1 Data Preparation

The first section of this project is to extract data from the given input video for training and testing the EM algorithm.

1.1 Extracting Data from the video

We have extracted training data for yellow, green and buoys by using opencv mouse handling events. By using $cv2.EVENTLBUTTONDOWN$ and $cv2.EVENTLBUTTONUP$ funcions, we were able to draw rectangles and crop out images of the buoys from the frames. Then, we saved the cropped images in Data folder. This functionality is carried put by *GenerateData.py* file.

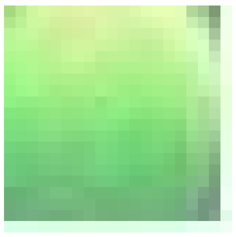


Figure 1: Cropped buoy

2 Gaussian Mixture Models and Maximum Likelihood Algorithm

2.1 Expectation Maximization Algorithm:

EM algoritm is used for clustering the data in this case clustering and detecting yellow, orange and green buoys. We used 1-D gaussian distributions for each buoy which provided the distribution of dominant channel of the RGB colour space for that buoy. For green buoy, green channel is considered. For yellow channel Red + Green channel is considered. For orange channel, R channel is considered for gaussian distribution.

$$\mathcal{N}(x \mid \underset{\text{mean}}{\mu}, \underset{\text{variance}}{\sigma}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Figure 2: 1-D Gaussian Distribution formula

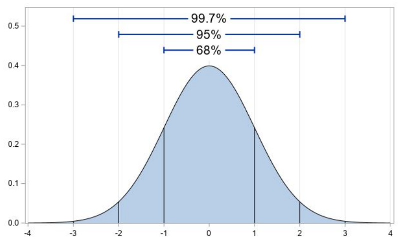


Figure 3: 1-D Gaussian Distribution

These are the steps involved in EM agoritm.

1. First step is to initialize mean and standard deviation with random values.
2. Then gaussian distribution is generated by passing data of R or G or B channel of colour space with the randomized means and variances.
3. E step: Then gaussian distibutions are combined along with their weightage. Weightage depends on number of clusters considered. Then, the probabilities that the points belongs

$$p(x) = \underbrace{\sum_{k=1}^{K}}_{\text{Number of Gaussians}} \underbrace{\pi_k}_{\text{Mixing coefficient: weightage for each Gaussian dist.}} \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

Figure 4: Combining gaussian Distributions

to a cluster must be found. This is found by using the formula:

$$= \frac{\pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x \mid \mu_j, \Sigma_j)} \qquad \text{where, } \pi_k = \frac{N_k}{N}$$

Figure 5: Probability of the cluster given points

4. M step: Now find the updated mean and standard deviation using the following formula:

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)} \qquad \Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)} \qquad \pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

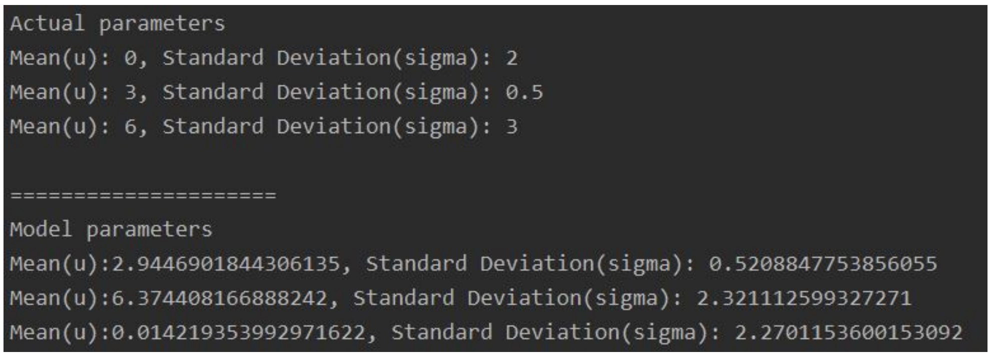
Figure 6: Updated mean and SD formula

5. Now we compute the log likelihood of the probabilitilies. This process is iterated until the difference between the consecutive log likelihoods is less than a threshold value.

$$\ln p(\mathbf{X} \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathrm{N}(\mathbf{x}_n \mid \mu_k, \Sigma_k) \right\}$$

Figure 7: Loglikelihood formula

First the EM algorithm is tested before the traning step. This is done by creating a guassian distribution with known means and standard distributions and comparing the means and standard distribution obtained from the output of EM algorithm.
We have considered three guassian distributions of numbers randomly generated having means 0, 3, 6 and standard deviations 2, 0.5, 3. We have provided these data to put EM algorithm and the following results are obrained. The means and standard deviations obtained from the EM algoritm are almost equal to the actual values.



```
Actual parameters
Mean(u): 0, Standard Deviation(sigma): 2
Mean(u): 3, Standard Deviation(sigma): 0.5
Mean(u): 6, Standard Deviation(sigma): 3

====================
Model parameters
Mean(u):2.9446901844306135, Standard Deviation(sigma): 0.5208847753856055
Mean(u):6.374408166888242, Standard Deviation(sigma): 2.321112599327271
Mean(u):0.014219353992971622, Standard Deviation(sigma): 2.2701153600153092
```

Figure 8: Test output

3 Learning Color Models

In this section, we applied the process to our training data. But, before that we need compute and visualize the color histogram for each channel of the sampled/cropped images gathered during the data preparation phase for each image.This will provide some intuition on how many Gaussians [N] to fit to the color histogram. The following are the histogram outputs:

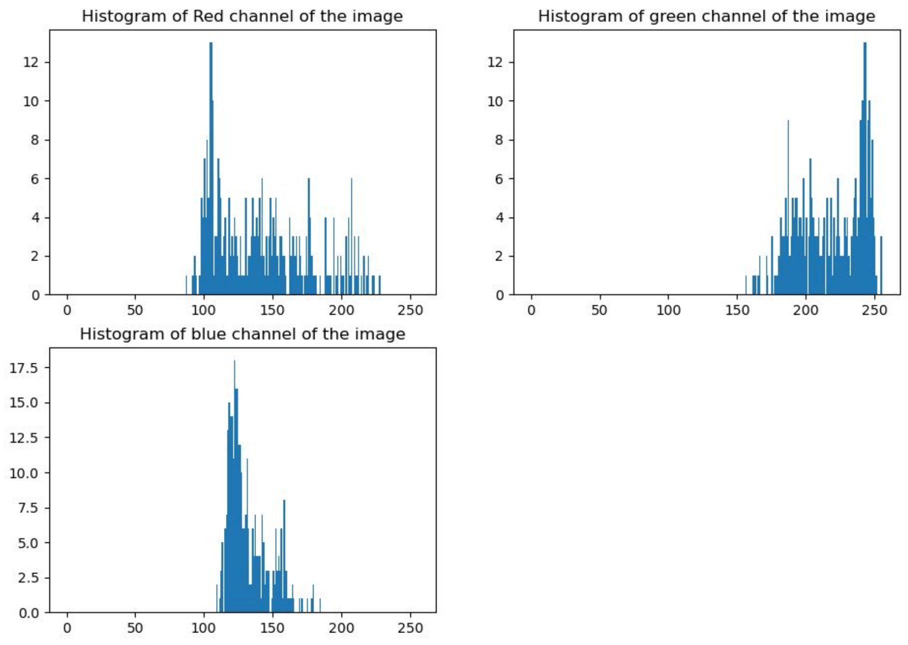


Figure 9: Green buoy channel distribution

For green buoy detection, 3 clusters are considered as there are three clusters formed in the Green channel distribution.

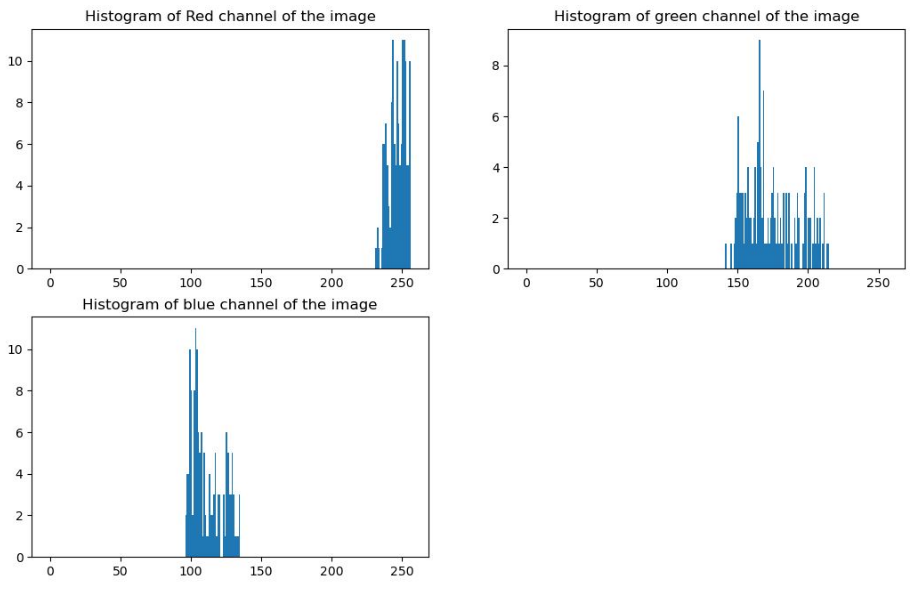


Figure 10: Orange buoy channel distribution

For orange buoy detection, 2 clusters are considered as there are two clusters formed in the Red channel distribution.

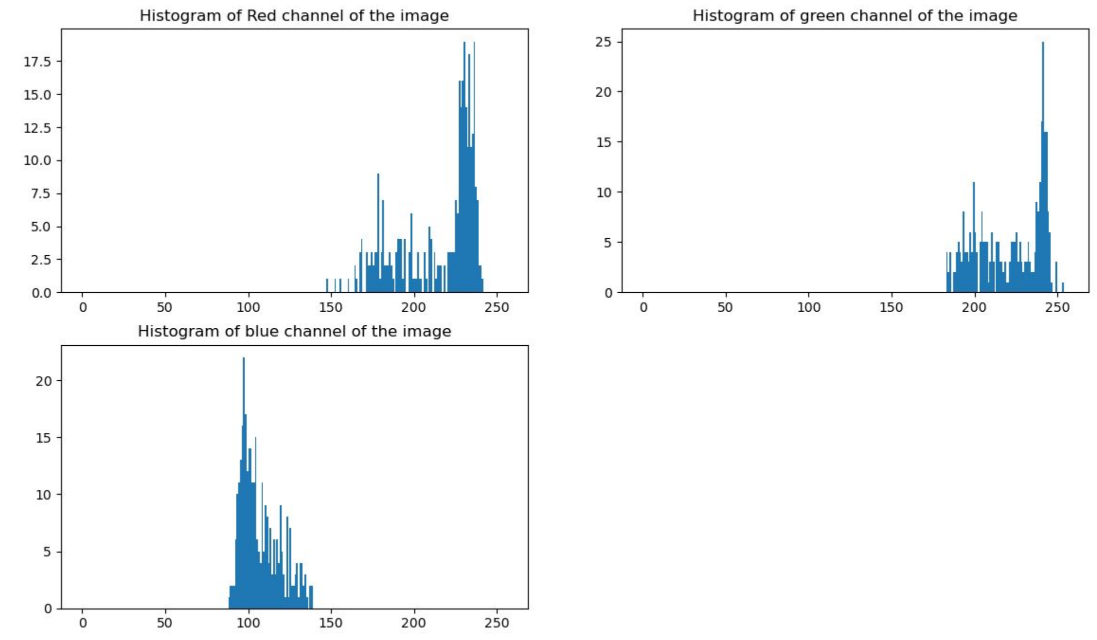


Figure 11: Yellow buoy channel distribution

For yellow buoy detection, 2 clusters are considered as there are two major clusters formed each in Red and Green channel distribution.

4 Buoy Detection

The following procedure is followed to detect the buoy:

1. First training and testing is performed on the obtained cropped images of the buoy in the Data preparation phase. We extract the channel of the image we want to work with as our data. As number of clusters are obtained from the histograms, we run the EM algoritm. The algorithm runs until it converges to the given value (bound = 0.0001) or until it reaches the maximum number of iterations specified (1000). For each cluster, the last mean and standard deviations are stored.

2. Now the given input video is processed frame by frame. The same procedure is employed for frames. The data now becomes the channels of the frame we want to with. For 1-D gaussian distribution, the means and standard distributions obtained from training data is considered. Then again the probabilities are calculated for each pixel in the image. If the probabilities are grater than a threshold value, that channel of the pixel is made 255.

3. Then, the frame is blurred and edges are determined using canny edge detection.

4. Then, contours are found using $findcontours()$ function and only the contour with minimum length is considered as the buoy is small. Then we obtain the boundary points of the contours using $convexHull()$ function.

5. Then, we enclose the hull with a circle using $cv2.Circle()$ function.

5 Output

The following are the output frames obtained after detecting the buoys:

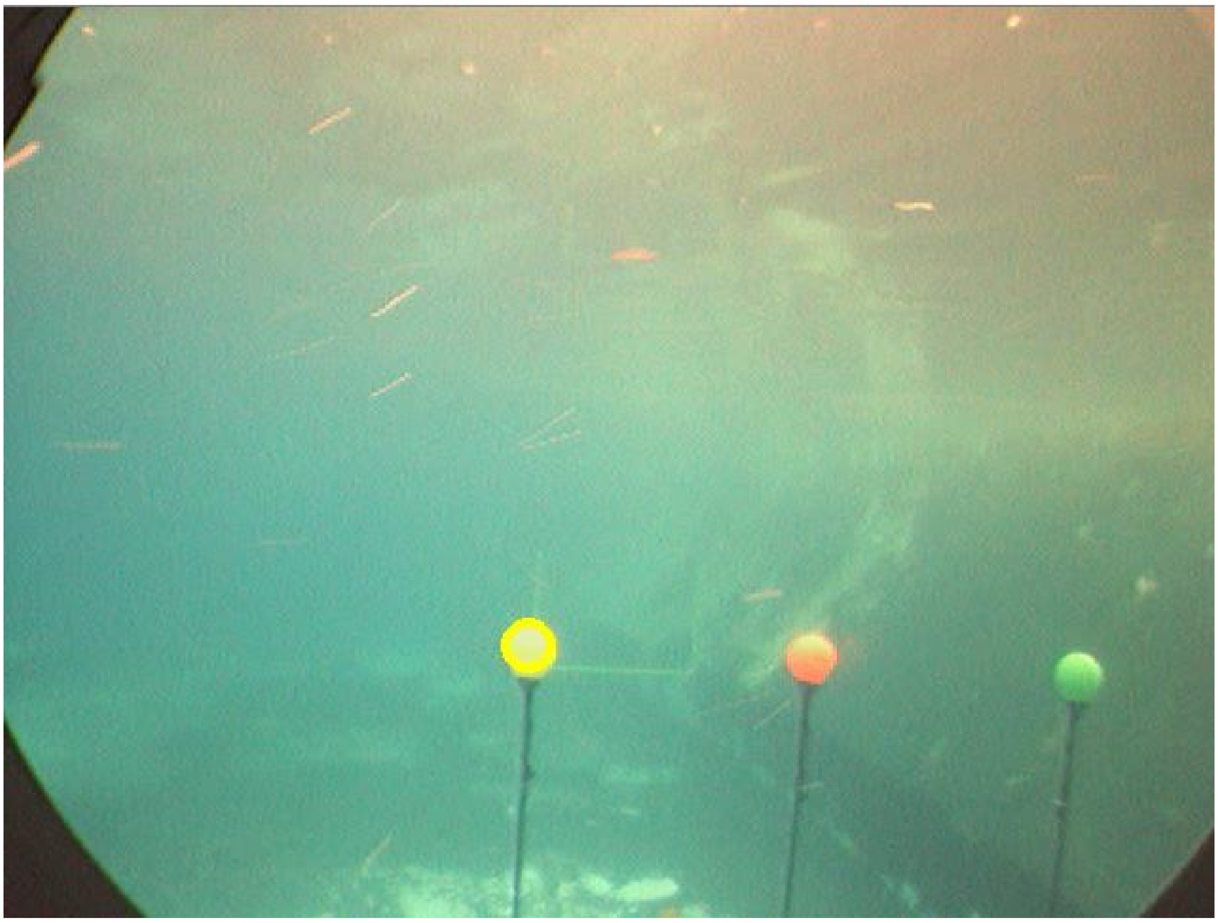


Figure 12: Yellow buoy detection

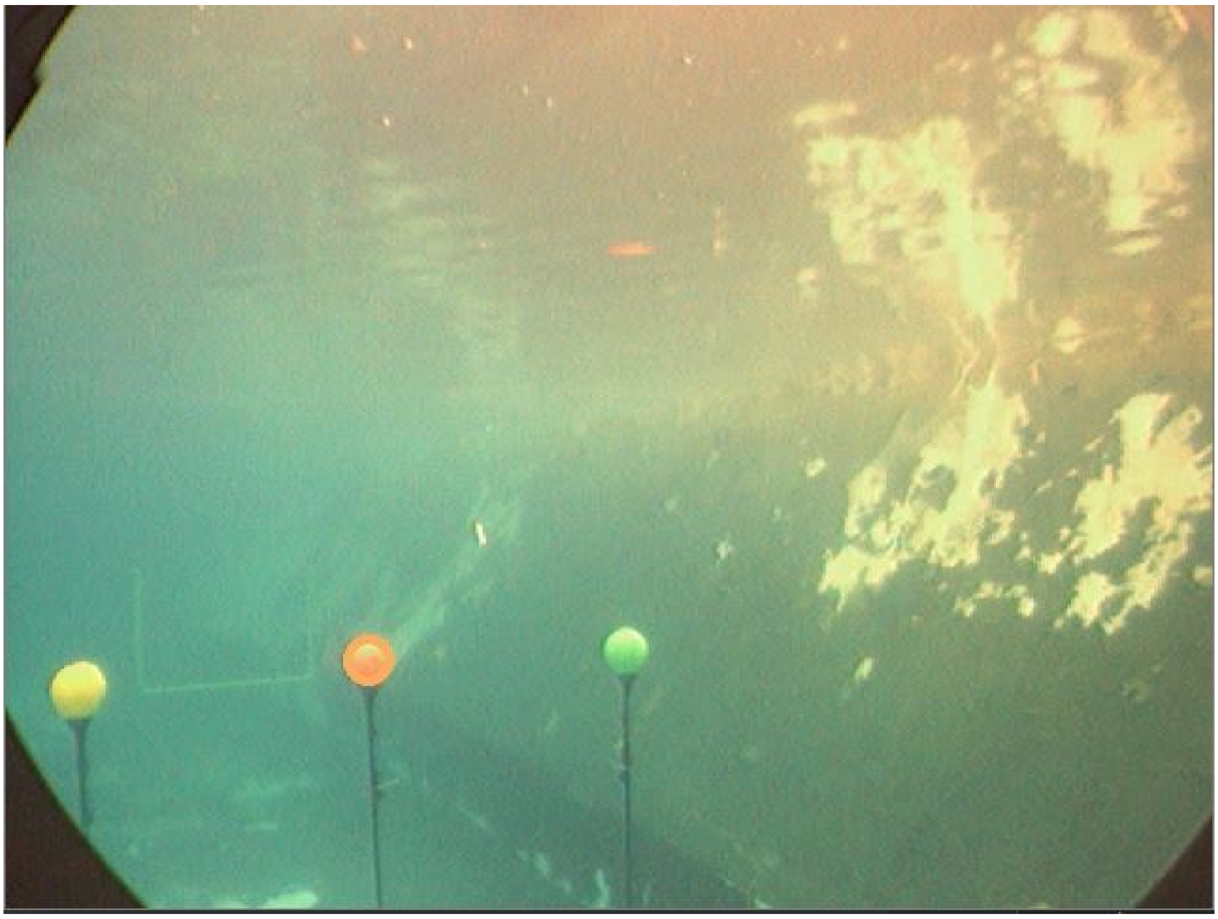


Figure 13: Orange buoy detection

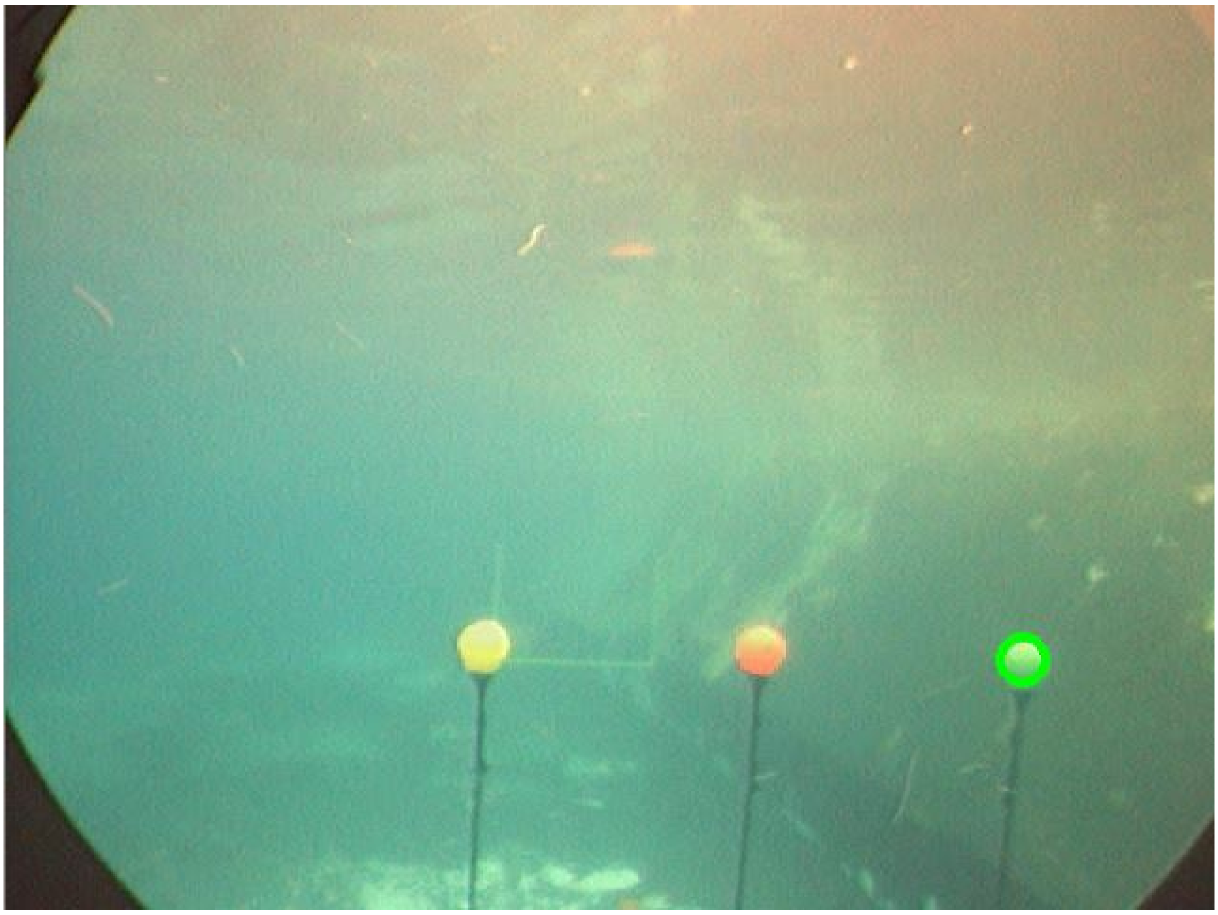


Figure 14: Green buoy detection

6 Challenges Faced:

Detection of green buoy was very difficult as the video has only few frames containing green buoy.

7 Team Members:

1. Eashwar Sathyamurthy 2. Akwasi A Obeng 3. Achal P Vyas

8 Output Videos:

To access output videos please use this link.