Project 3 Report

ENPM 673 - Perception for Autonomous Robots

Team 5

Prof. Dr. Mohammed Charifa

UNIVERSITY OF MARYLAND

Nikhil Lal Kolangara (116830768)
Kartik Venkat (116830751)
Kushagra Agrawal(116700191)

This paper represents our own work in accordance with University regulations.

Contents

1 Introduction

The aim of this project is to understand and implement the segmentation process in image processing of a color image using Gaussian Mixture Expectation Maximization techniques.

For this project, a video sequence containing three distinctively colored buoys under the water are provided. They are colored yellow, orange and green and circular in shape. Conventional Image Segmentation procedures fail in this scenario mainly because of the varying light intensities and noise.

The project involves learning the color distributions of the buoys and using this learned model to segment them. The segmentation must be kept tight for each buoy for the entire video sequence by applying tight contour around each individual buoy. The three different buoys are trained by using Expectation Maximization algorithm with regards to their color distributions to generate their Gaussian distributions of each channel for individual buoys. Furthermore, the buoys are identified using the probability distribution of each individual color channel's Gaussian distribution for each buoy.

1.1 Data Preparation

A video file is provided that consists the footage of three buoys of different colors under the water. The data set is prepared by collecting samples of each buoy by tightly cropping them from each frame of the video footage.

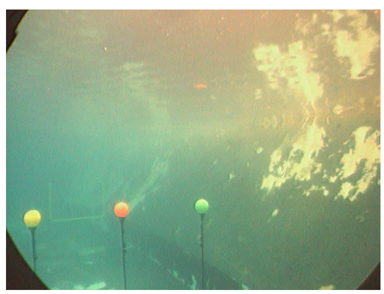


Figure 1: Sample image frame of buoys underwater

Steps Involved:

Step 1: With the help of Matplotlib event handling function, the x and y coordinates of the point where the mouse was clicked is stored in a Numpy array for segmenting the buoys from the image.

Step 2: These coordinates are used to crop the buoy region of interest from each frame of the video. Satisfactory amount of video frames were considered and cropped image of each individual buoy was stored in it's respective folder.

Step 3: The method used to crop the ROI involves using a mask which made the background of the ROI black.

Step 4: Majority of the black background is cropped out before the dataset is used to train the model.

2 Gaussian Mixture Expectation Maximization

2.1 Gaussian Mixture Models

Gaussian Mixture Model in accordance with supervised learning, models the distribution of each class as a set of weighted gaussians. $P(x_i, z_i|c) = P(x_i|z_i, c)P(z_i|c)$, where z_i is a hidden random variable that is not observed. One assumption that is made is that any distribution can be well modelled by a set of gaussians. In the figure below, you can see three weighted gaussians in $\mathbb{R}^1$ and their sum which models the distribution of a random variable. [] It is a function that comprises of various Gaussian distributions identified by k in 1,...,K, where K is the number of clusters of our dataset that is the cropped images of each buoys. Each Gaussian k in the mixture is comprised of the following parameters: A mean μ that defines its centre. A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario. A mixing probability π that defines how big or small the Gaussian function will be.[2]

Th mean and variances of the buoys are calculated using the formula below:

$P_r(x|A) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp \frac{-(x_i-\mu_b)^2}{2\sigma_b^2}$

$b_i = P_r(A|x) = \frac{P_r(x|A)P_r(A)}{P_r(x|A)P_r(A)+P_r(x|\ A)P_r(\ A)}$

$\mu = \frac{b_1x_1+b_2x_2+.....+b_nx_n}{b_1+b_2+....+b_n}$

$\sigma^2 = \frac{b_1(x_1-\mu_1)^2+....+b_n(x_n-\mu_n)^2}{b_1+b_2+....+b_n}$

2.2 Expectation Maximization Algorithm

Expectation-Maximization Algorithms is an iterative method to determine the maximum likelihood estimates of parameters in statistical models. Finding a maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values, the parameters and the latent variables, and simultaneously solving the resulting equations. [1]

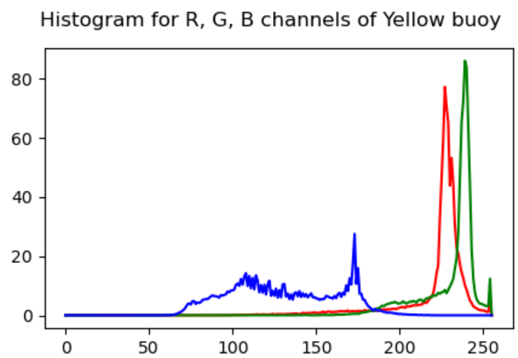
3 Buoy Detection Pipeline

Buoy Detection 1 *Implement a buoy detection pipeline that will take a frame from the video sequence as input and generate a color-segmented binary image, given the computed model parameters. Also compute the corresponding contours and center of each buoy. Draw the bounding contours around the buoy in the original frame, in their respective colors. Ensure this contour is as tight as possible for highest possible scores. Provide outputs for several non-consecutive frames in your report, and explain your pipeline and approach.*

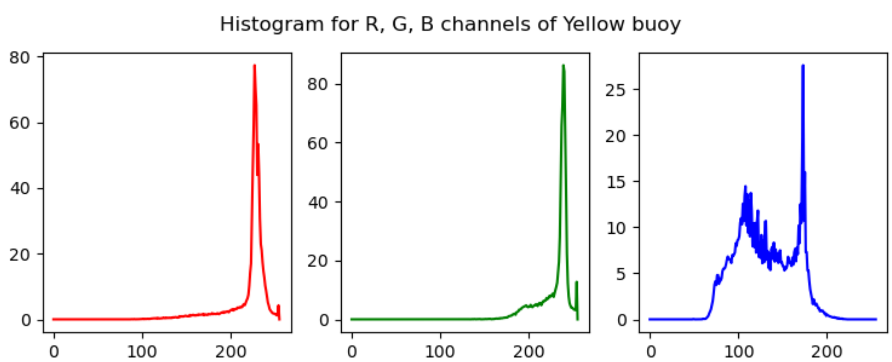
Approach 1 *For the buoy detection, the programmed code follows the given steps:*

Step 1: After the dataset has been generated, the next step is to plot the average histogram.

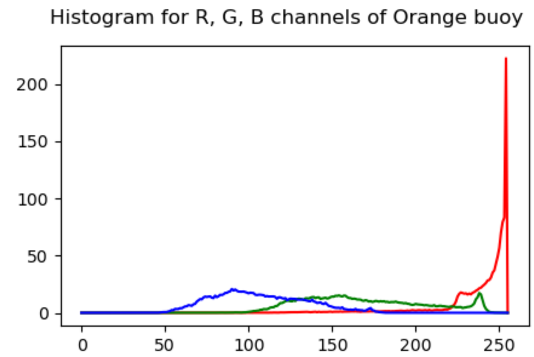
Step 2: To generate the average histogram, the histogram of each image from the dataset is first computed and then the average frequency of each pixel intensity is generated.



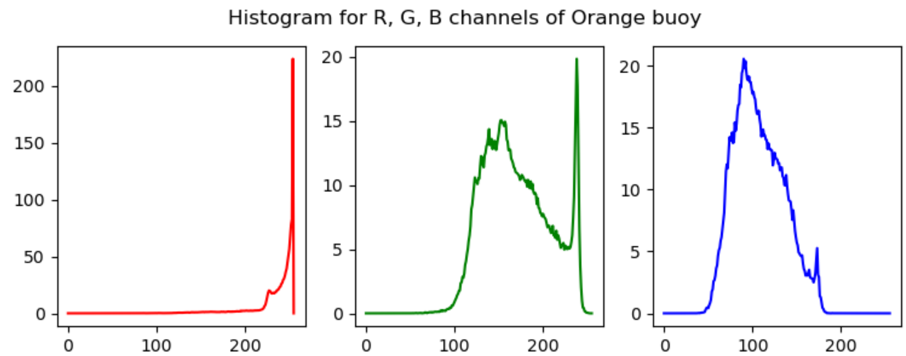
(a) Average Histogram for Yellow Buoy



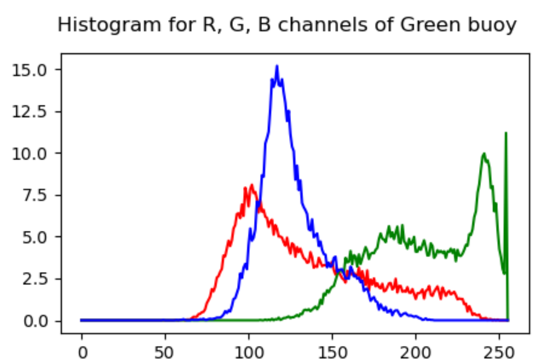
(b) Average Histogram for Yellow Bouy for Individual Channel



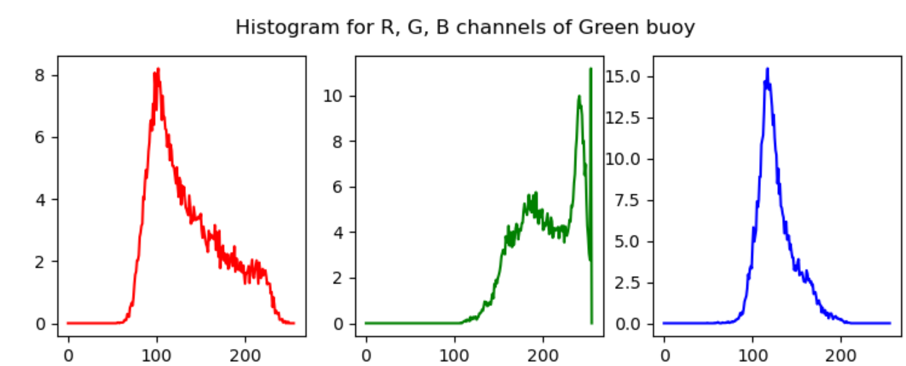
(a) Average Histogram for Orange Bouy



(b) Average Histogram for Orange Bouy for Individual Channel



(a) Average Histogram for Green Bouy



(b) Average Histogram for Green Bouy for Individual Channel

Step 3: The need to model the color distributions for each buoy is necessary so as to segment them from the background consisting of black or higher intensity pixels or noises.

Step 4: The above step was required to determine the number of Gaussian distribution to model the color distributions.

Step 5: The concept of finding the Gaussian Mixture Model using the statistical Expectation Maximization Algorithm is applied.

Step 6: In the Expectation step, the Bayesian probabilities of each color channel for every frame from the input video is computed.

Step 7: In the Maximization step, the new mean and standard deviation is computed as per the formula mentioned in section 2.1.

Step 8: The Expectation Maximization algorithm is run for 50 iterations at the end of which, the model parameters(mean and std) converge to their maximum likely-hood values.

Step 9: The model parameters trained in the previous step are stored in a '.csv' file so that these values can be accessed later to be used as pre-trained model parameters.

Step 10: Finally, we plot the 1D Gaussian Models of all three buoys from the model parameters generated in the previous step.

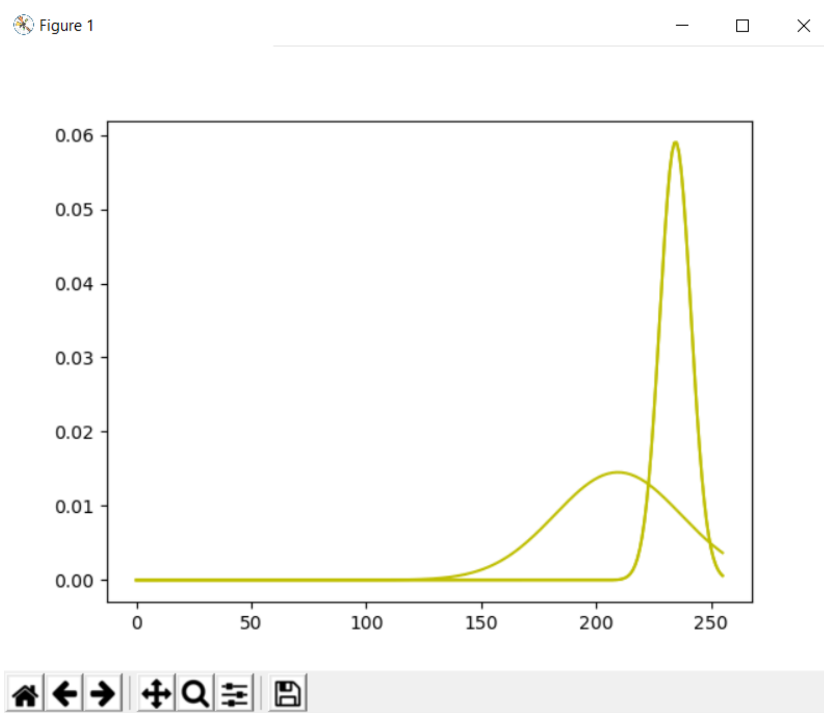


Figure 5: Gaussian Plot for Yellow Buoy

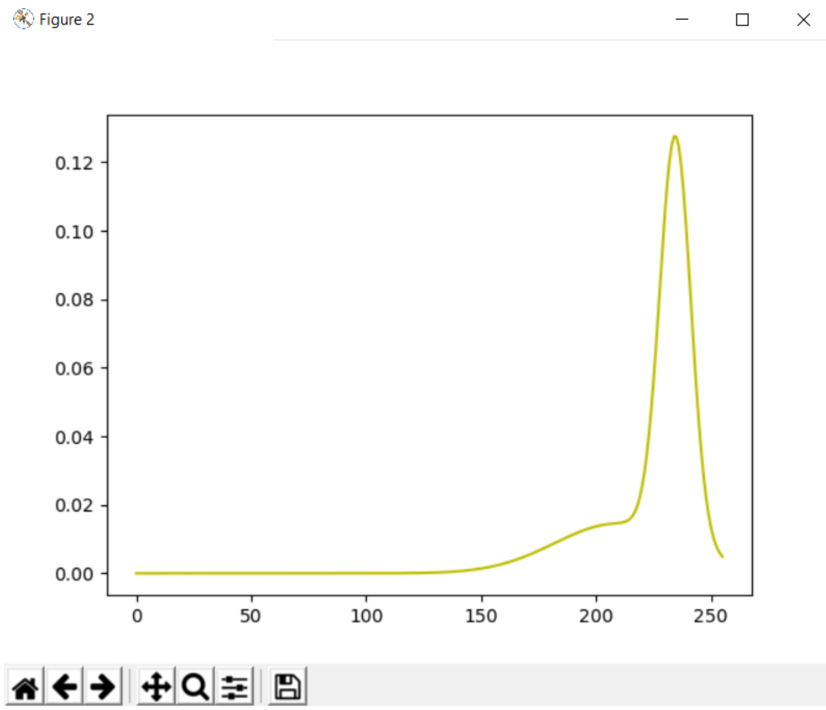


Figure 6: Gaussian Plot for Yellow Buoy for individual channel

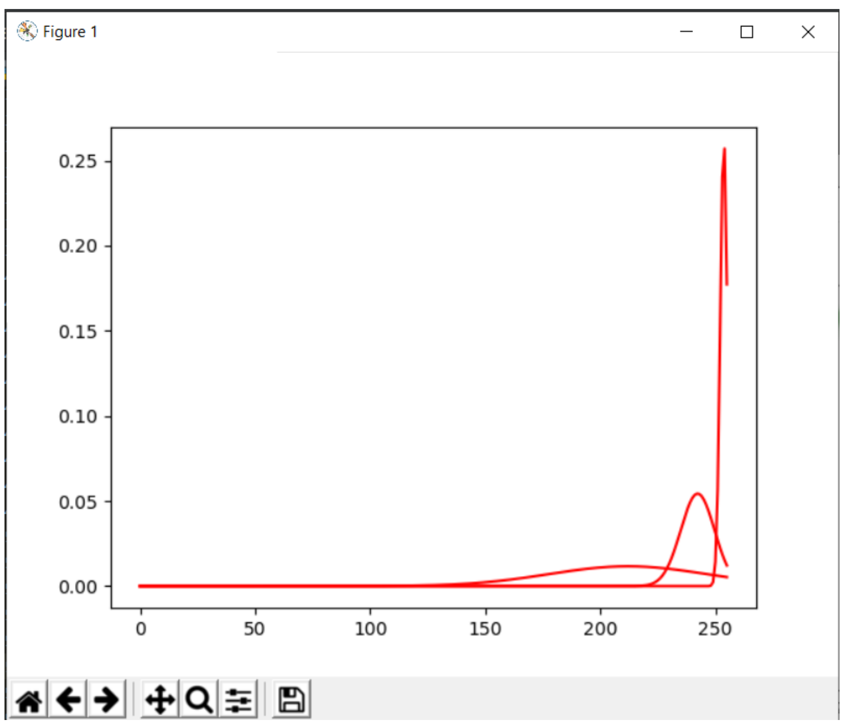


Figure 7: Gaussian Plot for Orange Buoy

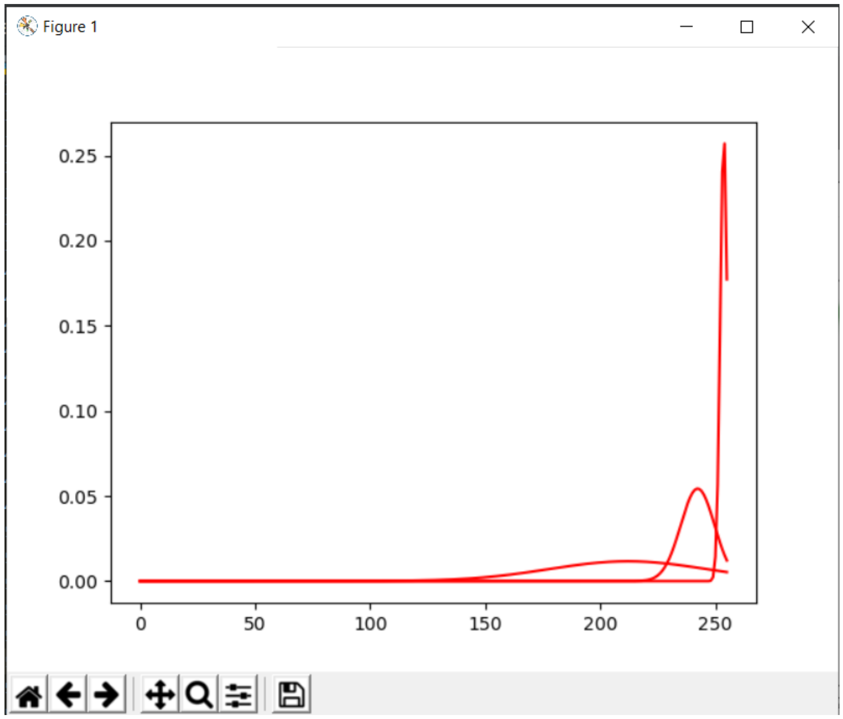


Figure 8: Gaussian Plot for Orange Buoy for individual channel

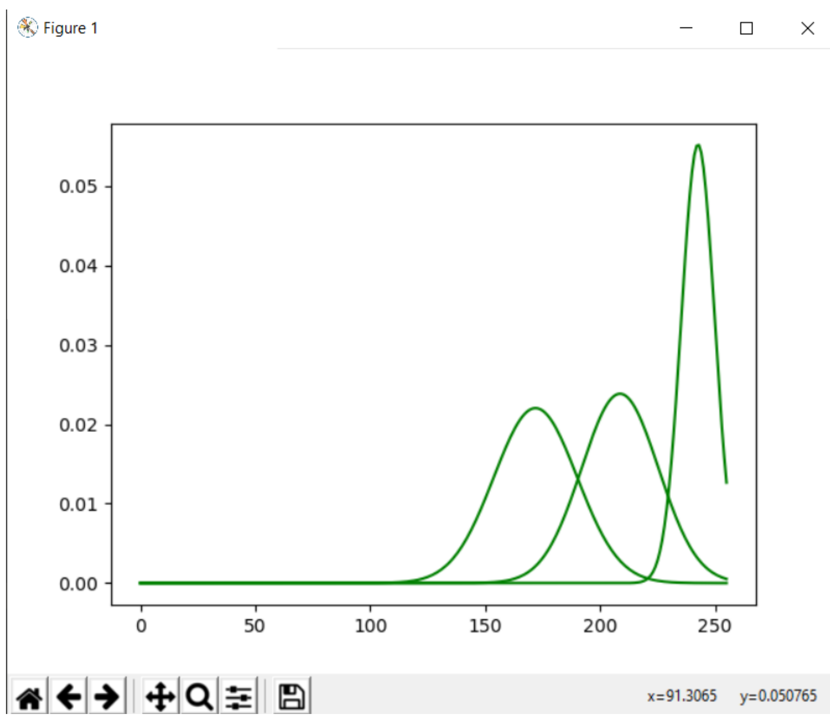


Figure 9: Gaussian Plot for Green Buoy

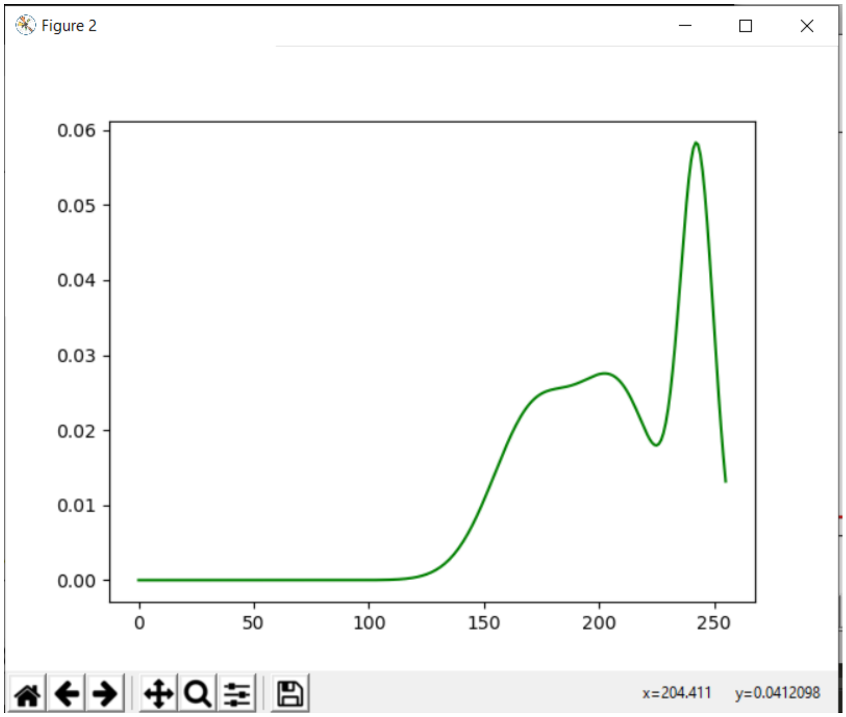


Figure 10: Gaussian Plot for Green Buoy for individual channel

Step 11: The average histogram generated in step 2 is useful in analyzing the number of pixels of each intensities of red, green and blue channels is present in each buoy. This determines which color channel is suitable for a given particular buoy.

Step 12: We observe that for the yellow buoy, both red and green channels provide useful information, for orange buoy, red channel provides maximum information and for green buoy, the green channel provides most information.

Step 13: From the observations made in the previous step we used the gaussians of red and green channels to determine thresholds for yellow buoy, the red channel gaussian was used to determine thresholds for orange buoy and the green channel gaussian was used to determine thresholds for green buoy.

Step 14: Once we obtain the thresholded images of each buoy from the previous step, we perform morphological operations to eliminate unwanted regions that may be present in the thresholded images.

Step 15: Once all the unwanted regions are eliminated, the cv2.findContours() function was used to find all the contour regions present in the thresholed images.

Step 16: Once the contours are obtained, the cv2.minEnclosingCircle() function is used to find the circle that encompasses the generated contour region. The center and radius of this circle is extracted and used to draw circles on the video frame.

Step 17: The previous two steps are performed for each buoy and then the final buoy detection video is generated.

4 Output

4.1 Sample Images from the generated dataset

Green Buoy:

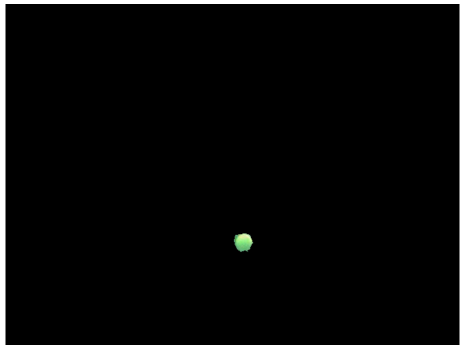


Figure 11: Green Buoy

Yellow Buoy:

Figure 12: Yellow Buoy

Orange Buoy:

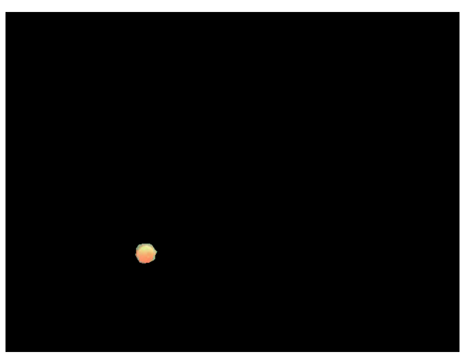


Figure 13: Orange Buoy

4.2 Sample Output from final buoy detection video

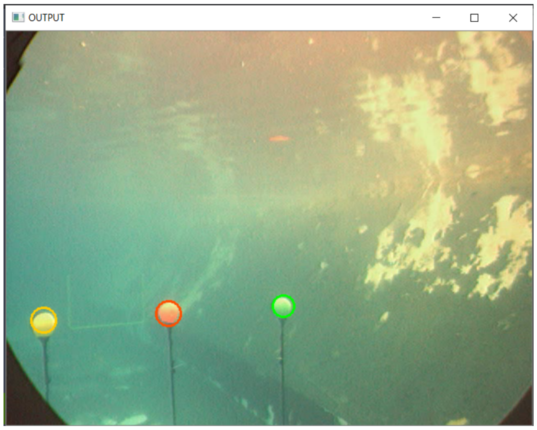
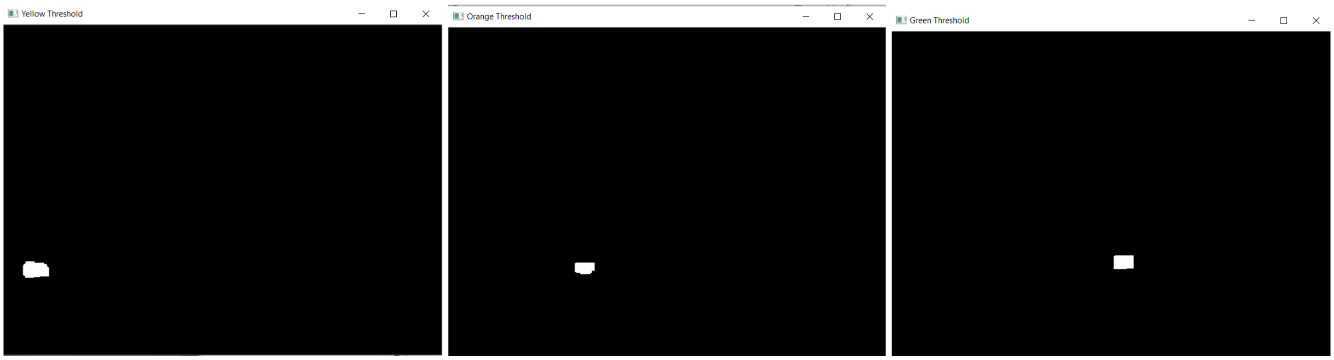


Figure 14: Sample Output 1



(a) Yellow buoy thresholded image (b) Orange buoy thresholded image (c) Green buoy thresholded image

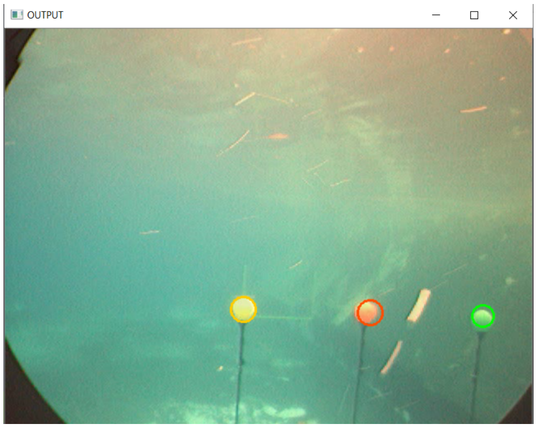
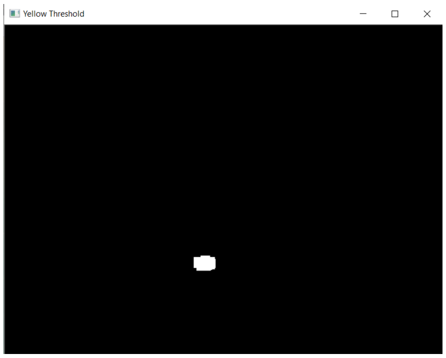
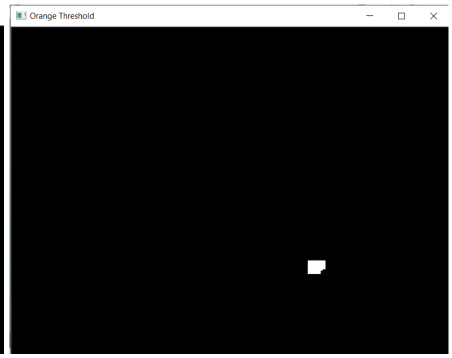


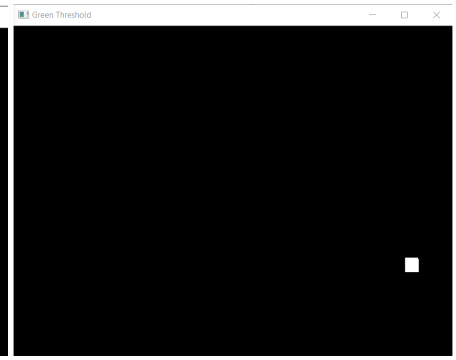
Figure 16: Sample Output 2



(a) Yellow buoy thresholded image



(b) Orange buoy thresholded image



(c) Green buoy thresholded image

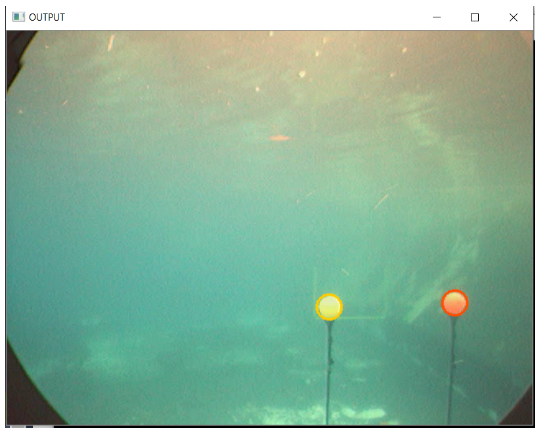
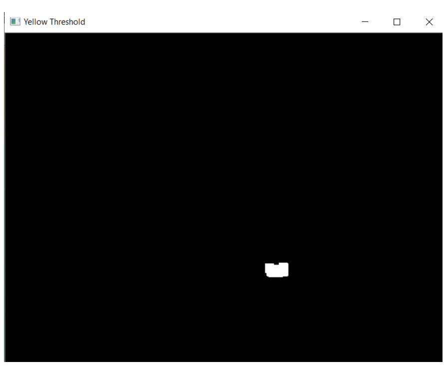
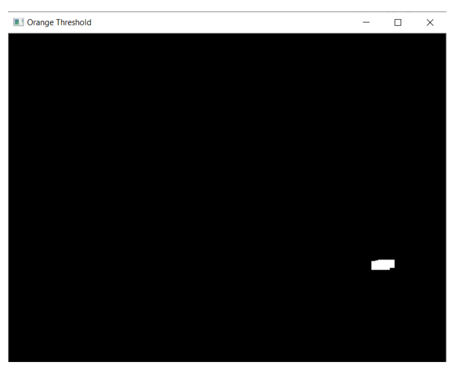


Figure 18: Sample Output 3



(a) Yellow buoy thresholded image



(b) Orange buoy thresholded image

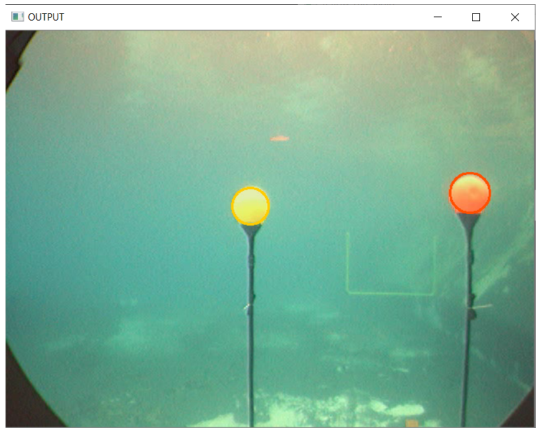
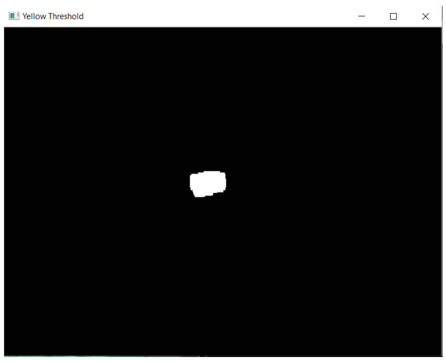
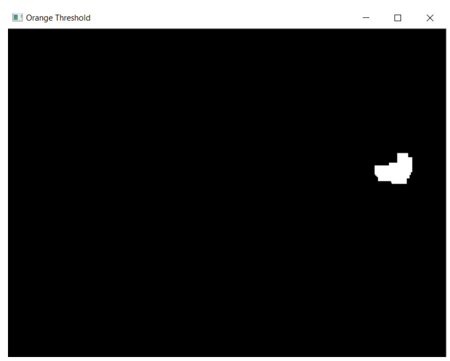


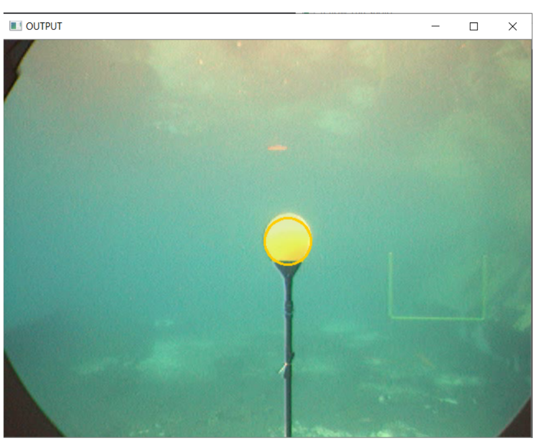
Figure 20: Sample Output 4



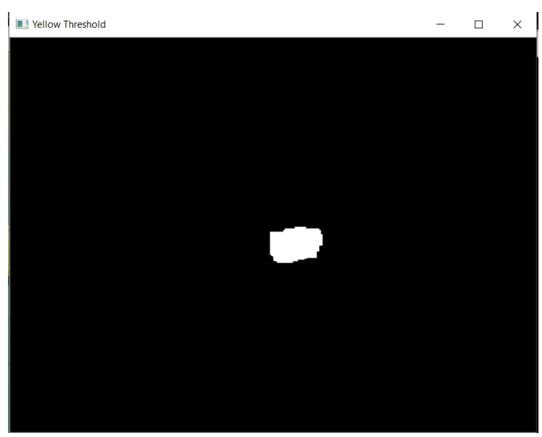
(a) Yellow buoy thresholded image



(b) Orange buoy thresholded image



(a) Sample Output 5



(b) Yellow buoy thresholded image

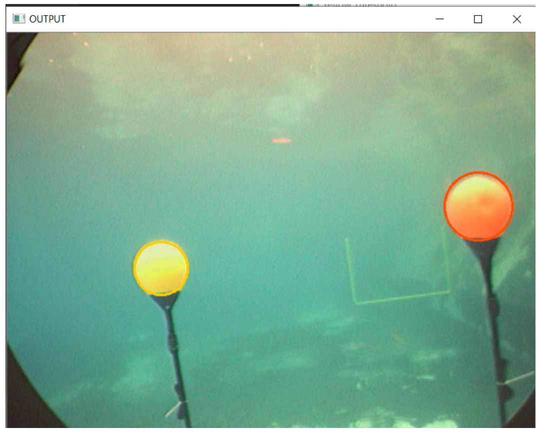
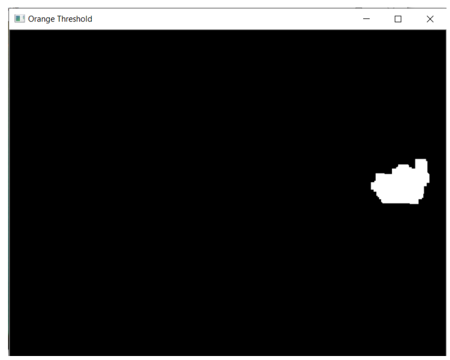


Figure 23: Sample Output 6

(a) Yellow buoy thresholded image



(b) Orange buoy thresholded image

4.3 Output Video Link

`https://drive.google.com/drive/folders/1A-jOclXAyxUIZvThQ-erImdqoCyHMtA6`

5 Challanges Faced

1. The first problem faced was the impact of black pixels from the dataset on the output. This problem was addressed by converting the black pixels to np.nan values.

2. The next problem was tuning the threshold values to detect the buoys. Detecting the green buoy was especially challenging since pixels from sea-bed region closely matched pixel values present in the green buoy. This problem was solved by adding several conditions to segment the green buoy alone without hard-coding the detection. But due to these added condtions, the detection of the green buoy flickers a lot and is not quite stable compared to the very stable detection of the other two buoys.

6 References

[1] https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

[2] https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

[3] https://www.youtube.com/watch?v=REypj2sy_5U

[4] https://www.youtube.com/watch?v=iQoXFmbXRJA