Perception For Autonomous Robots

Project 3

Color segmentation using Gaussian Mixture Models and Expectation Maximization

Pranali Desai (116182935) Nikhil Mehra (116189941) Sayan Brahma (116309165)

Contents

List of Figures

1 Introduction

The project is to implement Colour segmentation using Gaussian mixture model (GMM) and Expectation Minimization (EM) algorithm.

1.1 Video Link

The link for the video is "https://drive.google.com/file/d/1AddGe6rxmmTfrLJazHsYGO1XILcjEgcb/view"

2 Preparing the Data

1. First we determine the total number of frames in the video clip and divide the frames to 7:3 ratio for training and testing frames. In our case we had total of 200 frames, thus we chose to take 140 frames for testing purpose and extracted 28 frames at a certain intervals from all over the video.

2. After extracting the test frames we crop the different coloured buoys and save it in different folders. The cropping is performed tighter to the outline of the buoy for minimal noise inclusion.

3. After performing all these operations it is observe that we do not need so many data for training purpose, thus we removed some of the data and started visualizing the average histogram for each BGR channel for each buoy.

4. After deciding which channel gives the best output for each coloured buoy we start training the parameters i.e., mean and standard deviation for colour segmentation.

3 Average Histogram

1. Average histogram is drawn by averaging all the histograms for each frame i.e. first the histogram is drawn for each frame and then all data from all the histograms are averaged together to make an average histogram for the whole video.

2. Average histogram is shown for each of the channels i.e. BGR for each buoy.

3. The thing to notice in the average histograms was the number of pixels along the peak in each channel.

4. For orange buoy the number of pixels along the peak of the red channel were about 100, around 80 more than the blue and green channel, hence only red channel is used for the orange buoy for further processing.

5. Similarly for green buoy the peak along green channel was higher than the peak about blue and red channel, hence only green channel for the green buoy is used for further processing.

6. For the yellow buoy, the peak about the red and green channels were comparable, hence the average of the red and green channel is used for the further processing.

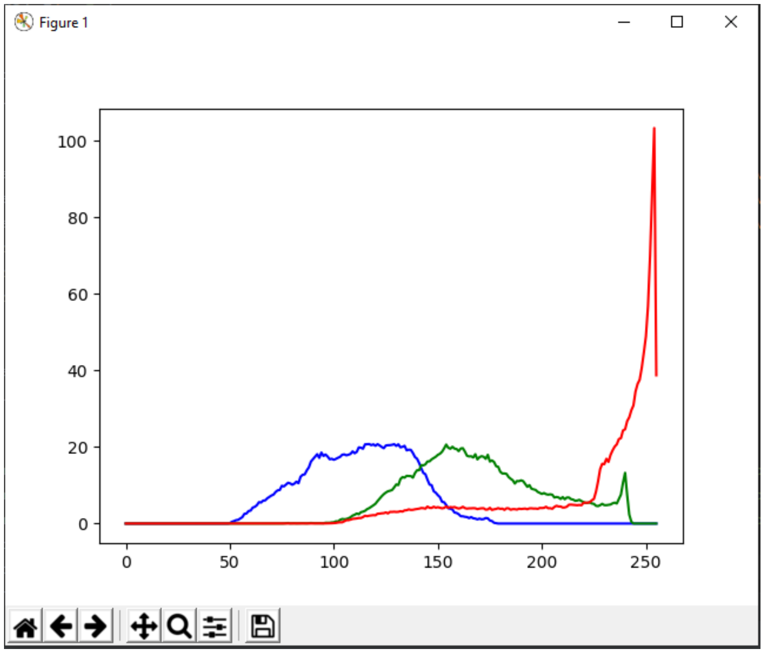

Figure 1: Orange Buoy Average Histogram

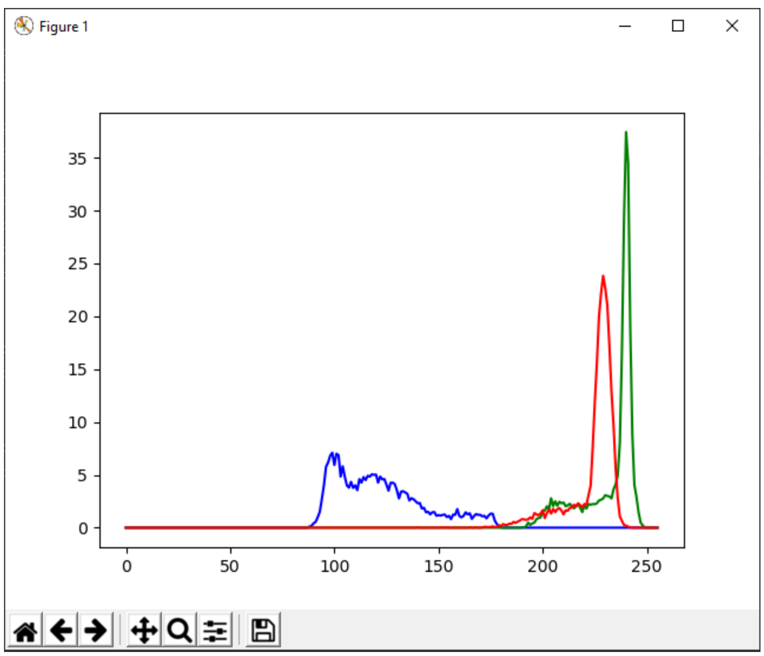

Figure 2: Yellow Buoy Average Histogram

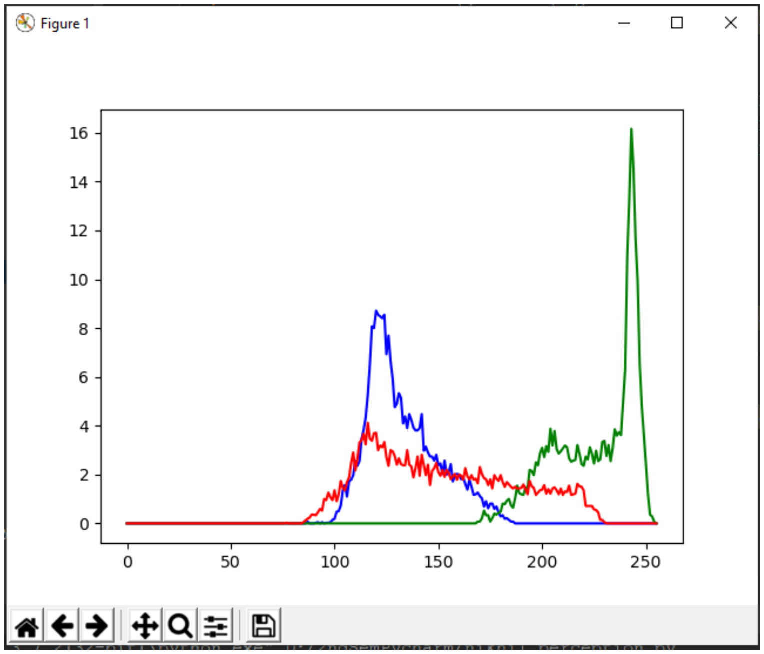

Figure 3: Green Buoy Average Histogram

4 One 1D Gaussian

1. After calculating the average histogram we now aim to fit a 1D Gaussian over it.
2. We first load the train data set and look at the channel of which the histogram is of maximum intensity for that buoy.
3. That is for the orange buoy we see that the average histogram has the maximum intensity in red channel. For yellow it is both red and green and for green it is only in green channel.
4. After loading that particular channel we calculate the mean and standard deviation of all the images of train set for that buoy.
5. This mean and standard deviation is then used to generate the Gaussian using the equation -
6. The images of the Gaussian generated has been depicted below.

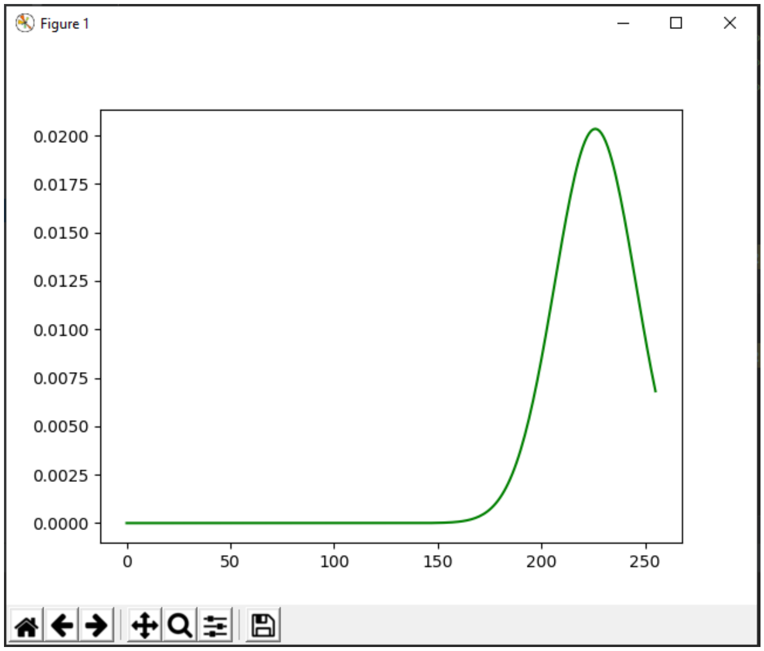

Figure 4: Green Buoy 1D Gaussian

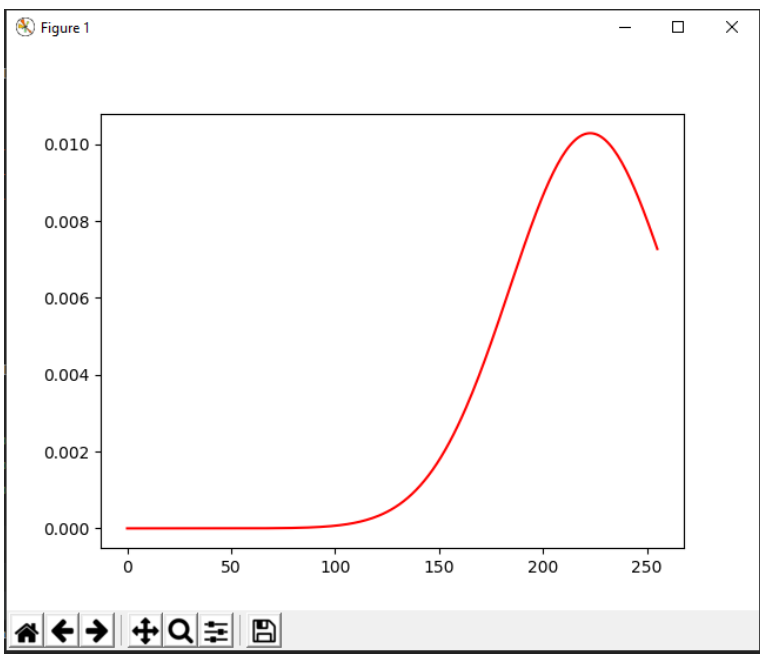

Figure 5: Red Buoy 1D Gaussian

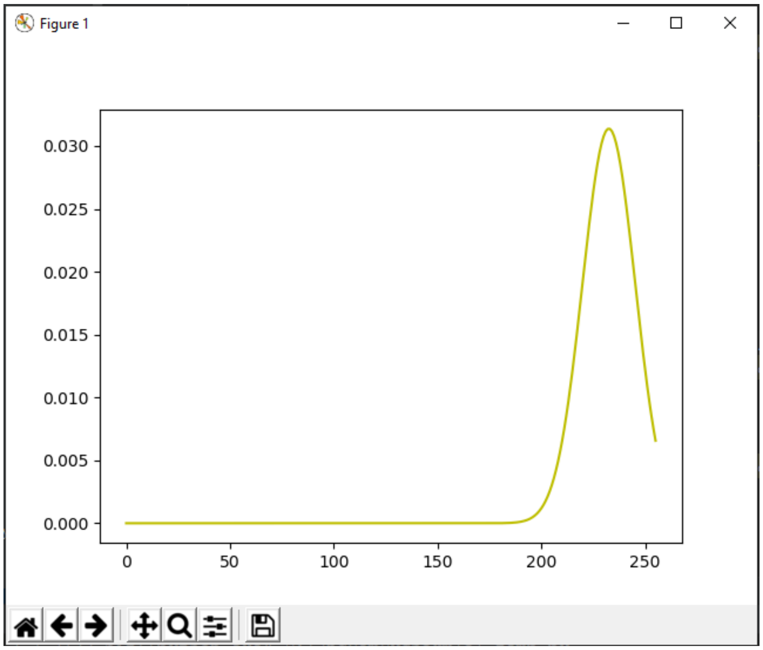

Figure 6: Yellow Buoy 1D Gaussian

7. Once the Gaussian are generates we put a threshold on the probability and thus detecting the buoy for that particular color.

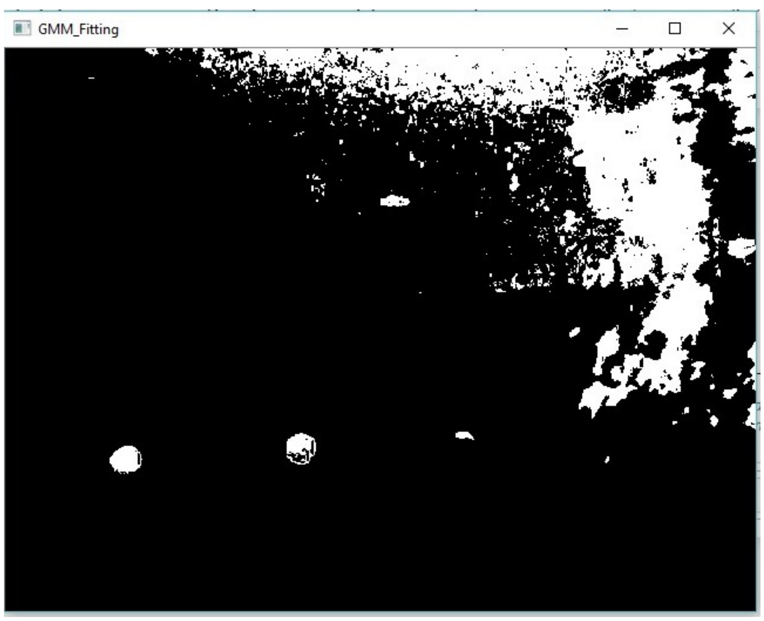

Figure 7: Red Buoy from 1D Gaussian

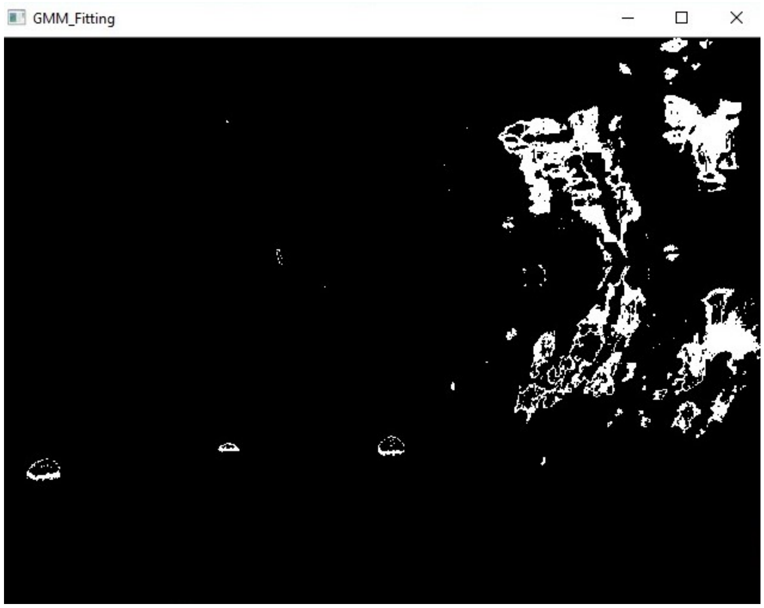

Figure 8: Green Buoy from 1D Gaussian

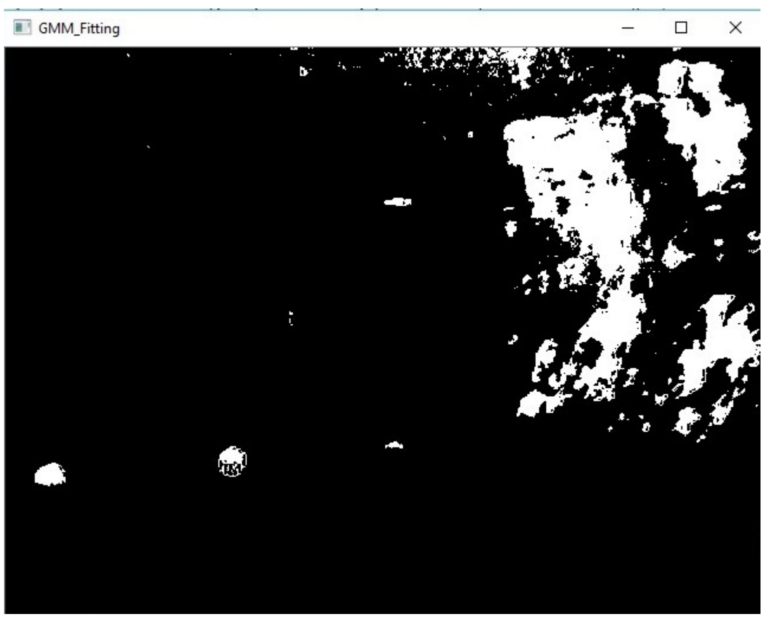

Figure 9: Green Buoy from 1D Gaussian

8. The images of the same are depicted below but we see that the images are not proper and thus we use the further approach.

5 Gaussian Mixture Model and Expectation Maximization

1. A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. It can be thought

that the GMM is a generalized k-means cluster to incorporate information about the variance structure of the data and the centers of the Gaussian formed.

2. The main difficulty in GMM algorithm is that the data is unlabeled, so one cannot determine which point came from which latent component. Expectation-maximization algorithm solves this problem through iterative process. At first we select random components (i.e., randomly selected center points and a common standard deviations) and compute for each point the probability of being generated by each component of the model. Then we tweak the parameters i.e., mean and standard deviations of each Gaussian formed for each considered channels. Repeating this process guarantees to always converge to a local optimum values.

The Gaussian/probability from the mean and standard deviation is found out using the following equation:

$$g(x) = (\frac{1}{\sigma\sqrt{2\pi}})e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

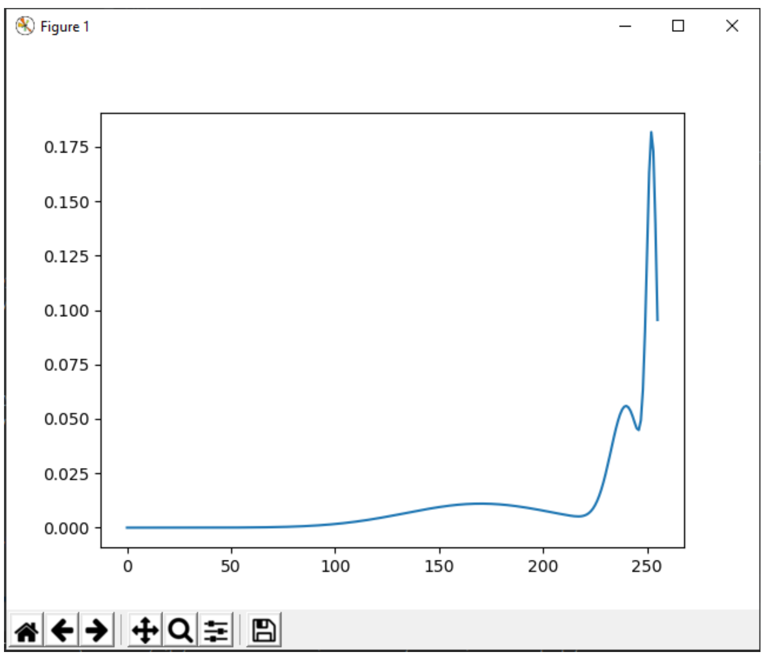

Figure 10: Red Buoy 3-1D Gaussian

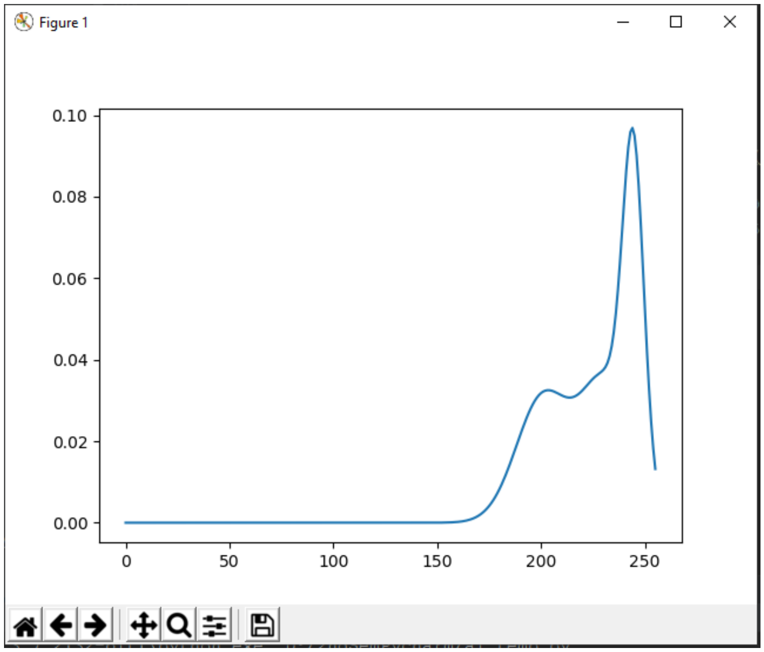

Figure 11: Green Buoy 3- 1D Gaussian

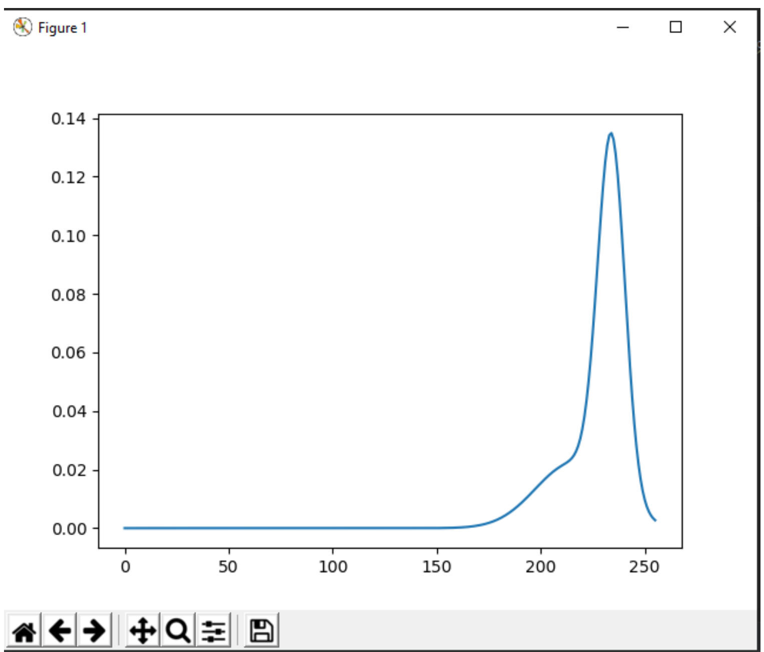

Figure 12: Yellow Buoy 3- 1D Gaussian

5.1 Using EM to extract GMM parameters

1. Generate 3 random samples for three 1-D Gaussian with different means and standard deviation.

2. The equations used are -

$$Pr(A|X) = \frac{Pr(X|A)Pr(A)}{Pr(X|A)Pr(A) + Pr(X|\sim A)Pr(\sim A)}$$

$$B1_2 = \frac{\frac{p12}{3}}{\frac{p12}{3} + \frac{p22}{3} + \frac{p32}{3}}$$

$$Mean = \frac{\sum(BayesianProbability)(pixelintensity))}{\sum(BayesianProbability)}$$

$$StandardDeviation = \sqrt{}(\frac{\sum(((BayesianProbability)(pixelIntensity)) - mean)^2}{\sum BayesianProbability})$$

3. These three 1-D Gaussian formed are applied to all the test images stored in the folder.

4. After running it through all the images we get a new mean and standard deviation.

5. These new values are used and again applied to all the images, and this step is iterated for 50 times.

6. After executing the process for 50 times the mean and the standard deviation converges and does not change, these values are the trained values and are used for further analysis. 50 has been chosen as the iteration number because the log-likelihood function converges in this time.

$$l(\Theta;\varepsilon) = -\frac{n}{2}ln(2\pi) - \frac{n}{2}ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

5.2 Buoy detection using GMM

1. Now we implement the model parameters obtained from the previous steps to all the frames of the video to generate a probability map.

2. Segment the buoy by thresholding the probability map by generate a binary image and compute the corresponding contours and center for each buoy taking into consideration the required channel or any combination of the channels.

3. Draw the bounding contours around the detected buoy. We need to repeat this process/step for each buoy for all the frames of the video. Thus, the final video would show all the different color buoy in all the frames of the video of interest.

6 Learning color model and Buoy Detection using N 1D Gaussian

1. This part of the project basically involves to identify how many Gaussian and dimension of Gaussian to be used to detect and segment the colored buoys.

2. After several iterations we have identified that one dimensional Gaussian will give better results and since we can fit it to the histogram of that particular colour will give us a better understanding of the segmentation in terms of noise and actual buoy.

3. For the orange buoy, we have taken the histogram and plotted the three 1-D Gaussian and we notice that it completely matches the average color histogram of red channel.

4. Similarly we compute the same process for the rest of the buoy to and we keep of generating the Gaussian until it clearly fits the histogram.

5. The images of the histogram along with the Gaussian used to fit has been depicted below.

6. Once the Gaussian have been generated to fit the histogram the Y axis represent the probabilities.

7. We segment every pixels according to the (probabilities) thus stating a threshold on the probability of that pixel being in that Gaussian.

8. We create a binary image after applying threshold according to the probabilities and find the contour of that buoy.

9. After this point we trace back the contour on the original frame of the video, and it is depicted in the images below.

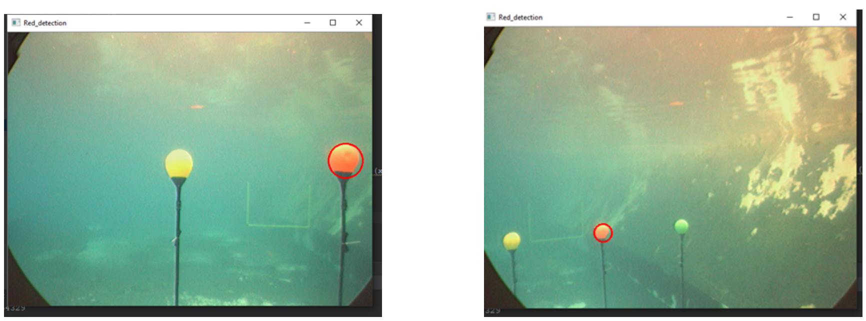

Figure 13: Red Buoy detection

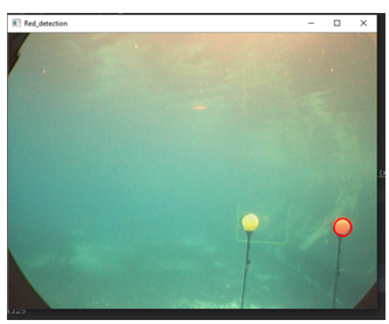

Figure 14: Red Buoy detection

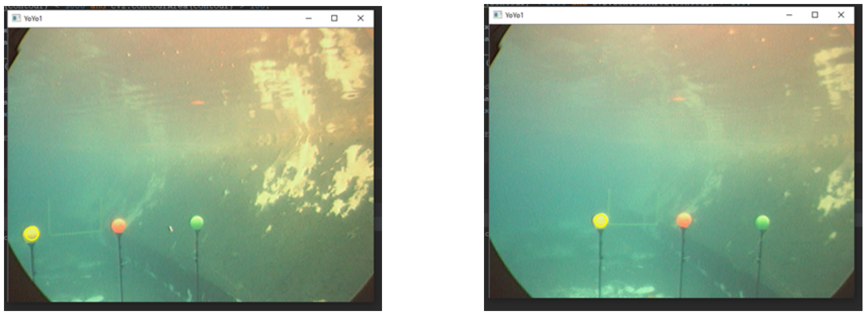

Figure 15: Yellow Buoy detection

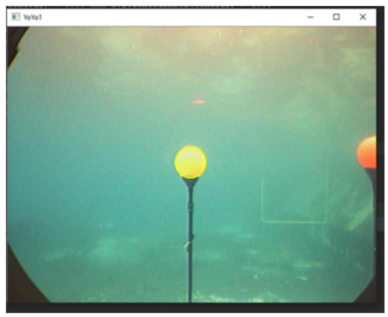

Figure 16: Yellow Buoy detection

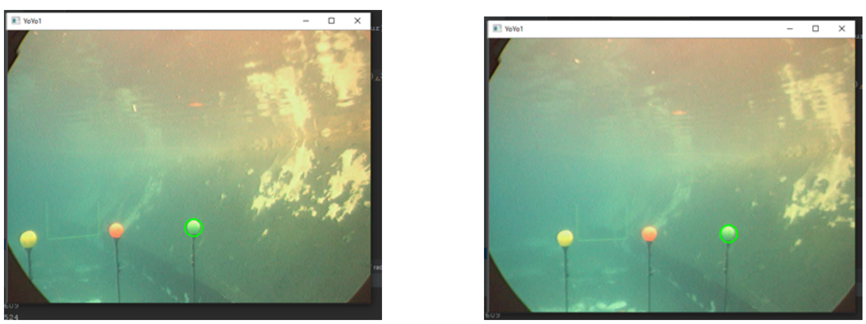

Figure 17: Green Buoy detection

Figure 18: Green Buoy detection

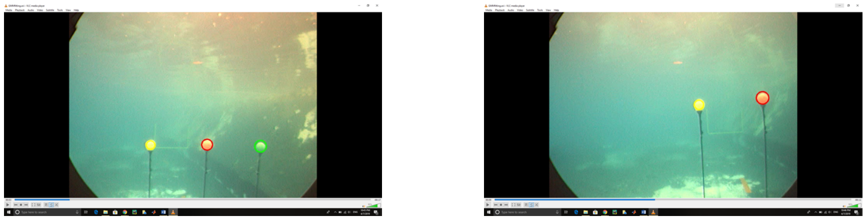

Figure 19: All Buoy detection

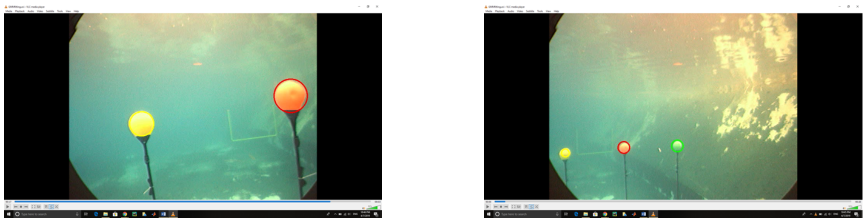

Figure 20: All Buoy detection

7 Analysis and Discussions

1. We only used BGR color space for the implementation of this project.
2. Only using three 1-D Gaussian on the red channel of the orange buoy gave excellent results.
3. Only using three 1-D Gaussian on the green channel of the green buoy gave good results with some background error. Background error can be removed by using morphological operations such as dilation and erosion. Background error can also be removed by using contouring operations. But the downside of only using contouring operation is that it does not work on all the frames. Using contouring operations after morphological operations gave us good results.
4. Yellow buoy was the most difficult to identify as the yellow color background error due to sunlight is present in the initial frames of the video.
5. Only using three 1-D Gaussian on on one of the channels of the yellow buoy gave no results, hence we used three 1-D Gaussian on the average of the green and red channel. Using contouring operations after morphological operations (erosion and closing) gave us good results, with only some background error in the initial frames.
6. We feel that different color space such as HSV and BGRY can be used for this project. BGRY has its own yellow channel, hence will make it easier to detect yellow color buoy in the input video.

8 References

1. Lecture notes and pdf

2. Bishop - Pattern Recognition and Machine Learing

3. https://opencv-python-tutroals.readthedocs.io/en/latest/py$_{t}utorials$