ENPM673
Project 3
Perception for Autonomous Robots

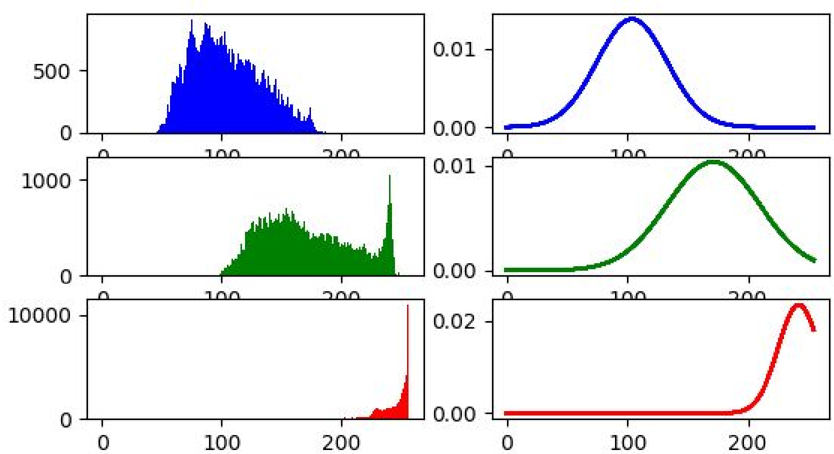
Project Group -

Amrish Baskaran (116301046)
Arpit Maclay (116314992)
Bala Murali Manoghar Sai Sudhakar (116150712)

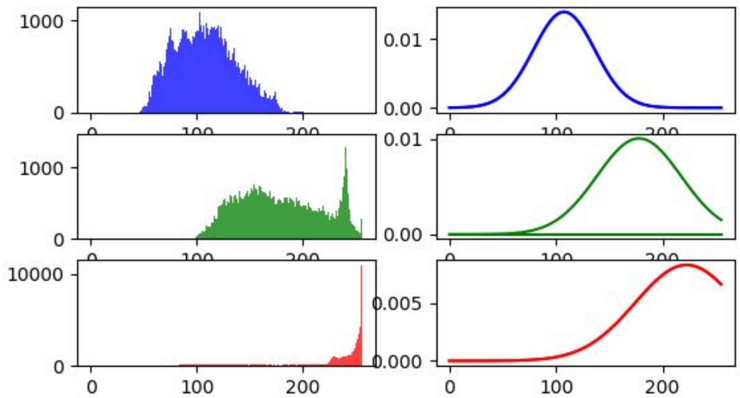
Aim of the project- Obtain a tight segmentation of each buoy for the entire video sequence by applying a tight contour around each buoy.

Data Preparation -

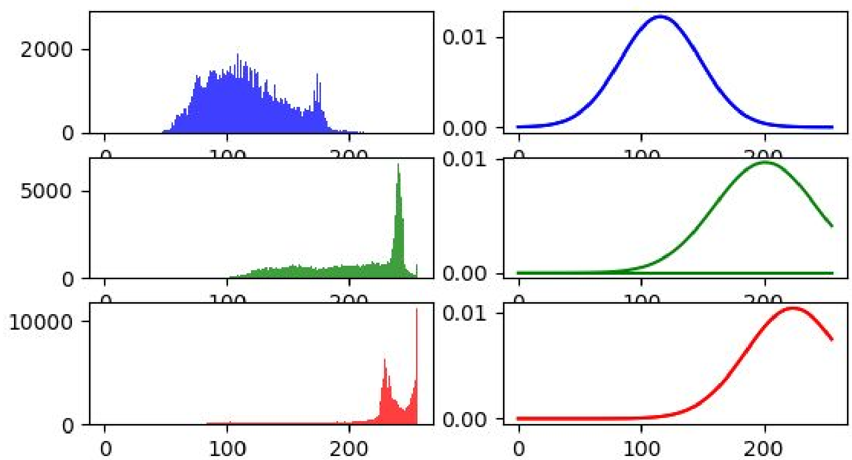
1. Cropping the data from multiple images and compiling into a single image. Cropping has been done manually using paint
   Number of Samples Used:
   Green Buoy - 9 Samples
   Red Buoy - 10 Samples
   Yellow Buoy - 10 Samples
2. Colour Histograms and 1 D Gaussian.
   For each buoy's training images all the blue, green and red channel values are grouped together and the histograms plotted.From the grouped color channel data mean and variance for each channel is calculated using Numpy and plotted as follows.
   a. Red Buoy



b. Green Buoy



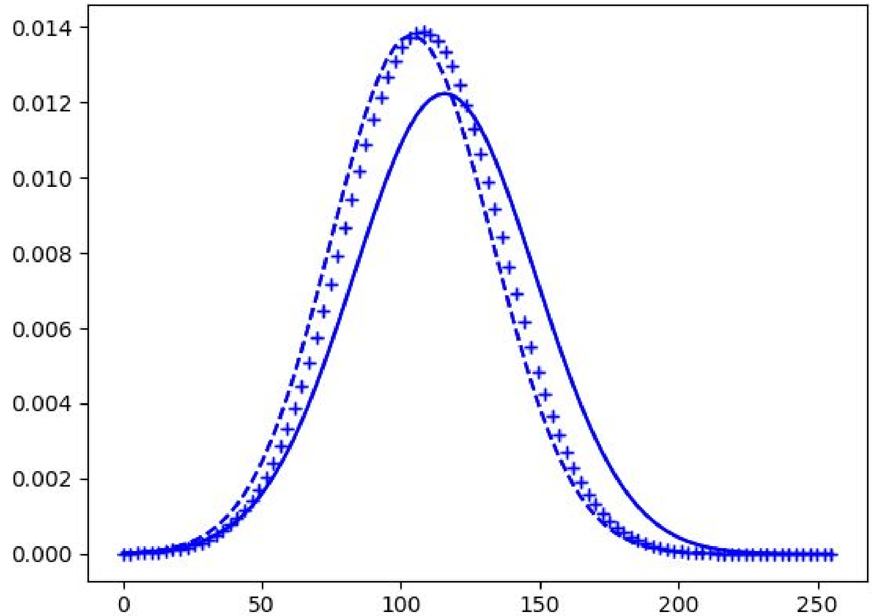
c. Yellow Buoy



3. Comparing 1D Gaussians of corresponding channels of each buoy.

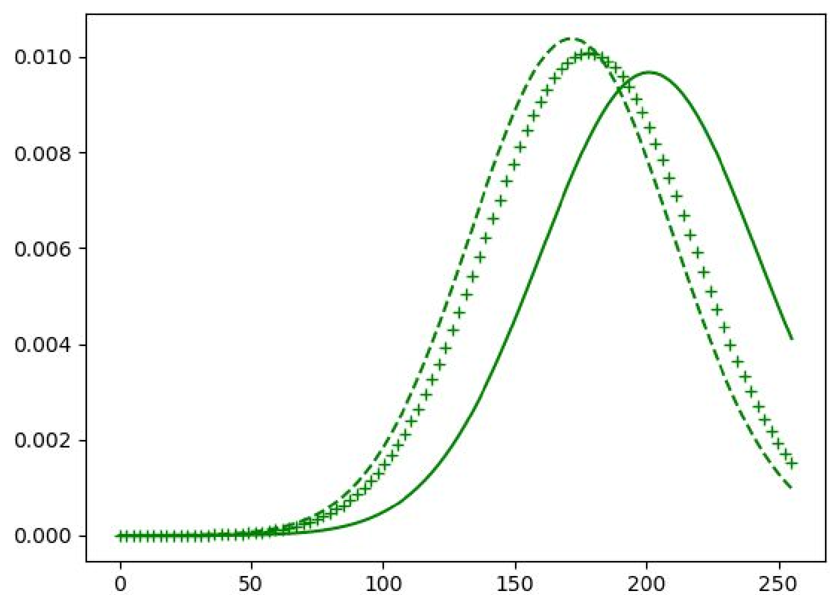
- - - Red buoy
+++Green buoy
----- Yellow buoy

a. Blue channels



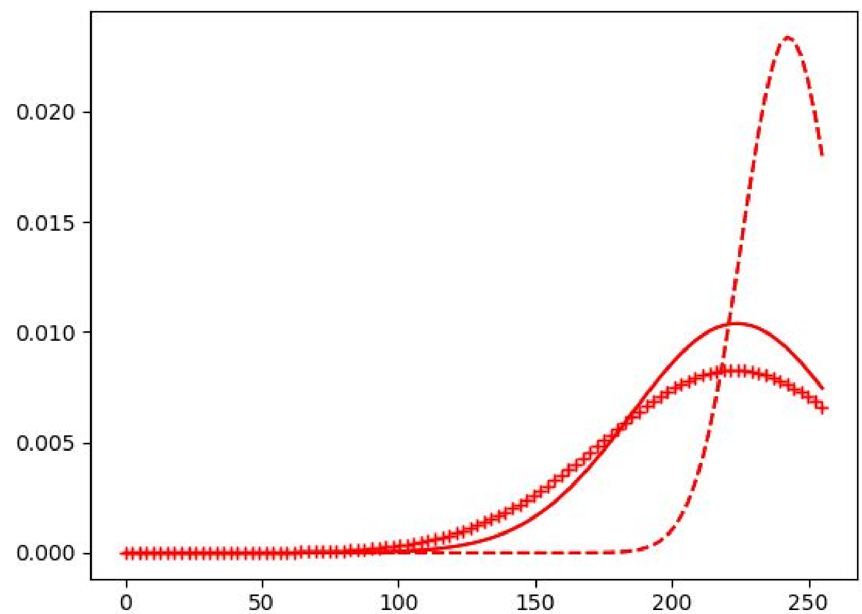
From the blue channel Gaussians, we can see that they do not have a significant difference for the same event. Hence using the blue channel to calculate the probability of belonging to the buoys is not feasible.

b. Green channels



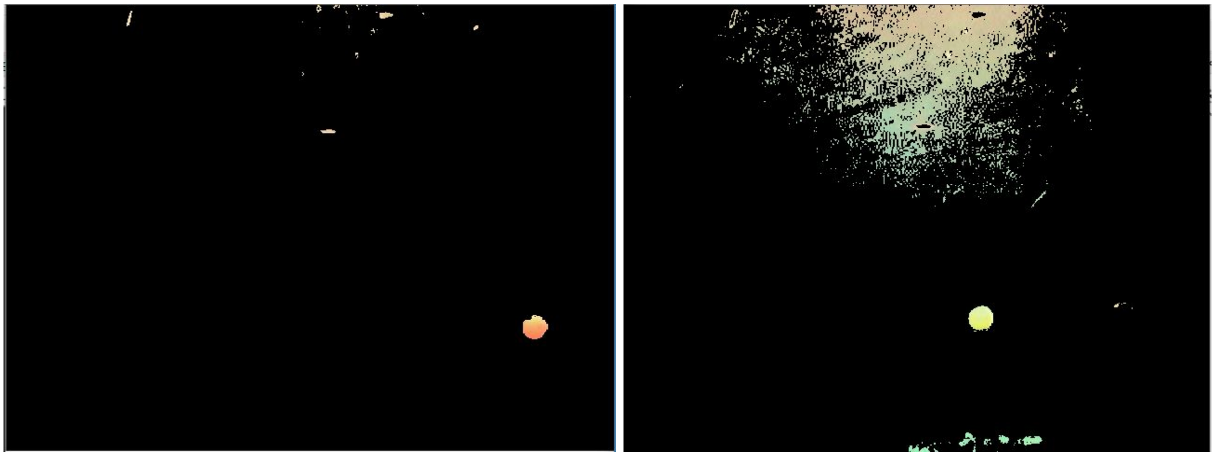
Similarly, for the green channel Gaussians, we can see that there is enough variation but may not be significant enough to distinguish between buoys.

c. Red channels

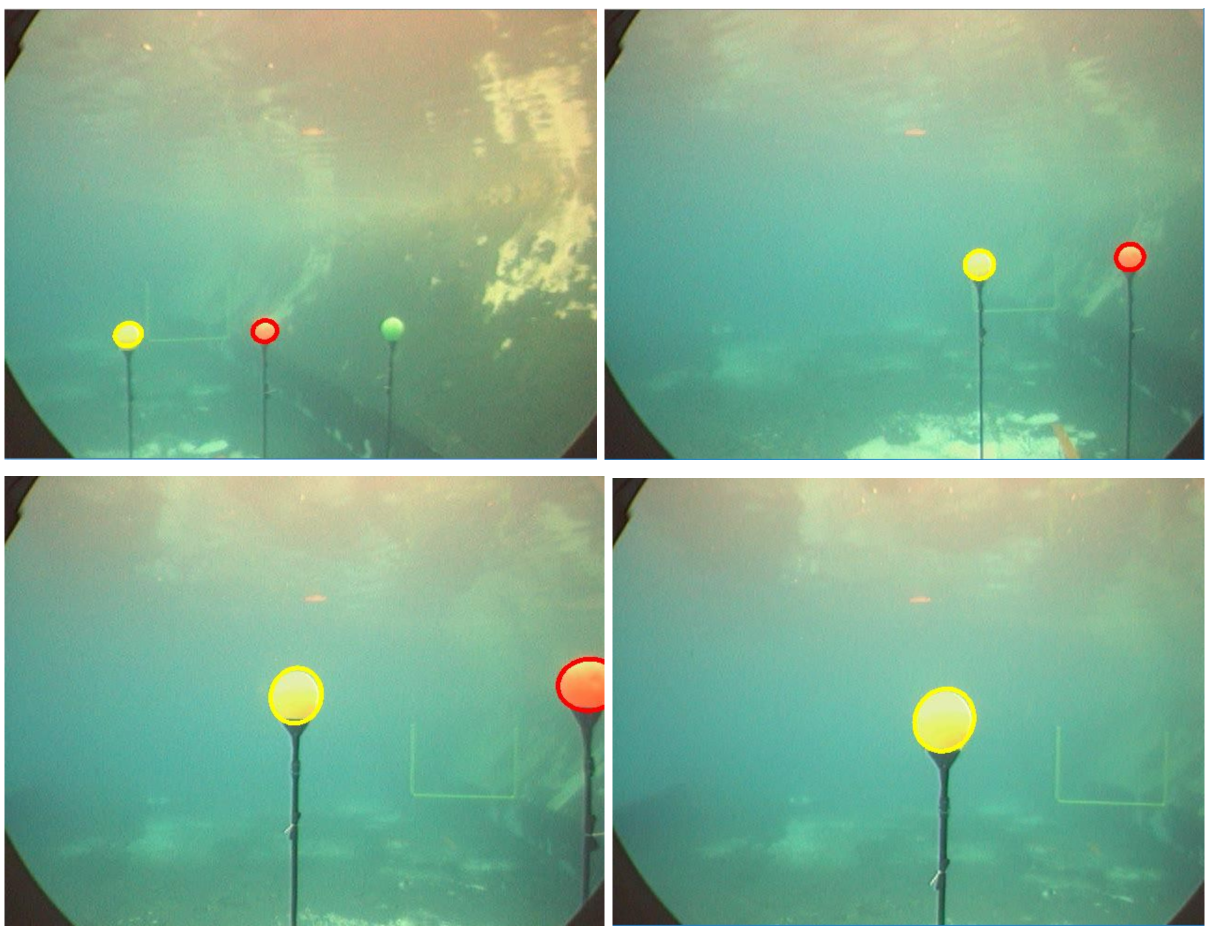


For the red channel, there is enough variation to distinguish between the red buoy and the green or yellow buoy.

4. Segmenting the buoys using 1D Gaussian
    a. Every pixel is checked with the previously calculated Gaussian distributions to decide which buoy that pixel belongs to.
    b. The stats.norm.pdf function is used to get the probability of the pixel channel value belonging to that group.
    c. Each pixel's, each color channel's probability is calculated using the mean and standard deviations obtained from the color histograms as above.
    d. For Red buoy, the total probability is taken as the product of red channel and green channel probability of that pixel.
    e. For Green buoy the total probability is taken as the product of blue channel and green channel probability of that pixel.
    f. For Yellow buoy, the total probability is taken as the product of red channel and green channel probability of that pixel.
    g. Now for each pixel, a threshold of 0.00001 is applied over its 3 buoy probabilities.
    h. And the buoy with the highest probability will have this pixel position added to its mask as white.
    i. The masks obtained for the red and yellow buoy and applied to the original image are as follows. The green bouy could not be segmented with just the 1D Gaussian as there is not much variation in the Gaussians of the color channels.



j. To display the segmentation these masks are used to fit ellipses using cv2.fitellipse. Depending on the area the noise and small artefacts are rejected.



To increase the accuracy of the segmentation we have also included another group over the buoys for the water/ tank on the right which has colors close to that of the red and yellow buoys. Now the pixels are distributed over 4 masks.

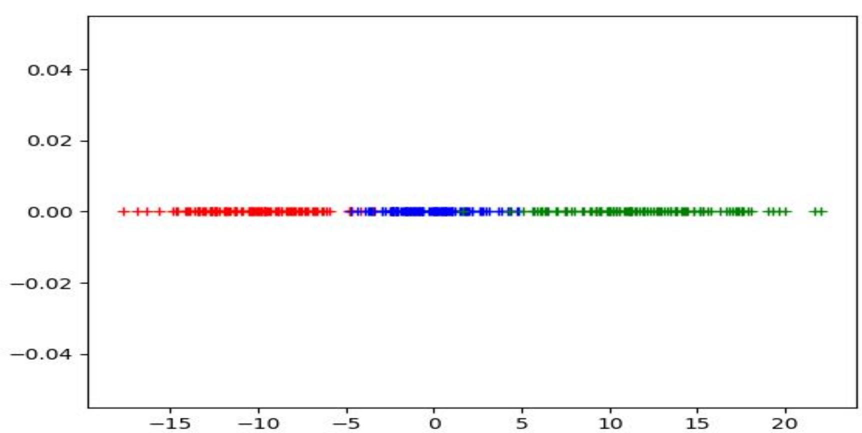
Expectation Maximization with Gaussian Mixture Models

To model one dimensional mixture data with a Gaussian Mixture Model-

Since there exists some complicated data where one gaussian is not enough to fit the data distribution with something that only represents a single peak to model the distribution correctly. A mixture of Gaussians is necessary for representing such data.
For a linear combination of Gaussian, the Gaussian function is modified as -

$$p(x|\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i \in [0,k)} \pi_i N(x, \mu_i, \sigma_i)$$

Here, πi, μi and Σi respectively define the scaling factor, mean and covariance of the kth Gaussian.
After modifying the function, we have elements like π which isn't observable from the data, this is where we introduce Expectation Maximization.
Step1- **Sample Data Preparation**
For the sample data set preparation, we first form three normalized clusters with 100 randoms points each with -10, 0, 12 and standard deviation: 3, 2, 4 respectively. Next, we combine them to one axis to get the following data set.



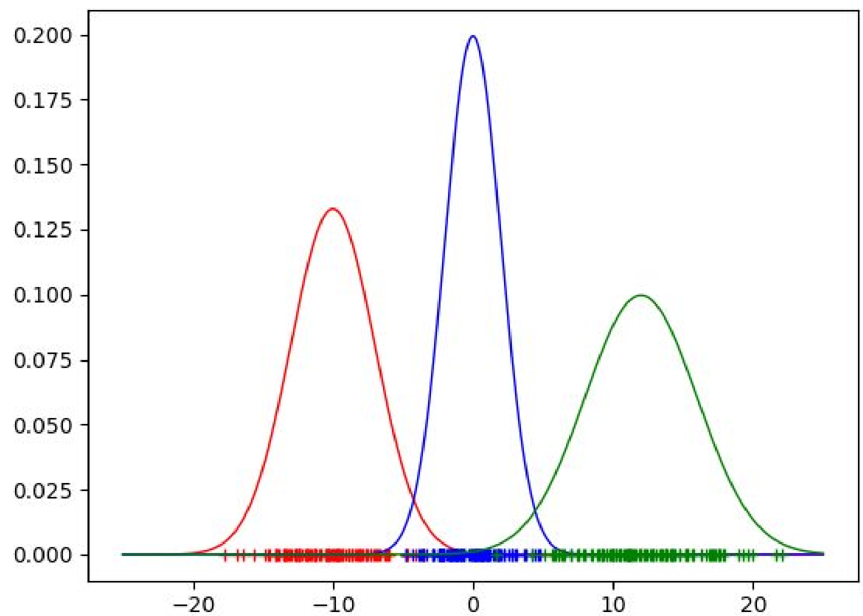
As per the given name, the expectation-maximization algorithm is divided into 2 parts- Expectation part and Maximization part-

1.) <u>The Expectation step-</u>

    Using the current mean and standard deviation guess, we calculate probabilities. We calculate these values for each Gaussian. This helps us predict which Gaussian is responsible for which datapoint

$$\alpha_{i,j} = \frac{\pi_i p(x_j|C_i)}{\sum_{i=1}^{k} \pi_i p(x_j|C_i)}$$

j is the data point index, i is the cluster index.



2.) <u>The Maximization Step -</u>
Now to evaluate the best parameters to best fit the points.
For the new mean for every curve c, we have

$$\mu_c^{new} = \frac{1}{N_c} \sum_i r_{ic} x_i$$

The denominator Nc takes into account this as well.

$$N_c = \sum_i r_{ic}$$

For the standard deviation -

$$\sigma_c^{new} = \frac{1}{N_c} \sum_i r_{ic}(x_i - \mu_c^{new})^2$$

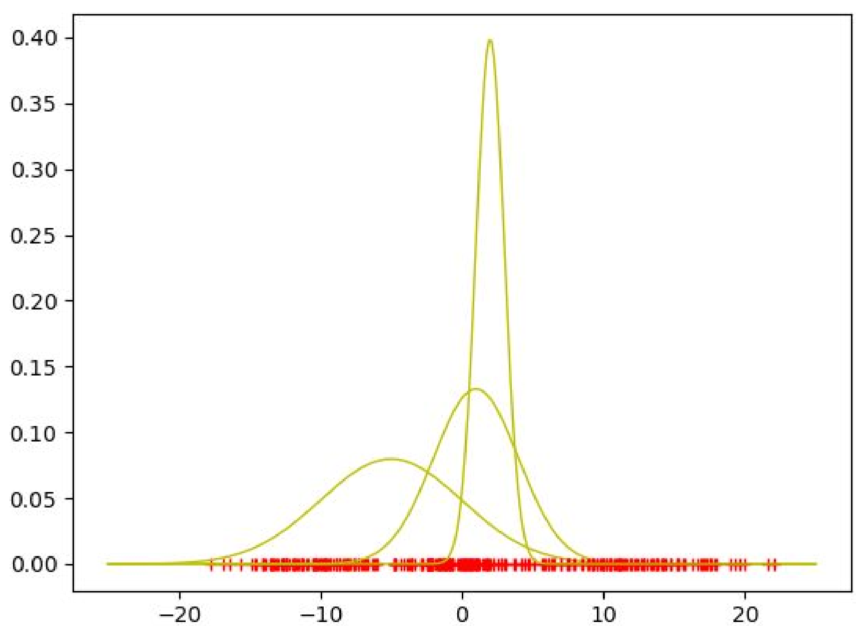
Similarly, we can calculate the new values of π as well-

$$\pi_c = \frac{N_c}{n}$$

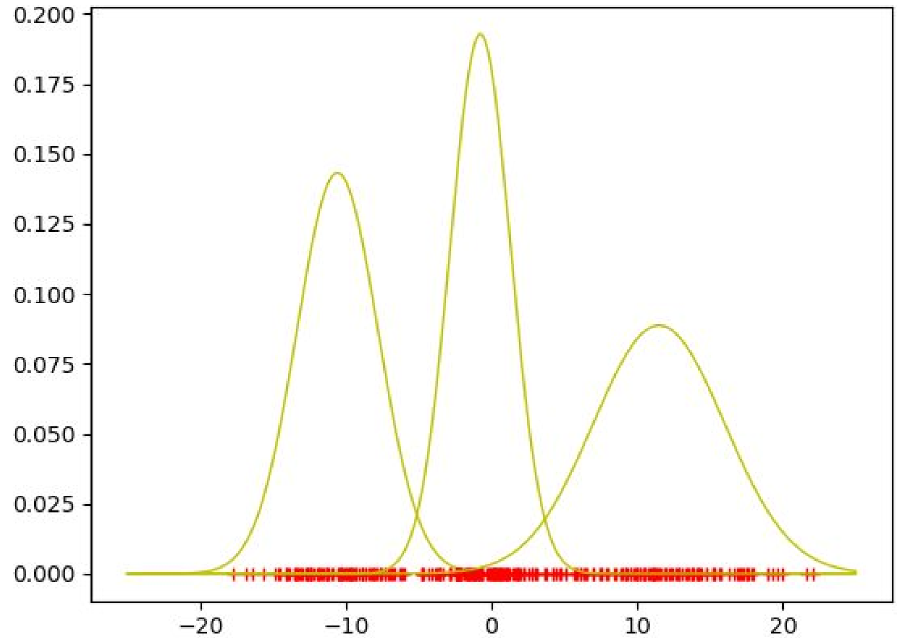
Here, n is the total number of data points in the dataset.

1.)Initialization of EM

This is the first iteration of the Gaussian fitting into the probability curves in which the Gaussians have not gone under maximization. As the iterations will go on the weights will attract the Gaussians to get the optimum mean and variance.



1. Results of EM



This is the final result of the expectation maximization or the last iteration having the optimum mean and variance guessed by the EM algorithm.

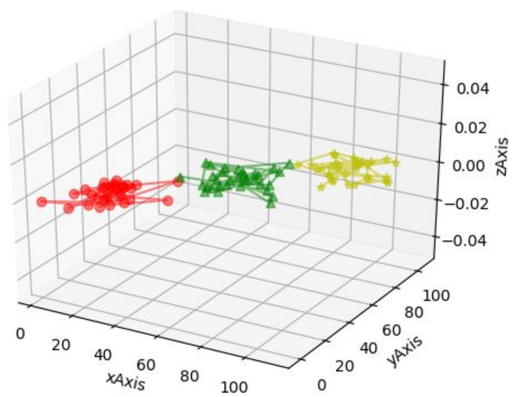
2. Mean values: -10.56650916, 11.49726333, -0.75946281
   Standard Deviation: 2.78570546, 4.49667308, 2.06911951

Expanding to 2D GMM

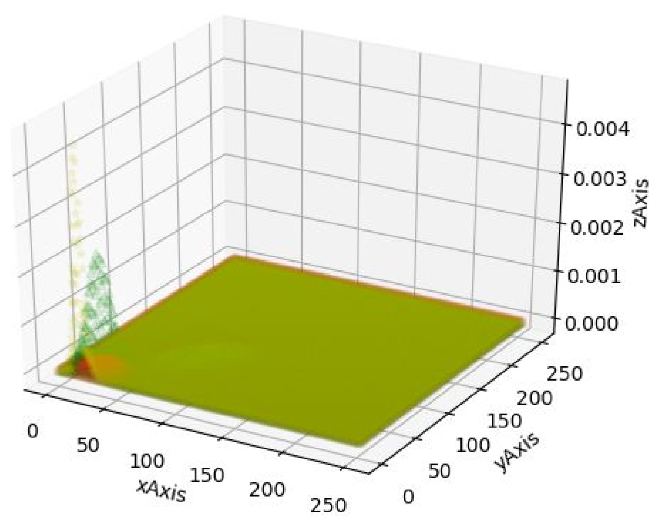
The image is a 3 Dimensional data in Red Green and Blue dimensions for which Gaussian has to be fit. Before going to 3 Dimensions, we test the EM algorithm with 2D data.

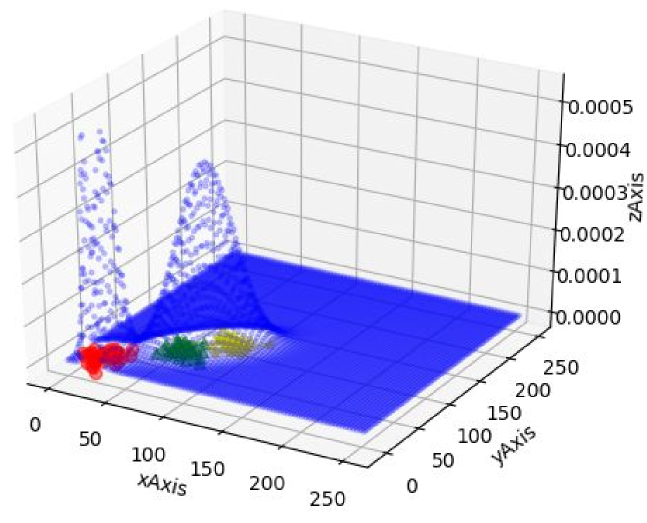
1. Sample Creation:
   Three random groups are created
   Mean = 20 ; Variance = 10
   Mean = 60 ; Variance = 10

Mean = 90 ; Variance = 10

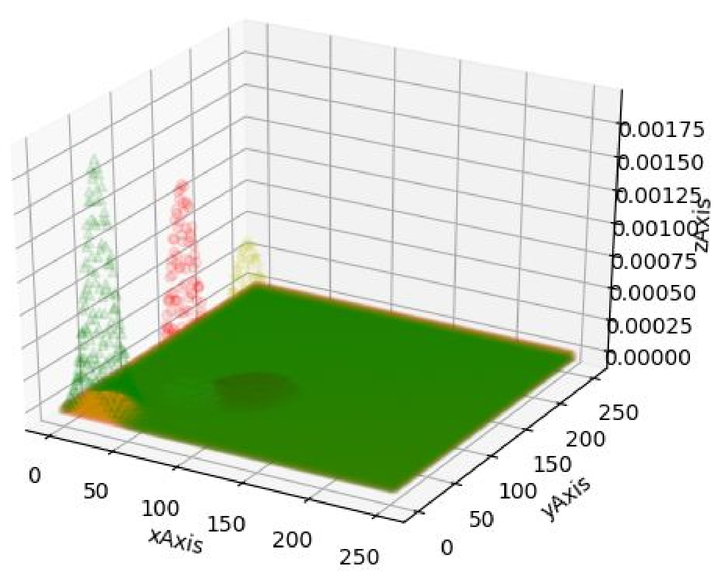


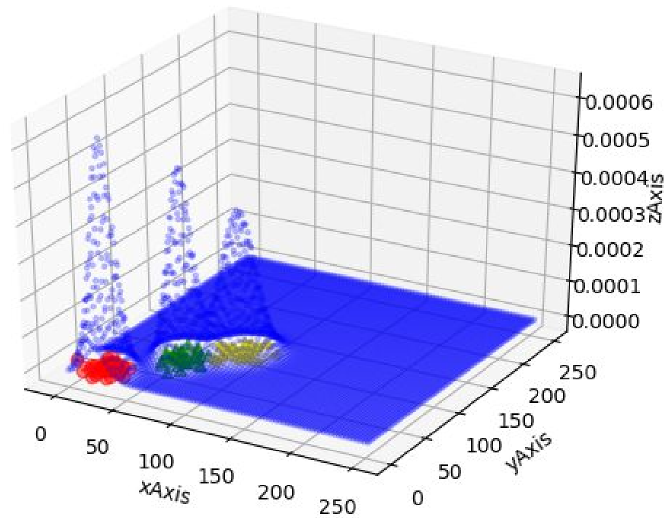
2. Training
   The convergence of mean and co-variance to real mean and co-variance is highly dependent on the seed values chosen. Currently, the initial values are chosen at random and hence the convergence might not be guaranteed.
   a. Trial 1
      The EM algorithm did not converge to expected mean and covariance values. The three Gaussians are as shown below





b. Trial 2
   In this Trial, the EM algorithm converged properly
   The three Gaussians are as shown below





The converged values arein range of expected numbers
Mean = [[59.25475349, 61.10332654], [20.66217873, 18.51801867], [91.19490224, 89.64325455]]
Co-variance = [[302.01488343, 195.35658639], [195.35658639, 324.79111761]]), array([[68.83965984, -1.50376434], [-1.50376434, 69.22888323]]), array([[70.26426383, -74.62285696],[-74.62285696, 96.07260495]]

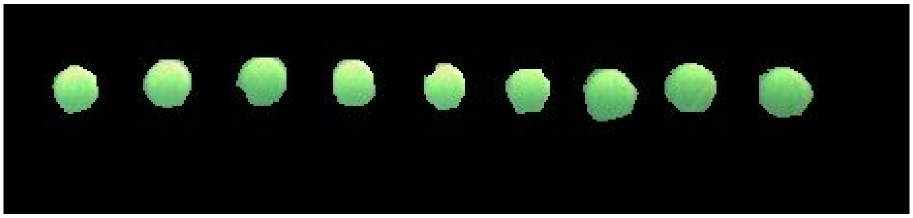
3. Initializing Mean and Co-Variance
   Since there the convergence is completely dependent on initial values, initializing the seed values randomly will not provide with the best results. So educated guess has to be taken for the initial values. To do so we first plot the data and see the clusters. Then seed values for Mean can be chosen such that at least one means is present inside the data cluster. For choosing seed values for co-variance, it is enough that it is not chosen too narrow or too wide.

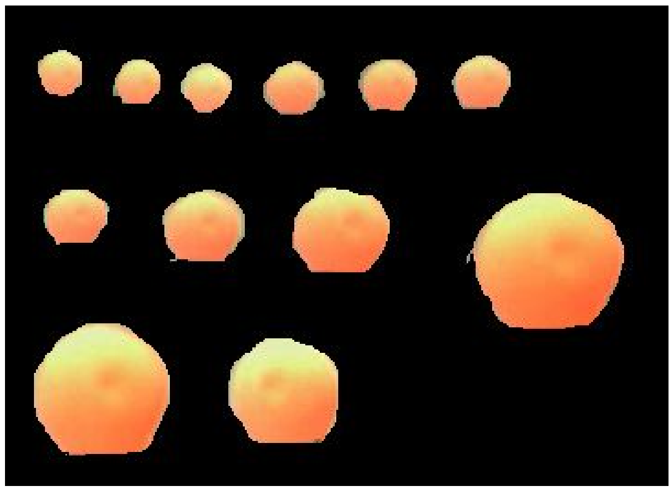
Fitting 3D Gaussian to detect Buoy

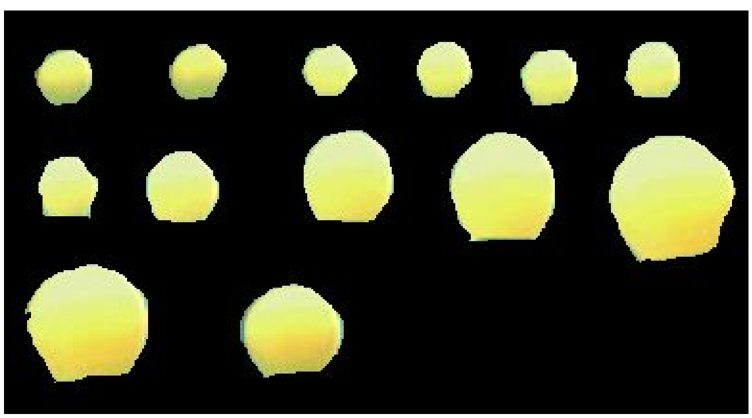
Once the EM working is ensured, it can be trained on the actual data set. The images are taken in RGB colour space. There is a possibility to go to a higher dimension to add more parameters such as shape, size etc along with the colour pixels but it takes a lot of time to generate tagged data. When such a data is created, we can use the current EM algorithm since it can handle N-dimensional gaussian and K clusters.

Data Preparation Step:

The requirement is not a lot of data but a good variety of data. For this purpose, 10 different frames which are far away and give a good variety of data are taken for training. From these frames, Red Buoy, Green Buoy and Yellow Buoy images are separated and used for training.

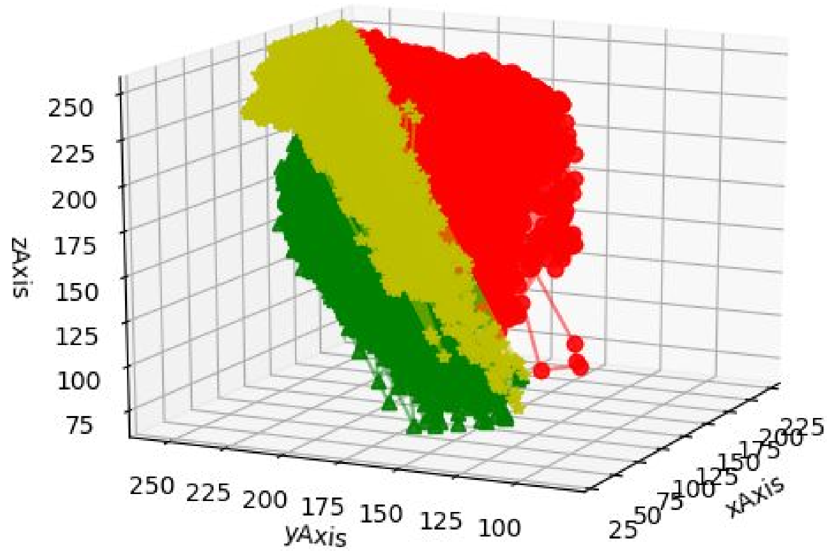






PIPELINE - 1 : Fitting 3 clusters, one for each category using 3D Gaussian (Red Buoy, Green Buoy, Yellow Buoy)

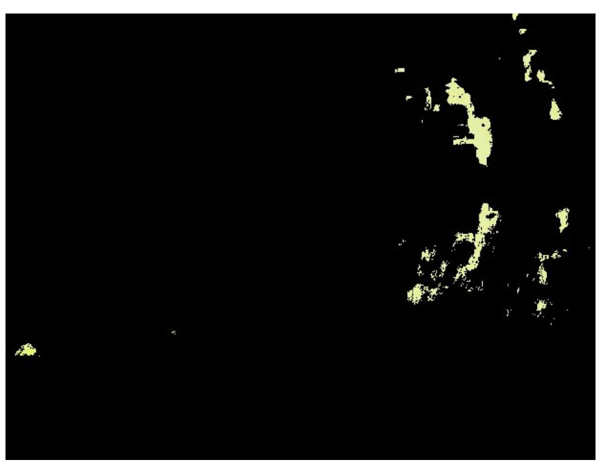
Data Plotting:



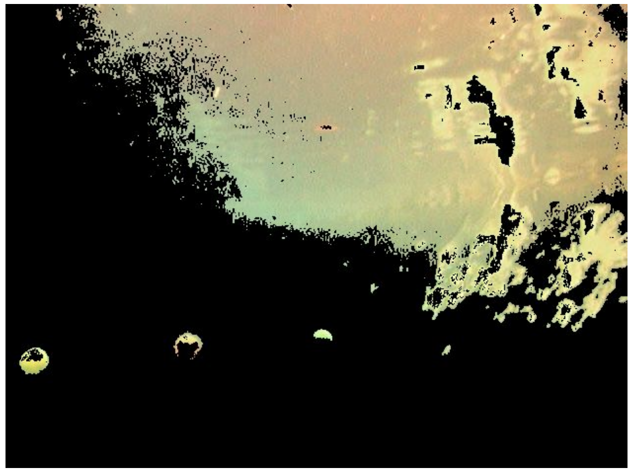
The three clusters represent green, yellow and red buoys. We can fit 3 Gaussians to categorize into 3 groups using the EM algorithm.

The prediction was based on softmax on these Gaussions and the below result was obtained

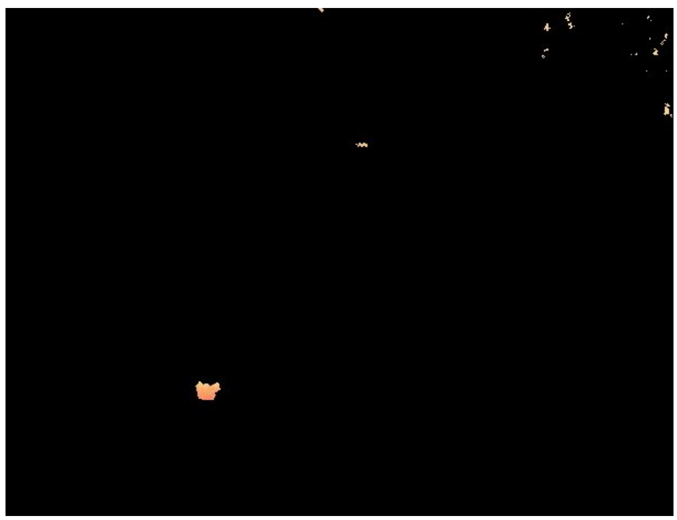
Results



Yellow Buoy Detection



Green Buoy Detection



Red Buoy Detection

Inference

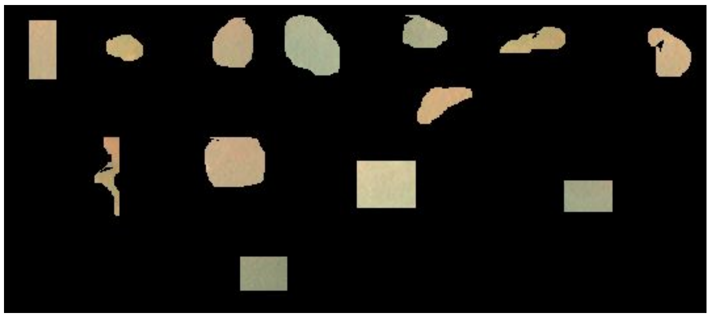
- The results for Red buoy was acceptable.
- The large portion of yellow buoy was detected in green buoy prediction
- The yellow and red buoy prediction were rejecting water area but green category included a large portion of water also
- Since water is also predicted along with color segmentation, we have to use other morphological techniques to reject water and circle buoys alone

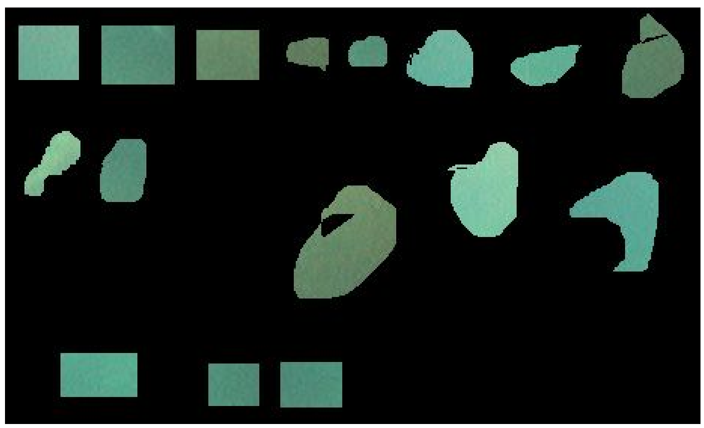
Improvements

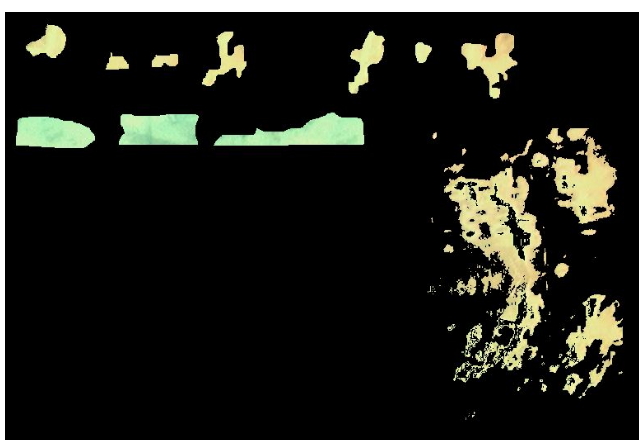
Instead of fitting into only 3 categories, we can include water also as a category and prediction can be based on all 4 categories.

PIPELINE - 2 : Fitting 4 clusters, one for each category using 3D Gaussian (Red Buoy, Green Buoy, Yellow Buoy, Water)

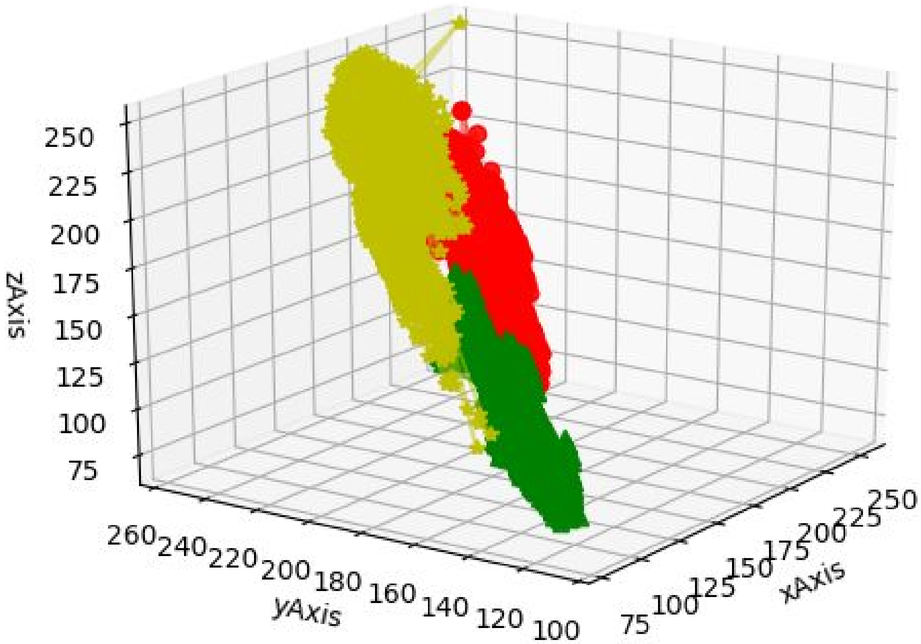
Training data set





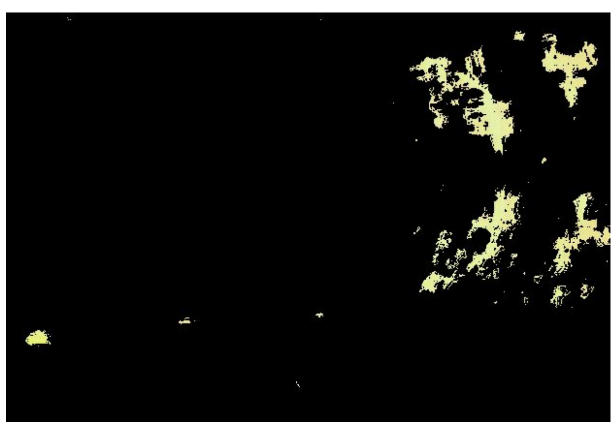


Data Plotting

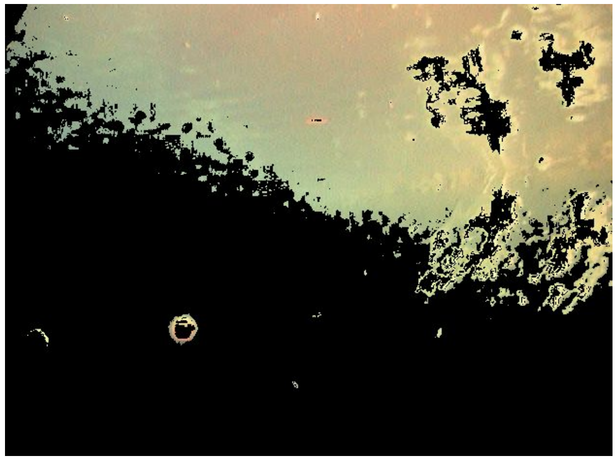


The above data represents the water data collected from video frames. Form the data it can be found that it will be better to fit 3 Gaussians only for water category for a better prediction.

Results Water Prediction

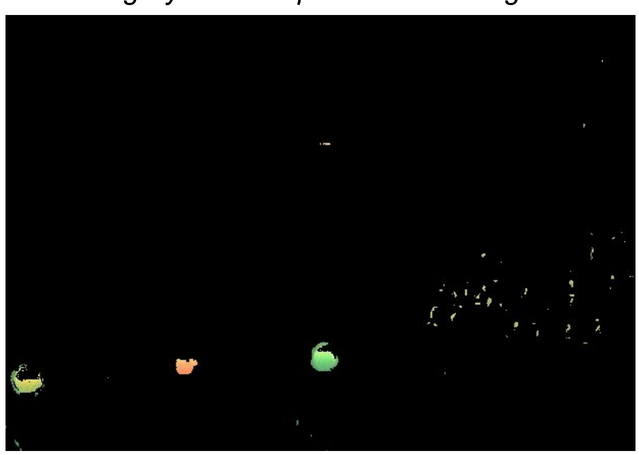


Water category 1 - white portions of background tank



Water category 2 - Red portions of background tank

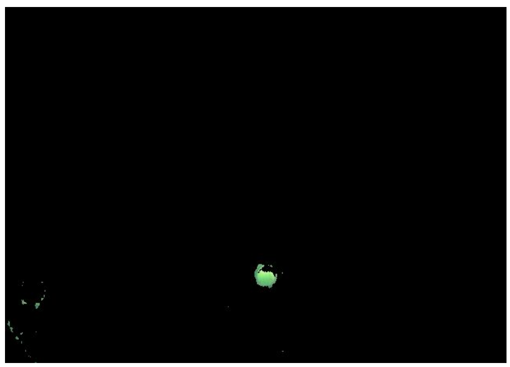
Water category 3 - Blue portions of background tank



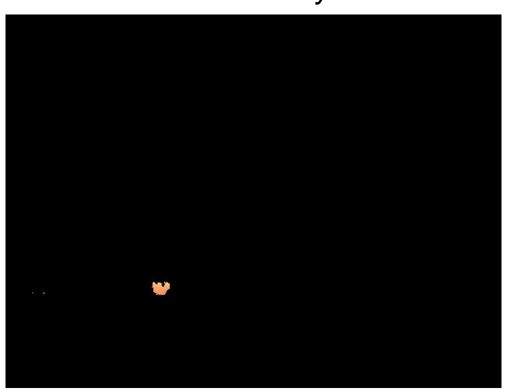
Water Removed

Results Buoy Prediction

Yellow Buoy



Green Buoy



Red Buoy

Inference

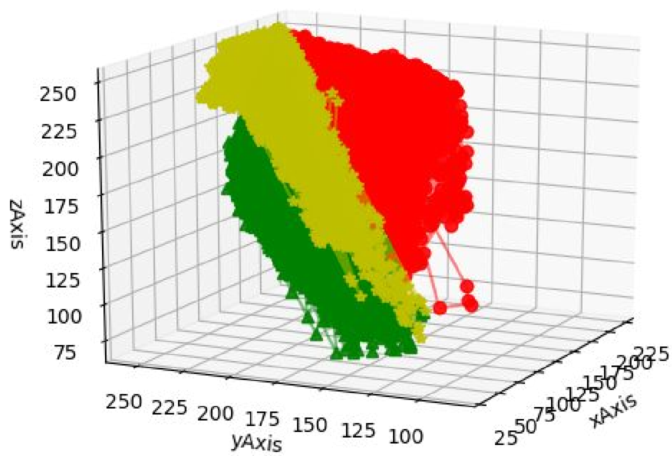
- The predictions are clear and the background is removed properly in the prediction of the buoy
- There are very few imperfections which can be removed using simple contour rejection techniques
- The green buoy was clearly separable from the other categories which is a good improvement

Improvements

Small portions for red and yellow are removed during water rejection. From the plotted data we can infer that if single Gaussian is used to fit each cluster there might be good overlap and there can be outliers. It would be better to fit the clusters tightly with multiple Gaussians for a single category. But this might lead to overfitting problem.

PIPELINE - 3 : Fitting 3 clusters each category using 3D Gaussian (Red Buoy, Green Buoy, Yellow Buoy, Water)

Data Plotting:



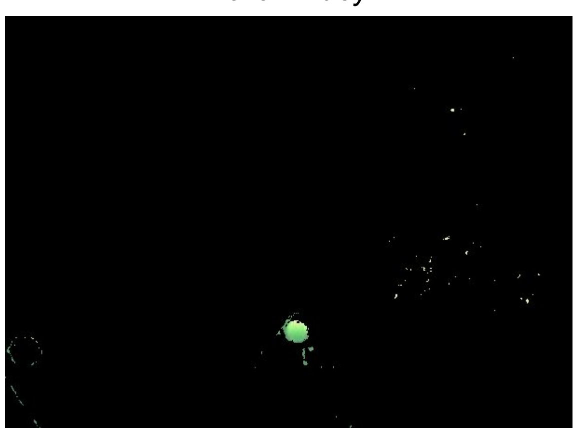
From the above plot, we can see that fitting a single Gaussian for each category will result in a good number of overlap. A tighter fit can be obtained when multiple Gaussians are fitted for a single category.

Result Removing Water

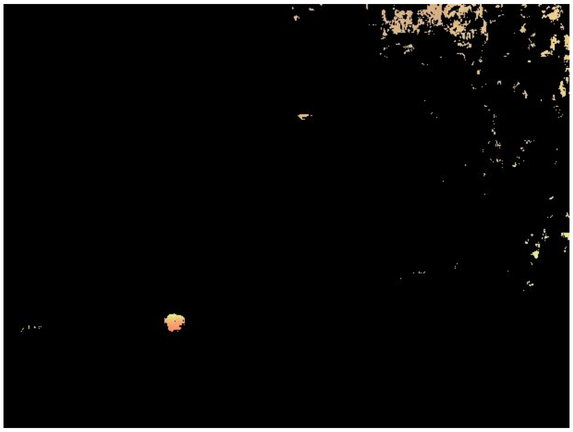
Water that is nearly same as yellow and has red tints

Result

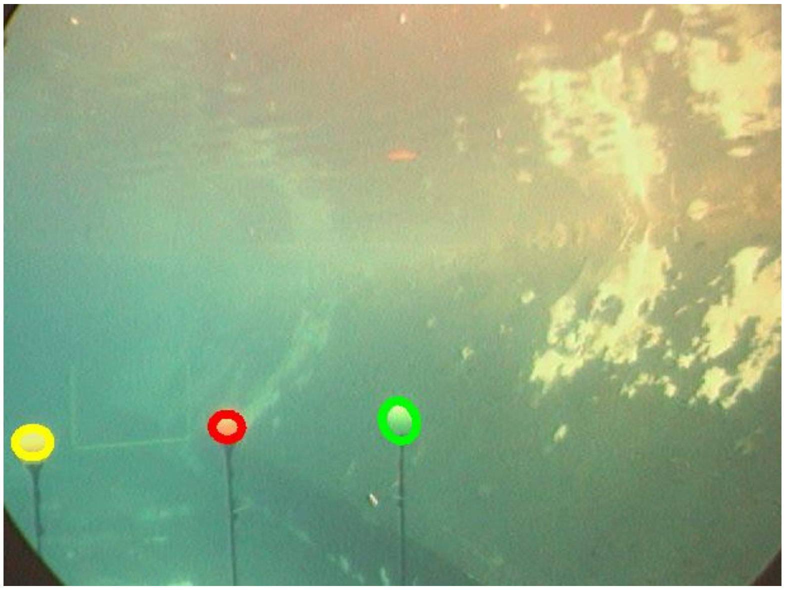
Yellow Buoy



Green Buoy



Red Buoy



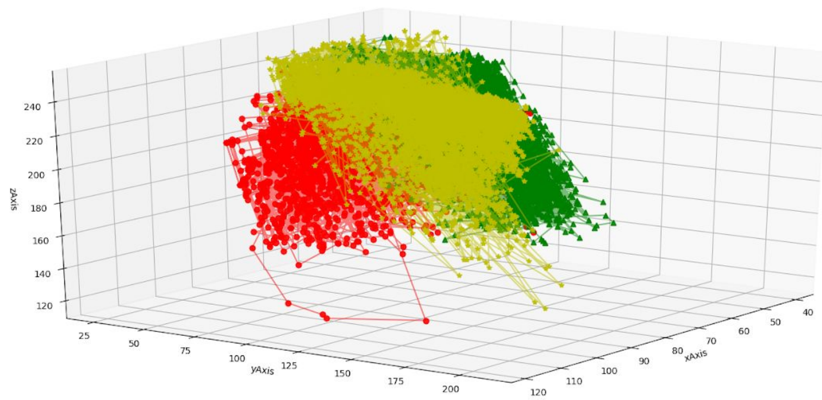
Final Result

Inference

- Detection area is greatly improved and now the aberrations can be easily removed using the area as a criterion.
- The green buoy result is so good that the entire buoy was clearly segmented using GMM.

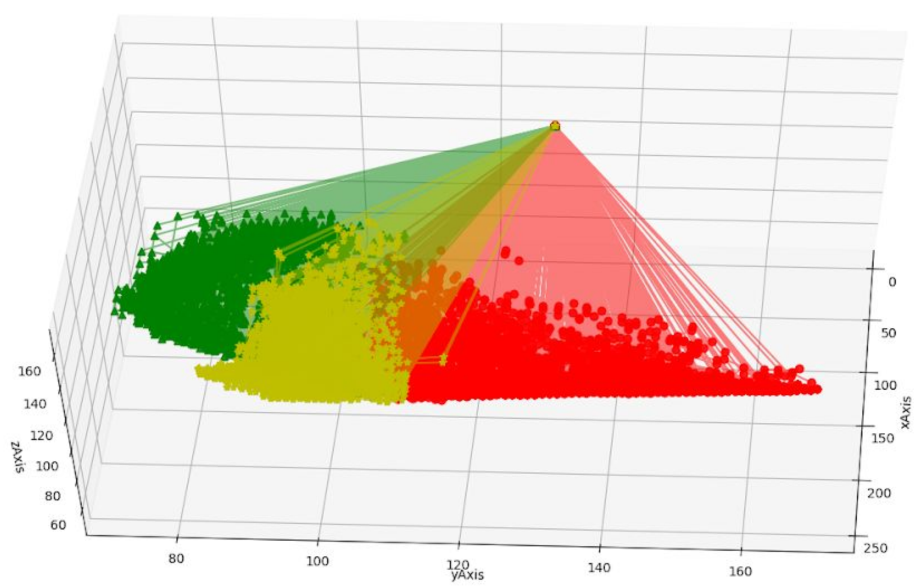
Exploring Other Color Spaces

HSV Color Sapce



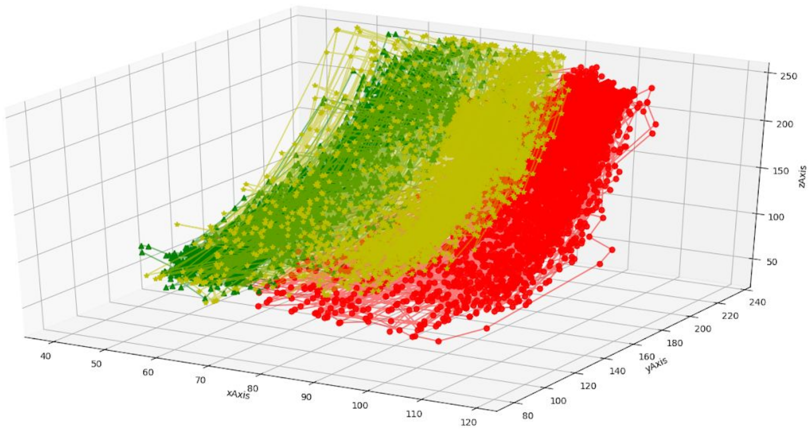
From the above data it is seen that there is a good amount of overlap in buoy data. Comparing with RGB color space, were data is more or less separate, RGB color sapace is better when compared to HSV.

LAB Color Space



LAB space has better separation of buoy data than HSV. But when compared to RGB color space which aso has better separation of data, there might not be a major improvement in segmation of buoys when LAB color space is used.

HLS Color Space



From the above data it is seen that there is a good amount of overlap in buoy data. Comparing with RGB color space, were data is more or less separate, RGB color sapace is better when compared to HLS.

Deliverables-

1. TrainingImages Directory
   Contains the cropped images that are required for training
2. Images Directory
   Test data.
3. Segmentation_1DGaussian.py
   Executable that will create the Histograms and 1D Gaussians for the training data and use it to segment the buoys.
4. Em_1D.py
   Executable for testing the Expectation Maximization algorithm on 1D data (comprising of 3 groups) to obtain a GMM of 3 groups.
5. Training_3d_Gaussian.py
   Executable for EM for n-Dimensional Gaussian for k- clusters. This file specifically fits 3 Gaussians for 3 categories(1 each belonging to each buoy)
6. Segmentation_3D_GMM.py
   Executable for segmenting the buoys with the calculated clusters from running the EM to fit 3 clusters per buoy.

Video Links-

1.1D Gaussian segmentation
https://drive.google.com/file/d/1fG4eq13SRAxhk0Mfl9am_DQCIU6PHvQi/view?usp=sharing
2.N D Gaussian segmentation
https://drive.google.com/file/d/1GhizL-lhZboy6LvRamKCqHTjRfdc5jly/view?usp=sharing