Perception For Autonomous Robots

Project 3

Color segmentation using Gaussian Mixture Models and Expectation Maximization

Raghav Nandwani (116321549) Varsha Eranki (116204569)

Sanket Goyal (116155144)

Contents

List of Figures

1 Introduction

The project is implementation of Gaussian mixture models and Expectation Maximization techniques for color segmentation.

The video sequence that is captured underwater has three buoys of different colors i.e. yellow, orange and green which are distinctly colored. However, conventional segmentation techniques, for example color thresholding did not work well in underwater scenarios. This is most probably because of various noise and varying lighting conditions.

The average *RGB Histograms* for each colored buoy is plotted. The images of the three buoys of yellow, orange and green are *trained* on the basis of their color distributions and their respective *Gaussian distributions* are generated using these *trained* dataset. Now, the buoys are identified in the video sequence using the probability distribution of each Gaussian distribution of their respective channels. The final video is uploaded 'https://drive.google.com/open?id=1Rz-nhmkflnfC9S4lGn5yjavLLXGcBju-'.

2 Preparing the Data

1. We extracted cropped samples of each colored buoys from a number of frames of the video and segregated those images to their respective folders.

2. We made compact frames by cropping most of outlines of each buoy for accurate colour detection and minimal noise.

3. This formed our datasets and we further saved their pixel intensity values to a *numpy* array.

4. For the Orange buoy, we had a dataset of 135 images, for the Green buoy we had a dataset of 42 images which is comparatively less because of its lesser time span in the video and for the Yellow buoy we had a dataset of 107 images.

5. For each colored buoy separately, we computed and visualized the average color histogram for each channel of the sampled RGB images.

3 Average Histogram

1. We performed smoothing of the image using Gaussian blur.

2. Then we calculated the histogram of one image and stored it as one column of *numpy* array and did the same for all the images using column stack to make a matrix of size *(255 X Number of images)*

3. Then we took the mean of all the rows of this matrix

4. This gave the average number of pixels of every intensity across all the images.

5. We plotted the histogram of the mean data.

6. After calculating the average histogram we fitted a 1D Gaussian over it.

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

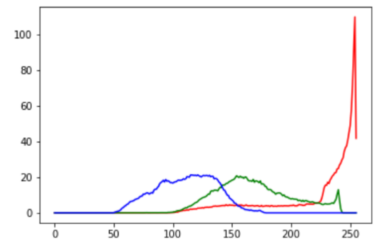

Figure 1: Average Histogram of Orange buoy

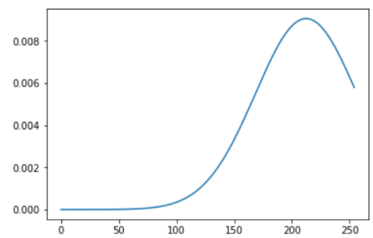

Figure 2: Gaussian distribution of Red Channel for Orange Buoy

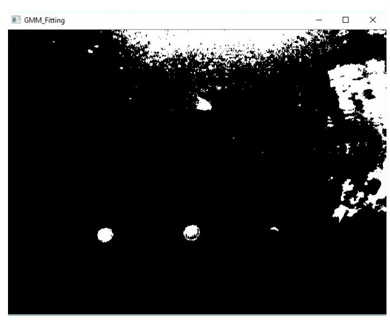

Figure 3: Red Buoy for 1D Guassian

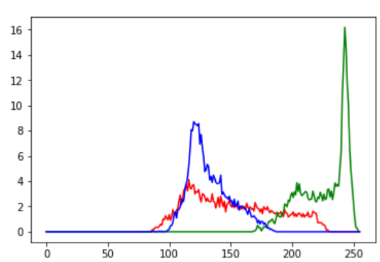

Figure 4: Average Histogram of Green buoy

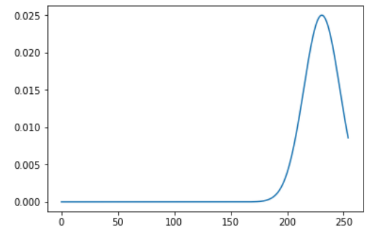

Figure 5: Gaussian distribution of green channel for grange buoy

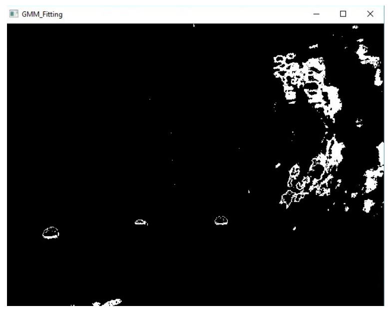

Figure 6: Green Buoy for 1D Guassian

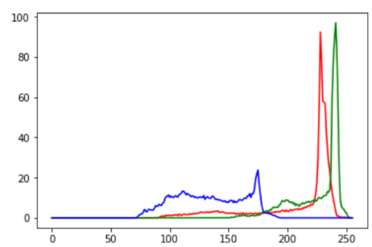

Figure 7: Average Histogram of Yellow buoy

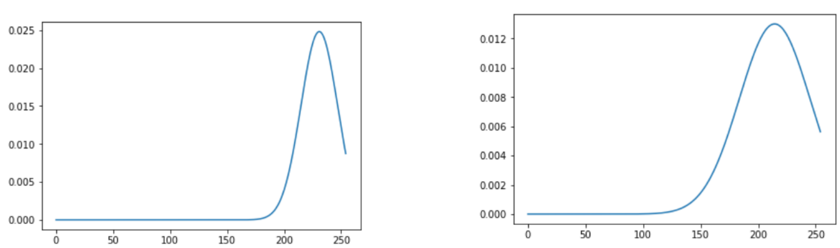

Figure 8: Gaussian distribution of Red Channel(left) and Green Channel(right) for Yellow Buoy

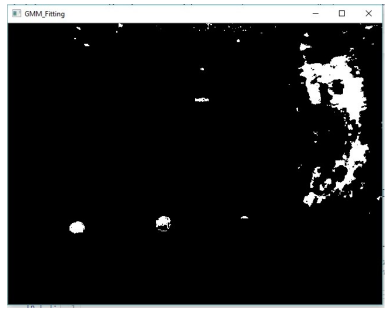

Figure 9: Yellow Buoy for 1D Guassian

4 Gaussian Mixture Models and Expectation Maximization

1. **Gaussian mixture models -** Gaussian mixture models are a probabilistic model for representing normally distributed Gaussian functions within an overall data as one function. In these mixture models the data is unlabelled i.e after obtaining the mixture model we cannot determine which point came from which Gaussian distribution. The means and variances for the colored buoys are calculated from this.

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

$$p(x) = \sum_{i=1}^{k} \phi_i \mathbb{N}(x|\mu_i\sigma_i) \quad , \quad \sum_{i=1}^{k} \phi_i = 1$$

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

2. **Expectation Maximization** - The Expectation Maximization algorithm assumed random starting points and thus computed for each point a probability that the point was generated by each one of the distributions. It iteratively found the maximum likelihood of the data in a given dataset and tried to maximize the likelihood of the data given those assignments to calculate the means and variances. With these iterations, we arrived at the local optimum and the respective means and variances.

Steps to compute GMM parameters using EM algorithm:

1. Initiating the algorithm by generating 3 random samples for 3 1D gaussian with different means and standard deviations.
2. These gaussian distribution is applied to all the training images.
3. With these probablities, new standard deviation and mean is calculated.
4. These new values are then again applied to all the images, and this step is repeated for 50 times as after that there was no significant change in mean or standard deviation.

5. These values are considered as trained values. 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 Maximum Likelihood Algorithms and Learning Color Models

Firstly, we computed the color distributions for each colored buoy and used these distributions to segment the buoys from background. For each colored buoy, we computed and visualized the average colored histogram for each channel of the cropped images. This provided some intuition on how many Gaussians to fit to the color histogram and the dimension of each Gaussian for our model. Then we finally implemented the EM algorithm and computed the model parameters, i.e. the means and variances of the Gaussians.

6 Thresholding and Buoy detection using 1D Gaussian

1. The means and variances of the Gaussian distributions of the RGB channels of each buoy is calculated using Expectation maximization algorithm after training the dataset.

2. Through the average Histograms, we analyzed the number of pixels of each R, G and B channels for each buoy .This step helps in choosing the most suitable color channel for each buoy.

3. Now, based on which R, G and B channels are influencing each buoy color, we gave a certain threshold value for the each Gaussian distributions in order to determine the probability of being a particular colored buoy.

   (a) For yellow buoy, we considered average of red and green channel.

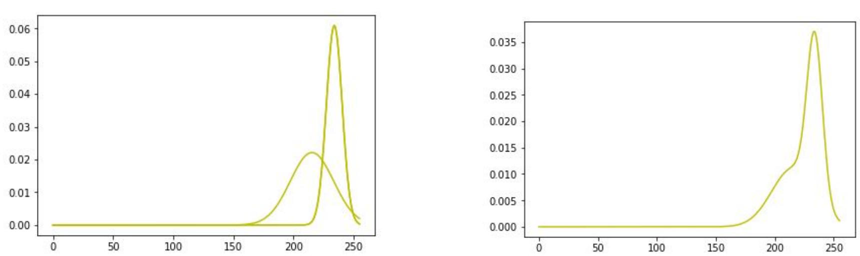

Figure 10: 2 Clusters in red and green channel for yellow buoy

   (b) For orange buoy, we considered only red channel.

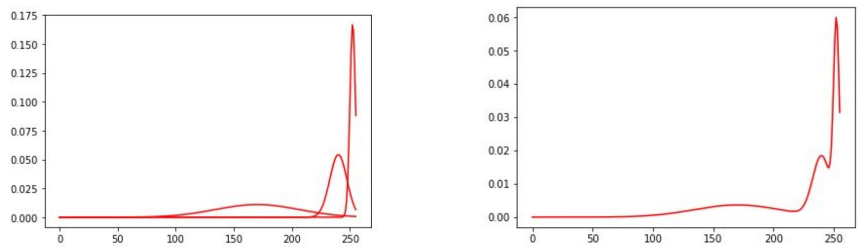

Figure 11: 3 Clusters in Red channel for Orange Buoy

(c) For green buoy, we considered only green channel.

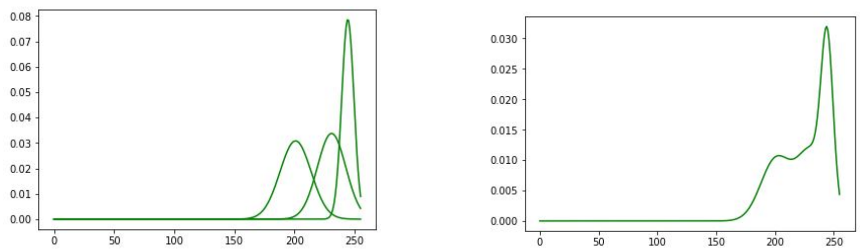

Figure 12: 3 Clusters in Green channel for Green Buoy

7 Buoy Detection

1. For this phase, we implemented the obtained model parameters on the video frames.

2. Then we applied thresholding to the GMM curve to obtain the intensity values in a particular channel, this gave us the pixel intensity that has the highest probability value of being a pixel of the buoy.

3. Through this we generated a binary image corresponding to each buoy.

4. *cv2.findContours()* is used to detect the contours and with that center of each contour is detected.

5. *cv2.minEnclosingCircle()* is used to draw the circle on the frames.

8 Results

8.1 Green buoy detection

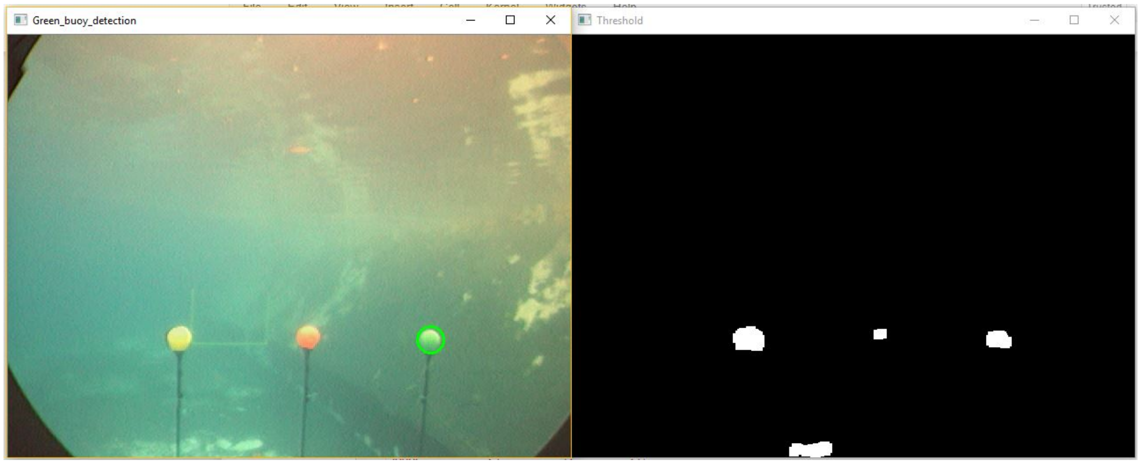

Figure 13: Green buoy detection - 1

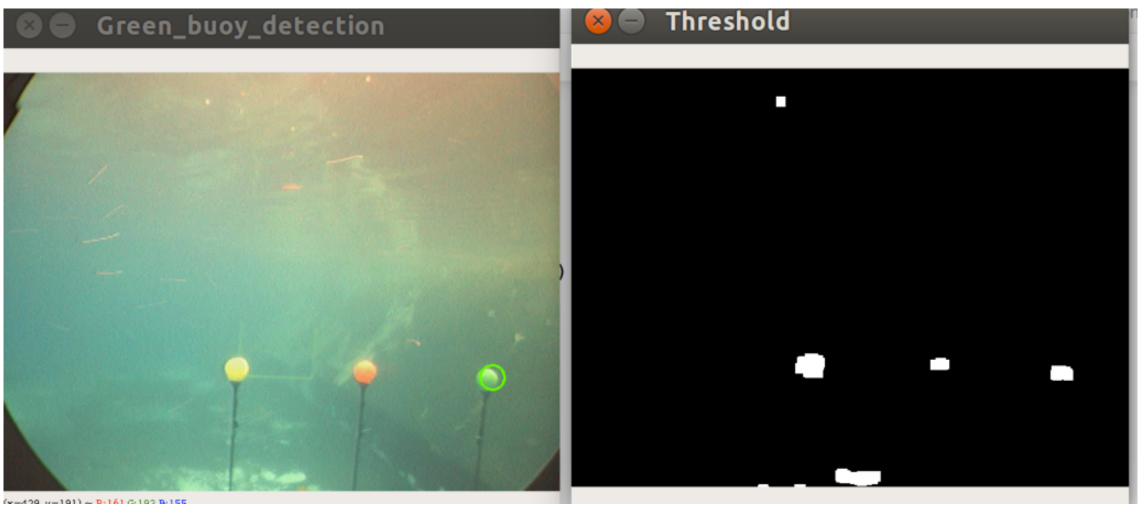

Figure 14: Green buoy detection - 2

8.2 Orange buoy detection

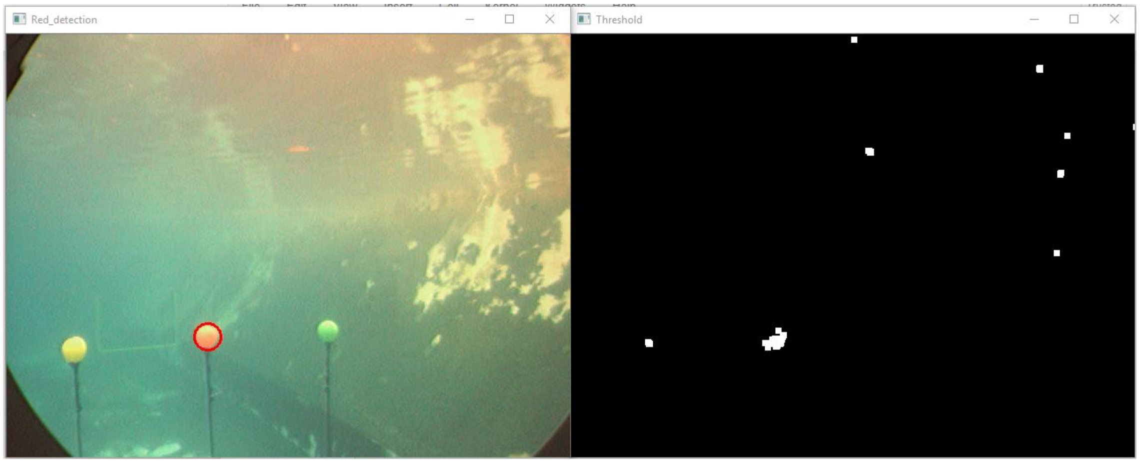

Figure 15: Orange buoy detection - 1

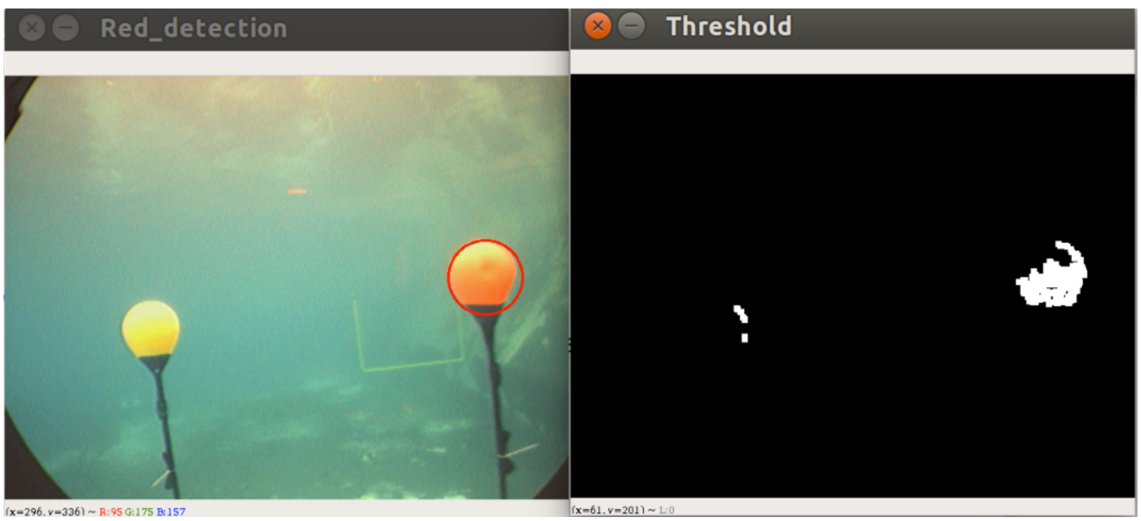

Figure 16: Orange buoy detection - 2

8.3 Yellow buoy detection

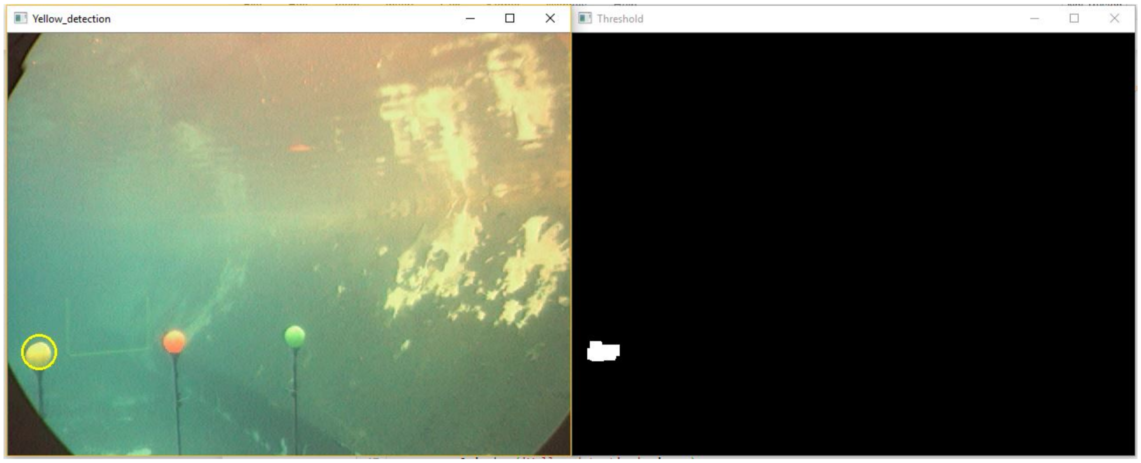

Figure 17: Yellow buoy detection - 1

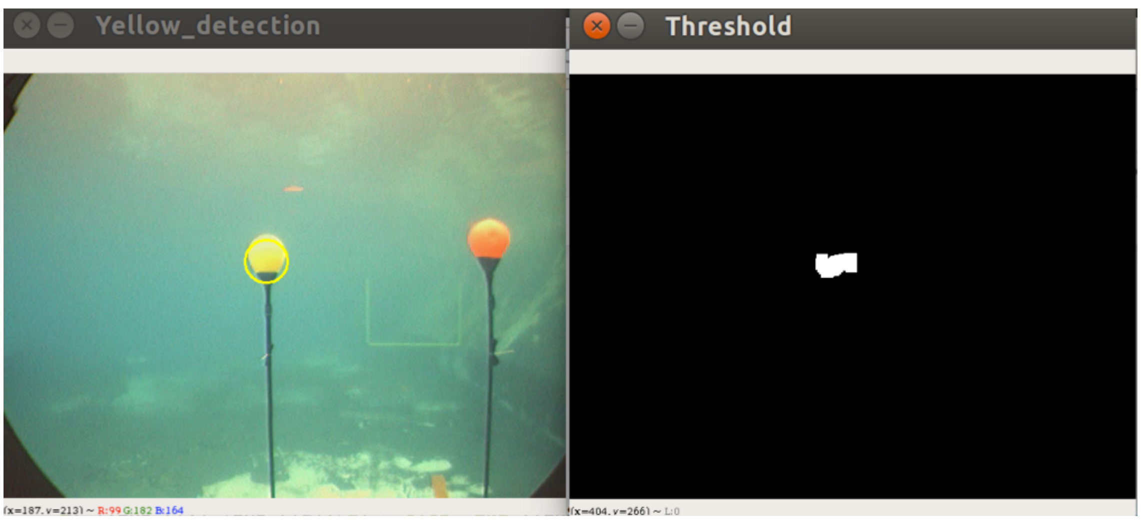

Figure 18: Yellow buoy detection - 2

8.4 All buoys detection

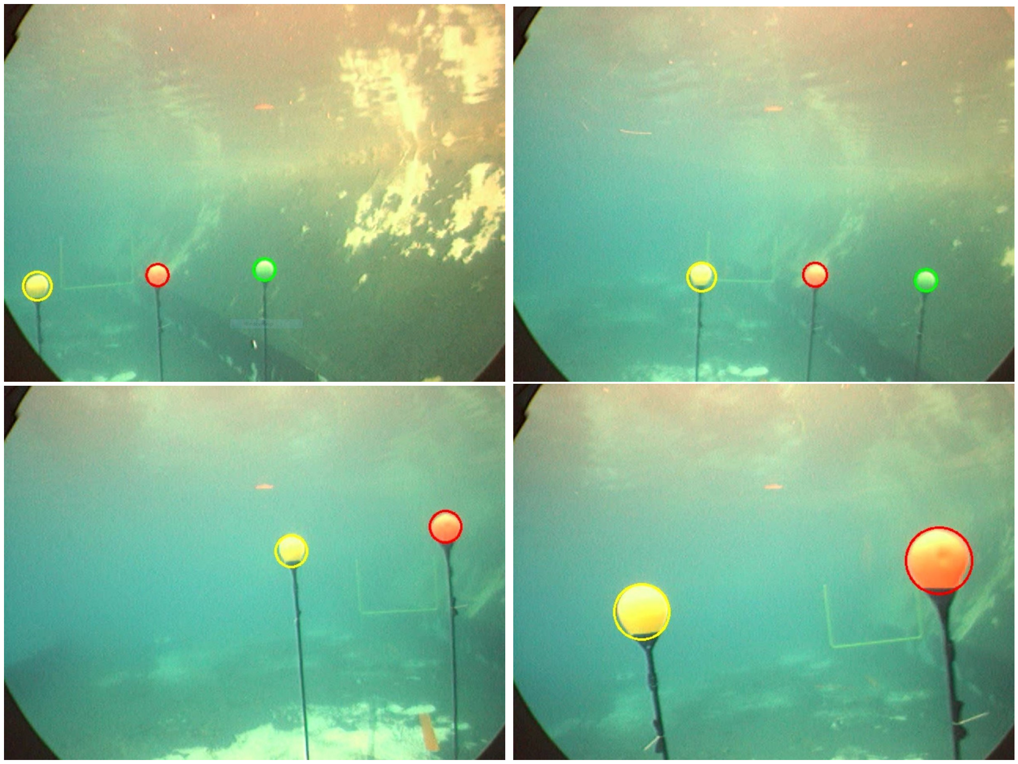

Figure 19: Final detection of all 3 buoys in the video

9 Analysis of the Model

There were some challenges faced during the project, they are mentioned below:

1. Initially our dataset images had more noises due to boundaries around the buoy which caused a problem in obtaining color distributions for the average histograms. We overcame this by improving our dataset images to more compact and tighter buoy surface areas.

2. For the buoy detection, we tried different morphological techniques to segment the foreground from background such as *backgroundsubtractor* from CV2 module. This was time consuming in order to identify the accurate function.

3. Buoy detection was slightly problematic as the varying lighting underwater conditions caused same intensity noises. This was tackled by contouring and setting a threshold value for the RGB channels based on their respective buoy colors.

4. Buoy detection for yellow was slightly time consuming because of the consideration of two RGB channels, i.e, green and red. So the thresholding phase had to be computed diligently for accurate detection.

10 References

1. EM.ppt from Coursework

2. https://www.youtube.com/watch?v=REypj2sy5U

3. https://www.youtube.com/watch?v=iQoXFmbXRJA

4. https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

5. https://docs.opencv.org/3.3.0/db/d5c/tutorial_py_bg_subtraction.html

6. http://www.ee.bgu.ac.il/ haimp/ml/lectures/lec2/lec2.pdf