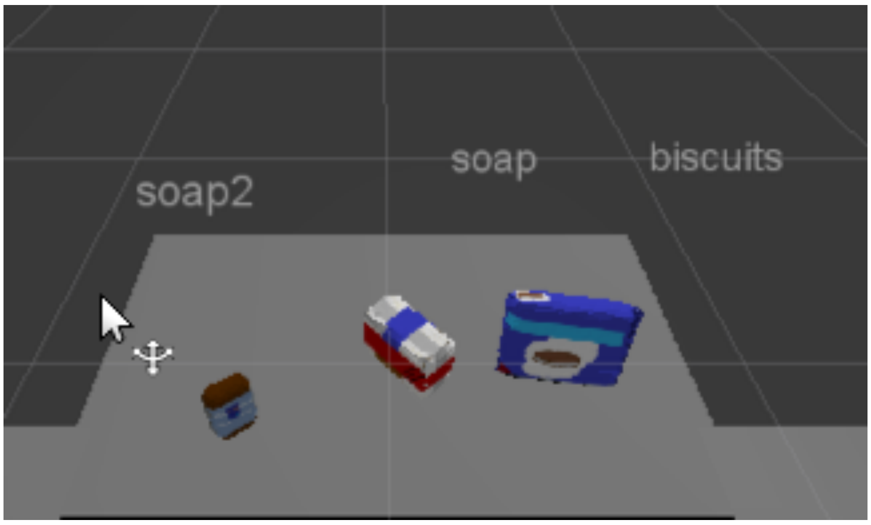
confusion matrix



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

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

All items recognized correctly. (100%)

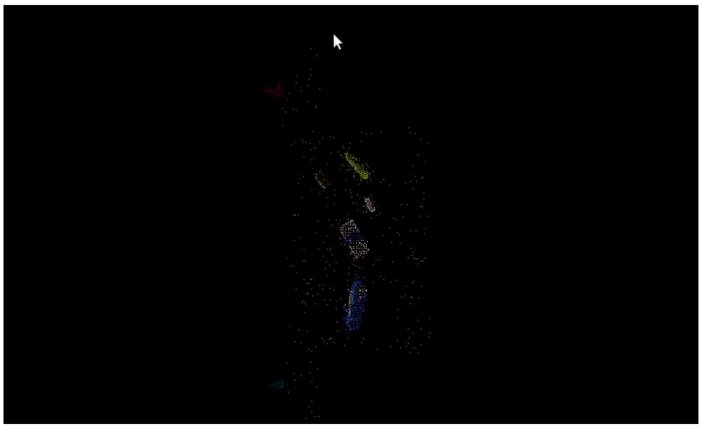


output1 yaml file content:

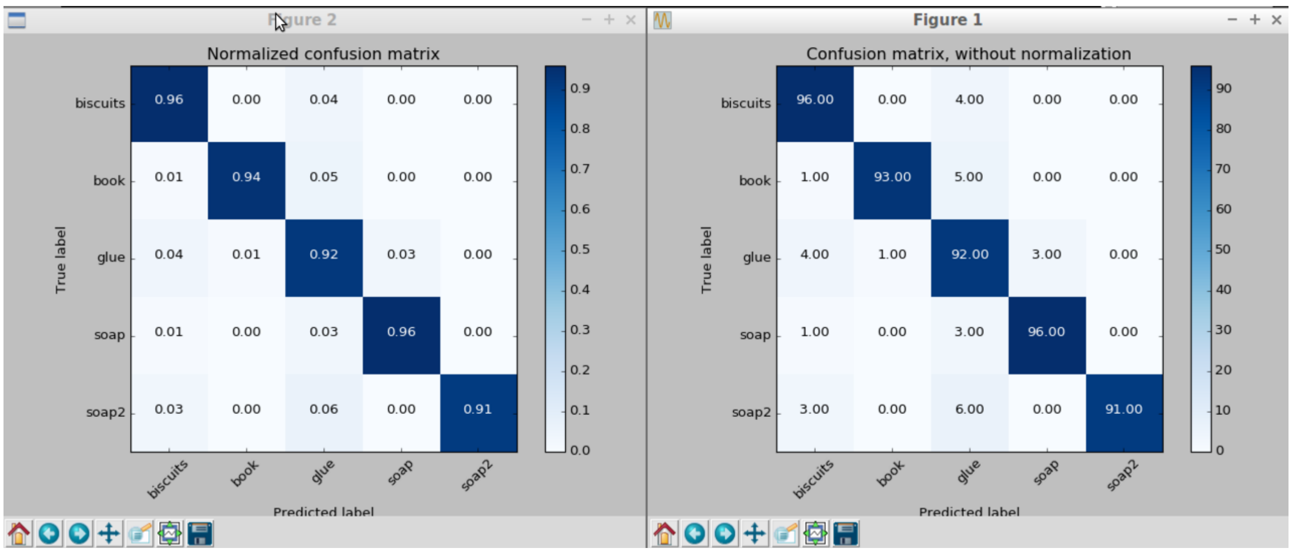
```yaml
biscuits:
- 0.2793441414833069
- 0.5603342056274414
- 0.7238720655441284
soap:
- 0.5441828966140747
- -0.018730824813246727
- 0.6747274994850159
soap2:
- 0.4457259476184845
- 0.22262020409107208
- 0.6771800518035889
```

test world 2:

Extracted outliers:



confusion matrix:



Normalized confusion matrix

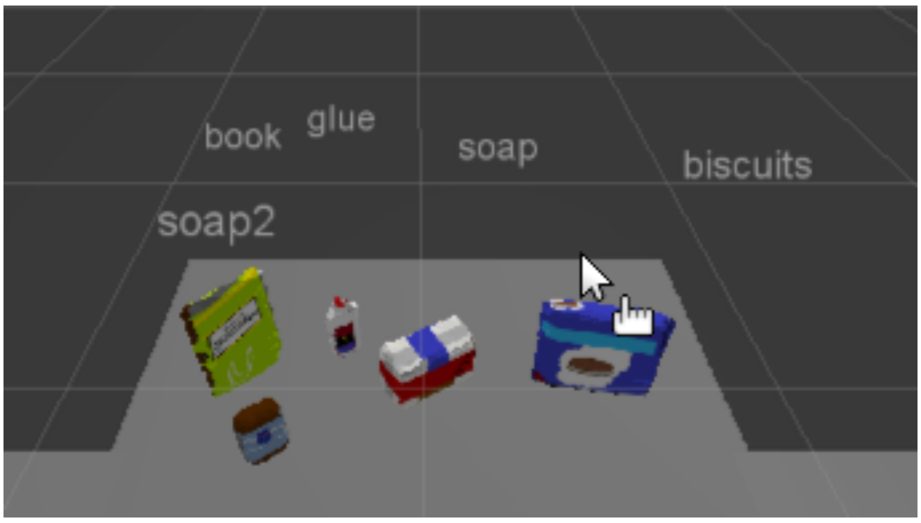
True label \ Predicted label	biscuits	book	glue	soap	soap2
biscuits	0.96	0.00	0.04	0.00	0.00
book	0.01	0.94	0.05	0.00	0.00
glue	0.04	0.01	0.92	0.03	0.00
soap	0.01	0.00	0.03	0.96	0.00
soap2	0.03	0.00	0.06	0.00	0.91

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

items:

All items recognized correctly (100%)

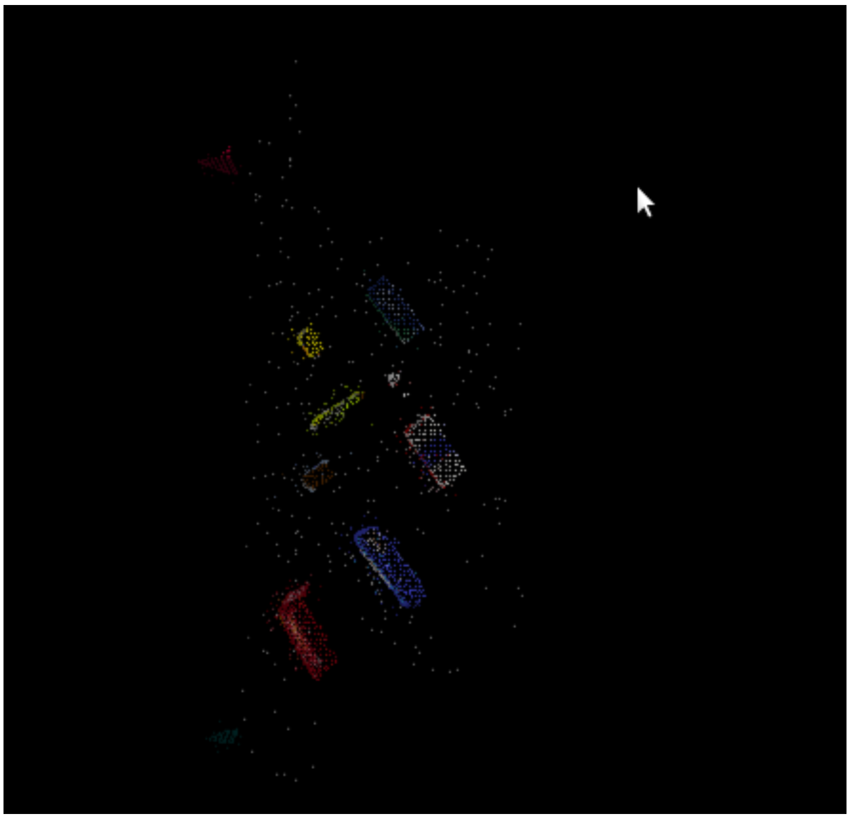


output2 yaml file content:

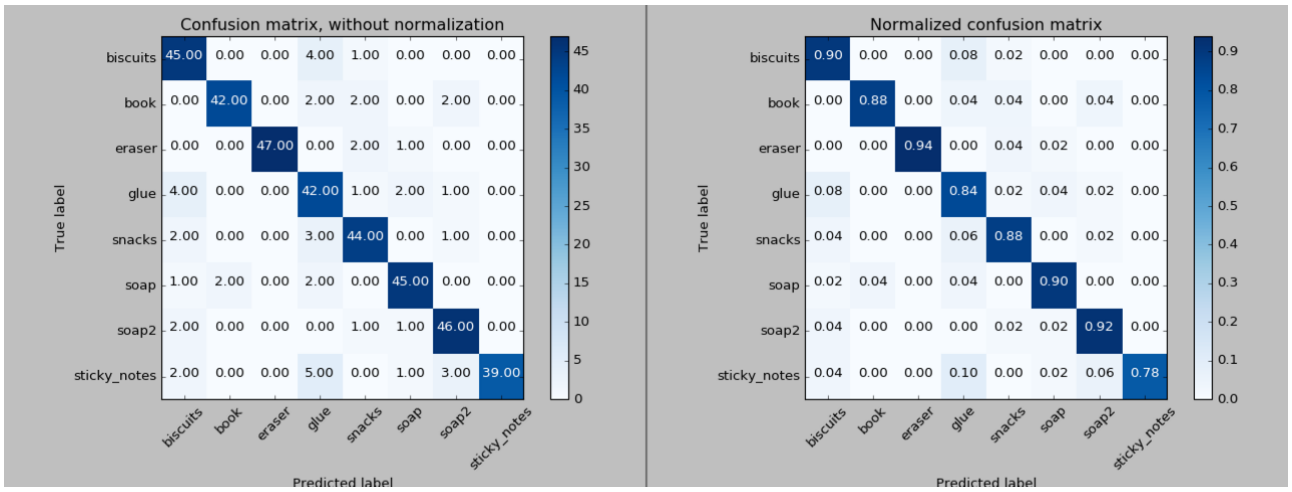
```
biscuits:
- 0.5708536505699158
- -0.24835027754306793
- 0.7036103010177612
book:
- 0.5802746415138245
- 0.27999553084373474
- 0.7224472165107727
glue:
- 0.27720972895622253
- 0.5618106722831726
- 0.7263579368591309
soap:
- 0.5581965446472168
- 0.0036523318849503994
- 0.6726524829864502
soap2:
- 0.4439592957496643
- 0.22644901275634766
- 0.6769988536834717
```

test world 3:

Extracted outliers:



confusion matrix:



Confusion matrix, without normalization

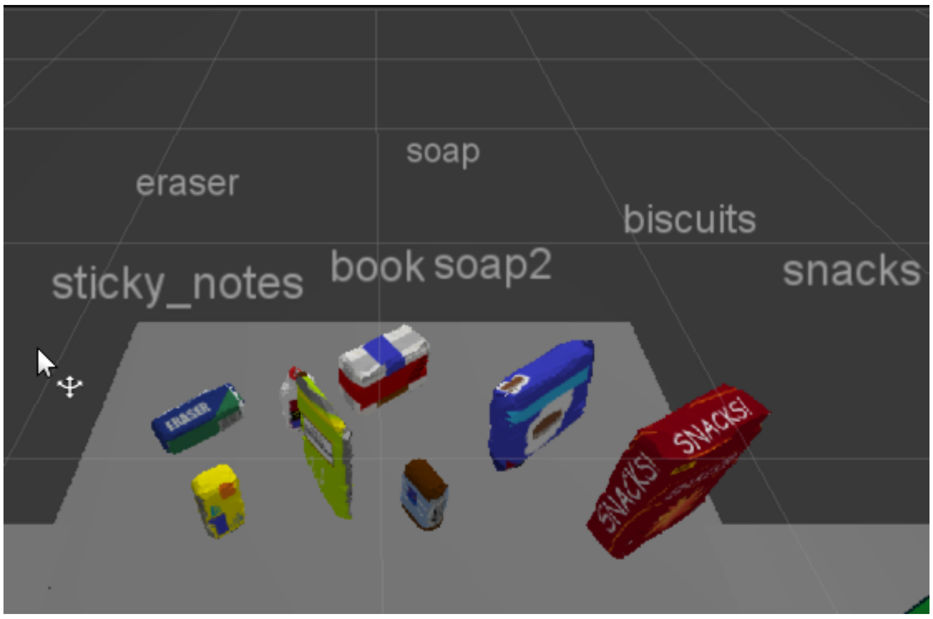
True label \ Predicted label	biscuits	book	eraser	glue	snacks	soap	soap2	sticky_notes
biscuits	45.00	0.00	0.00	4.00	1.00	0.00	0.00	0.00
book	0.00	42.00	0.00	2.00	2.00	0.00	2.00	0.00
eraser	0.00	0.00	47.00	0.00	2.00	1.00	0.00	0.00
glue	4.00	0.00	0.00	42.00	1.00	2.00	1.00	0.00
snacks	2.00	0.00	0.00	3.00	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

Normalized confusion matrix

True label \ Predicted label	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

items

All items recognized correctly except glue, I think it is behind another object and that is what it could not be recognized. It is also hard for a human too. Success 87.5%



Output3 yaml file

```
biscuits:
- 0.589409589767456
- -0.22033236920833588
- 0.702957272529602
book:
- 0.49326598644256589
- 0.08309326320886612
- 0.7257049679756165
eraser:
- 0.6067849397659302
- 0.28088369965553284
- 0.6459604501724243
snacks:
- 0.27838391065597534
- 0.5610676407814026
- 0.7266518473625183
soap:
- 0.679608106613159 2
- 0.004904804285615683
- 0.6734455823898315
soap2:
```

```
- 0.4544113874435425
- -0.04611271992325783
- 0.673527717590332
sticky_notes:
- 0.43928566575050354
- 0.21655531227588654
- 0.6822288036346436
```