“[ENPM 673] PROJECT - 3”

ENPM 673 – PERCEPTION FOR AUTONOMOUS ROBOTS

PROJECT 3 REPORT

ADITYA VAISHAMPAYAN -116077354

ISHAN PATEL – 116169103

NAKUL PATEL – 116334877

Instructor: Dr. Cornelia Fermuller

Contents

INTRODUCTION:

A buoy is uniquely colored floating device, which is anchored to the bottom underwater for marking purpose in navigation. We are given the video sequence of the distinctly colored buoys from the camera-view of some underwater vehicle. Our goal for this project is to segment and detect the buoys of each color in the scene, from the given video sequence.

As for segmentation based on colors, one can use hardcoded thresholding values for respective colors, and mask the image with this thresholded image to get the final detected colored buoy. But since the buoy is underwater, it would be surrounded by varying light intensities and noises that would render this conventional segmentation techniques ineffective.

Thus, to overcome this drawback, Gaussian Mixture Models are used to learn the color distribution and use the model to segment and detect buoys.

DATA PREPARATION:

The first step is to extract the frames out off the video sequence. Since the frame rate of given video is 5 frames/ second, and video is of 40 seconds length, we finally obtain 200 frames as our dataset. Now, these frames are divided into training set and test set in the ratio of 70% - 30%, to avoid the overfitting.

The training set is cropped further to get the images of cropped buoys of each color, to record the color intensities of buoys and extract the model parameters using Expectation-Maximization algorithm. And the test set will be used to evaluate the performance of the model developed based on training set.

So, for test set, the frames with all buoys have been considered, while for training set, we will use the cropped images of each buoys and learn the color model.

CALCULATING AVERAGE HISTOGRAM:

The next step is to calculate the average color histogram for each channel of the cropped image of buoys. Since the color space is RGB, we will get color histogram for Red, Blue and Green channels for each image of buoys. Thus, we will get the intuition of average color distribution for each buoy. So, consider following general pipeline to achieve this objective:

For each colored buoy:

- Split the color channels of each image.
- Calculate the histogram of channels using cv2.calcHist() function of OpenCV.
- Finally, take the average of corresponding channels' histogram for all images
- Thus, we get average color histogram for red, blue and green channels of multiple images of same colored buoys.

After getting the average color histogram, we plot them using matplotlib.pyplot function plot().

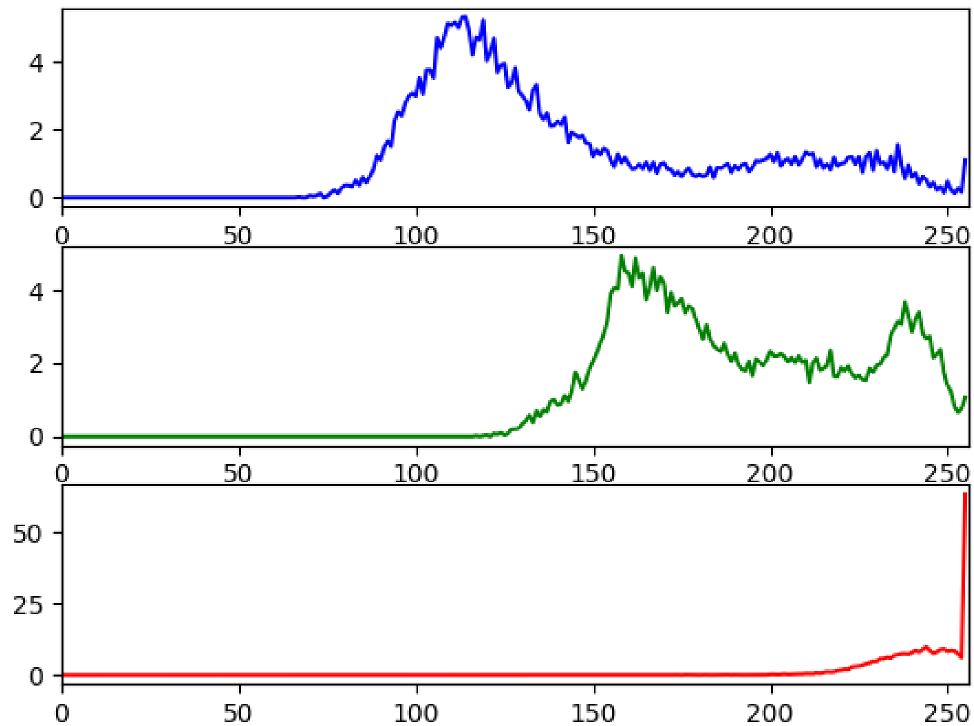


Fig: Average histograms for the red buoy

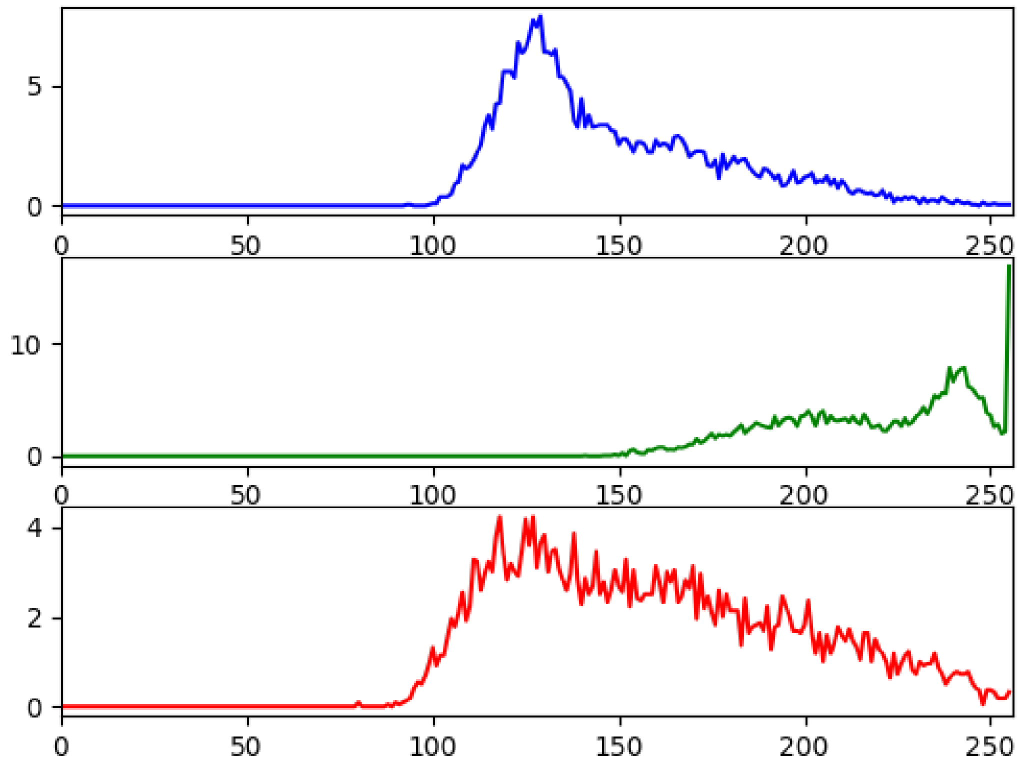


Fig: Average histogram for green buoy

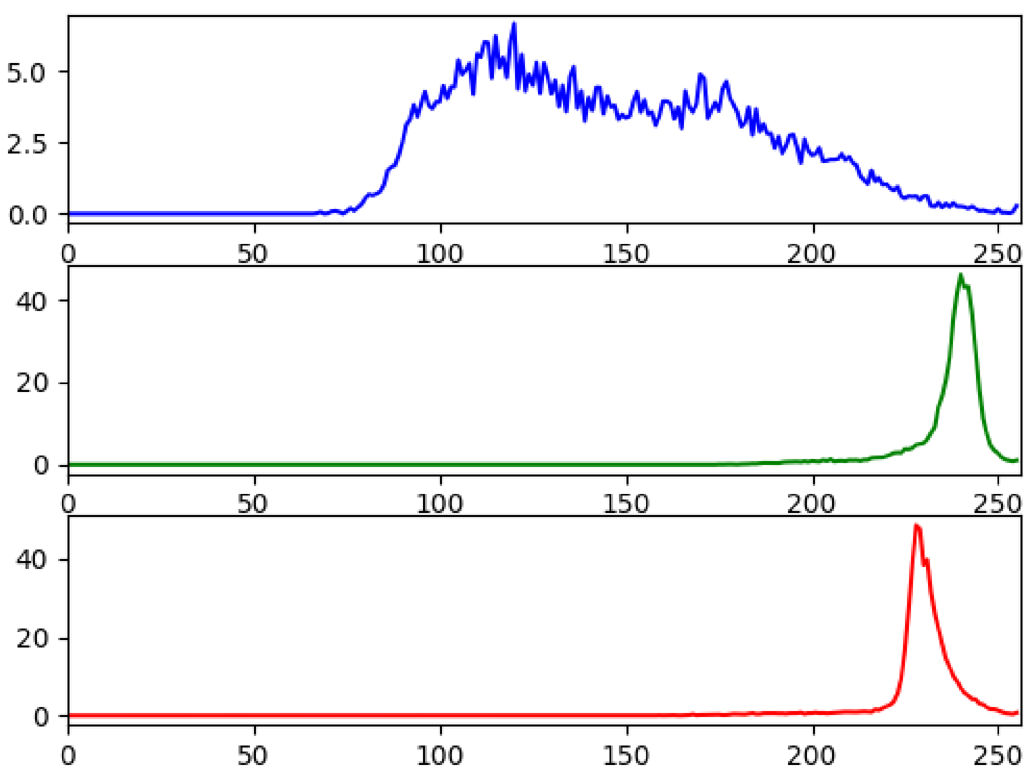


Fig: Average histogram for Yellow buoy

SEGMENTING USING 1-D GAUSSIAN:

After the average color-histogram have been obtained, next step is to build the 1-D Gaussian from them, so that the pixels of image can be segmented based on this Gaussian distribution to yield color segmented image. Thus, we will generate a model to segment the buoys using this 1-D Gaussian.

As interpreted from the above histograms, we can extract the combination of appropriate channels for detecting the buoys. For example, for detecting red buoy, we can use the histogram of red channel to generate 1-D Gaussian, since that channel is dominant for red color. Similarly, for green buoy, we use the histogram data of green channel to generate 1-D Gaussian. And for yellow buoy, we will extract the histogram of red and green color.

The Gaussians are generated by calculating the mean and variance from the histogram data of main color channel of respective buoys (Eg, Red channel for red buoy, green channel for green buoy and red + green channel for yellow buoy)

Once the gaussians are generated, the pipeline for segmentation of buoys is as follows:

- Iterate over each pixel of each images in test dataset.
- Extract the intensity value of the color channel corresponding to the buoy which is considered for detection.
- Check whether that intensity falls (after taking a certain threshold) under the Gaussian distribution of that color channel.
- If yes, then consider that pixel for segmentation in one group.
- Finally, the detected buoy is encircled with the contour of same color.

And after repeating steps 1 through 5 for all frames, we generate a video sequence from the images. Here, for each buoy, we used 1-D data input for each major color.

5. EXPECTATION-MAXIMIZATION ALGORITHM:

Most of the data in real world cannot be best characterized by simply a Gaussian. The data points are generated from a finite number of Gaussian distributions, whose parameters are unknown.

Suppose that we are trying to learn the mixture of Gaussians for our training data(frames), but we don’t know the parameters like mean and variances of the Gaussians involved, and the probabilities that which point came from which Gaussian distribution. To solve this problem, we employ Expectation-Maximization algorithm.

EM Algorithm:

It is an iterative optimization technique which consist of two major steps:

To start with, one first considers random datapoints with some K distributions with predefined mean and variances. Then compute the probability of each point that it was generated by each of the distributions. And repeatedly update the parameters (mean and variance) by maximizing the log-likelihood of he data. Sufficient number of iterations for running this algorithm will make the log-likelihood function to converge. Finally, we get the mixture of Gaussians which accurately describes the considered data.

Given the initial dataset and number of Gaussians (**Symbol: K**) for modelling the data, the steps of the algorithm along with the formulae are as follows:

- Initialize the mean $\boldsymbol{\mu_k}$, covariances $\boldsymbol{\Sigma_k}$ and mixing coefficients $\boldsymbol{\pi_k}$ and evaluate the initial value of log-likelihood.

- **E- step:** This is expectation step. Here, we compute the expected values of the latent variable for given parameter values.

$$\gamma_j(x) = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_k, \Sigma_k)}$$

- **M-step:** This is maximization step. Here, update the parameters of our model based on the latent variable calculated using above step.

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

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

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

- Evaluate the log-likelihood. And if there is no convergence, return to E-Step:

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right\}$$

Here, N refers to the number of samples of the data and K refers to the number of Gaussians.

Using the above steps, we calculate the true parameters of each Gaussian, and hence model the data into defined number of model.

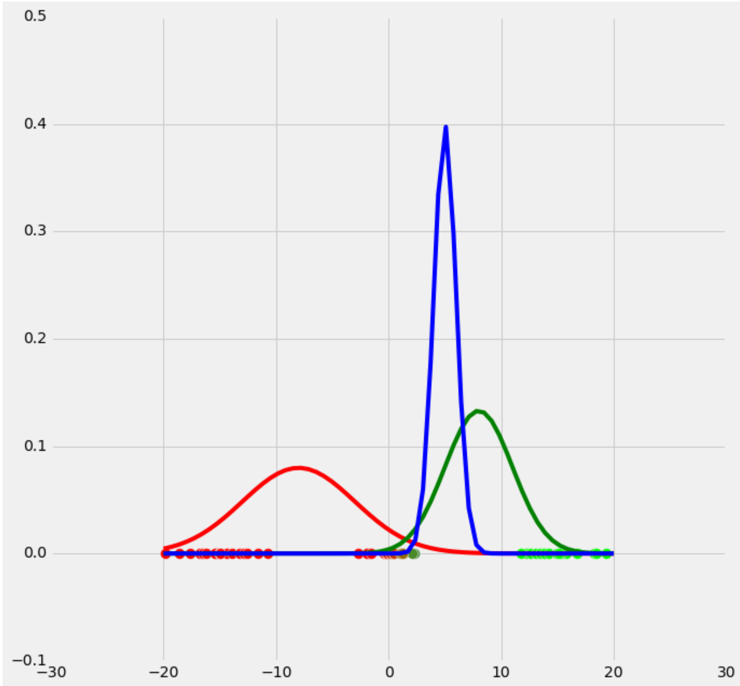


Fig: Random gaussians initialized

The random gaussians initialized have a mean of around 5, 7 and -8.

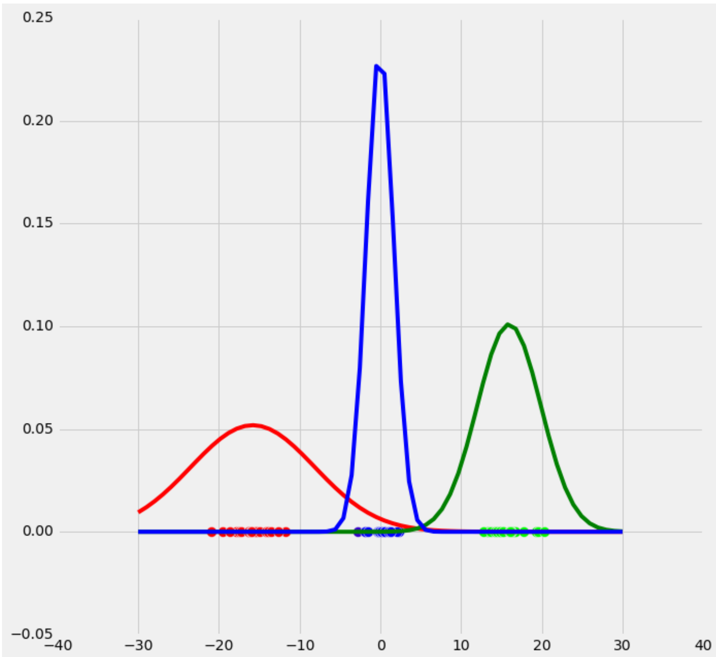


Fig: Recovered gaussian curves

The recovered gaussians have a mean of 0, 16 and -18.

LEARNING COLOR MODELS:

In the first section we will see how many gaussians are needed to fit 1D data input for each major color of thee buoy. In the second section we will calculate the model parameters for the 3D input of (RGB colors) for each buoy with different gaussians.

This part of the project focuses on the use of the clustering concept developed in Section 4 of this project for the goal of color segmentation. We first visualize the color histograms for each buoy and based on this data, we decide upon the number of Gaussian to be used to model a particular buoy. We also decide the dimension of the Gaussian to be used. It was much difficult to just decide upon the desired model looking at the color histograms. So, instead, I applied all the possible models with K ranging from 1 to 5 on the images and decided upon the model based upon the quality of output per model.

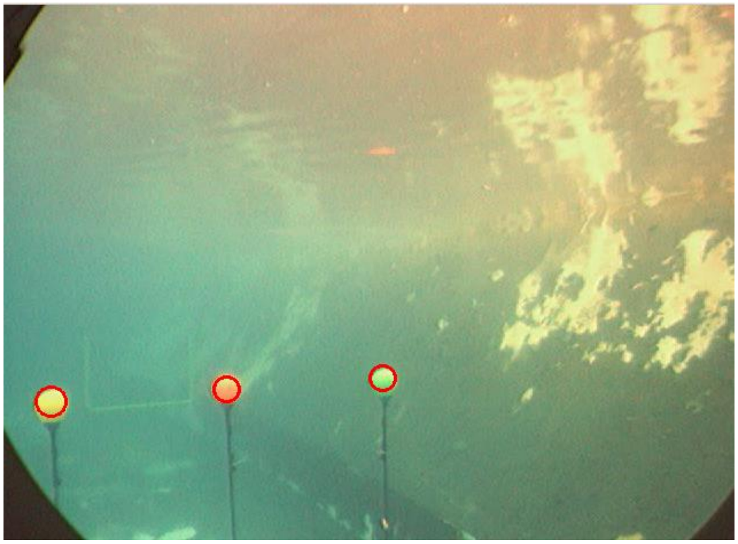
BUOY DETECTION:

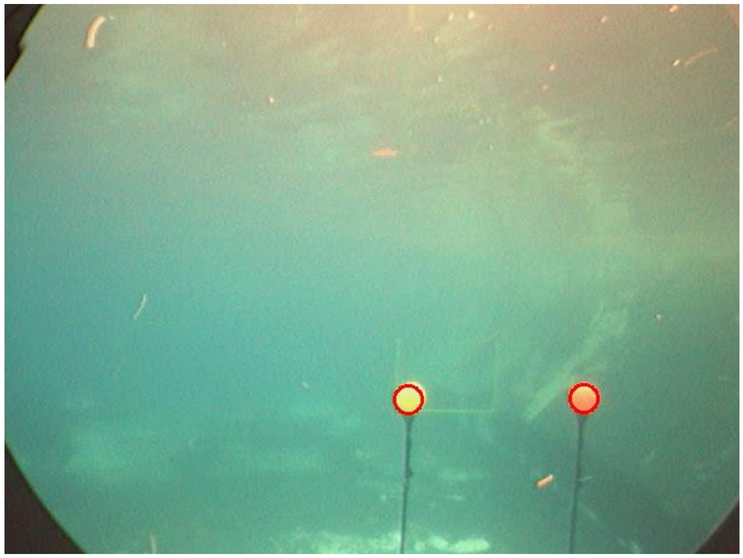
When given a 3 D input of [R G B] intensity values a total of three gaussians give the best binary image output. Upon comparing the binary image output of 3Dimensional input with 1 Dimensional input it is seen that the binary image posses a lot of noise near the ROI i.e. the buoys and thus causes difficulty in segmenting. Upon adding thresholds of standard deviations, we couldn’t remove the noise.

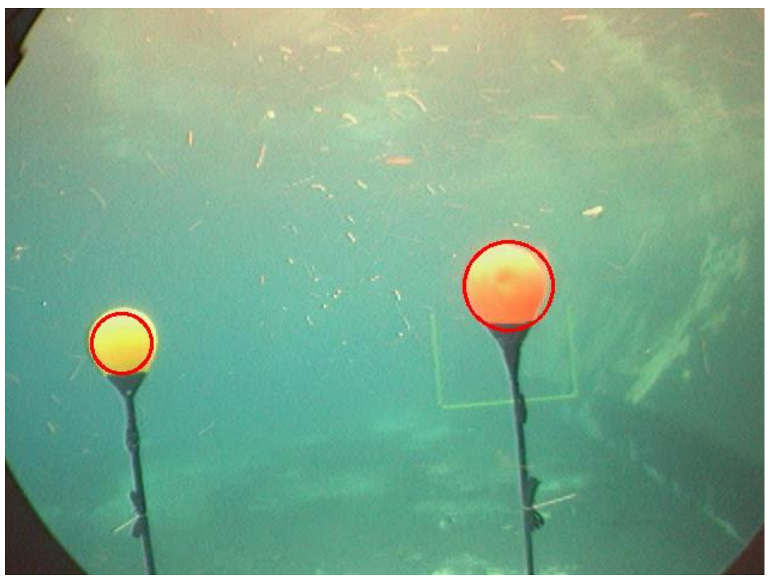
Thus the final buoy detection method is as follows:

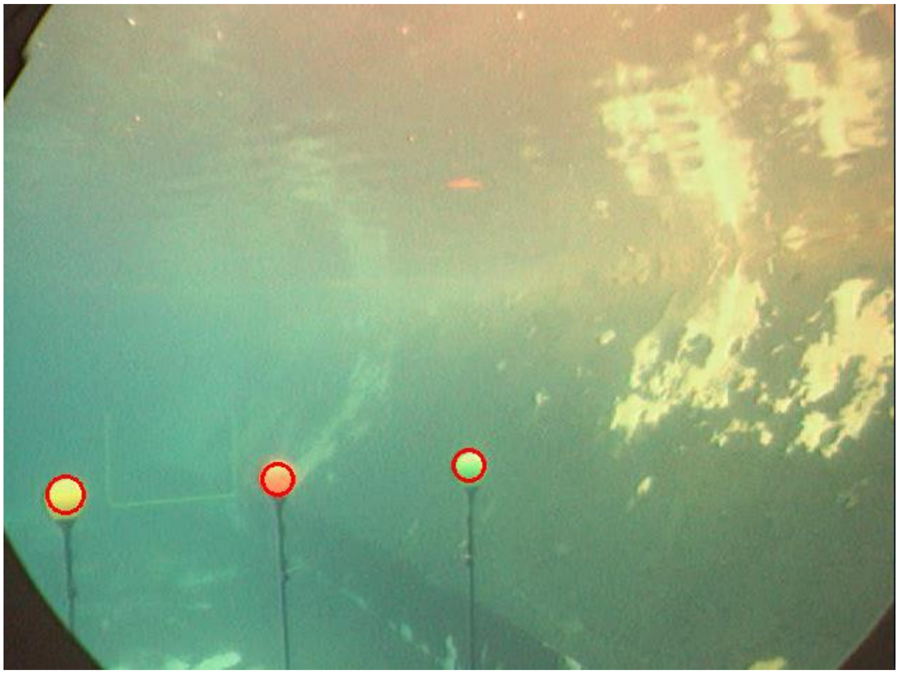
- We have used 1D input with multiple gaussians. Thus, the input has red intensity values for red buoys, green intensity values for green buoys and (red+ green)/2 intensity values for yellow buoys.
- Also, we have chosen 4 gaussians for red color of red buoys and similarly 4 gaussians for green and yellow color buoys.

The following figures show the detection of all three buoys:









FURTHER ANALYSIS:

RGB images have an inherent problem such images not being pure in color. Because of which noises such as background color, lighting and motion makes it difficult to segment the buoys. I think that HSV and LAB color spaces are more adaptive to changes in images such as lighting, shadow etc. Upon taking the HSV color space, the hue generally remains constant and differences in lighting are represented through the value component and the white intensity of each color is represented through the saturation component. The hue and saturation values remain stable despite changes in lighting.

- I think using HSV color space can result in difficulty segmenting the green buoy because HSV works on HUE saturation values, and because of the green color background, segmenting the green buoy will become more difficult.

I also think that using LAB color spaces will give much better output because, the lab workspace has specific planes for green /red channels and yellow/blue channels. An alternative to the conventional color spaces is the RGBY color space because it adds an additional value of yellow colour. As yellow is a combination of red and green, sometimes the yellow buoy gets detected as green buoy if the green tint becomes too high. Also, adding the yellow color would compensate for the noise effects because of the sunlight as it is generally a yellow tint. Resulting in negation of the RGB values.

GOOGLE DRIVE LINK:

https://drive.google.com/drive/folders/1Cx2OyAKTExJG2ChtNDQmFzM7o2jSYaDB

REFERENCES:

1) Bishop Textbook- Pattern Recognition and Machine Learning
2) Lecture presentation (EM.pptx)
3) http://www.rmki.kfki.hu/~banmi/elte/bishop_em.pdf
4) http://www.cse.iitm.ac.in/~vplab/courses/DVP/PDF/gmm.pdf