Lianming Shi
A99097650
ECE 276A
HW1

Project 1 Report

Introduction

In this project, I am supposed to extract useful data of the red barrel and other colors' from the given pictures by using roipoly function. The resulting information, which includes red barrel area, barrel distance, colors, and etc., generated a database for me to classify a testing picture. By using the database I got from previous training pictures, I trained a Gaussian Mixture Model to find where the barrel is located and then I took out the red region by using Find Contour function. At last, I trained a linear regression model to approximate the barrel's distance.

Problem Formulation

I trained Gaussian Mixture model to get several colors' probability and then get a mask picture for red barrel.

Gaussian Mixture:

$$P_{X|Y}(x \mid i) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_i \mid}} \exp\left\{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right\}$$

I trained linear regression by using the red barrel area I got form the sample images, and use the resulting linear parameters to predict distance in testing images.

Linear Regression:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$$

Technical Approach

Color Segmentation

(1) Getting data:

At first, I used roipoly function to hand label the given image set. Then, I used getmask function to get a binary mask image that only contains 'True' and 'False'. True is my wanted region, so I used numpy where to get the position of 'True' in the mask image. I divided the 3D sample image into red[:, :, 0], green = [:, :, 1], and blue[:, :, 2] layers. Then I created a list and only stored those layer's pixels that position == True. After that, I stored the data into a pkl file. After that, I was able to calculate the mean and covariance of each color, such as red, brown, notbarrelred, and yellow. In addition, I also used regionprops to get the red barrel area in each sample image, which was useful for later calculation, such as linear regression.

(2) Building model

I trained a Gaussian Mixture model by using the mean and covariance of colors. Then, I was bale to use the Gaussian Mixture formula for matrix to get the Gaussian probability for each color.

(3) Segmentation Part

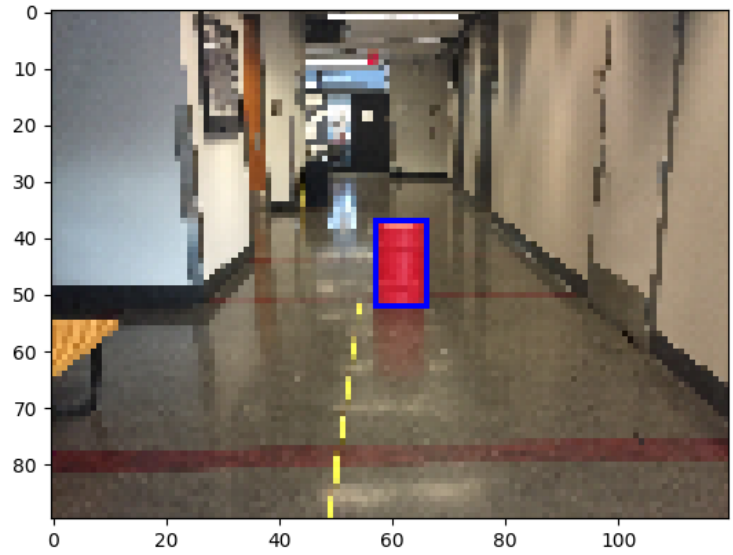
Because the execution time for a normal image is too long, I divided the test images sizes by 10. I use the trained Gaussian Mixture here to find the probability of each color. If the red color's probability of a pixel is the highest, I set the pixel equal to 1. If one of other colors' probabilities of a pixel is highest, I set the pixel equal to 0. After doing this, I got a black-white image that only show red barrel on it. I wrote a for loop to segment the original image based on the black-white image. At last I got the segmented image.

Barrel detection

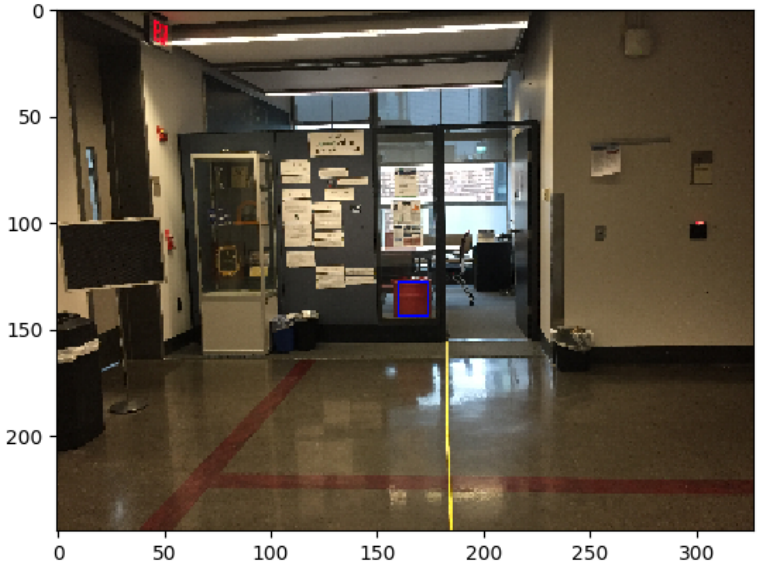
I convert the segmented image into gray scale and used cv2threshold to find the threshold value. I used getcontour to find the contour of the image. Then, I used cv2.contourArea to find areas in each contour. My algorithm was to compare every area in the image and the contour that has max areas will be my red barrel. Moreover, I used cv2.minAreaRect and cv2.drawcontour to get and to draw a bounding lox for my red barrel. I used cv2boxPoints function to get the four corners, and I wrote a algorithm to find top-right and bottome-left coordinates.

Result

Overall, my algorithm works well on most test images. However, there is still several pictures that I cannot draw a bounding box on the barrel. Such as image#3 vending machine with barrel and image #6 barrel in an elevator and image#8 barrel with stairs. I may need larger database and more algorithms to analyze those tricky pictures. Thus, next time I will spend more time on how to enlarge my database and on improving my barrel-detecting algorithm.

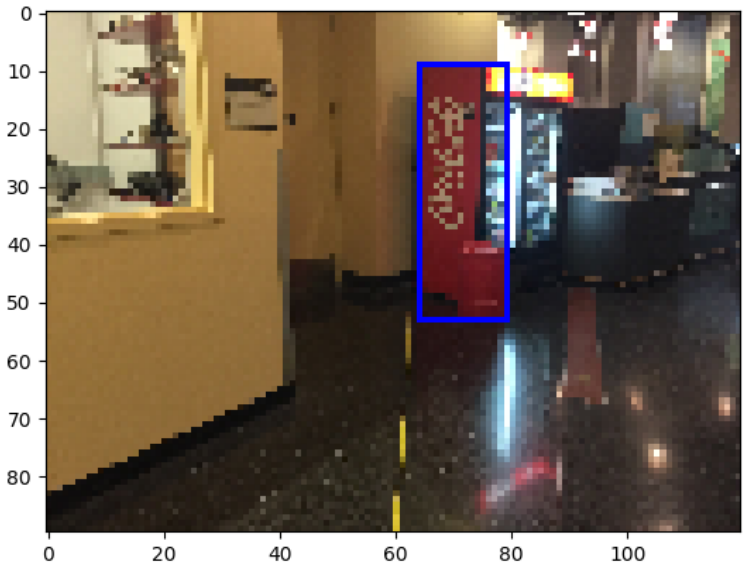


ImageNo = [01], BottomLeftX=57, BottomLeftY=52, TopRightX=66, TopRighty=37 , Distance= 5.7434913

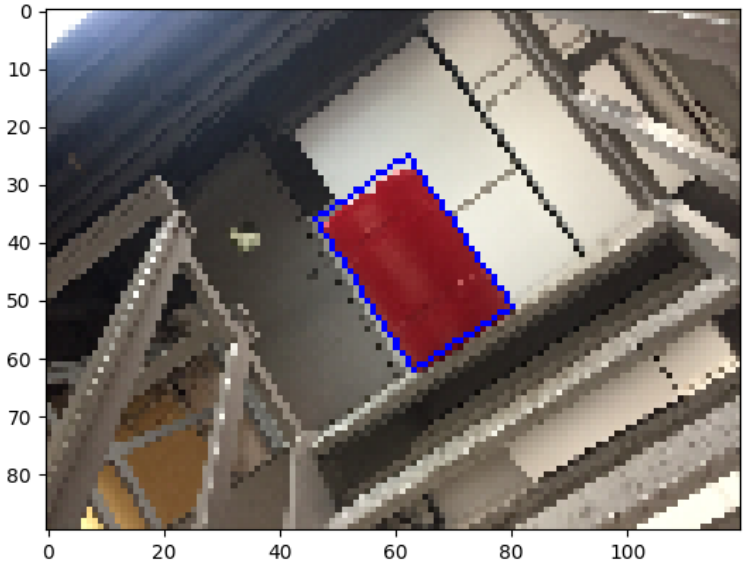


ImageNo=[02],BottomLeftX=160,BottomLeftY=144,TopRightX=174,TopRighty=

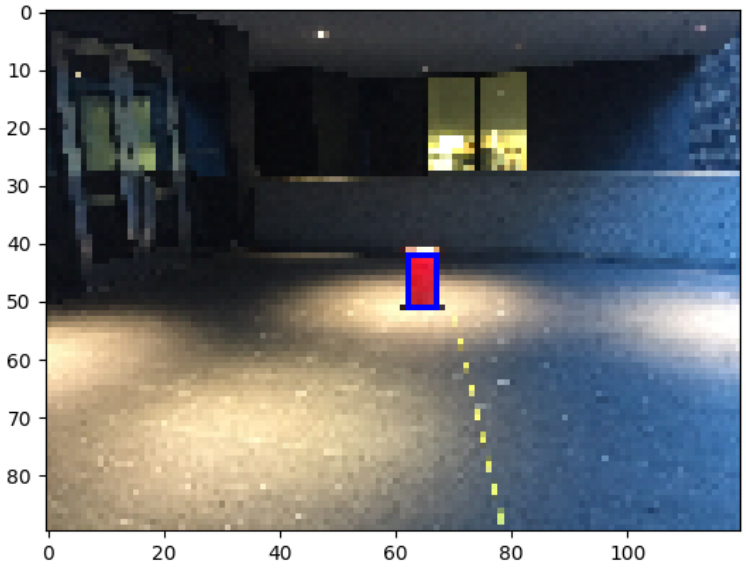
128 , distance= 5.85811532



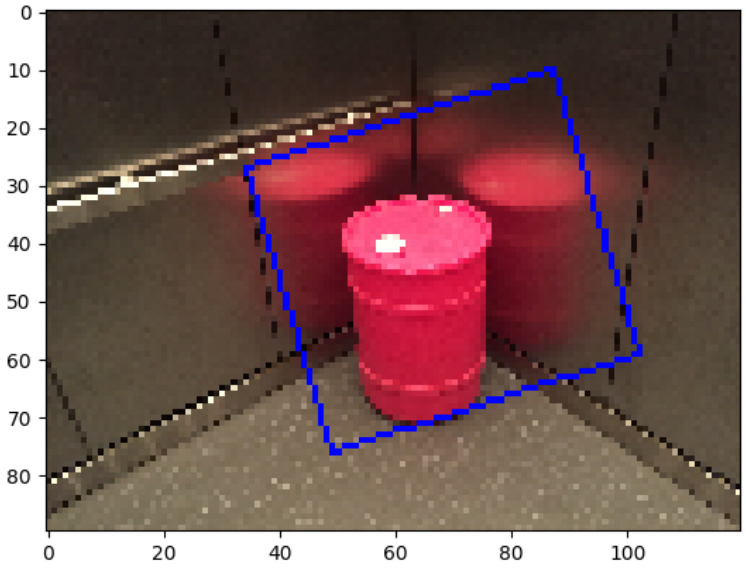
ImageNo = [03], BottomLeftX=64, BottomLeftY=53, TopRightX=79, TopRighty=9 , Distance= 2.67854467



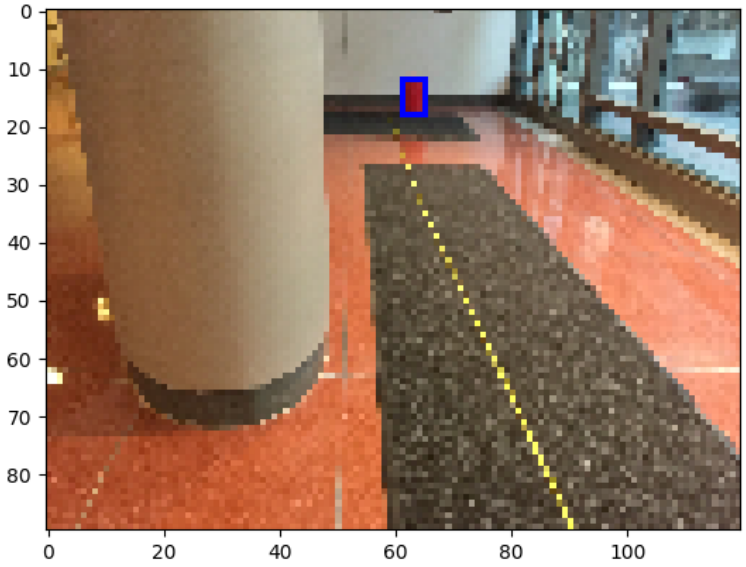
ImageNo = [04], BottomLeftX=63, BottomLeftY=63, TopRightX=62, TopRighty=25, Distance= 1.36286027



ImageNo = [05], BottomLeftX=62, BottomLeftY=51, TopRightX=67, TopRighty=42, Distance= 6.43123542

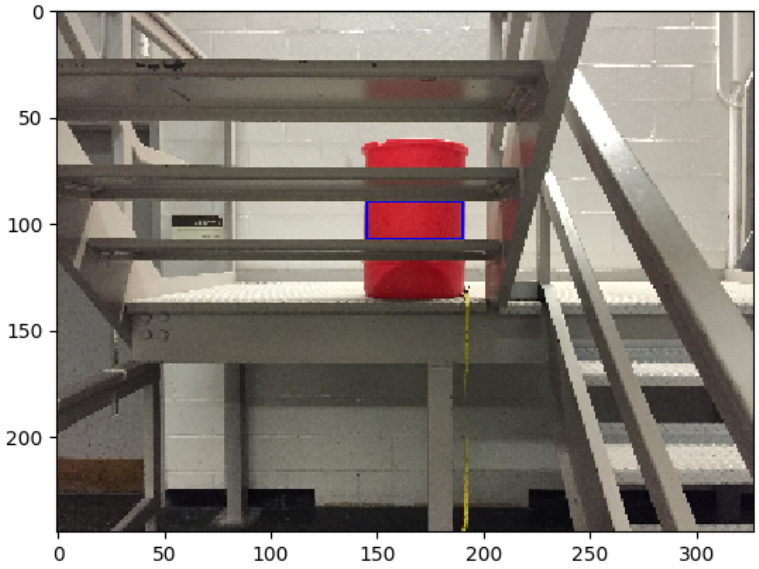


ImageNo = [06], BottomLeftX=49, BottomLeftY=76, TopRightX=87, TopRighty=10, Distance= -11.61457228

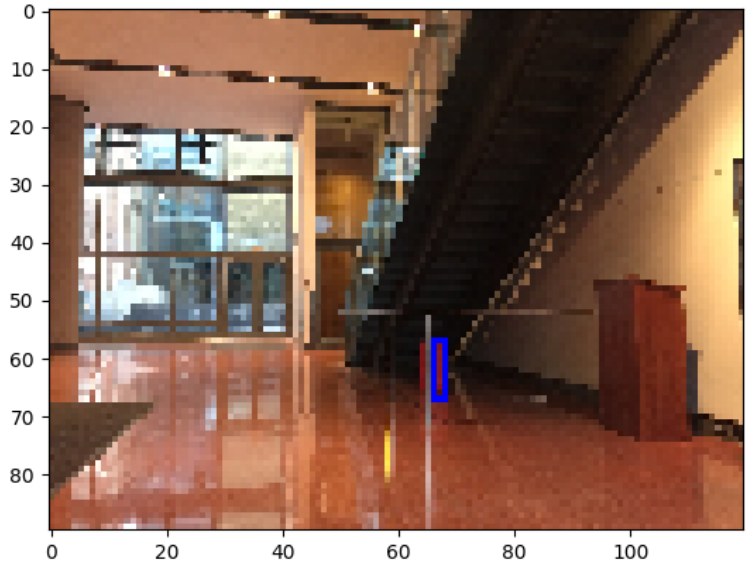


ImageNo = [07], BottomLeftX=61, BottomLeftY=18, TopRightX=65,

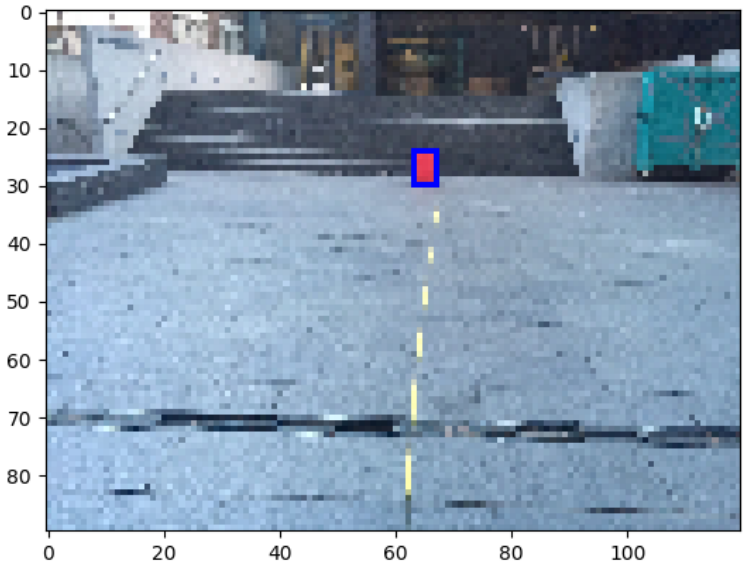
TopRighty=12, Distance= 6.63058154



ImageNo = [08], BottomLeftX=145, BottomLeftY=107, TopRightX=190, TopRighty=90, Distance= -0.64056826



ImageNo = [09], BottomLeftX=66, BottomLeftY=67, TopRightX=68, TopRighty=57, Distance= 6.69038538



ImageNo = [10], BottomLeftX=63, BottomLeftY=30, TopRightX=67, TopRighty=24, Distance= 6.66546711