ENPM673-Project 3

Mushty Sri Sai Kaushik, Alexandre Filie, Santhosh Kesani

April 7, 2020

1 Introduction

This project performs colour segmentation underwater with the use of methods like implementation of Gaussian Mixture Models and Expectation Maximization. The video provided includes 3 colours of Yellow, Orange and Green. However, due the variation of the shades of colour in each frame, conventional techniques could not be implemented for segmentation. Thus, we use the Gaussian techniques to train the model using the data set that is generated to help the model detect the colour based on the variations. The output images of all three colours being detected are shown along with the histograms for each colour. The final video with the overlap of the detection frames for each colour is also attached in this file.

2 Data Preparation

Data preparation for this project is required to be done from the video in order to train the Gaussian distribution for each colour. There were steps implemented in the Training code to prepare the individual frames for each colour from the video provided: 1) Individual frames were extracted from the video and used as sample frames for the system training.
2) The user is prompted to select a colored buoy and the coordinate of the click is recorded for evaluation.

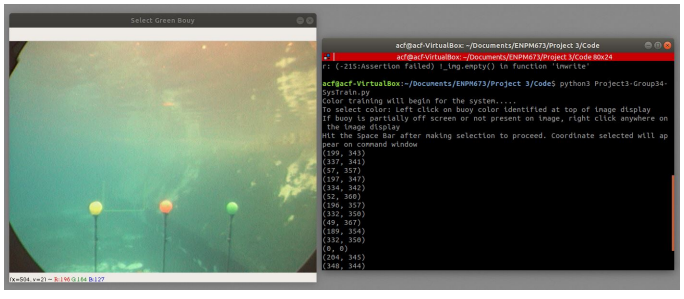

Figure 1: Screenshot of user operated color training

3) The sample frames are cropped around the user selection for the respective color to obtain a better color sample. By removing unwanted zones from each cropped frame, it was possible to develop a more accurate color representation of the three buoys.
4) The resulting data is then converted to a np.array format to obtain and process the data.
5) The data was then separated to training and testing data in a 70-30 ratio.
6) The training images are stored for future use by other processes. A .npy file is stored containing data respective to the color histograms generated from the training process.
7) The data collected from the training process is evaluated for each color to generate a mean value and standard deviation for the color spread representative of the buoys. These histograms and statistical values can be used by future processes.
8) The images were further made smooth and clear using Gaussian blurring.

3 Gaussian Mixture Models

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 the general equation.

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 finds the maximum likelihood of the data in a given data set 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 three 1-D Gaussian with different means and standard deviations.
2. These Gaussian distribution is applied to all the training images.
3. With these probabilities, 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.

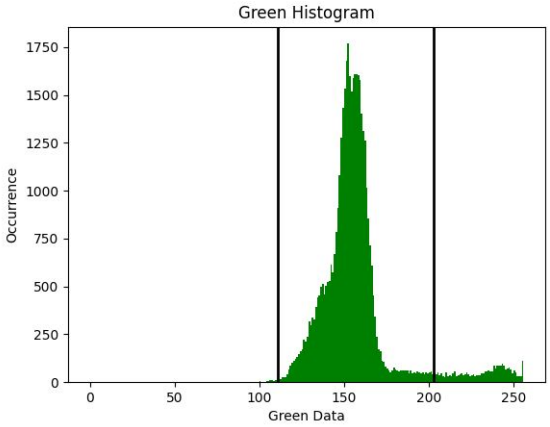

Figure 2: Histogram of Green buoy

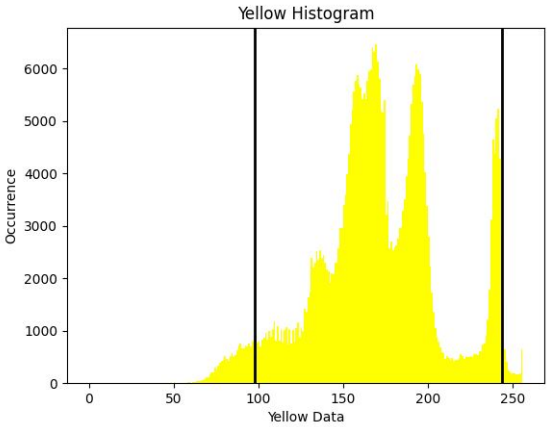

Figure 3: Histogram of Yellow buoy

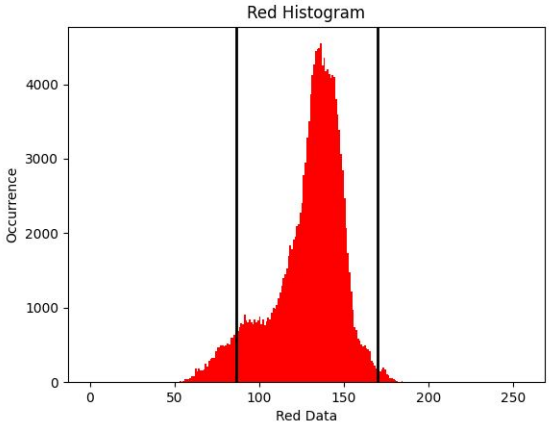

Figure 4: Histogram of Red buoy

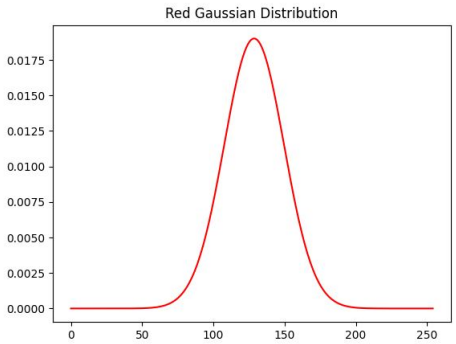

Figure 5: Image of Red 1-D Gaussian

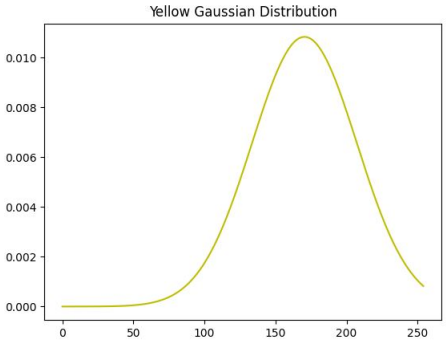

Figure 6: Image of Yellow 1-D Gaussian

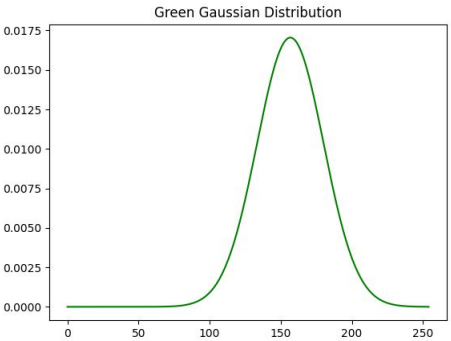

Figure 7: Image of Green 1-D Gaussian

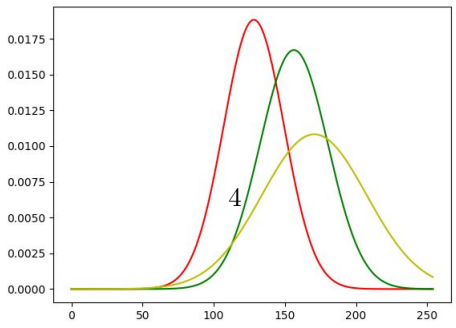

Figure 8: Image of Green 1-D Gaussian

4 Buoy Detection

For this phase, we implemented the obtained model parameters on the video frames. Following this, we apply the optimum threshold 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 the buoy. Through this we generated a binary image corresponding to each buoy. We then use cv2.findContours() is used to detect the contours and with that center of each contour is detected. Finally, cv2.minEnclosingCircle() is used to draw the circle on the frames.

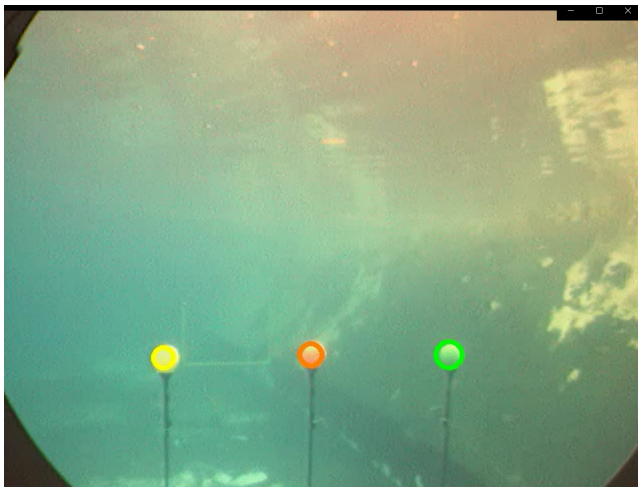

Figure 9: Image of Buoy detection

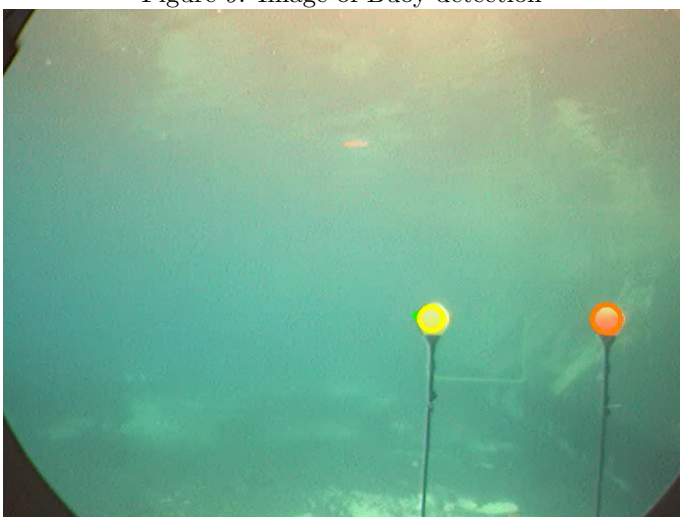

Figure 10: Image of Buoy detection

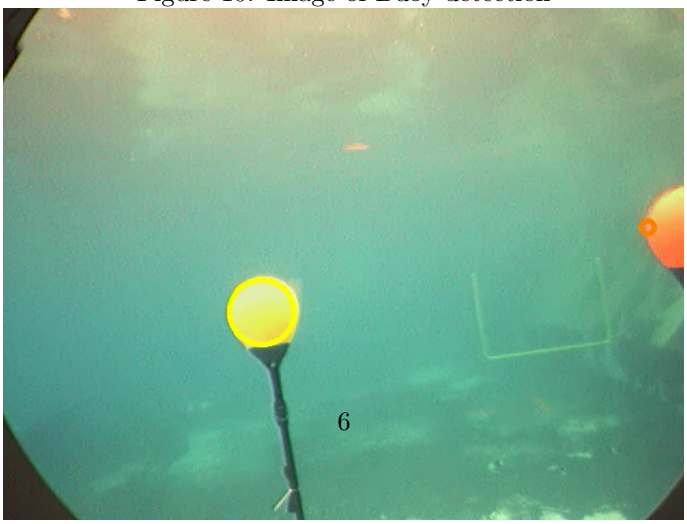

Figure 11: Image of Buoy detection