University of Maryland College Park

Professional Master of Engineering

Robotics Engineering

ENPM673 - Perception for Autonomous Robots - Project 3

Students:
Akash Agarwal (116904636)
Nupur Nimbekar (116894876)
Anubhava Paras (116905909)

Lecturer:
Dr. Mohammed Samer Charifa

April 6, 2020

Color image segmentation simplifies the vision problem by assuming that objects are colored distinctively, and that only gross color differences matter. It therefore discards information about color and brightness variations that provides many valuable cues about the shapes and textures of 3D surfaces. But the resulting simplified (and impoverished) image can be processed very rapidly, which can be important in mobile robot applications.[1]

The main idea of the project was to explore the Gaussian Mixture Model and Expectation Maximization techniques for color segmentation, so we have tried our best to understand and implement the ideologies of the methods mentioned above.

A Gaussian Mixture is a function that is comprised of several Gaussians, each identified by $k \in \{1, \ldots, K\}$, where K is the number of clusters of our dataset. Each k is comprised of the following parameters:[2]

1. A mean which defines the centre (μ)
2. A covariance that defines width (Σ), which will be equivalent to the dimensions of an ellipsoid in a multivariate Scenario (σ)

Graphically we can represent Gaussian Mixture as

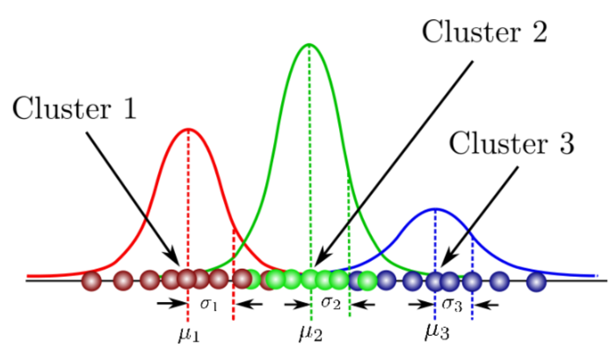


Figure 1: Representation of Gaussian Mixture Model

The Expectation-Maximization (EM) algorithm is a way to find maximum-likelihood estimates for model parameters when your data is incomplete, has missing data points, or has unobserved (hidden) latent variables. It is an iterative way of approximation. [3]

Problem Statement: -

The aim is to detect the buoys from the input video and using the proper color segmentation techniques detect the buoy.

Link to Drive to access data and output along with Codes

Stage 1: Data Preparation

The data has to be pre-processed so that we can extract particular information from the given Video input.

Let's look at the pipeline followed here to extract data and pre-process it:

1. We run the video through the preproccess.py code, to extract the frames from video, and to maintain the ratio but still filter the training data set, we save the frames in Even and Odd sequences.
2. Then we run the code through traintest. py code to seperate the fles from either even or odd folder to divide them into train dataset and test dataset. (75-25 ratio is maintained)
3. Then the frames from train dataset are run through testcrop.py code to crop the regions with clustered information regarding the yellow, red and green blob from every frame.
4. An average color histogram is plotted for every RGB frame
5. These Histograms are further analyzed to learn the behaviour associated with the extracted data.

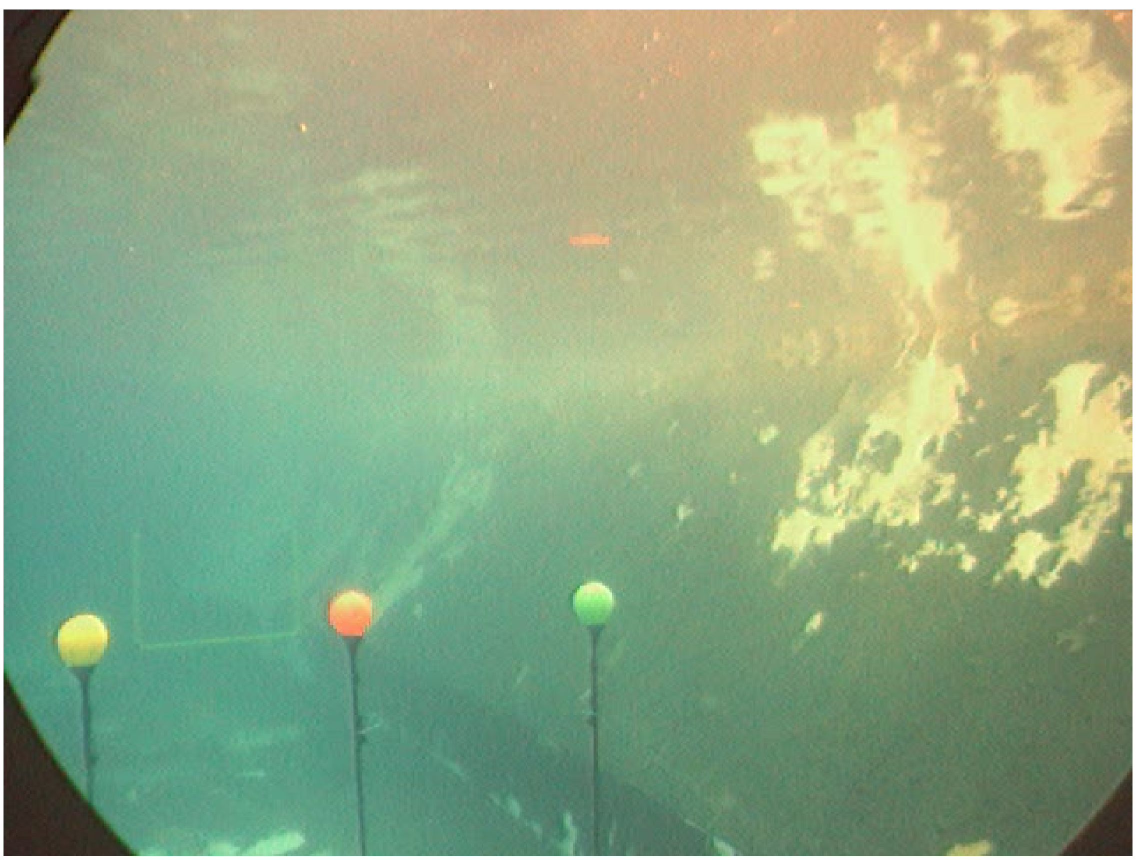
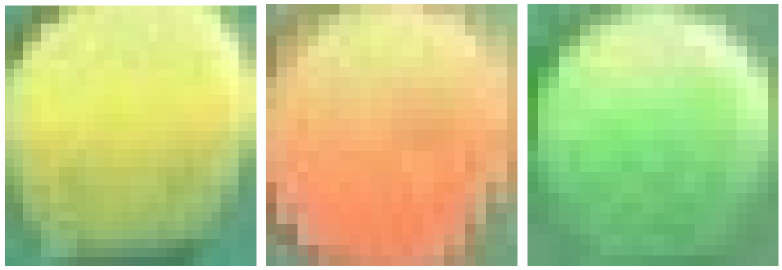


Figure 2: Frame fed to extract particular information from the frame.



(a) Yellow buoy data (b) Red buoy data (c) Green buoy data

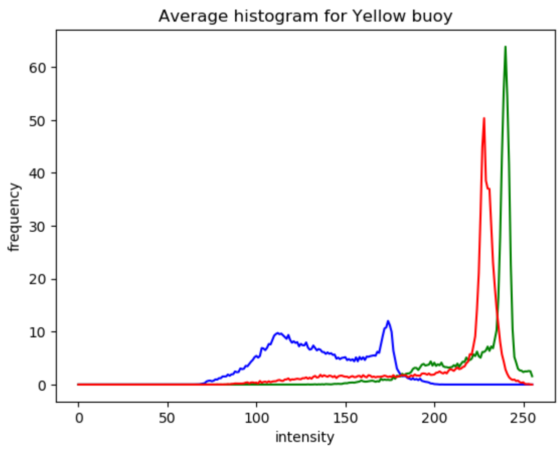
Figure 3: Data extraction from train dataset to create cropped dataset

The Average study of Histogram for all the colours individually from every frame can be seen in the graphs in Figure 4.

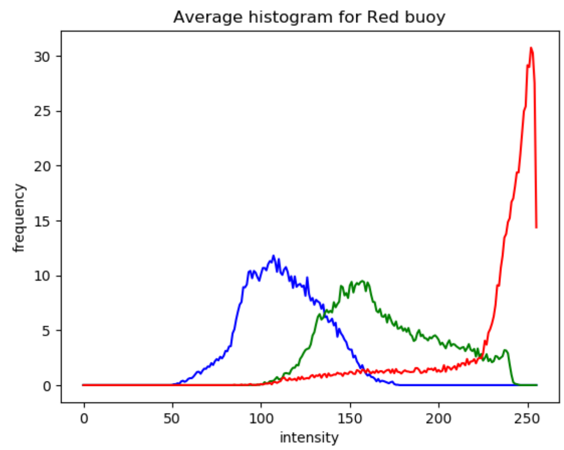
From the Histograms we can note down following traits for individual colour based on the frames.

1. For Yellow: We can say that as we check from first frame to the last one, Red pixels and Green pixels with high intensity are present for majority of the frames where as the pixels for Blue have intensity of less 200 and are merely present in less than 30% frames. Red and Green colour pixels are dominating.
2. For Red: We can say that as we check from first frame to the last one, Red pixels with high intensity are present for majority of the frames where as for Green pixels and Blue Pixels we can see that we have intensities varying from 50-250 in 33% of frames and not present in the rest of the frames. Red colour pixels are dominating.
3. For Green: We can say that as we check from first frame to the last one, Greed pixels with high intensity are present for majority of the frames, and Blue Pixels with the intensities varying from 80-200 are present for most of the frames where as Red pixels of low intensities are present in 50% of the frames. Green and Blue colour pixels are dominating.

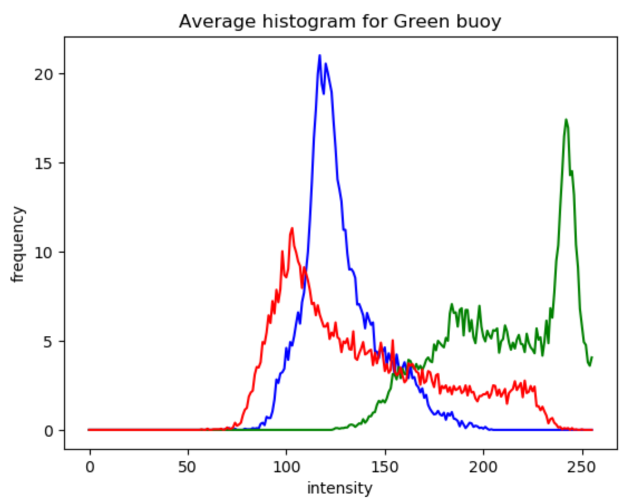
Analysis of the histogram assists us in finding the number of Gaussians and their dimensions required to fit the histogram.



(a) Average Histogram for Yellow buoy data



(b) Average Histogram for Red buoy data



(c) Average Histogram for Green buoy data

Figure 4: Average Histogram for each color with respect to every frame

Stage 2: Gaussian Mixture Model and Maximum Likelihood Algorithm

As discussed before, we will use the pre-processed data to create 1D and 3D Gaussians using Histograms.

Let us discuss the flow of what we do in this stage:

1. We start by plotting 1D Gaussians for 3 buoys.
2. Sample Data was tested with following specifications to test the EM algorithm.

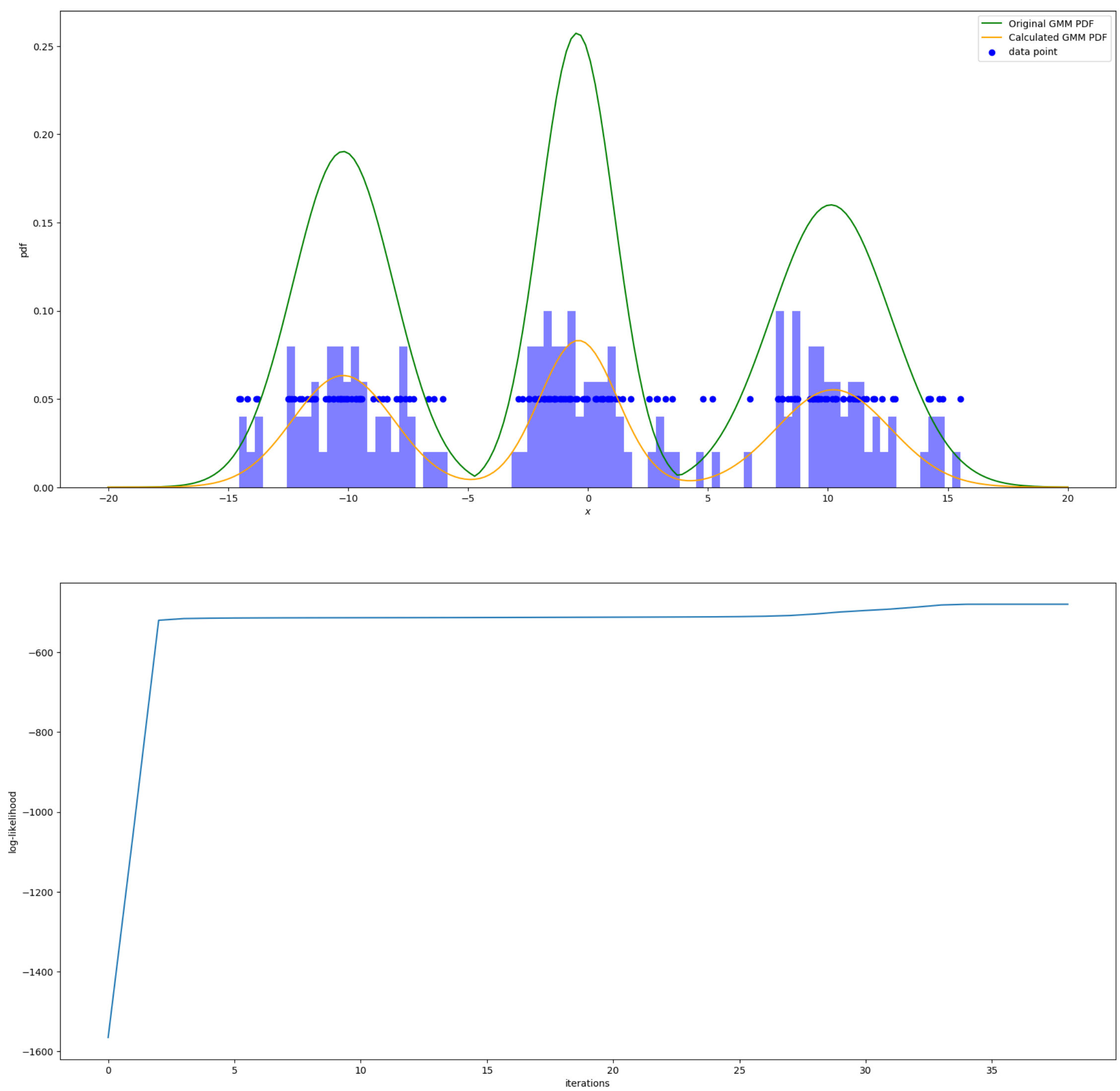


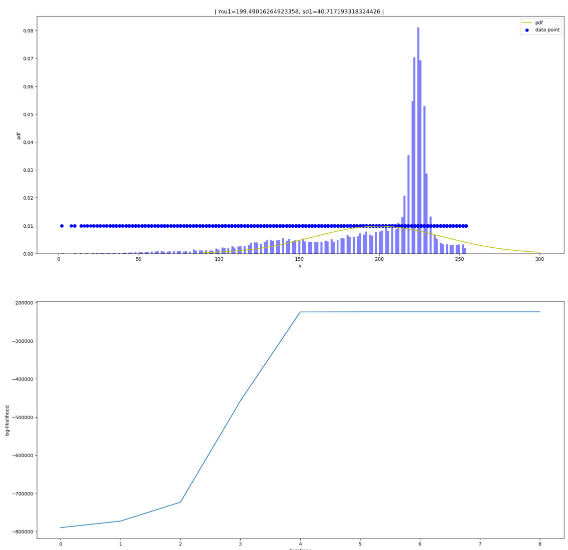
Figure 5: Sample which plots all GMM PDF for Original and Calculated data to test the EM Algorithm.

Data regarding the Gaussians:-

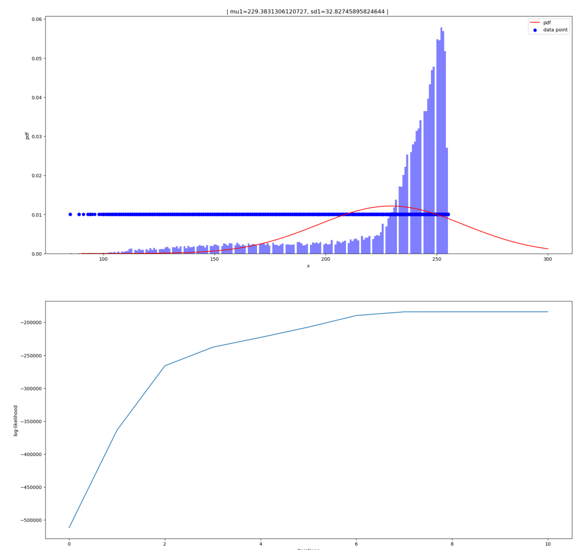
Original values:-

μ_1 = 0; σ_1 = 2
μ_2 = 10; σ_2 = 3
μ_3 = -10; σ_3 = 2

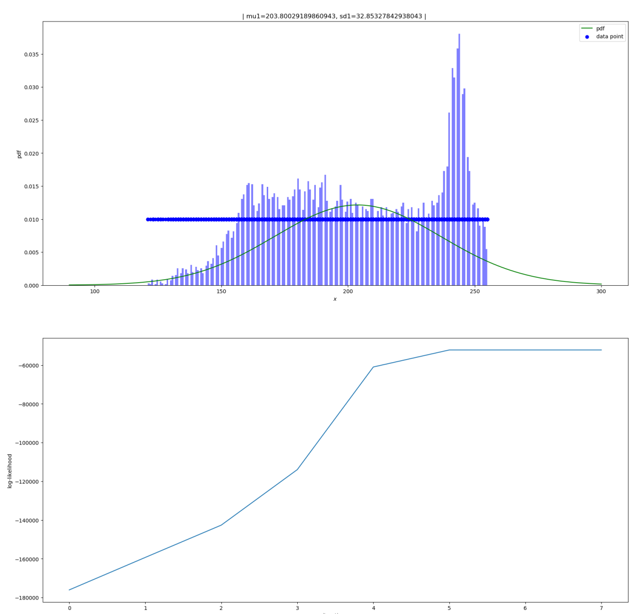
Calculated values:-
μ_1 = -9.634; σ_1 = 2.012
μ_2 = 0.04; σ_2 = 1.836
μ_3 = 9.101; σ_3 = 3.053



(a) 1D Gaussian for Yellow buoy data



(b) 1D Gaussian for Red buoy data



(c) 1D Gaussian for Green buoy data

Figure 6: 1D Gaussian for individual buoys when the number of gaussians is 1

Stage 3: Learning Color Model

We tried fitting various number of Gaussians in the Histogram to see which one would cover the maximum number of data points increasing the accuracy.

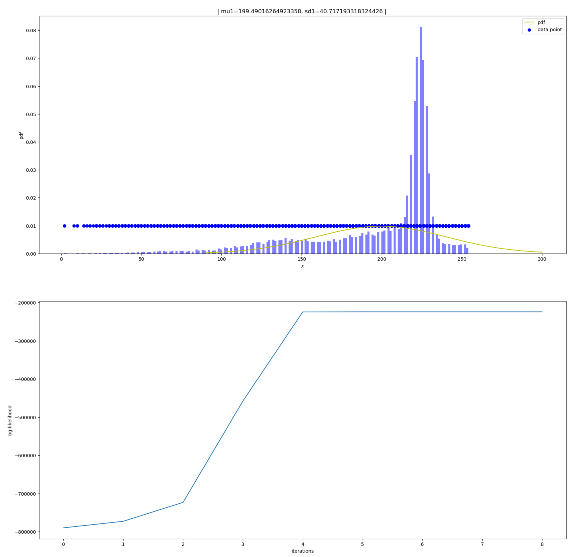
Observations:

1. As the number of Gaussians increase, the coverage of data points increases.
2. The maximum number we tested was 4, but that also made us realize that as the number of Gaussians increase calculations also become complex
3. We trained the model till our log likelihood doesn't converge.

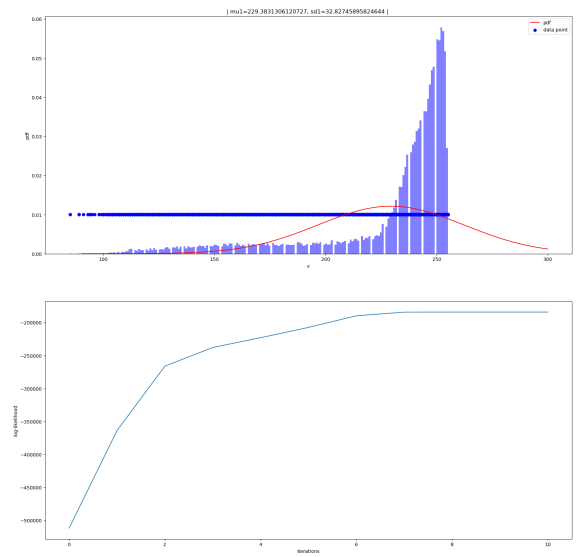
To maintain simplicity in calculation and achieve the output we decided to use 3 gaussians to cover the histogram.

Let us plot the Gaussians for the Graphs.

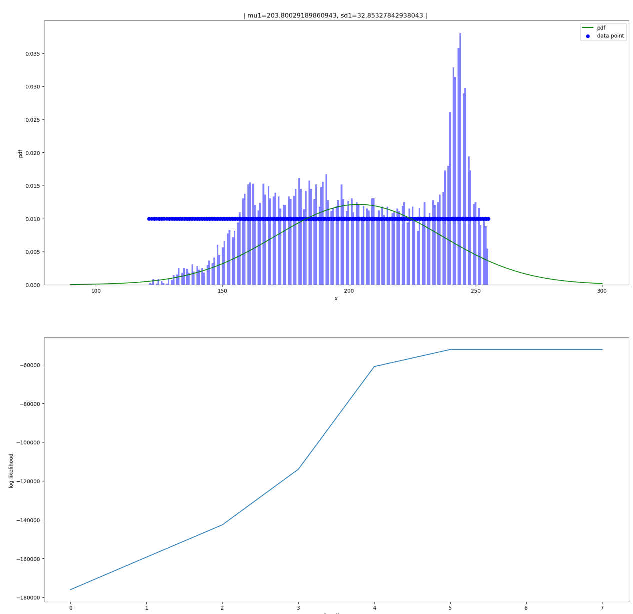
Note: - Number of peaks in the graph symbolizes the number of Gaussians.



(a) 1D Gaussian for Yellow buoy data

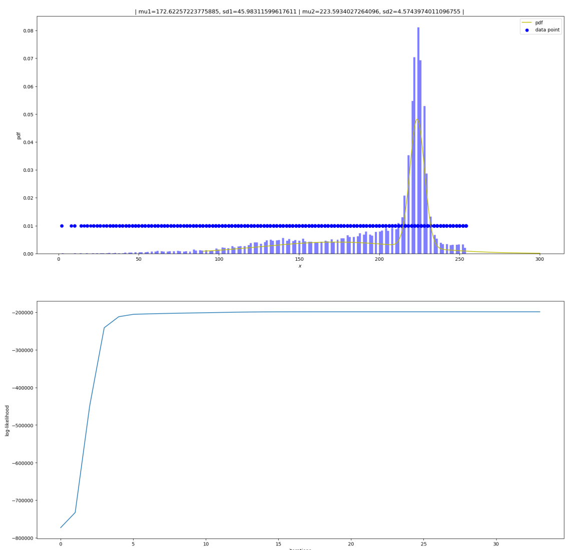


(b) 1D Gaussian for Red buoy data

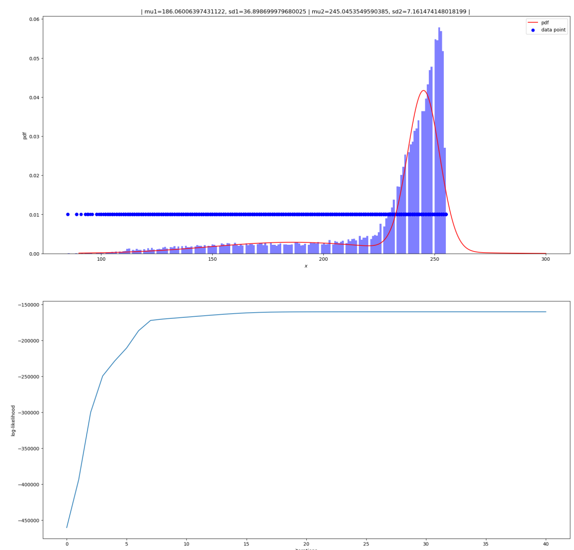


(c) 1D Gaussian for Green buoy data

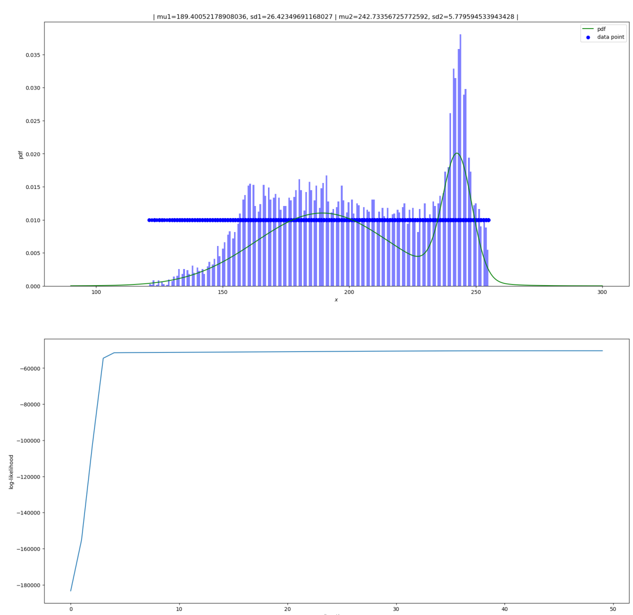
Figure 7: 1D Gaussian for individual buoys when the number of gaussian is 1



(a) 1D Gaussian for Yellow buoy data

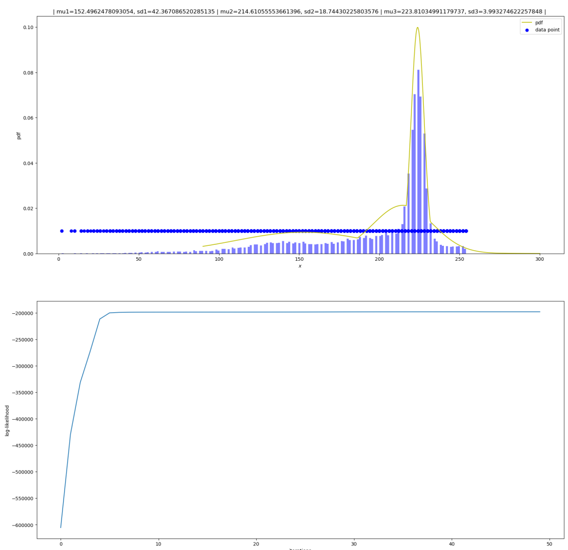


(b) 1D Gaussian for Red buoy data

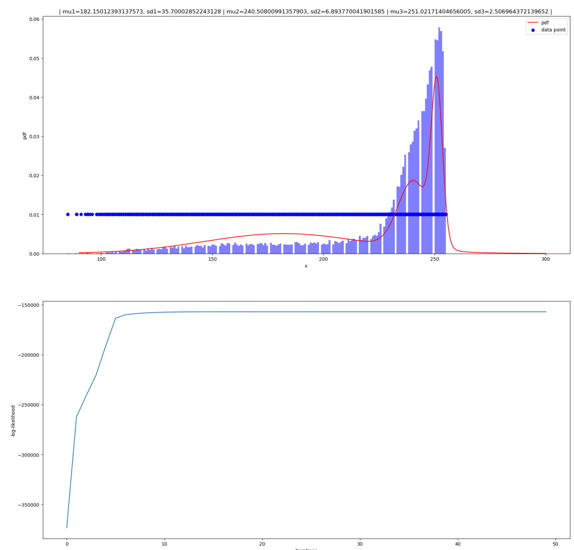


(c) 1D Gaussian for Green buoy data

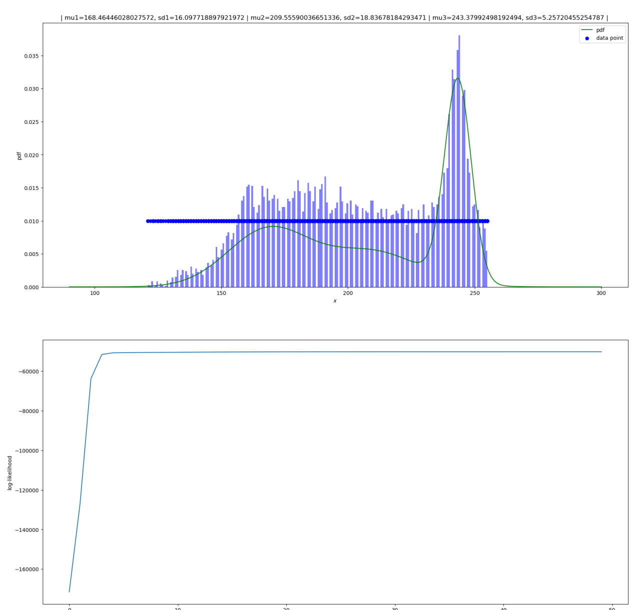
Figure 8: 1D Gaussian for individual buoys when the number of gaussians are 2



(a) 1D Gaussian for Yellow buoy data

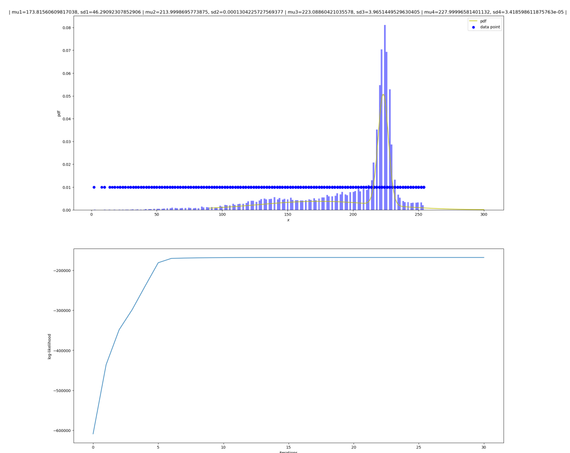


(b) 1D Gaussian for Red buoy data

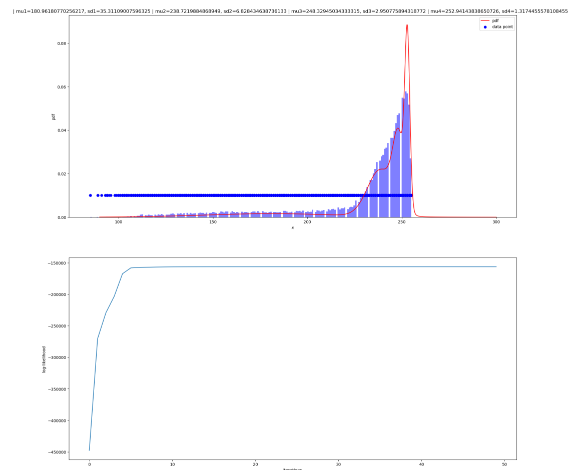


(c) 1D Gaussian for Green buoy data

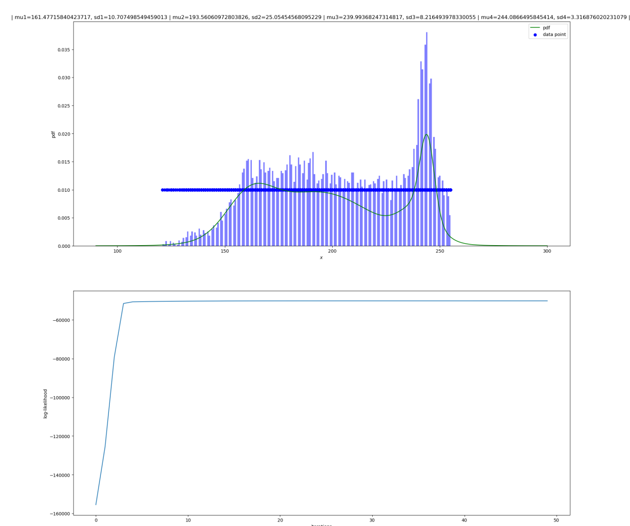
Figure 9: 1D Gaussian for individual buoys when the number of gaussians are 3



(a) 1D Gaussian for Yellow buoy data



(b) 1D Gaussian for Red buoy data



(c) 1D Gaussian for Green buoy data

Figure 10: 1D Gaussian for individual buoys when the number of gaussians are 4

Stage 4: Buoy Detection

In this stage we test the Buoy detection for individual Buoys and then run the algorithm to detect all possible buoys in a frame.

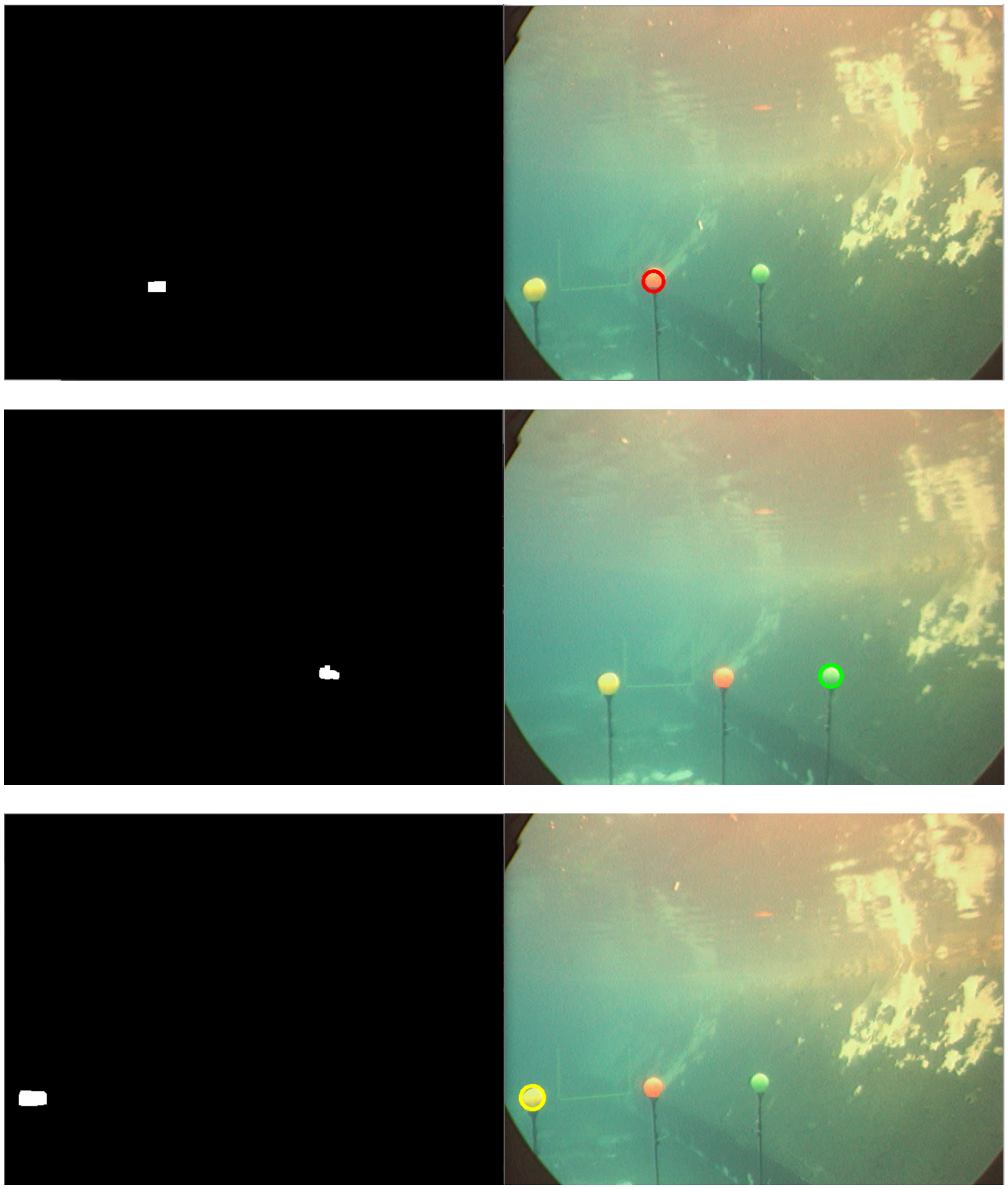


Figure 11: Buoy Detection for each individual buoy along with its binary image

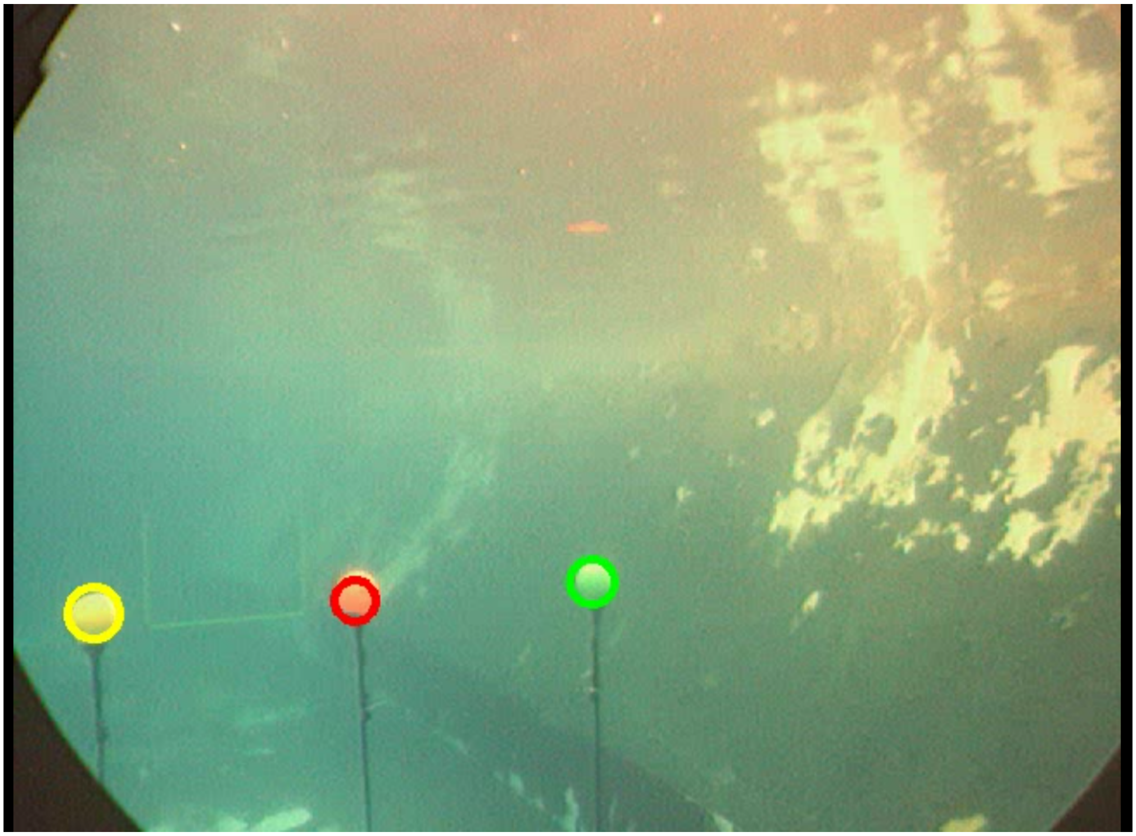


Figure 12: Frame in which all three buoys are detected

Comments:-

- Performing the morphological operations and setting the threshold values for the various colours are time consuming processes used to achieve the desired result.
- The 1D Gaussian needs to be tuned more as due to the extra noise, the Gaussian does not fit properly.

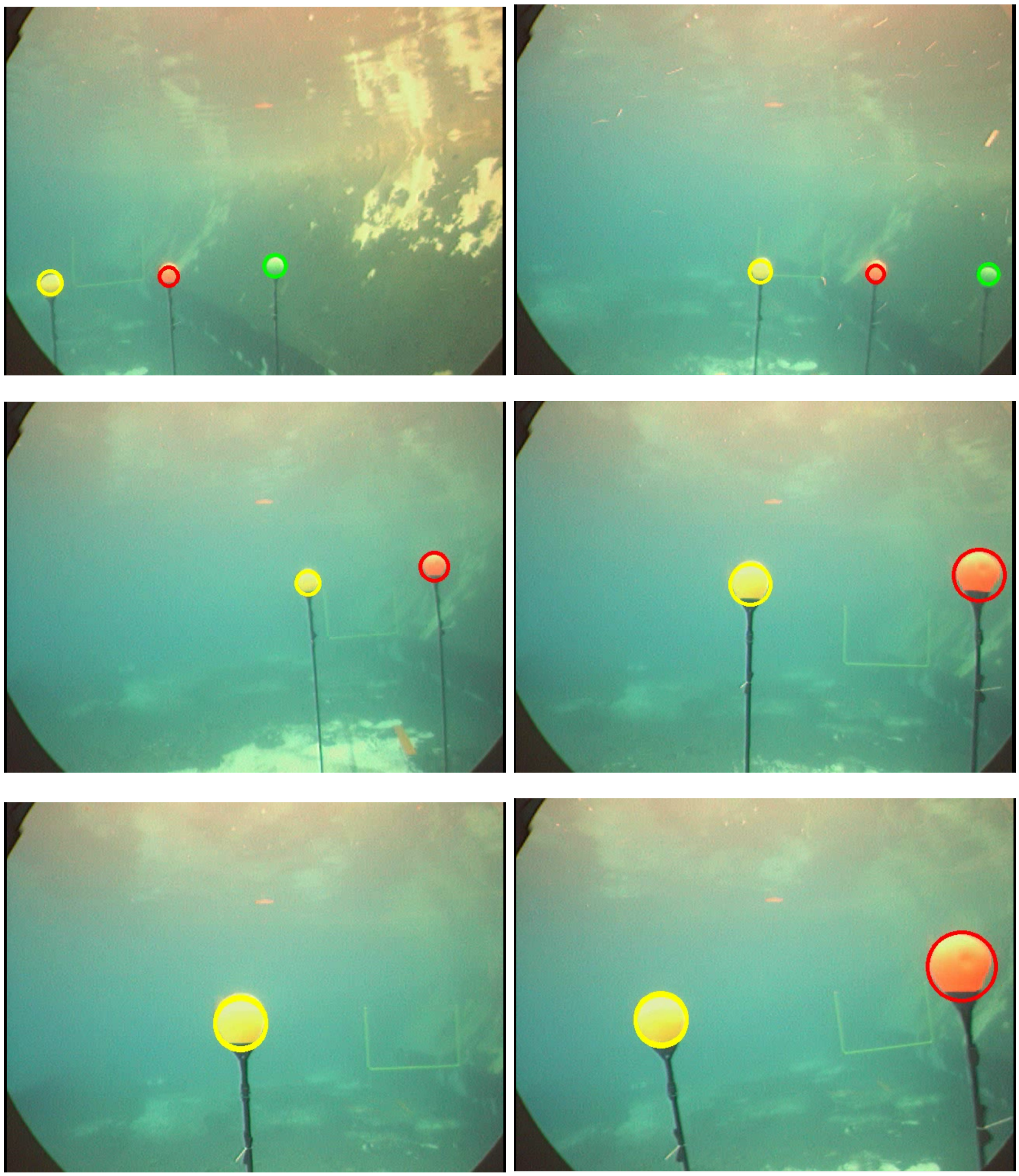


Figure 13: Buoy Detection when various numbers of Buoys are present in the Frame