Robotics in the University of Maryland

ENEE673 Perception of Autonomous Robots

Report of Project 3

An Li (116156356)

Lih-Narn Wang (116415682)

Yu-Kai Wang (116349802)

1. Data Preparation

The most important factor for all kinds of learning method is the training data. In this project, the training data are crop images of buoys, which contain three attributes: values in R, G and B channels. To best extract possible variations of these attributes (variance within one channel and covariance between two channels) without interference of noises (RGB values from background), we have to crop buoys' images with background being filtered. To best filter the noise of the background, we crop the image by circle, so that it'll have maximum ratio of ROI and noise.

To implement this, on python we can use OpenCV with cv2.EVENT_LBUTTONDOWN to connect mouse clicking:

- When detecting mouse clicking, the program saves the current coordinates of the clicking point.
- Once the mouse is clicked again, the coordinate at the edge of the circle will be saved to the program. Then, we calculate the radius of the circle and crop the image.
- Due to the reason every image is in square shape, we assigned zero to all the points outside the circle, so that we can filter the noise and get the best training data.

Figure 1: Cropped training data.

2. Expectation Maximization

In real life, many datasets can be modeled by Gaussian Distribution (Univariate or Multivariate). So, it is intuitive to assume that the clusters come from different Gaussian distributions (hereinafter called Gaussians). In other words, it is tried to model the dataset as a mixture of several Gaussians.

EM algorithm is a method to cluster data based on the above ideology. Suppose we have several data points, we can find K Gaussians to describe the sources of each point. However, since the means and variances for these Gaussians are unknown, we will have to extract these values from data points. What we have here is a chicken-egg problem. Therefore, an appropriate initial guess is important: the variance of each Gaussian should be large enough such that all data points can be assigned to different Gaussian appropriately, while the converge time won't be affected too much by such guess. In other words, if the initial guess of variances is too small, many data points will be assigned with extremely small posterior probabilities from certain Gaussian, which yields overflowing of posterior probabilities to be zero in program and thus they won't be considered during calculation.

To show our EM algorithm is feasible, we first implement it on a one-dimensional case. First, we generate data samples from three 1-D Gaussians with different means 0, 3 and 6 and standard deviations 2, 0.5 and 3, respectively. Each Gaussian generates 50 samples. The following image shows the result.

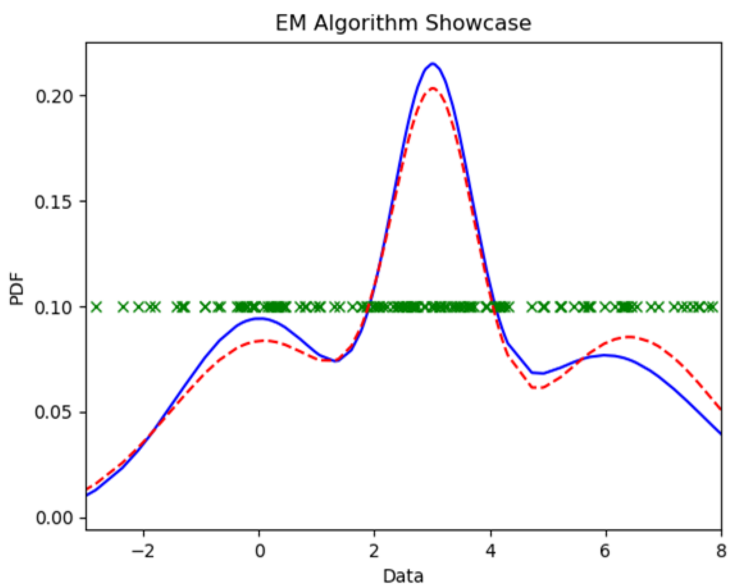


Figure 2: EM Algorithm Showcase.

As shown in the figure, blue line indicates the ground truth and red line indicates the fitted GMM. Green points indicate the truth data points. The following table compares the ground truth means and standard deviations.

Ground Truth Value (Mean, Standard deviation)	Fitting Result (Mean, Standard deviation)
(0, 1.4142135623730951)	(0.08040429209711256, 1.590743198649424)
(3, 0.7071067811865476)	(3.0313214895054763, 0.7373009807211791)
(6, 1.7320508075688772)	(6.41386194327583, 1.5570376447060392)

3. Learning Color Models

Because a 3-Dimensional histogram would be too hard to intuitively understand, we use 1D and 2D histogram to decide how many clusters do we need for a 3-Dimensional Gaussian-Mixture Model.

a. Yellow buoy:

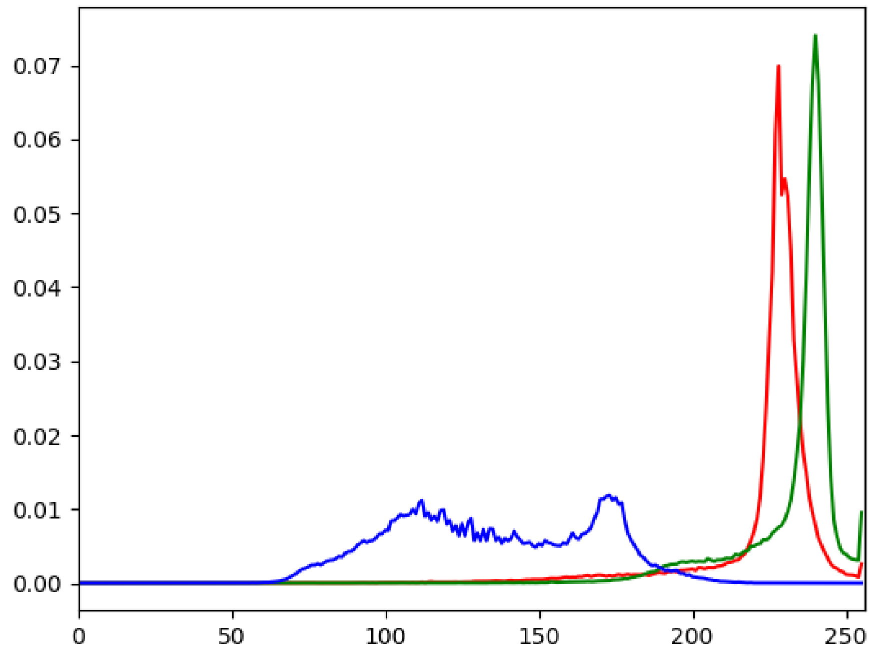


Figure 3: 1-D histogram of the yellow buoy.

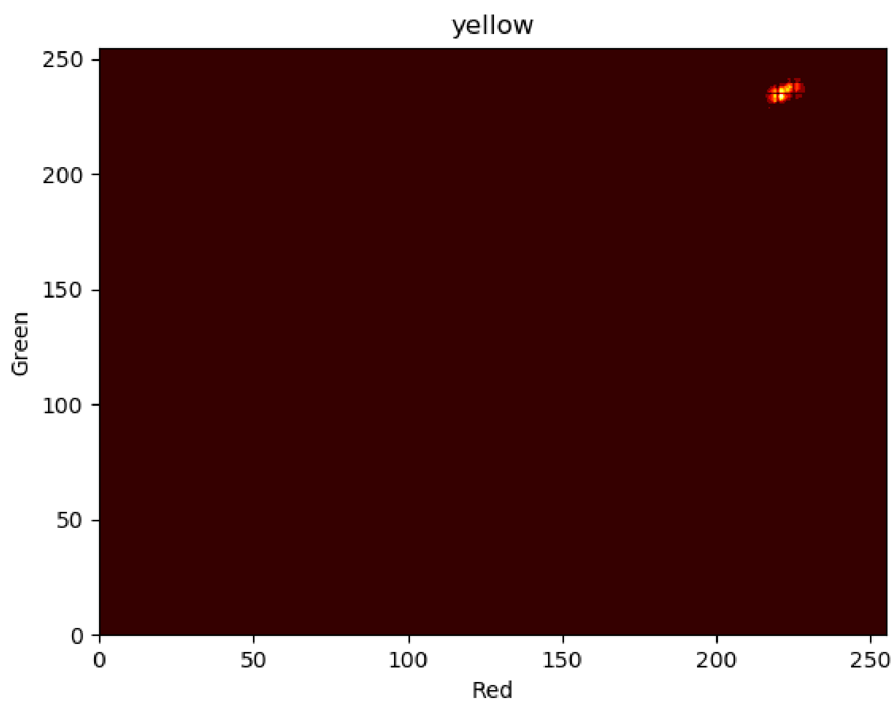


Figure 4: 2-D (red-green) histogram of the yellow buoy.

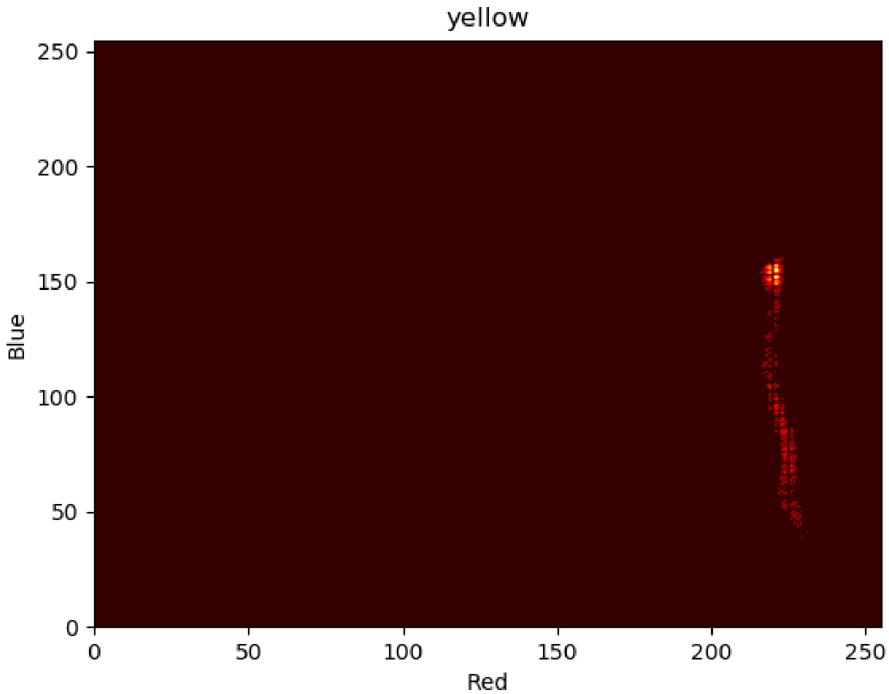


Figure 5: 2-D (red-blue) histogram of the yellow buoy.

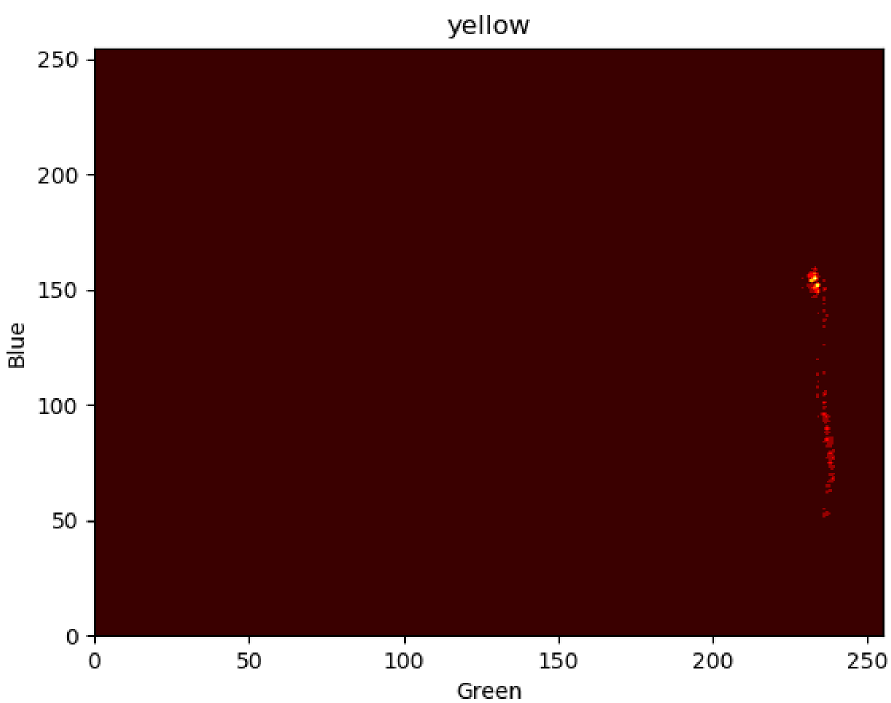


Figure 6: 2-D (green-blue) histogram of the yellow buoy.

Firstly, we can know form these histograms that the pixel values in blue channel is irrelevant compared to other channels. So, we'll neglect its impact to the process of the color segmentation. Secondly, we noticed that the pixel values of green and red channels are concentrated in the 3D space. Therefore, we decided to use one 3-D Gaussian model to model the yellow buoy.

b. Orange buoy:

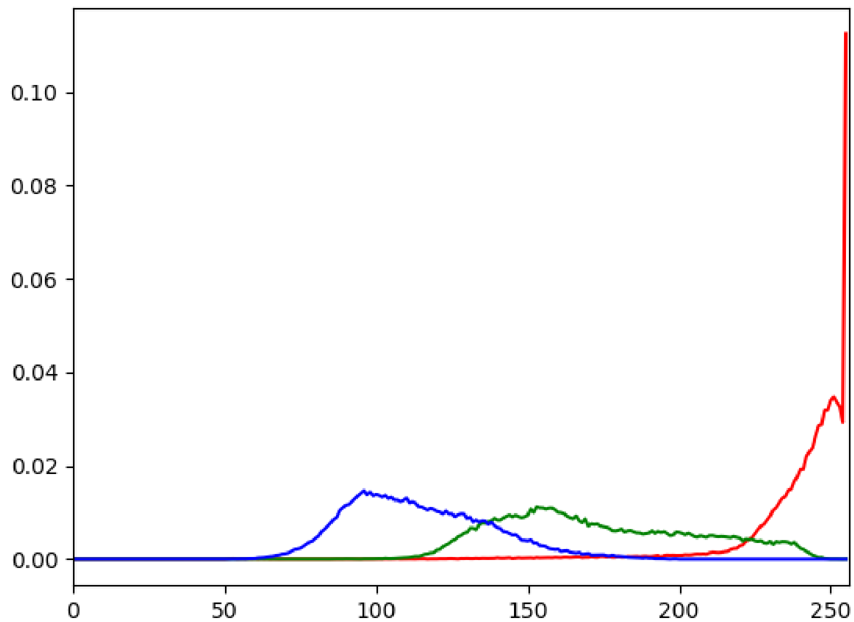


Figure 7: 1-D histogram of the orange buoy.

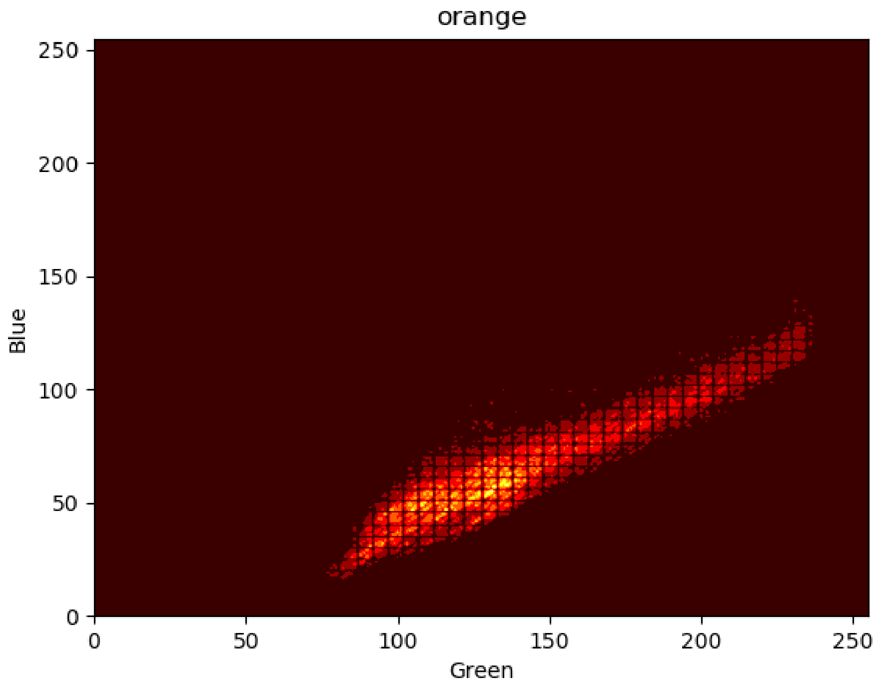


Figure 8: 2-D (green-blue) histogram of the orange buoy.

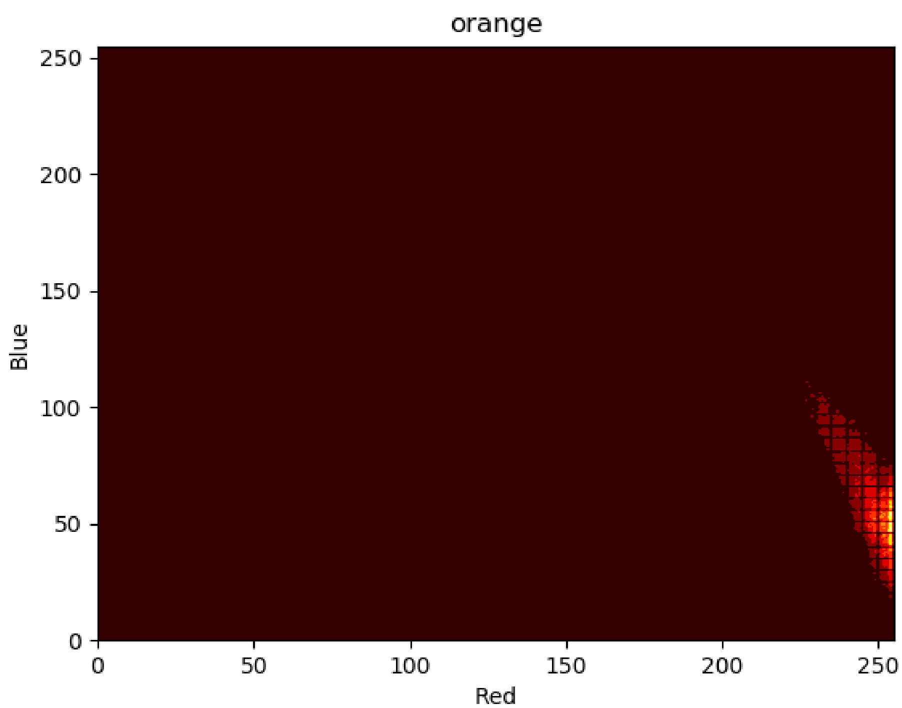


Figure 9: 2-D (red-blue) histogram of the yellow buoy.

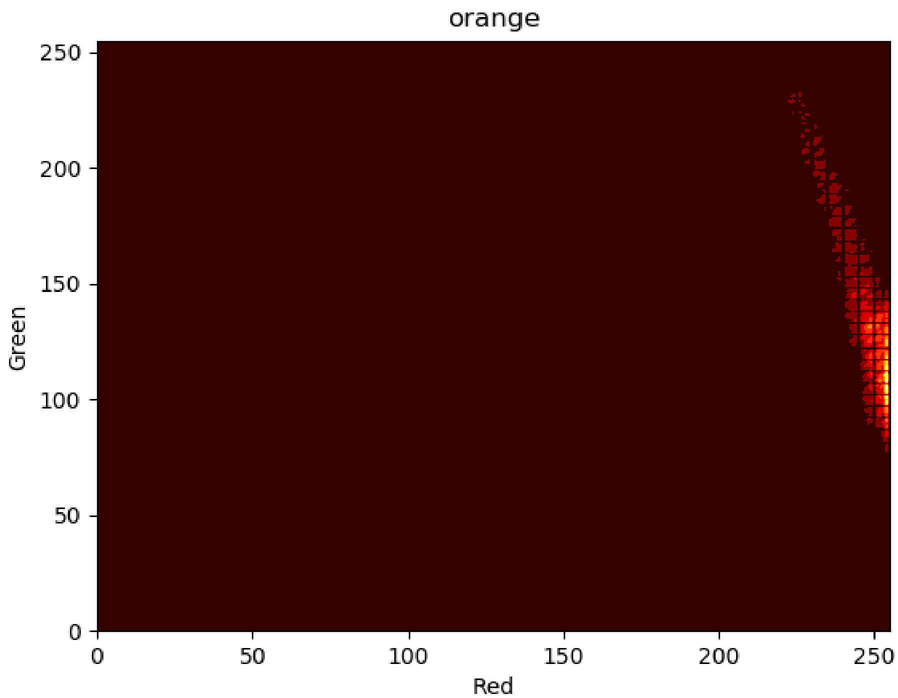


Figure 10: 2-D (red-green) histogram of the yellow buoy.

From these histograms, we know that the red pixel values are significant compare to others. But from 2-D histograms, the correlation between channels are also significant compare to yellow buoy, so we decided not to neglect the effect of green and blue channels. Therefore, we decided to use three 3-D Gaussian Mixture model to model the orange buoy.

c. Green buoy:

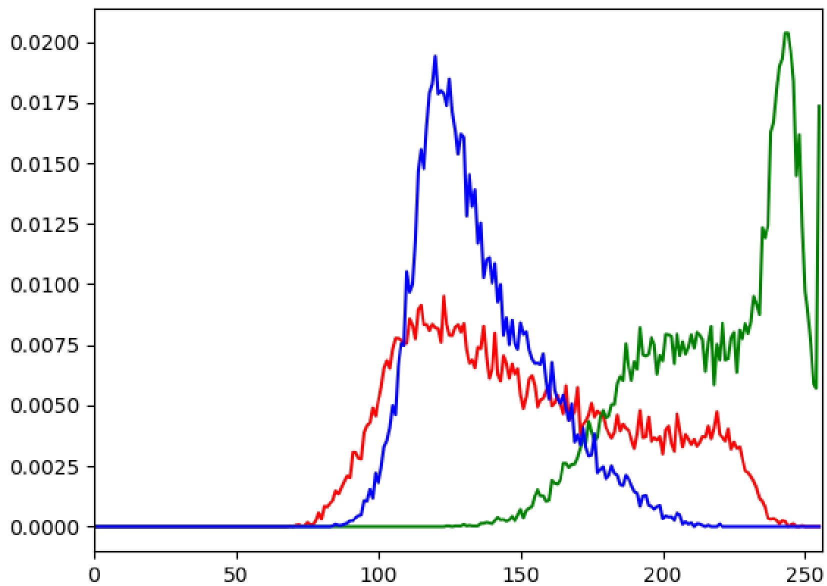


Figure 11: 1-D histogram of the green buoy.

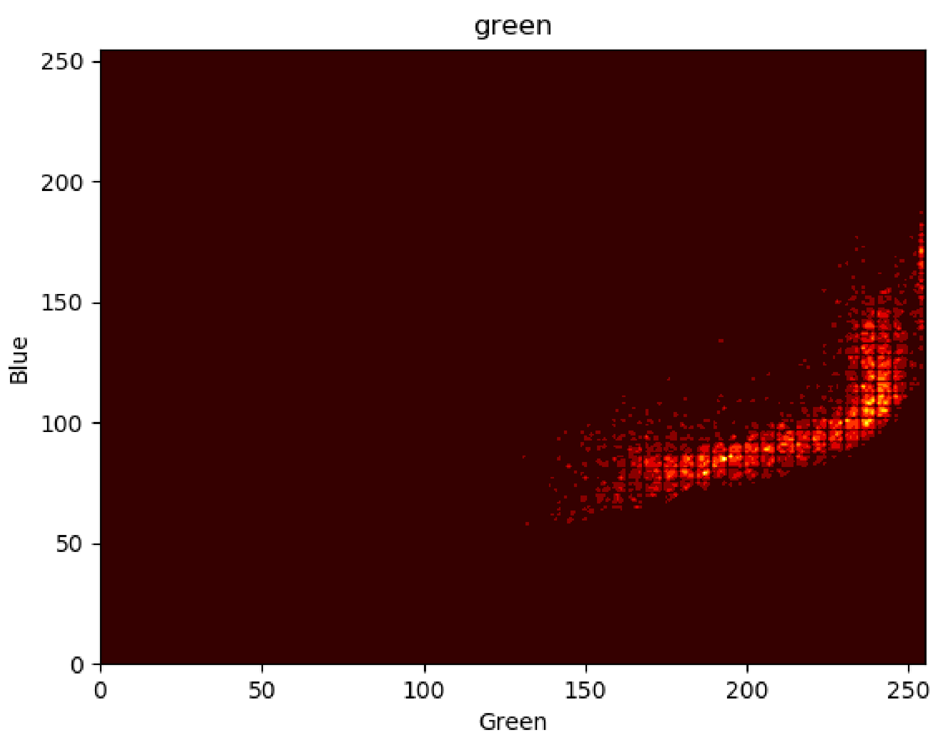


Figure 12: 2-D (green-blue) histogram of the green buoy.

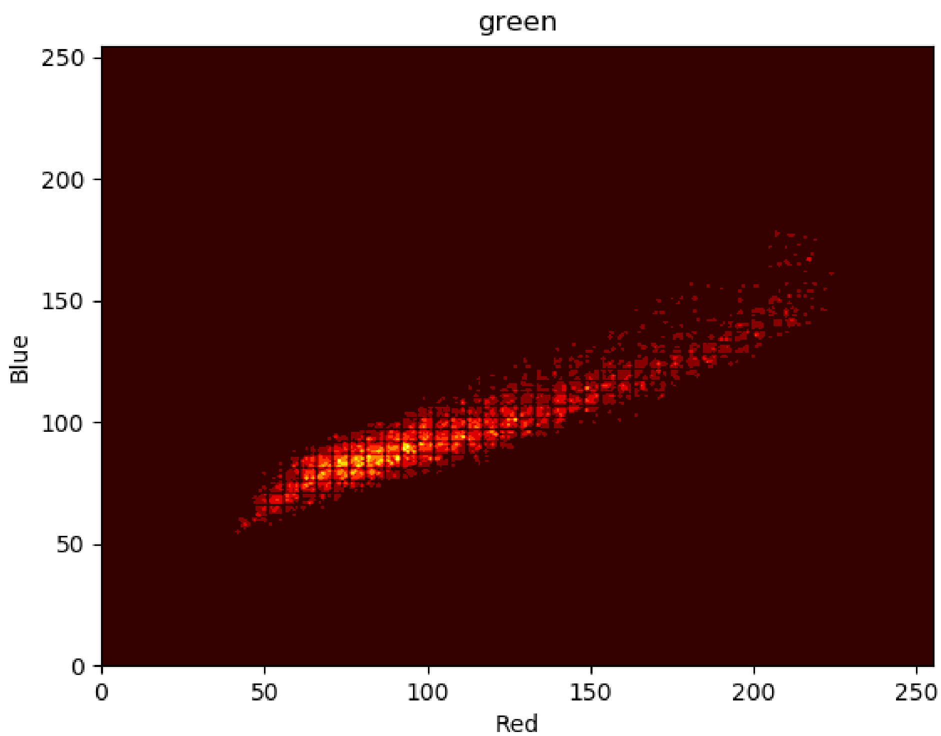


Figure 13: 2-D (red-blue) histogram of the green buoy.

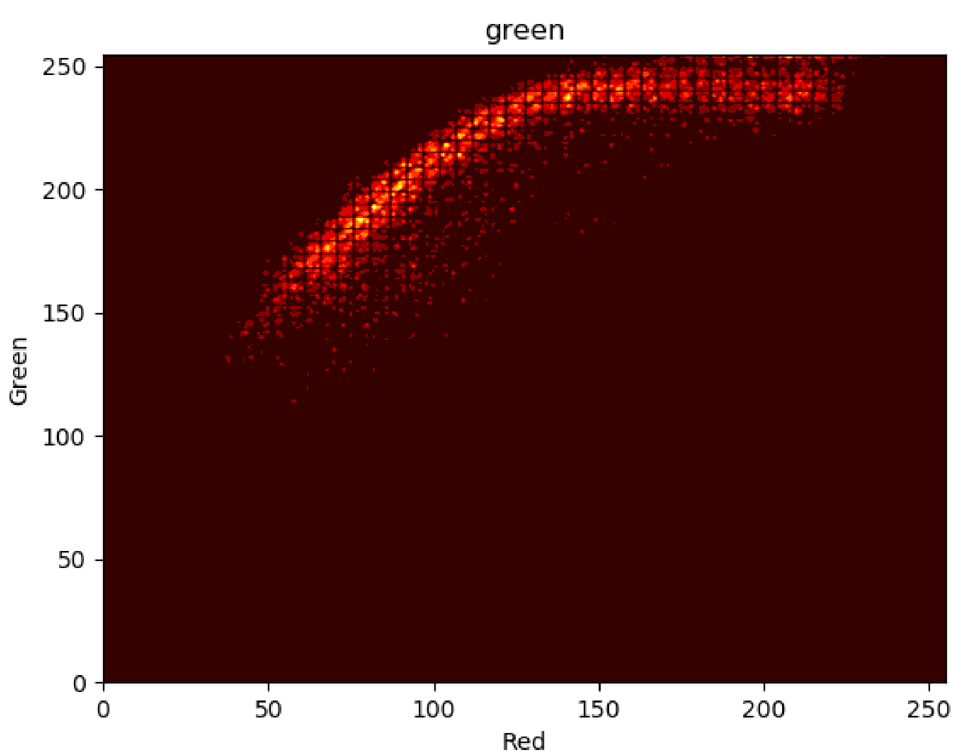


Figure 14: 2-D (red-green) histogram of the green buoy.

From 1-D histogram, it is obvious that at least two clusters (k = 2) should be assigned in GMM. 2-D histogram also implied the same message. When implementing, however, the result of two clusters is not satisfying. On the other hand, by assigning k = 1 (one cluster), the green buoy can be segmented well at the first time. Therefore, in the final version of our implementation, we use one cluster for green buoy training. A possible explanation of the situation of the situation is that the distribution of R, G and B in 3-D histogram can be described by a cluster, while this information cannot be intuitively visualized on 1-D and 2-D decompositions.

After we construct the histogram of the training data, we divided it by the total number of pixels, so the value in the histogram represent the PDF of the image in BGR space. Then, we need to transfer the probabilities into training data. We divided the maximum of the PDF into some ranges, and we sample all the coordinates (in BGR space) of points whose probability is in the range. In this way, we create the training data of each buoy for the Gaussian Mixture Model.

4. Buoy Detection

Two step prediction refinement

We use a tight threshold (probability) to locate the position of the target buoy and predict a bonding box around the buoy. Then, we apply a slack threshold in the bonding rectangle to allow more pixels in our final segmentation. Last, we use the smallest circle fitting to draw the final contour of our detected buoy.

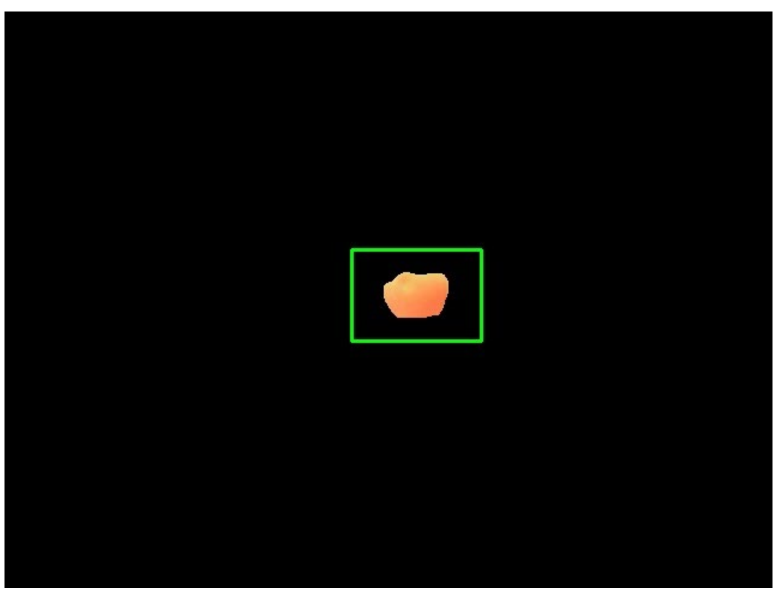


Figure 15: The first segmentation results with stricter threshold and the corresponding bonding box of the first threshold.

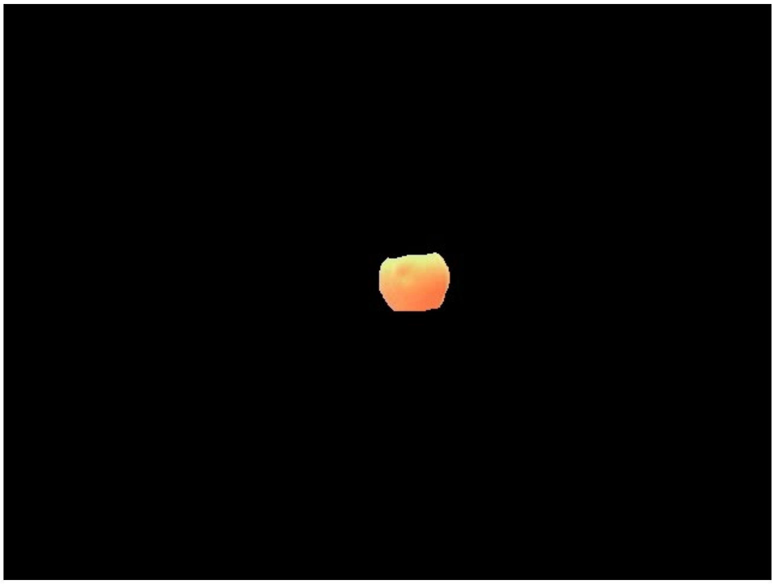


Figure 16: The segmentation results with slacker threshold.

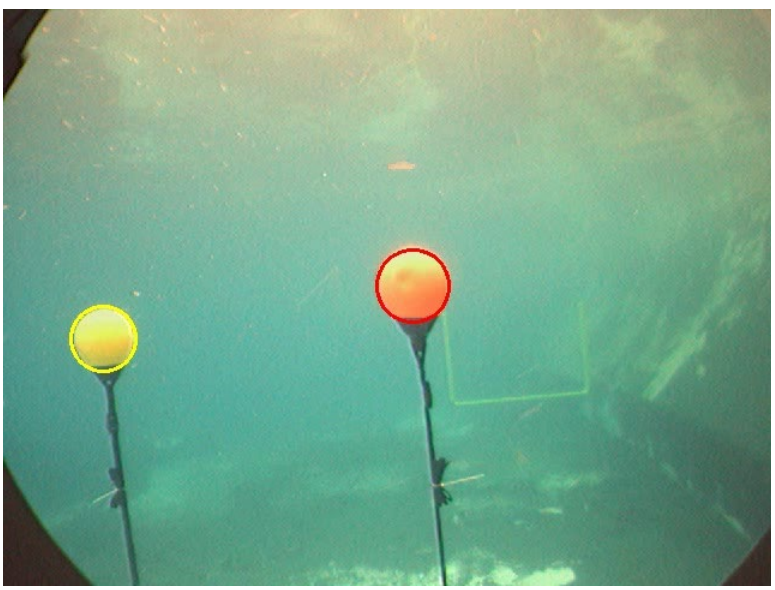
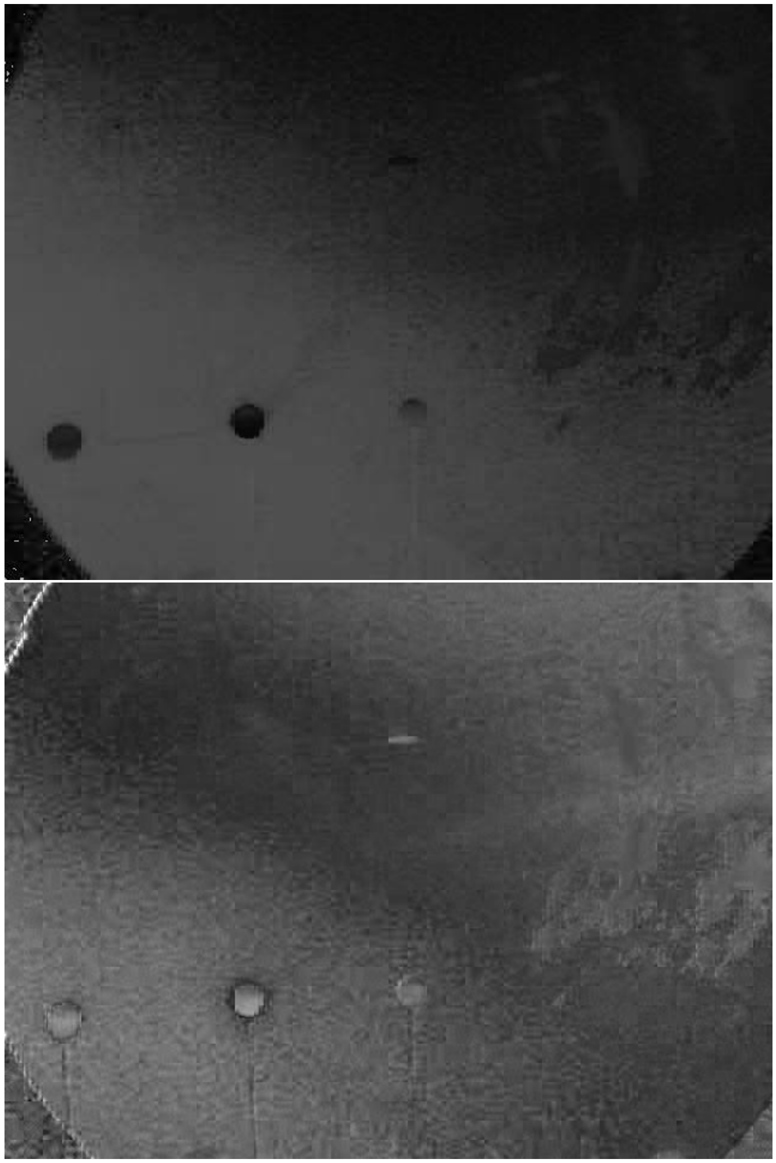
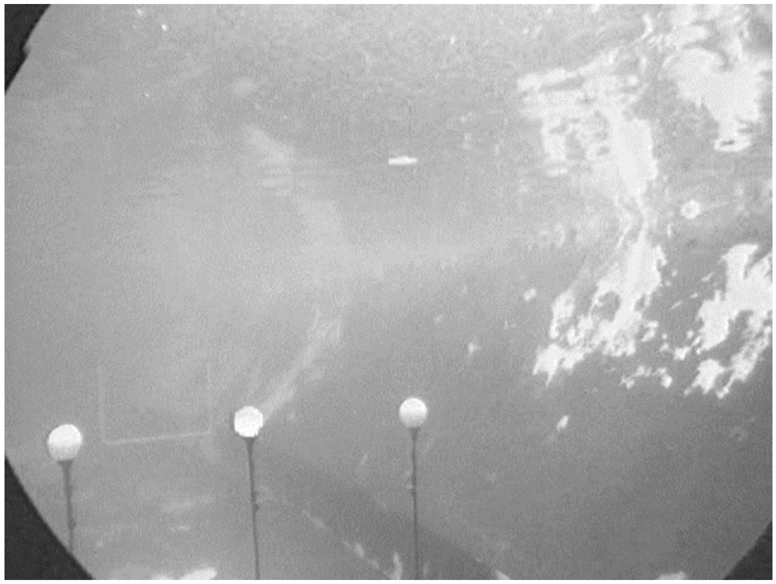


Figure 17: The final results.

5. Further Analysis

In this case, only RGB channels are used for learning. While in applying GMM for color segmentation, using HSV channels is an popular alternative. The following images are H, S and V for a frame from a video, respectively.





As the figure of the hue channel, using color-based system will encounter some issues when the background has similar color with targets, which can be seen on the green buoy (right most). In addition, if the target surface has sever changing on light source, such as underwater surrounding, using HSV channels can better segment the targets because the light effect and the true color of targets can be better separated in HSV.