I/ Introduction & Problem Statement

In this project, I aim to quickly read an image then detect a red barrel with its distance & centroid.

The approach includes: 1) Annotate training images by hand; 2) Train Gaussian models to segment out the barrel's red pixels; 3) Detect the barrel in test images using region analysis & geometry.

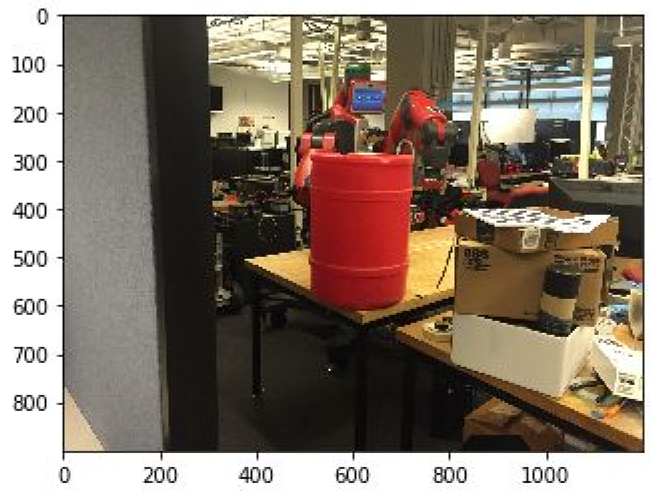
As a result, I successfully detected all red barrels in all test images very quickly, without mis-detecting any other red objects.

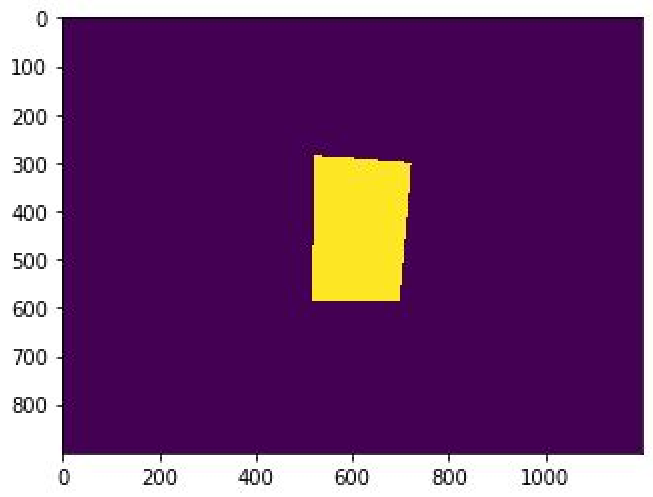
II/ Approach

1) Annotate by hand

I use roipoly tool to annotate the red barrel in training images. The annotation output are image masks that are saved in folder "Annotate".

Refer to file "annotate_barrel.py" for full code. Below is an example of image & mask:

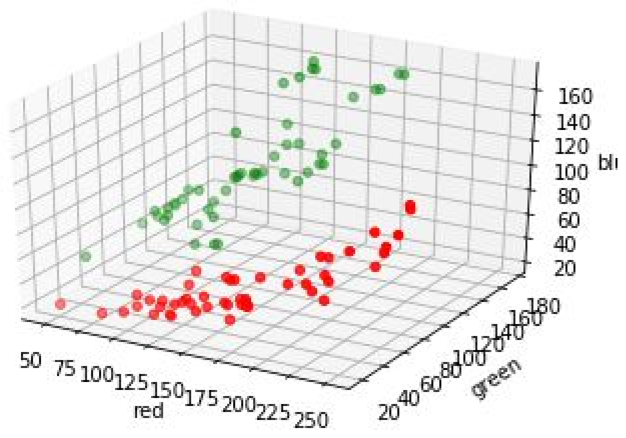




2) Train Gaussian models to segment out red pixels

With the annotated image masks, I have 2 classes of pixels in each training image: i) Barrel pixels; and ii) Other pixels.

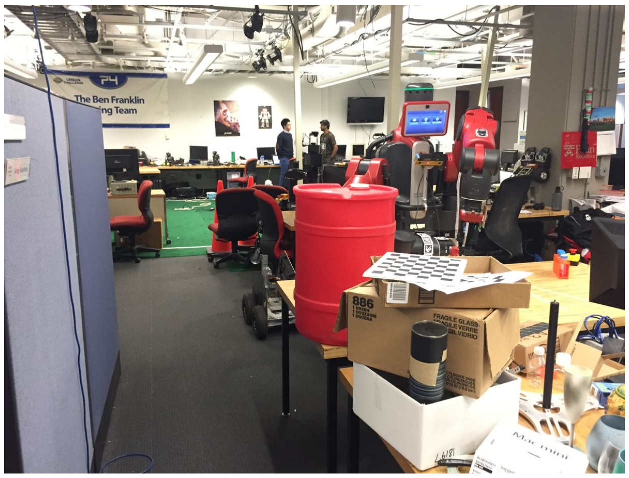
I decide to use RGB color space since the pixels seem to be separated in this 3D space:



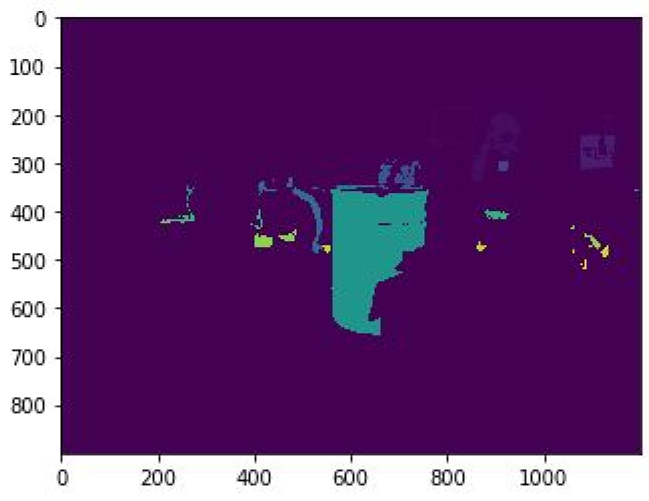
I fit one single Gaussian model to each of the 2 pixel classes, and end up with 2 models: GMM_barrel and GMM_other. The trained models can be found in folder “Model”.

For each pixel in a test image, I calculate the log probability that it will fall into GMM_barrel or GMM_other. The pixel will then be classified as “barrel” or “other” depending on which log probability is higher.

Refer to file “train_barrel.py” for full code. Below is one example of this step:



The red pixels are classified as “barrel” and segmented out. Some red pixels that are not really barrel will also be caught here:

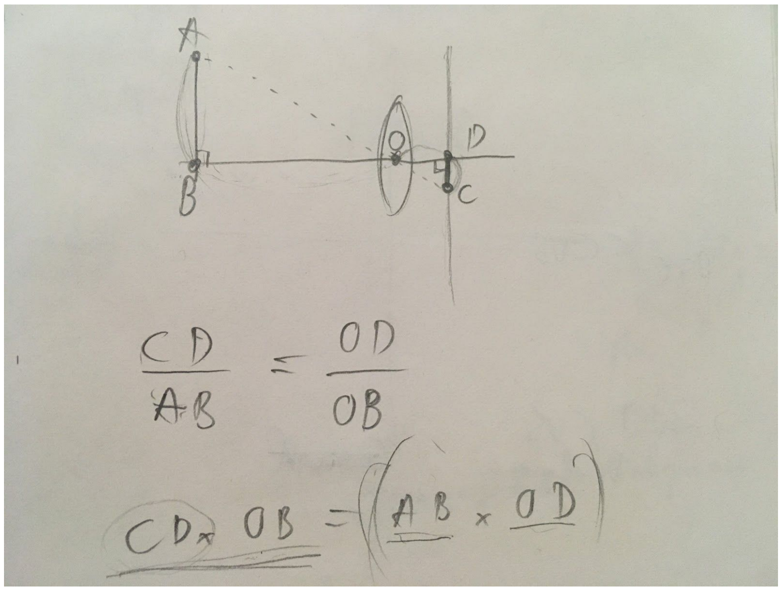


3) Detect barrel using region analysis:

I then use skimage’s regionprops to analyze the geometry of different regions of red pixels. The region will be classified as a barrel if it meets the following criterias:

- The region is mostly solid: region.solidity > 0.7
- The dimension is similar to the barrel: (side_ratio>1.15 and side_ratio<2.6) or (axis_ratio>1.3 and axis_ratio<1.6)

The distance of the barrel can be calculated using geometry:


A
B
O
D
C
$\frac{CD}{AB} = \frac{OD}{OB}$
$CD \times OB = (AB \times OD)$

In the image above, AB is the height of the barrel. O is the center of the camera's lense. CD is the image of the barrel on the sensor. We will have:

CD * OB = AB * OD

Since AB (barrel's height) and OD (distance between lense & sensor) are fixed

→ AB * OD is fixed

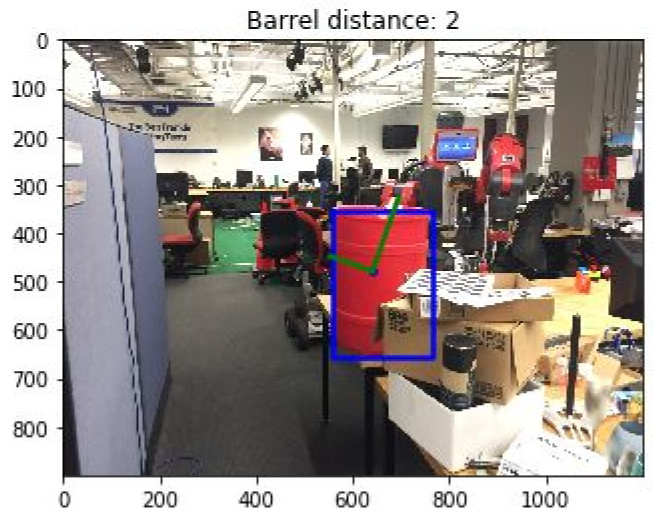
→ CD * OB is fixed

→ the product of barrel's image height * barrel's distance is fixed

Now we just need to calculate this product from the training images. Then use that product to calculate the barrel's distance for any new test image. The final formula is:

Barrel distance = 650 / barrel's image height (in pixels)

Refer to file "detect_barrel.py" for full code. Below is one example after this step:



III/ Result

The final result of test images can be found below:

Image: 001.png
Barrel distance: 2
Centroid X: 641.0; Centroid Y: 479.0

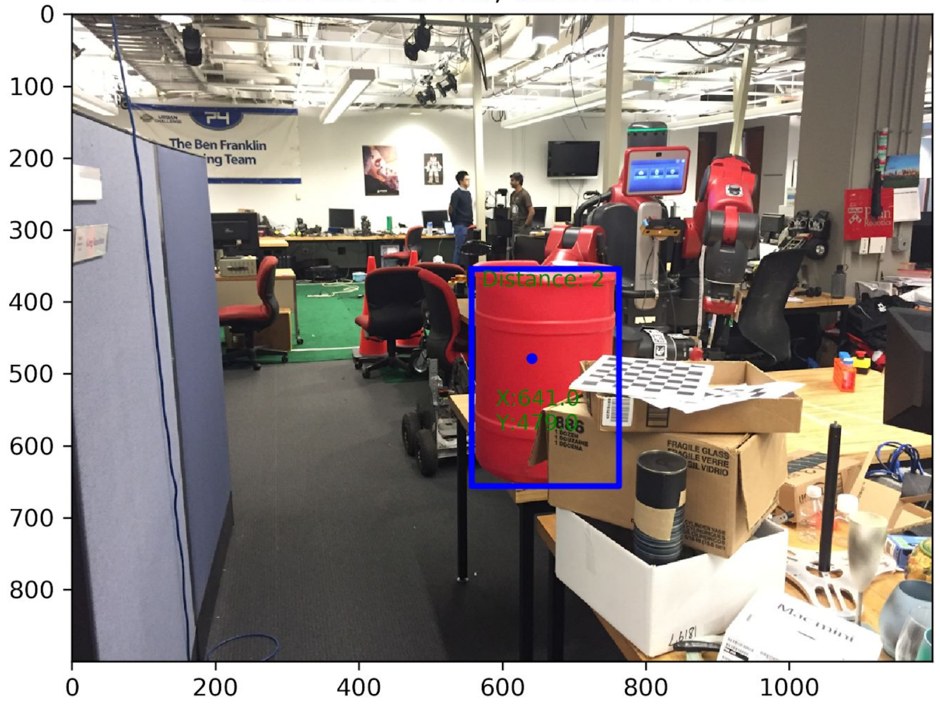


Image: 002.png
Barrel distance: 3
Centroid X: 631.0; Centroid Y: 514.0

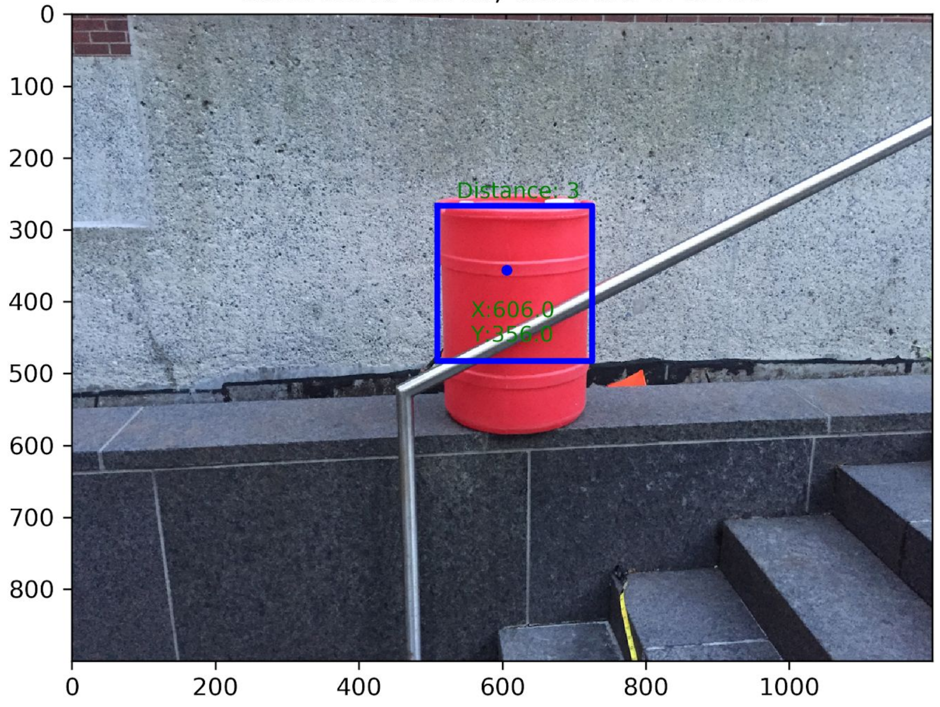


Image: 003.png
Barrel distance: 3
Centroid X: 699.0; Centroid Y: 449.0

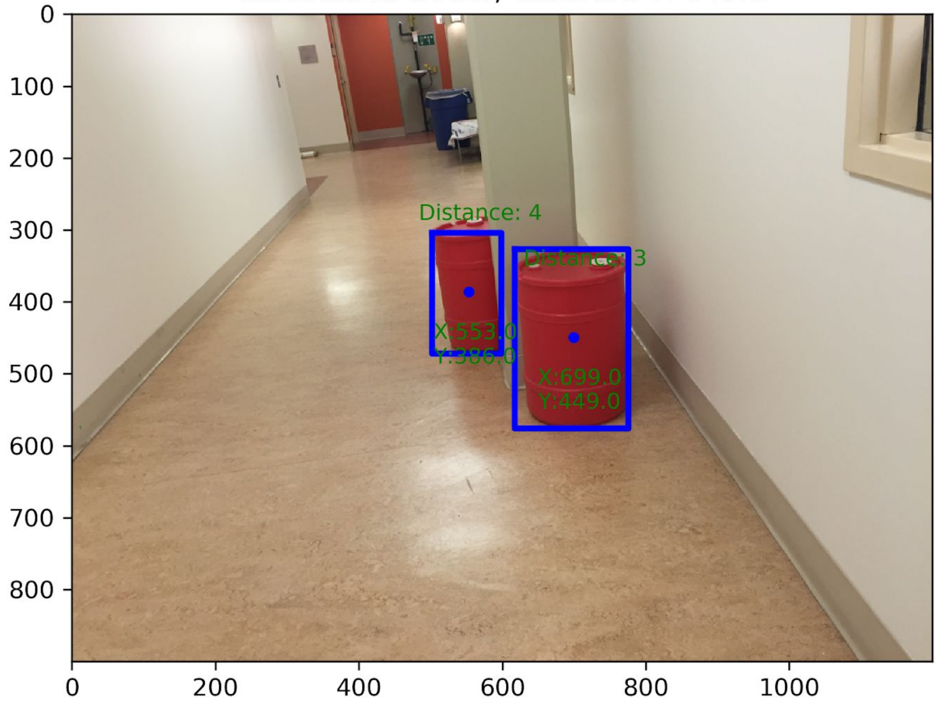


Image: 004.png
Barrel distance: 2
Centroid X: 634.0; Centroid Y: 445.0

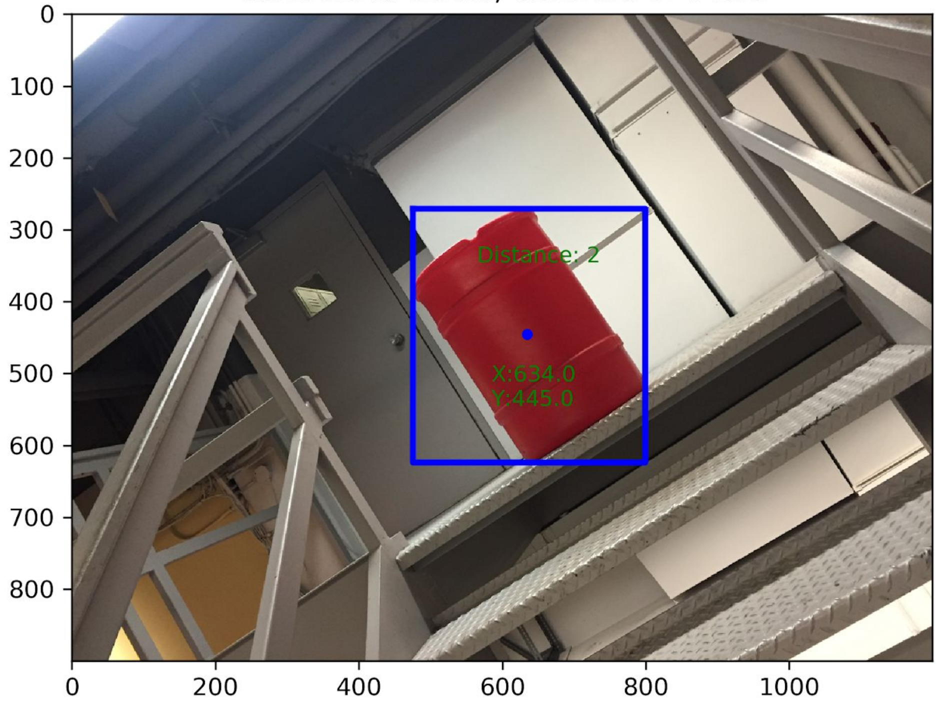


Image: 005.png
Barrel distance: 6
Centroid X: 644.0; Centroid Y: 469.0

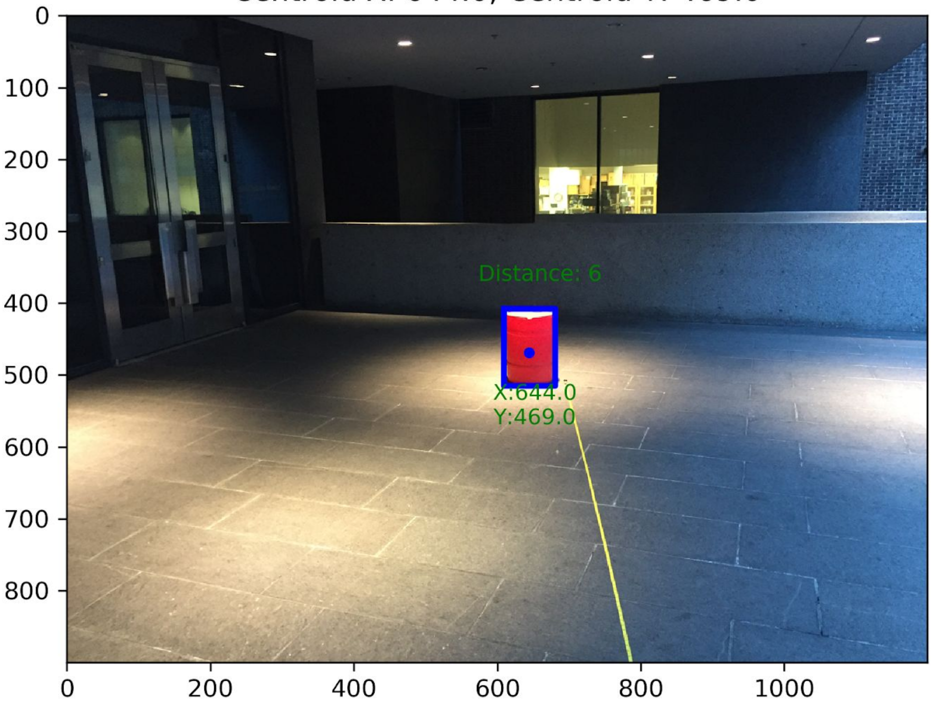


Image: 006.png
Barrel distance: 10
Centroid X: 632.0; Centroid Y: 427.0

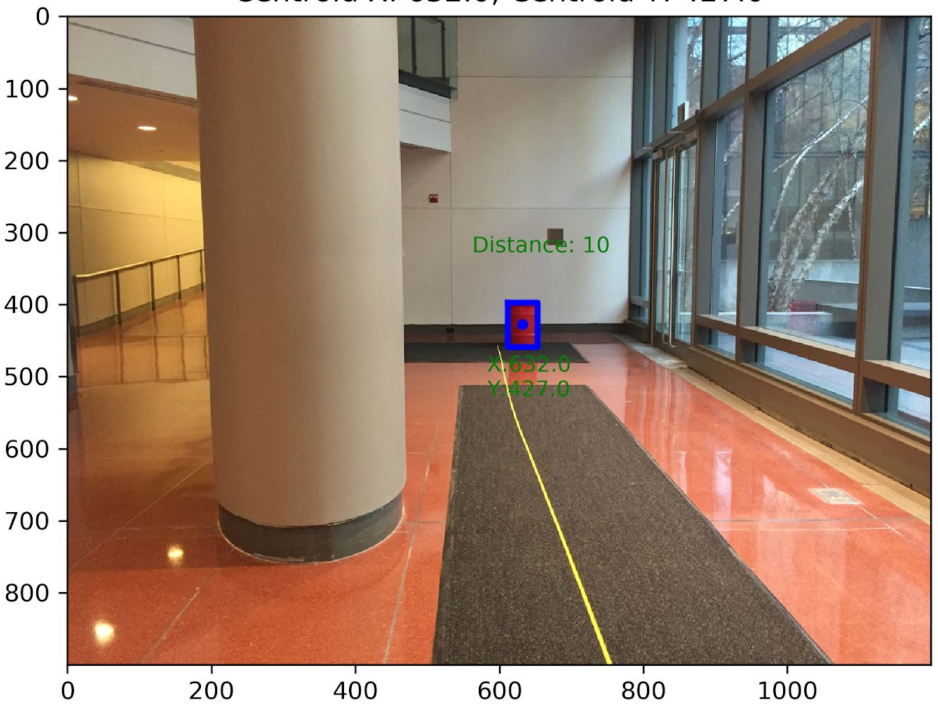


Image: 007.png
Barrel distance: 3
Centroid X: 568.0; Centroid Y: 628.0

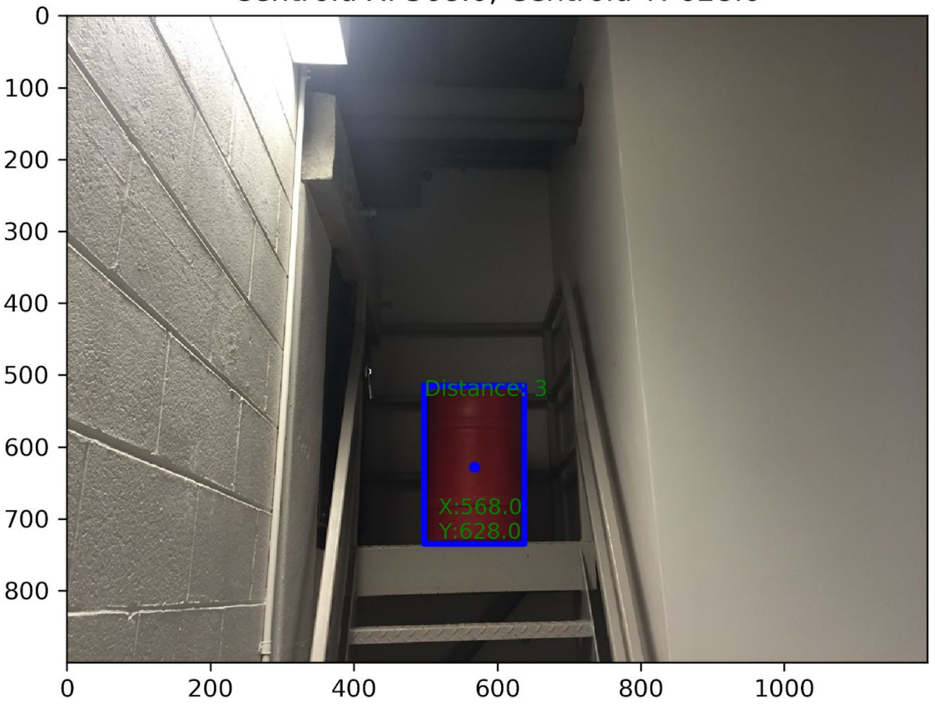


Image: 008.png
Barrel distance: 8
Centroid X: 738.0; Centroid Y: 479.0

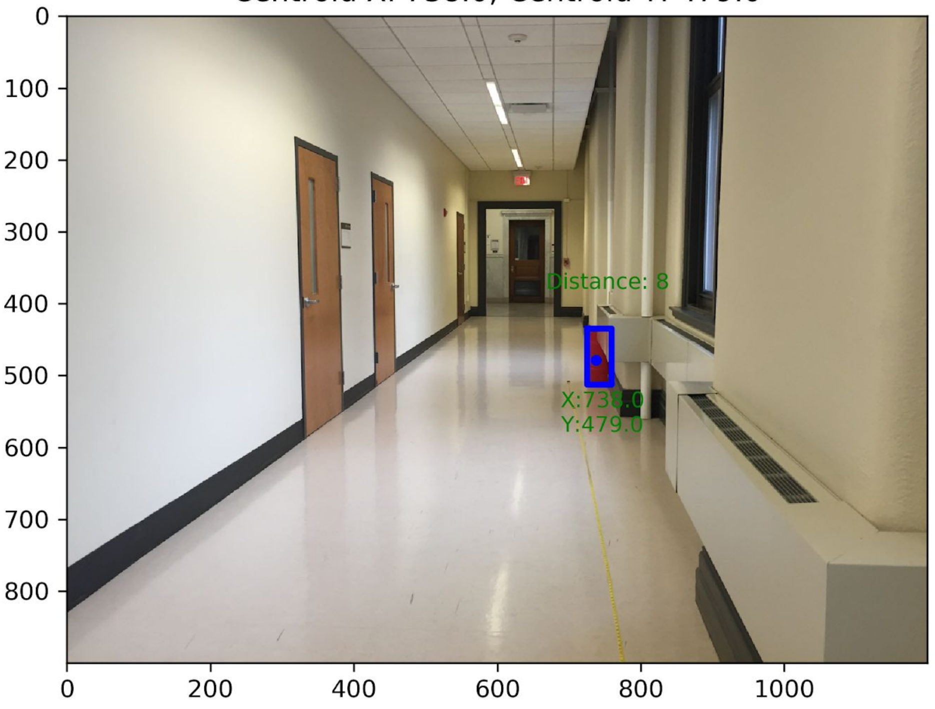


Image: 009.png
Barrel distance: 11
Centroid X: 673.0; Centroid Y: 428.0

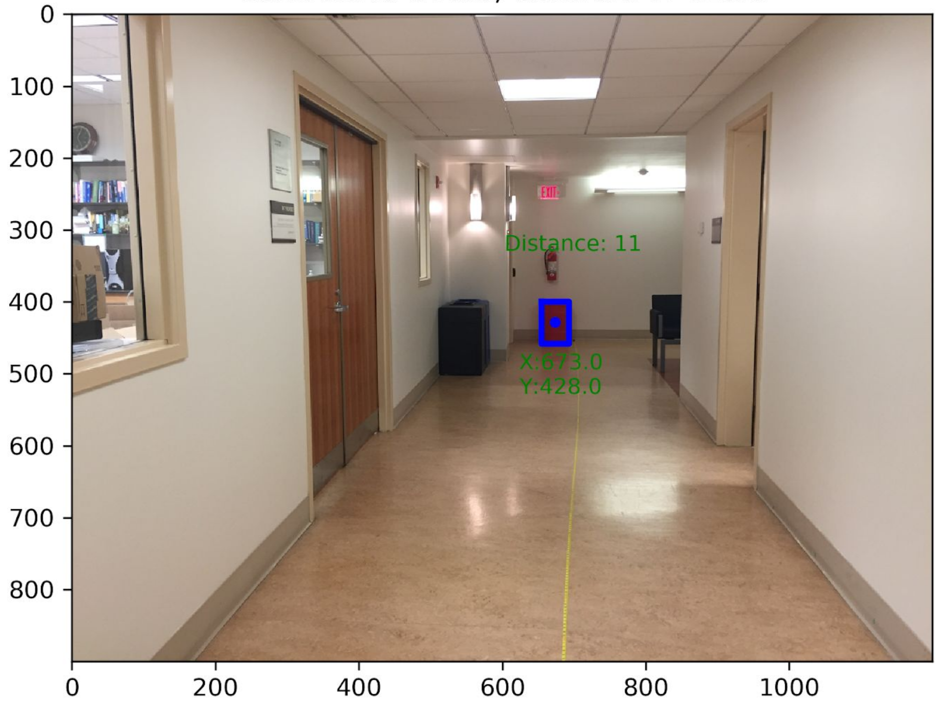
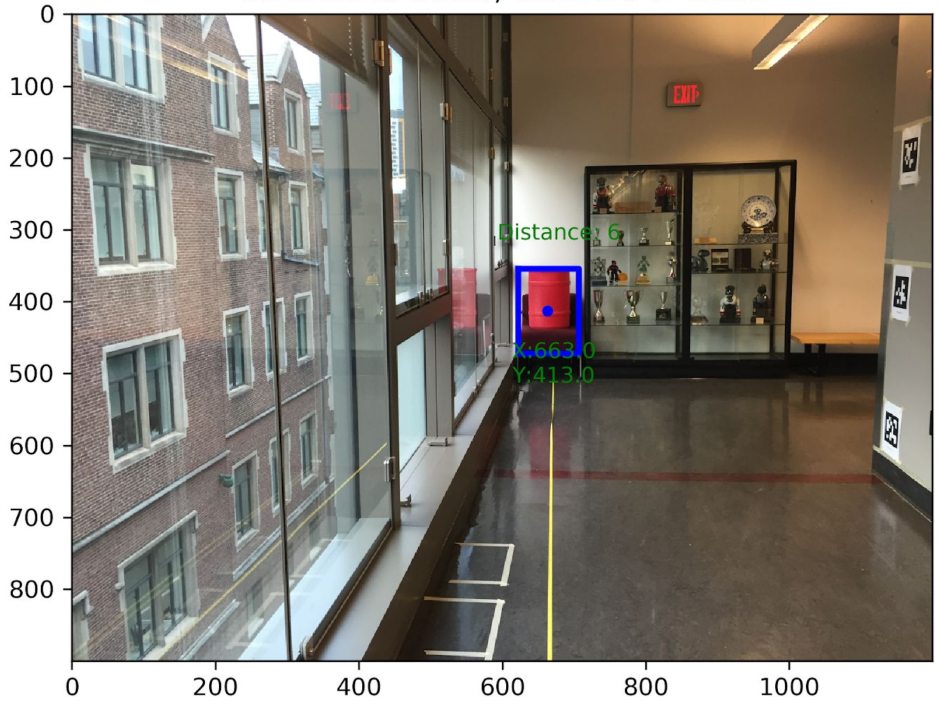


Image: 010.png
Barrel distance: 6
Centroid X: 663.0; Centroid Y: 413.0



IV/ Discussion:

Future improvements to this project include:

- Better region analysis to detect barrel that is behind a smaller object. As seen in test image 002.png, the current algorithm only detects half of the barrel, since the barrel is behind a steel pipe.
- Better feature analysis to avoid mis-detect non-barrel red objects. In test image 001.png, there are many red objects around the barrel. While the current algorithm successfully avoids mis-detection for that image, another red & solid object may easily fool the algorithm into thinking that it’s a barrel.