Lab 05: OpenCV and Depth Sensing

By Michael Su, Wolf Chen, last modified on 03/29 2020

The objective of this tutorial is to introduce the concept of point cloud, depth camera, and examples for OpenCV.

In topic 1, you will learn how to use Realsense camera SR300 or D435 and understand the the content of published topics.

In topic 2, we will use OpenCV functions and detect objects with HSV color space.

# Hardware and software setup

## **Laptop setup**

Update the latest repo to laptop

**laptop$ cd ~/sis\_lab\_all\_2020 (If you haven’t clone repo, please clone it first.)**

**laptop$ git checkout devel-[student\_id] (create and switch to your branch)**

**laptop$ git stash**

**laptop$ git pull origin master**

**laptop$ git stash pop**

**laptop$ cd ~/sis\_lab\_all\_2020/05-Opencv\_and\_Depth\_Sensing/Dockerfiles/PC**

**laptop$ docker build -t="lab5" .**

**^ there is a dot here.**

## **Nano setup**



DC 電源 亮小綠燈 右下角

網路線接電腦

dongle 連外網

裝D435深度相機 現在盒子裡 插恐在右邊 是反著放進盒子 不然會卡住

Please turn on Nano and plug your Realsense Camera into Nano.

Then, open a terminal and SSH into Nano.

**laptop$ ssh arg@[hostname].local**

First, set Jetson Nano from 5 Watt mode into 10 Watt mode to increase the computational speed

**Nano$ sudo nvpmodel -m 0**

Delete all the docker images, containers on Nano

**Nano$ docker rm -vf $(docker ps -a -q) 使用完畢 要刪除**

**Nano$ docker rmi -f $(docker images -a -q)**

**Nano$ cd ~/sis\_lab\_all\_2020 (If you haven’t clone repo, please clone it first.)**

**Nano$ git checkout master**

**Nano$ git stash && git pull**

**Nano$ cd ~/sis\_lab\_all\_2020/05-Opencv\_and\_Depth\_Sensing**

**Nano$ docker build -t="lab5" -f=Dockerfiles/Nano/Dockerfile .**

**^ there is a dot here.**

[**Hardware and software setup**](#_f15pq99wza2b) **1**

[Laptop setup](#_gvr5aa3blg1z) 1

Nano [setup](#_dn5qfhmecxov) 1

[**Overview**](#_eq0k1tw0l9qb) **2**

[**Topics and Activities**](#_lg1g6up84yjy) **3**

[Topic 1: Use RGB-D Camera](#_7t0w87d2e32w) 3

[Discussion and Check point](#_j25ic61clf18) 5

[Topic 2: Pin hole projection](#_nkxhd3dzloq4) 5

[Topic 2.1: Get point 3D coordinate from depth](#_uno5ncujlavg) 5

[Discussion.](#_hkh0bbgwsnz5) 6

[**Bonus. Analyze detected object**](#_nu6i5adk81ls) **8**

[Task 1: Detect multi objects and draw contours](#_qgwfheu4lu0) 8

[Task2: Determine object shape](#_qgwfheu4lu0) 8

[Task3: Calculate object size](#_b86qsgjvrcw) 9

[**Reference**](#_sxcm0uqez0at) **10**

# 

# Overview

Estimated Time to Finish: 1.3 hours

After completing this tutorial you should

* be able to
  + run RealSense Camera
  + use ROS topics from RealSense Camera
  + use OpenCV
* know
  + how pointcloud is generated.
  + what is Morphological Image Processing
  + what is camera matrix
  + what is HSV color space

# 

# Topics and Activities

## **Topic 1: Use RGB-D Camera**

Open a terminal and SSH into Nano.

**Laptop$ ssh arg@ [your nano hostname].local**

**Nano$** **export DISPLAY=:0 && xrandr --fb 1800x900 ( Resolution depends on your screen)**

**Nano$ /usr/lib/vino/vino-server**

Open the other terminal

**laptop$ vncviewer -quality 5 -encodings "tight" [your nano hostname].local**

Open a terminal in Nano

**Nano$ xhost +**

**Nano$ source ~/sis\_lab\_all\_2020/05-Opencv\_and\_Depth\_Sensing/Dockerfiles/docker\_run\_nano.sh**

**Nano Container$ source devel/setup.bash**

**Nano Container$ roslaunch realsense2\_camera rs\_rgbd.launch**

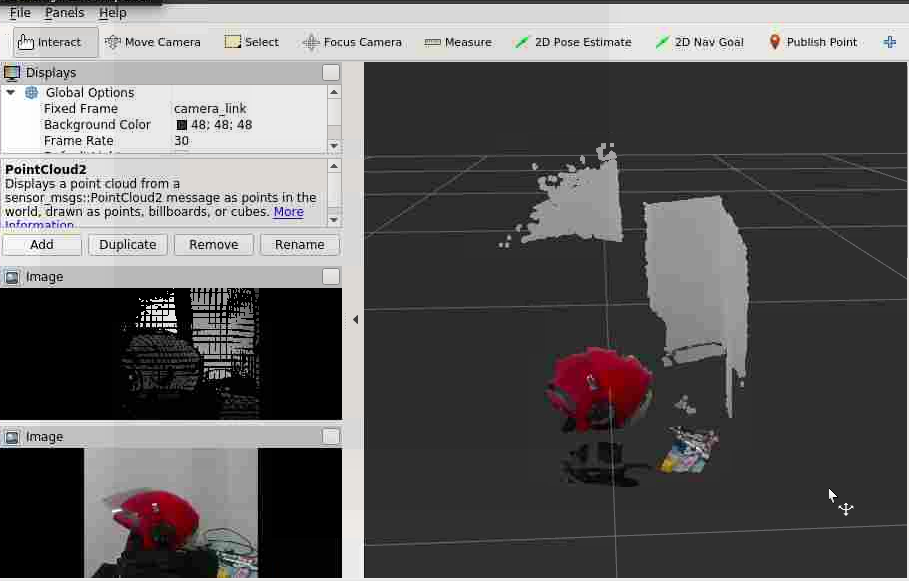
Open the second terminal in Nano

**Nano$ rviz**

Please set fixed frame as camera\_link

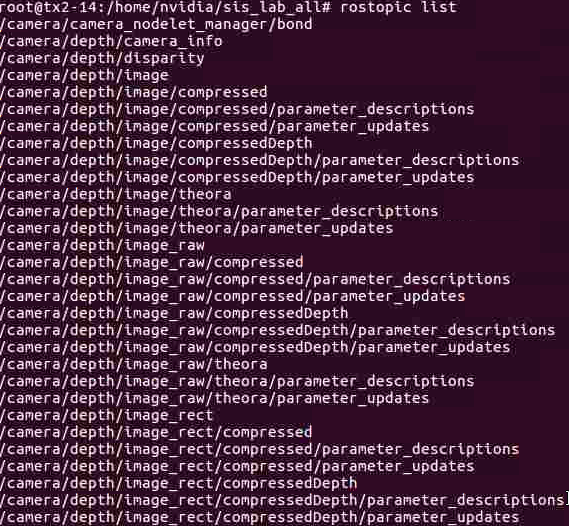
**Add topic “/camera/depth\_registered/points”, “/camera/aligned\_depth\_to\_color/iamge\_raw/raw”, “/camera/color/image\_raw/raw”**

**You need to modify fixed frame from map to camera\_link, otherwise, you won’t see the point cloud.**

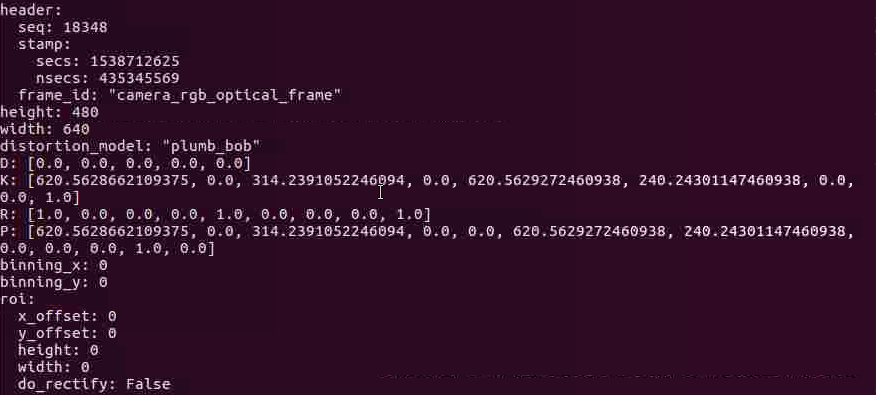
****

Open the third terminal in Nano

**Nano$ docker exec -it lab5 bash (Enter container bash)**  
**Nano Container$ rostopic list (List all ROS topics)**

****

**Nano Container$ rostopic echo /camera/color/camera\_info**

****

### 

### Discussion and Check point

1. Show depth image, rgb image, and pointcloud in RVIZ
2. Please explain the definitions of image rectification and registration and compare them to the ROS image topic “**/camera/aligned\_depth\_to\_color/image\_raw/raw**” and “**/camera/depth/image\_raw/raw**” (hint: <http://wiki.ros.org/rgbd_launch>)
3. Please explain the data in topic ”**/camera/color/camera\_info”.** You only need to explain “D”, “K”, and “R”(hint: [sensor\_msgs/CameraInfo Messag](http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html) , [https://silverwind1982.pixnet.net/blog/post/153218861-pinhole-camera%3A-相機校正-%28camera-calibration%29](https://silverwind1982.pixnet.net/blog/post/153218861-pinhole-camera%3A-))

Note: After you finished topic 1, please exit docker container, so the docker container will be automatically removed.

## Topic 2: Pinhole projection

## Topic 2.1: Get point 3D coordinate from depth

Suppose we have a depth image, the value of each pixel represents the real depth between camera and the captured point in the real world. Now we get random pixel of the image. With algorithm 1, we can get 3D coordinate from point depth.

Algorithm 1

|  |  |
| --- | --- |
| Pin hole projection illustration. | Transform matrix |

Open a terminal in Nano vncviewer

**Nano$ source ~/sis\_lab\_all\_2020/05-Opencv\_and\_Depth\_Sensing/Dockerfiles/docker\_run\_nano.sh**

You can go to checkout the information in this file

**Nano Container$ source devel/setup.bash**

**Nano Container$ roslaunch realsense2\_camera rs\_rgbd.launch**

Open a terminal in Nano vncviewer

**Nano$ cd ~/sis\_lab\_all\_2020/05-Opencv\_and\_Depth\_Sensing/catkin\_ws**

**Nano$ catkin\_make**

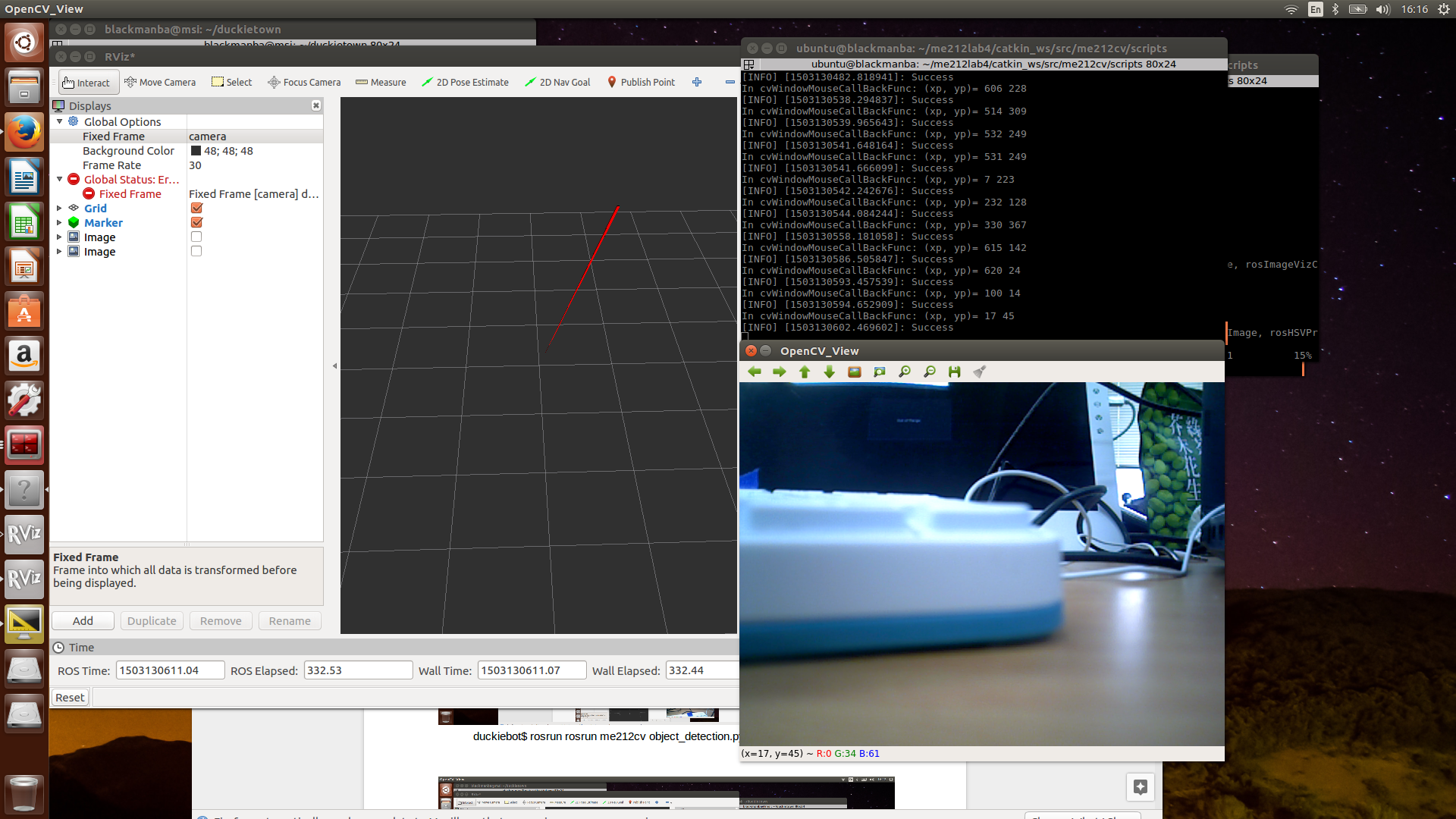
Please ignore the error

**Nano$ source devel/setup.bash**

**Nano$ rosrun me212cv object\_detection.py**

You will see the the following image:

OpenCV View: RGB image



### Discussion.

1. Show the results of the topic 2.1
2. Please explain the code in function “getXYZ”
3. Given a depth image, in image(20,20), the value is 0.5. Please use algorithm 1 and parameters from ROS topic “/camera/color/camera\_info” to calculate the real world coordinate (x,y,z)

### Topic 3.1: HSV filtering

HSV filtering is a common and easy method to detect colored objects. In topic 3.1, we will filter red object by using HSV color space. Then, we will use morphology image process skills to denoise and make mask complete. Finally, with separate object mask, we can get object contour and its bounding box.

In Nano container, change the variables in 05-Opencv\_and\_Depth\_Sensing/catkin\_ws/src/me212cv/scripts/object\_detection.py “useHSV” as True, “useDepth” value as False

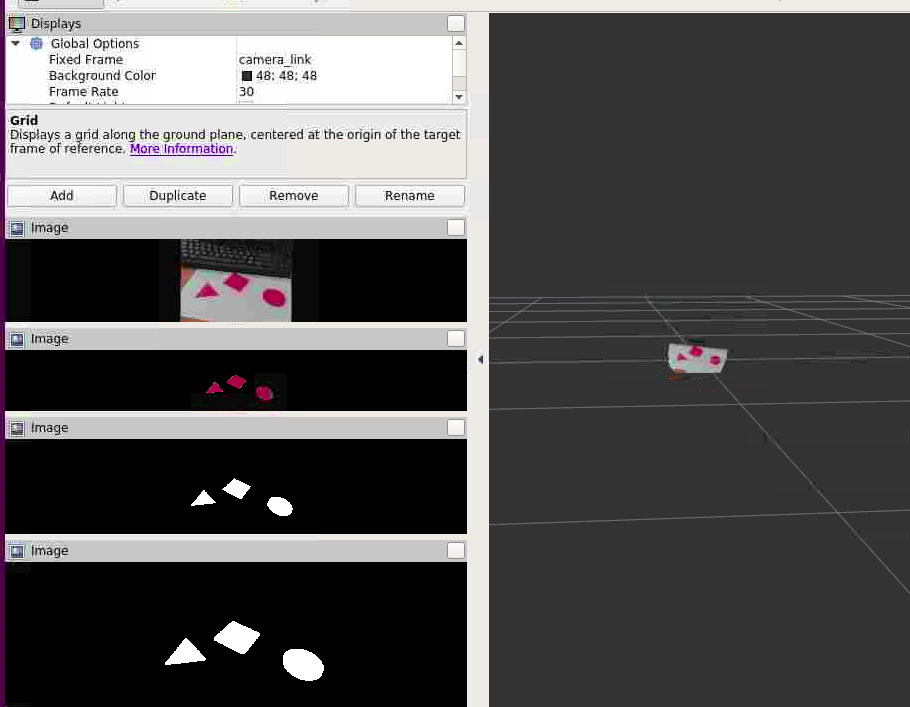
Open a camera launch file in Nano container

**Nano Container$ rosrun me212cv object\_detection.py**

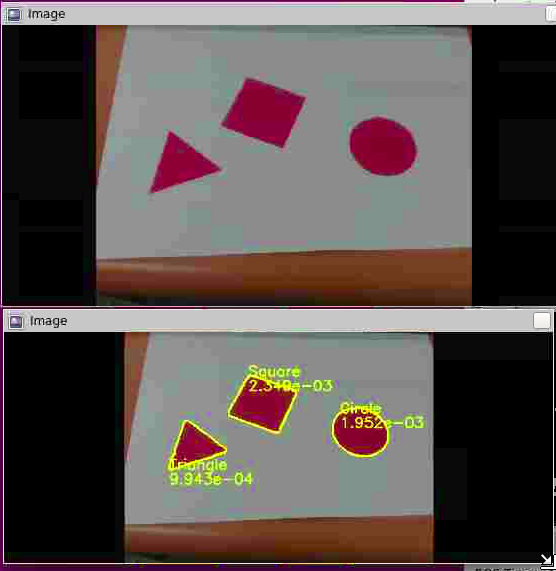
Add the topic “**/camera/color/image\_rect\_color**”, “**/object\_detection/mask\_eroded**”, “**/object\_detection/mask\_eroded\_dilated**”, “**/object\_detection/img\_result**”

You wiil see the the following image:

Rviz: the detected object



# Bonus. Analyze detected object



In tutorial, we are able to detect objects by using HSV color space. In bonus, we will start to analyze objects. Each group will be given a paper where red square, equilateral triangle, circle are printed. Our mission is to detect each object, get the shape, and calculate the area.

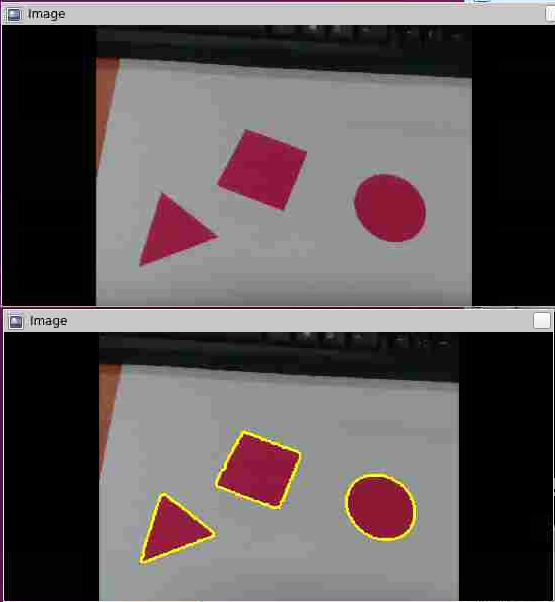
In task 1, we will detect objects by using the same method as tutorial, but we will draw the contours instead of bounding box.

In task 2, the shape of object needs to be determined. That is, you need to get the number of object vertex. Now you have had the contours of the object. All you need to do is approximate the contours as polygon. (hint: use opencv function **approxPolyDP**)

In task 3, after determining the object shape, we can know which formula to use to calculate object area. Your mission is to get side length or radius and then you can use the already prepared function to calculate object area. (You don’t need to implement algorithms for calculating area.)

## Task 1: Detect multi objects and draw contours

In task 1, you will revise object\_detection.py and finish line 144 - 148. You also need to set variable “draw\_contours” as “True” if you want to demo.



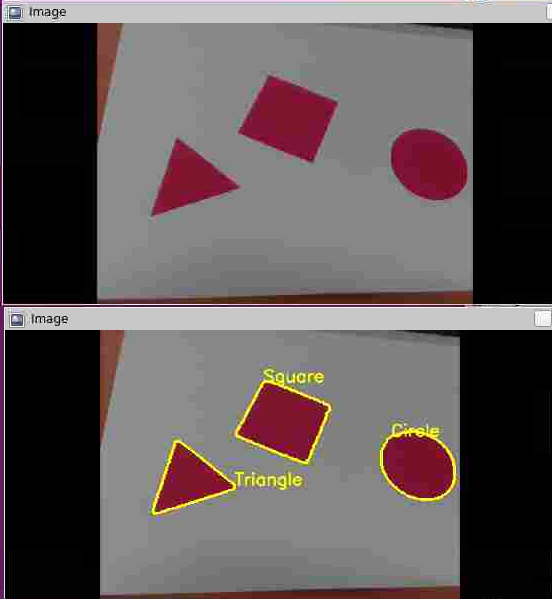
Topic “/object\_detection/img\_result”

## 

## 

## Task2: Determine object shape

In task 2, you need to detect object shape for each object (contour). You also need to set variable “draw\_contours” and “detect\_shape” as “True” if you want to demo.



Topic “/object\_detection/img\_result”

## 

## 

## 

## Task3: Calculate object size

In task 3, we offer the methods to calculate area. If you want to use the method TA offers, you will need to assign vertex list of single object to variable "vtx\_list" and finish functions "get\_side\_length", "get\_radius". You can also write your own code to calculate area. If so, please ignore and comment the line 193 - 218.

You also need to set variable “draw\_contours”, “detect\_shape”, and “calculate\_size” as “True” if you want to demo.

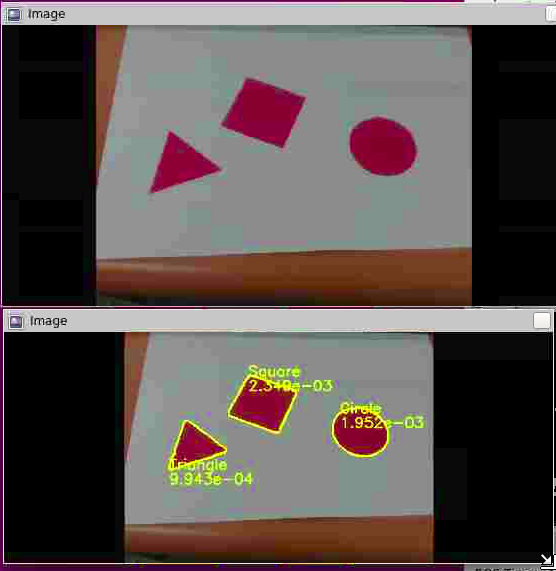
**(Note: The area doesn’t need to be the same as expected result. The result will be regarded to be successful if error is lower than 10%.)**

**Groundtruth:**

**Equilateral triangle: 0.0010825m^2**

**Square: 0.0025 m^2**

**Circle: 0.0019635 m^2**



Topic “/object\_detection/img\_result”

## Grading rule

The homework template will be announced on New E3

# Reference

* <https://www.jetsonhacks.com/2019/04/10/jetson-nano-use-more-power/>
* <http://people.csail.mit.edu/peterkty/teaching.htm>
* <http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=calibratecamera>
* <http://wiki.ros.org/rgbd_launch>
* <http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html>