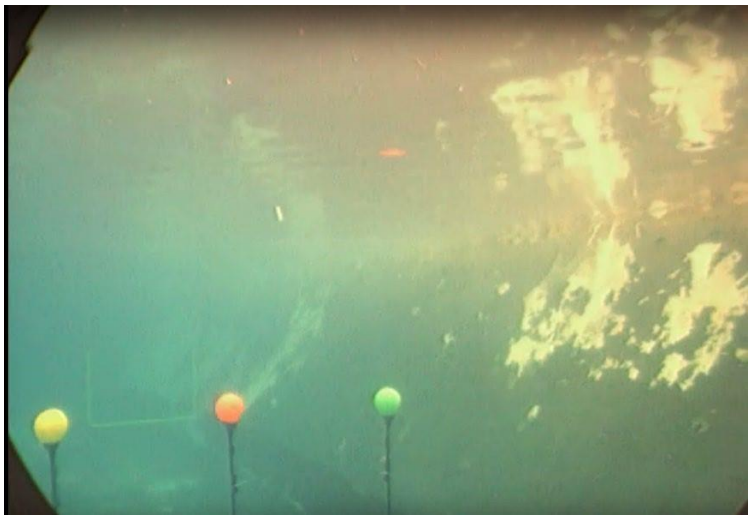
"[ENPM 673] Project-3"

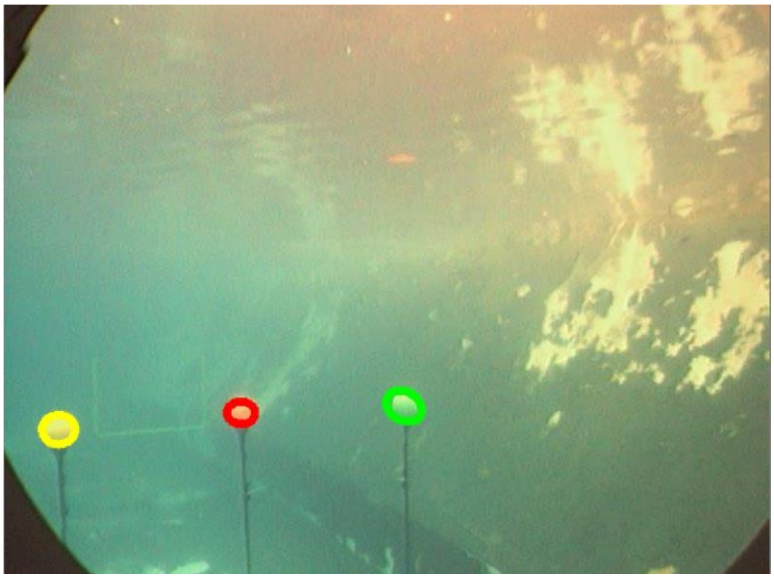
Perception for Autonomous Robots

PROJECT REPORT

Objective of the Project-

Obtain a tight segmentation of each of the three colored (Red, Green, Blue) underwater buoys for the entire given video sequence by applying a tight contour (in the respective color of the buoy being segmented) around each buoy.





Approach technique in focus-

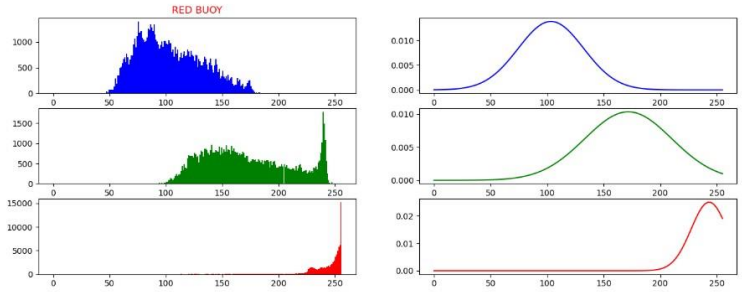
Color segmentation using Gaussian Mixture Models(GMMs) and Expectation Maximization techniques.

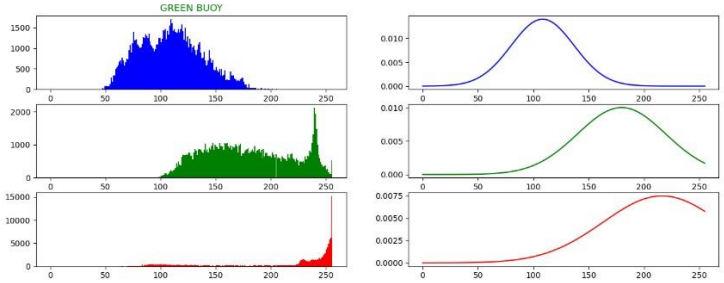
Implementation-

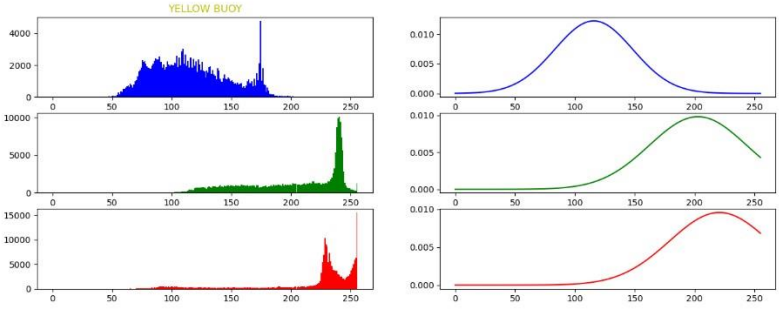
1) Data Preparation-
    a) Cropped a set of images(10) for each buoy

b) Separating RGB components and histograms-

For each training set we computed the mean and covariance from the histogram and plotted it

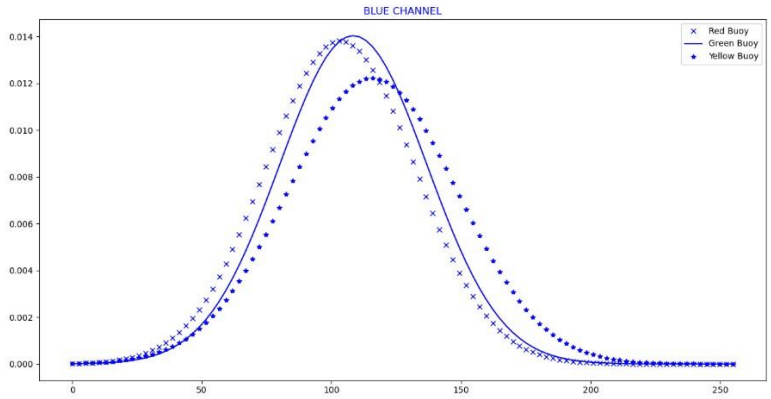


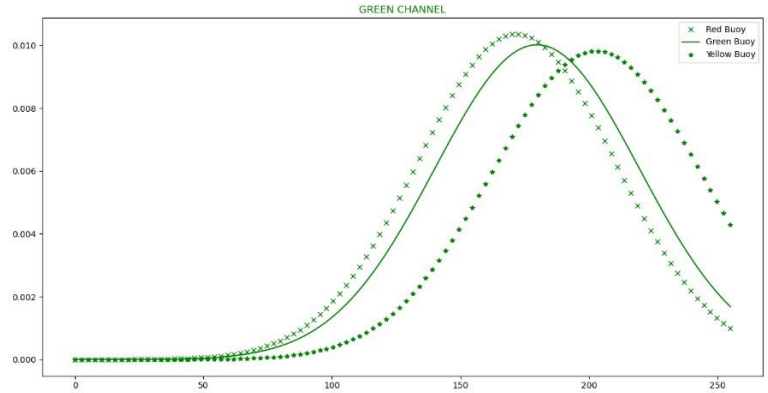


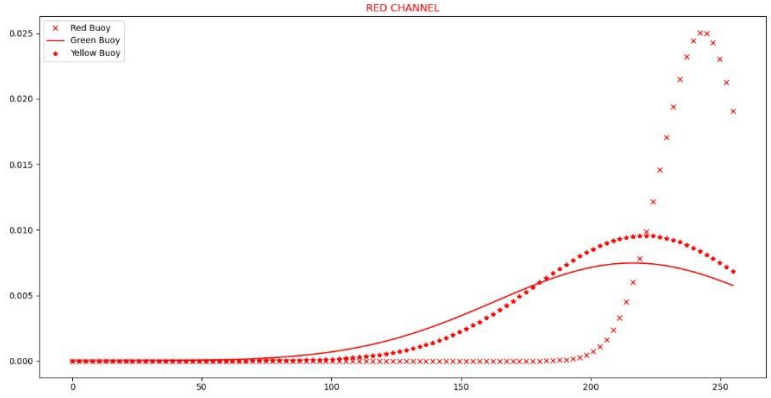


c) Choosing the best gaussian distribution-

We plotted the gaussian distributions for all 3(Red, Green, Blue) channels and based on the results chose the most distinguishing one.







Based on the results, only the Red channel gaussian distribution has enough distinction to separate the buoys.

d) 1-D Gaussian to segment buoys-

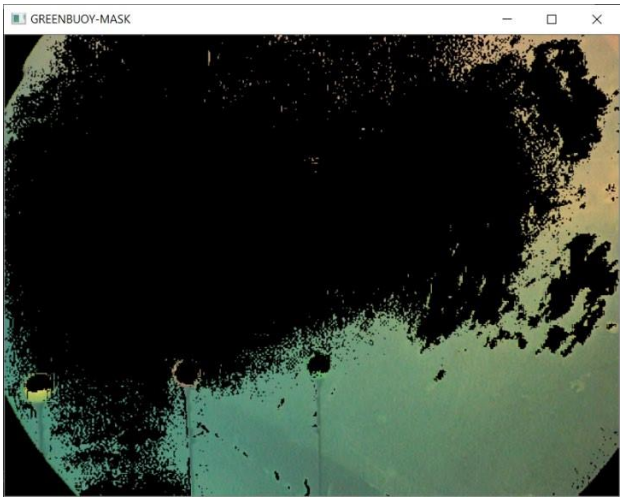
1) The color channels-

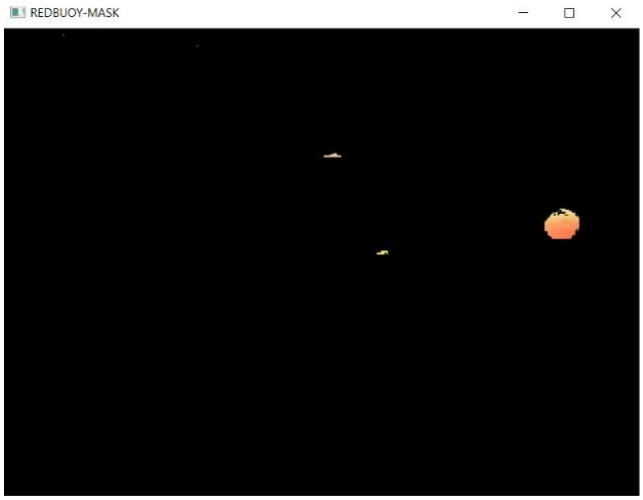
   Red Buoy- Product of Red and Green channels.

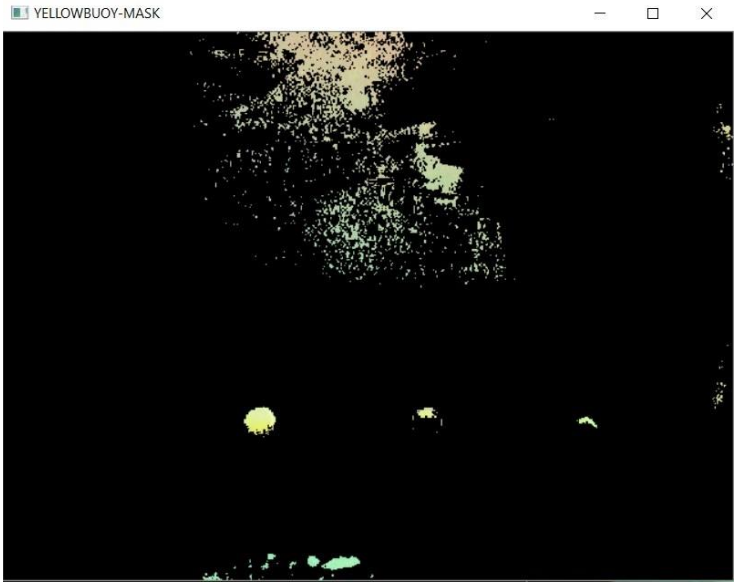
   Green Buoy- Product of Blue and Green channels.

   Blue Buoy- Product of Green and Blue channels.

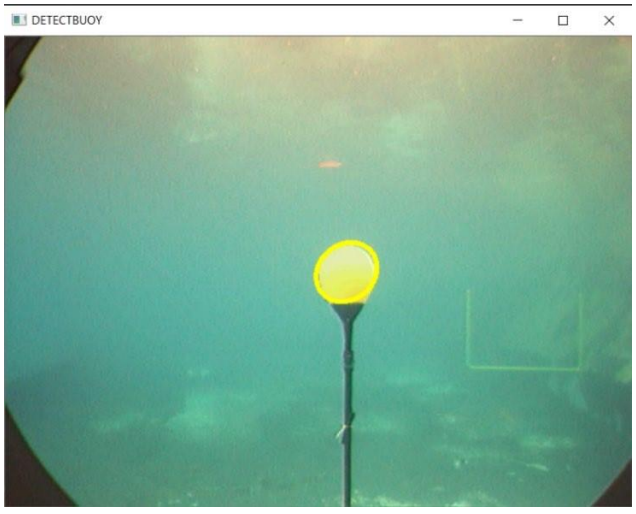
2) Start with the first pixel and as proceeding compare with each previously calculated gaussians and group on best matching gaussian.
3) Using mean and standard deviation color channel probability is calculated based on the training set results.
4) To separate the buoys mask matrices are constructed and each pixel belonging to the buoy are marked white and added to the mask.
5) The masks are applied to the original image, yielding the results-

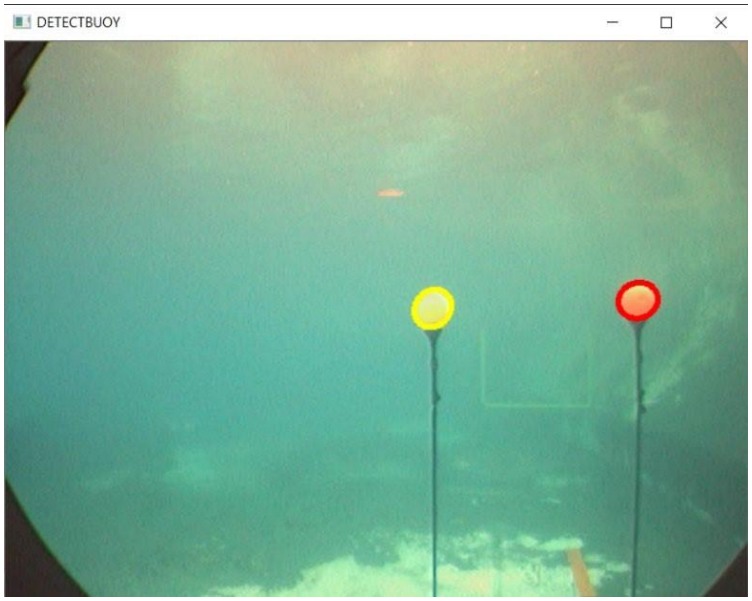


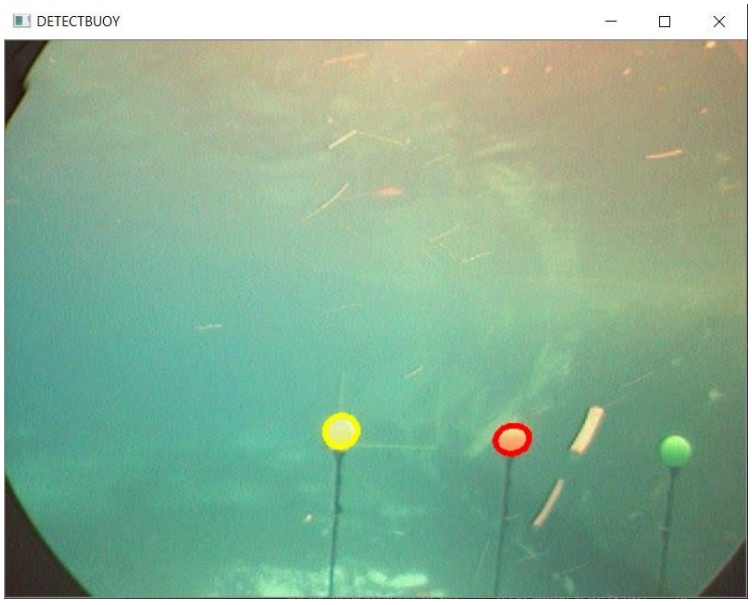




6) Based on the results, the green mask does not offer much insight due to the color of water, the yellow offers some separation but also has other yellow components due to presence of light, the red channel offers the best fit separation.
7) Using the cv2.fitellipse routine built in OpenCV which uses Greens Formula to draw contours based on the masks following results are yielded,







Gaussian Mixture Models Overview-

Gaussian Mixture Models(GMMs) are a way of segmenting an image into sub-population of existing normal gaussian distribution. GMMs provide unsupervised learning technique to segment the image without no prior knowledge of the sub-population.

In this case of problem statement there exist 3 different buoys and hence single gaussian or unimodal approach would yield a poor fit for the buoys. The multimodal approach i.e. existence of more than one peaks would yield in desired results.

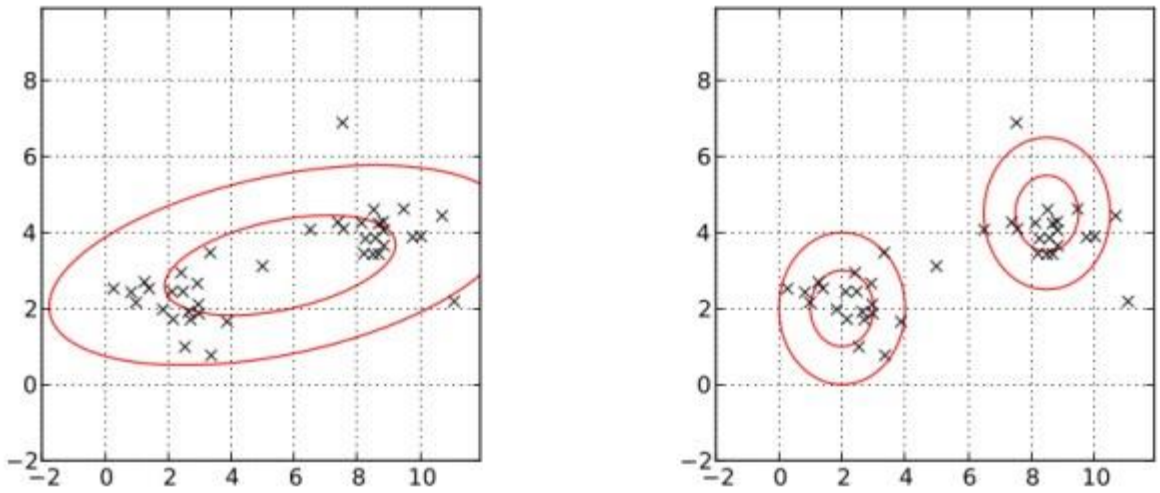


Fig. Multimodal Gaussian Distribution

A Gaussian mixture model is parameterized by two types of values, the mixture component weights, and the component means and variances/covariances. For a Gaussian mixture model with KKK components, the kk^{tth}

component has a mean of μ_k and variance of σ_k for the univariate case and covariance matrix of $\Sigma_k$for the multivariate case.

One-dimensional Model

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \mid \mu_i, \sigma_i)$$
$$\mathcal{N}(x \mid \mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)$$
$$\sum_{i=1}^{K} \phi_i = 1$$

Multi-dimensional Model

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$
$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^{\mathrm{T}} \Sigma_i^{-1} (\vec{x}-\vec{\mu}_i)\right)$$
$$\sum_{i=1}^{K} \phi_i = 1$$

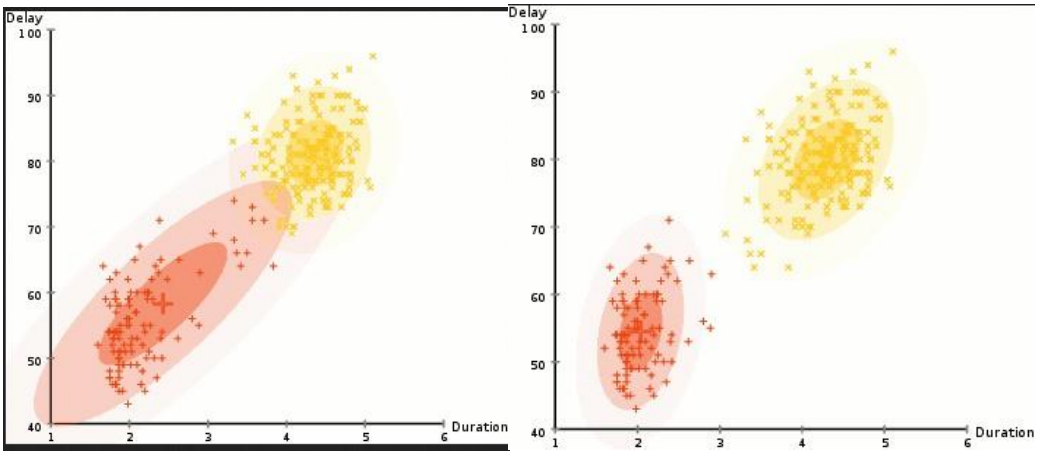
Expectation Maximization for Gaussian Mixture Models

Expectation maximization for mixture models consists of two steps.

The first step, known as the **expectation** step or **E** step, consists of calculating the expectation of the component assignments CC_{kk} for each data point $xx_{ii} \in XX$ given the model parameters ϕ_k μ_k $aaaaaa$ σ_k.

The second step is known as the **maximization** step or **M** step, which consists of maximizing the expectations calculated in the E step with respect to the model parameters. This step consists of updating the values ϕ_k μ_k $aaaaaa$ σ_k.

The entire iterative process repeats until the algorithm converges, giving a maximum likelihood estimate. Intuitively, the algorithm works because knowing the component assignment CC_{kk} for each $xx_{ii} \in XX$ makes solving for ϕ_k μ_k $aaaaaa$ σ_k easy, while knowing ϕ_k μ_k $aaaaaa$ σ_k makes inferring $pp(CC_{kk}|xx_{ii})$ easy. The expectation step corresponds to the latter case while the maximization step corresponds to the former. Thus, by alternating between which values are assumed fixed, or known, maximum likelihood estimates of the non-fixed values can be calculated in an efficient manner.



Algorithm for Univariate Gaussian Mixture Models

Initialization Step:

- Randomly assign samples without replacement from the dataset $X = \{x_1, ..., x_N\}$ to the component mean estimates $\hat{\mu}_1, ..., \hat{\mu}_K$. E.g. for $K = 3$ and $N = 100$, set $\hat{\mu}_1 = x_{45}, \hat{\mu}_2 = x_{32}, \hat{\mu}_3 = x_{10}$.
- Set all component variance estimates to the sample variance $\hat{\sigma}_1^2, ..., \hat{\sigma}_K^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$, where $\bar{x}$ is the sample mean $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$.
- Set all component distribution prior estimates to the uniform distribution $\hat{\phi}_1, ..., \hat{\phi}_K = \frac{1}{K}$.

Expectation Step:

Calculate $\forall i, k$

$$\hat{\gamma}_{ik} = \frac{\hat{\phi}_k \mathcal{N}(x_i \mid \hat{\mu}_k, \hat{\sigma}_k)}{\sum_{j=1}^{K} \hat{\phi}_j \mathcal{N}(x_i \mid \hat{\mu}_j, \hat{\sigma}_j)},$$

where $\hat{\gamma}_{ik}$ is the probability that x_i is generated by component C_k. Thus, $\hat{\gamma}_{ik} = p(C_k|x_i, \hat{\phi}, \hat{\mu}, \hat{\sigma})$.

Maximization step

Using the $\hat{\gamma}_{ik}$ calculated in the expectation step, calculate the following in that order $\forall k$:

- $\hat{\phi}_k = \sum_{i=1}^{N} \frac{\hat{\gamma}_{ik}}{N}$
- $\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$
- $\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}(x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$.

Once EM is complete we run clustering to isolate the buoys.

Implementation of GMM-

Steps involved-

1) Data Preparation
2) Training
3) Fitting and clustering

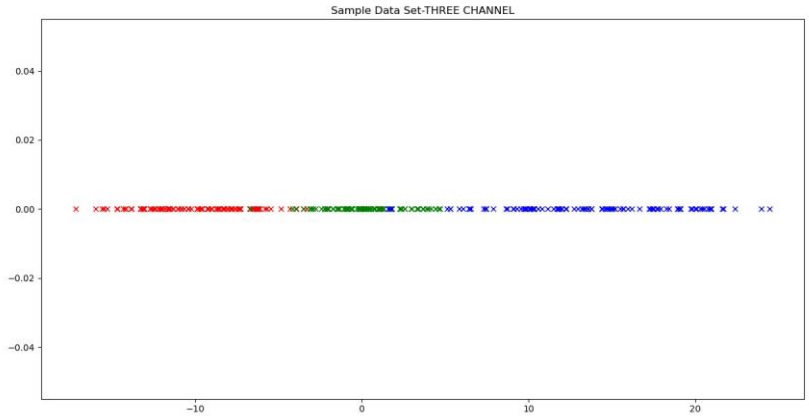
1-D Gaussian-

Data Preparation-

We form normalized clusters using random 100 test points each with following parameters-

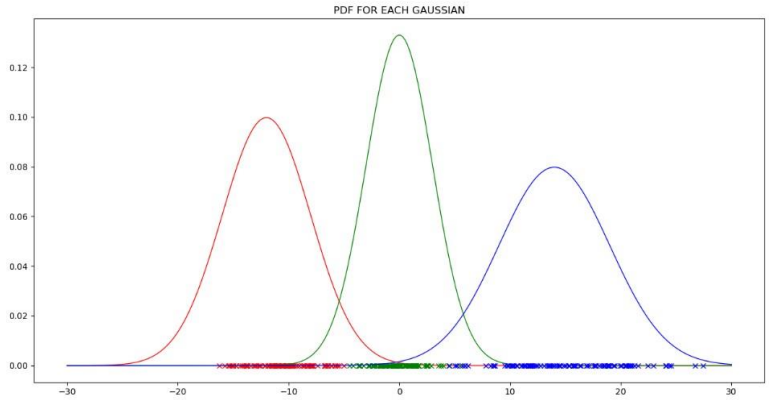
$$MMMMaaaa = 10,0,14$$
$$SSSSaaaaaaaaSSaa\ DDMMDDDDaaSSDDDDaa = 3,2,5$$

The results are plotted on a single axis yielding,



Expectation Maximization- **Expectation Step-**

We calculate probabilities based on the mean and standard deviation in the previous step.

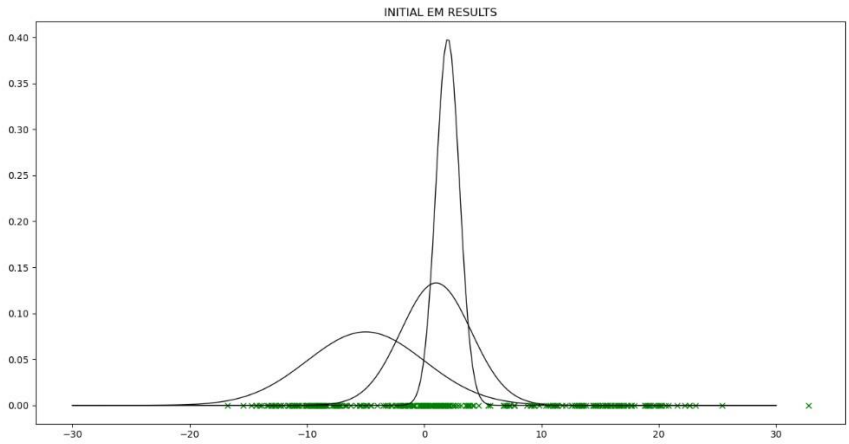


Maximization Step-

Using the probabilities calculated in the expectation step we find ϕ_k μ_k $aaaaaa$ σ_k.

1) Initialization of EM-

The initial run of Gaussian filtering with guessed mean and standard deviation yields following results. In subsequent iterations weights will be attracted to the curves and optimum mean and variances will be found.

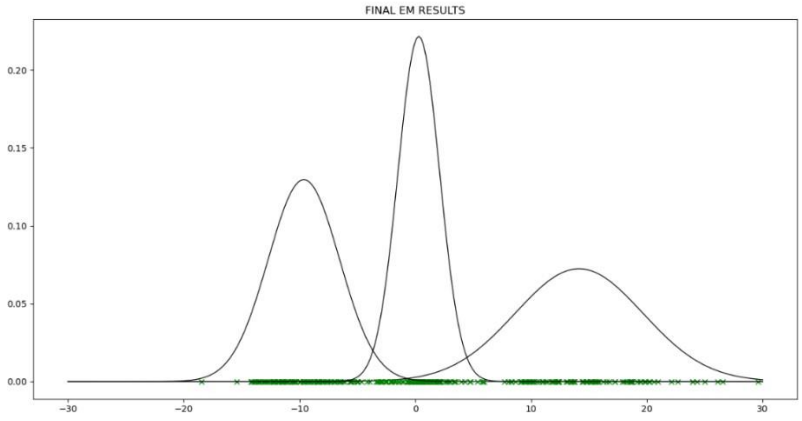


2) Final Results-

The last iteration of EM gives the optimum mean and variance values as-

$$MMMMaaaa = 14.07, -9.65, -0.37$$

$$VVaaSSDDaaaaVVMM = 4.78, 2.87, 1.79$$



Expanding the 1D gaussian to 2D-

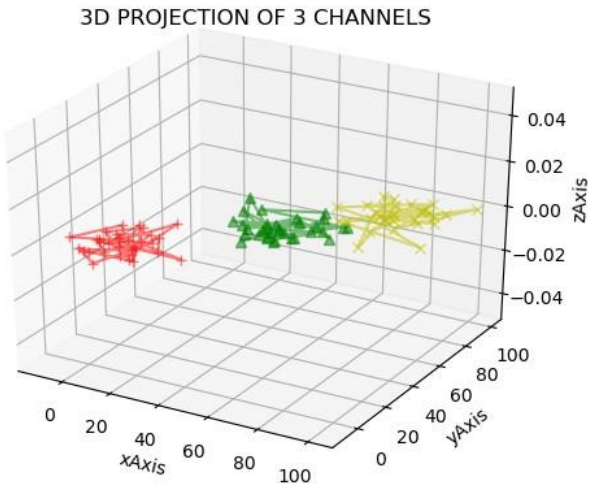
Testing the data on 2D planes using the EM algorithm.

1) Data preparation-

We test the data on 3 randomly generated mean and variances

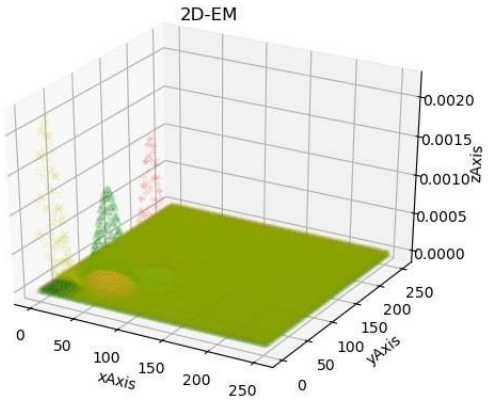
$$MMMMaaaa = 10,50,80$$

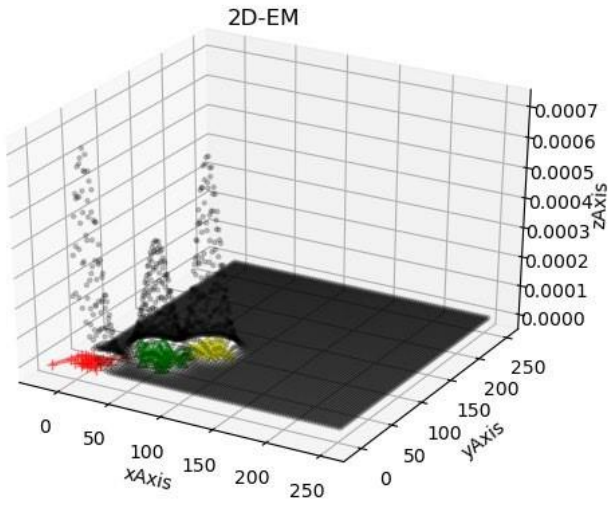
$$VVaaSSDDaaaaVVMM = 10{,}10{,}10$$



2) Training-

As the mean and variances are chosen randomly and not optimized the convergence of mean and co-variance might not be guaranteed. The EM algorithm converged the Mean and Covariance. The 3 gaussians are plotted below





The converged values are-

```
[[80.8844337  79.48348291]
 [50.19543272 49.18984544]
 [ 6.55045638 11.36677474]]
[array([[94.64880514, -3.59386475],
       [-3.59386475, 80.02640024]]), array([[158.96730018, -44.90470818],
       [-44.90470818, 117.05716829]]), array([[93.8789409 , 18.9874421 ],
       [18.9874421 , 61.04917846]])]
```

3. **Initializing Mean and Co-Variance**

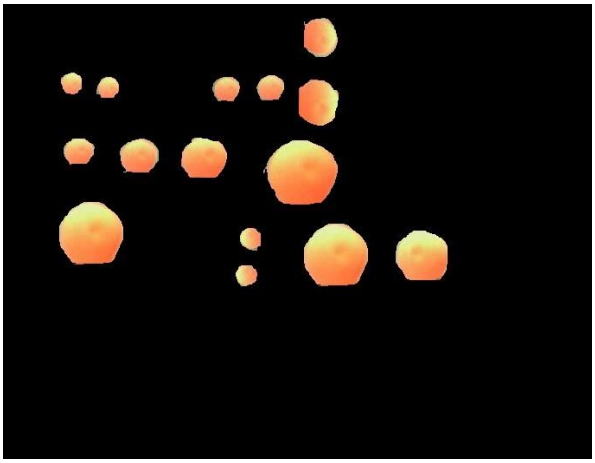
Since the convergence is completely dependent on initial values, initializing the seed values randomly will not provide with the best results. So educated guess must be taken for the initial values. To do so we first plot the data and see the clusters. Then seed values for Mean can be chosen such that 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.

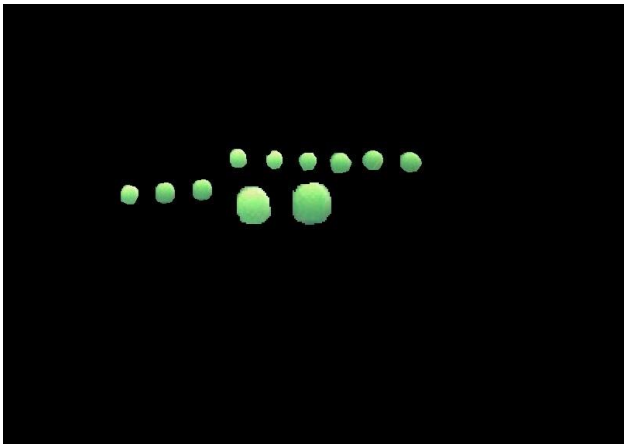
Fitting 3D Gaussian for Buoy detection-

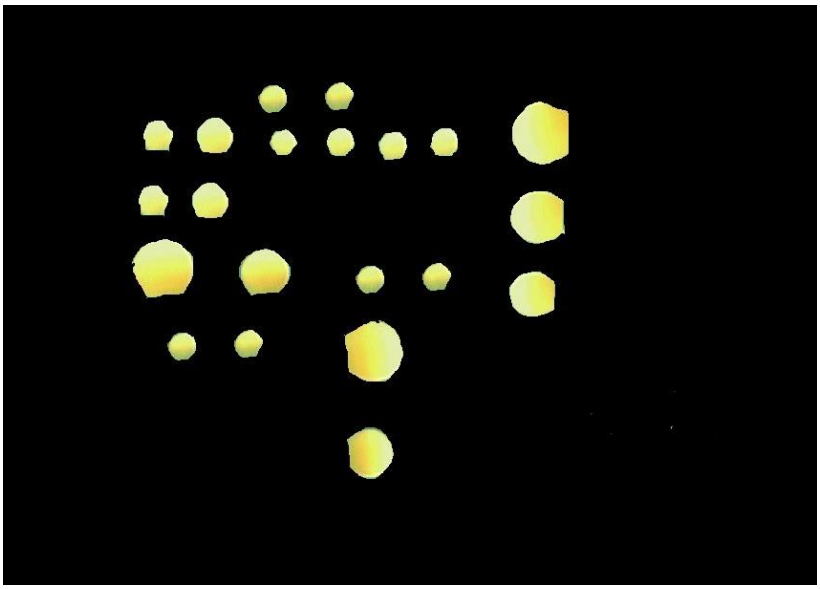
Once the EM working is ensured, it can be trained on the actual data set. The images are taken in RGB color space. There is a possibility to go to a higher dimension to add more parameters such as shape, size, etc. along with the color pixels but it takes a lot of time to generate tagged data. When such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

Step1- Data Preparation-

The training set was developed using 10 frames with buoys placed far away with variety of positions were randomly chosen. The Red, Green and Yellow buoys were separated and masked.







Training Data set-

Fitting clusters based on the dataset and separating components.

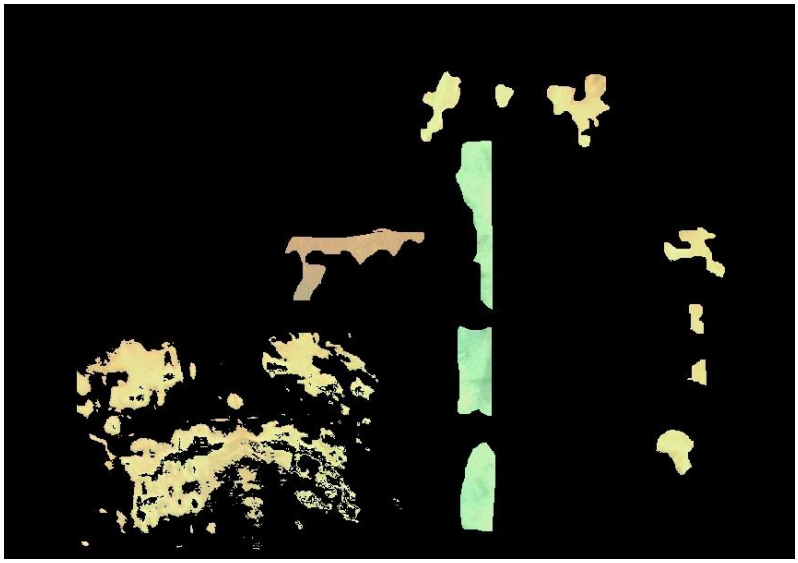
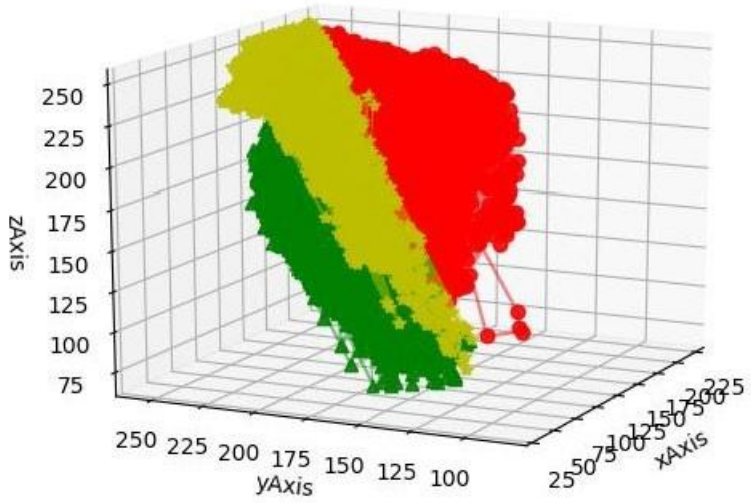


Fig. Water separation by clustering based on the training dataset

Pipeline-

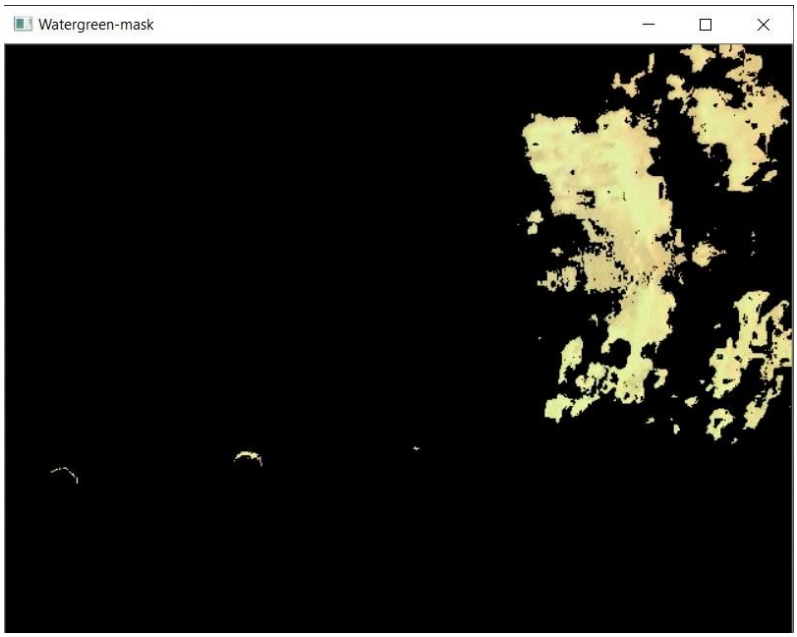
Data plotting-



Fitting a single Gaussian plot results in good number of overlaps. Fitting multiple gaussians for a single category will result in a tighter fit.

Removing water-

a) Initial Masking



b) Final Result-

Buoy detection after removing water

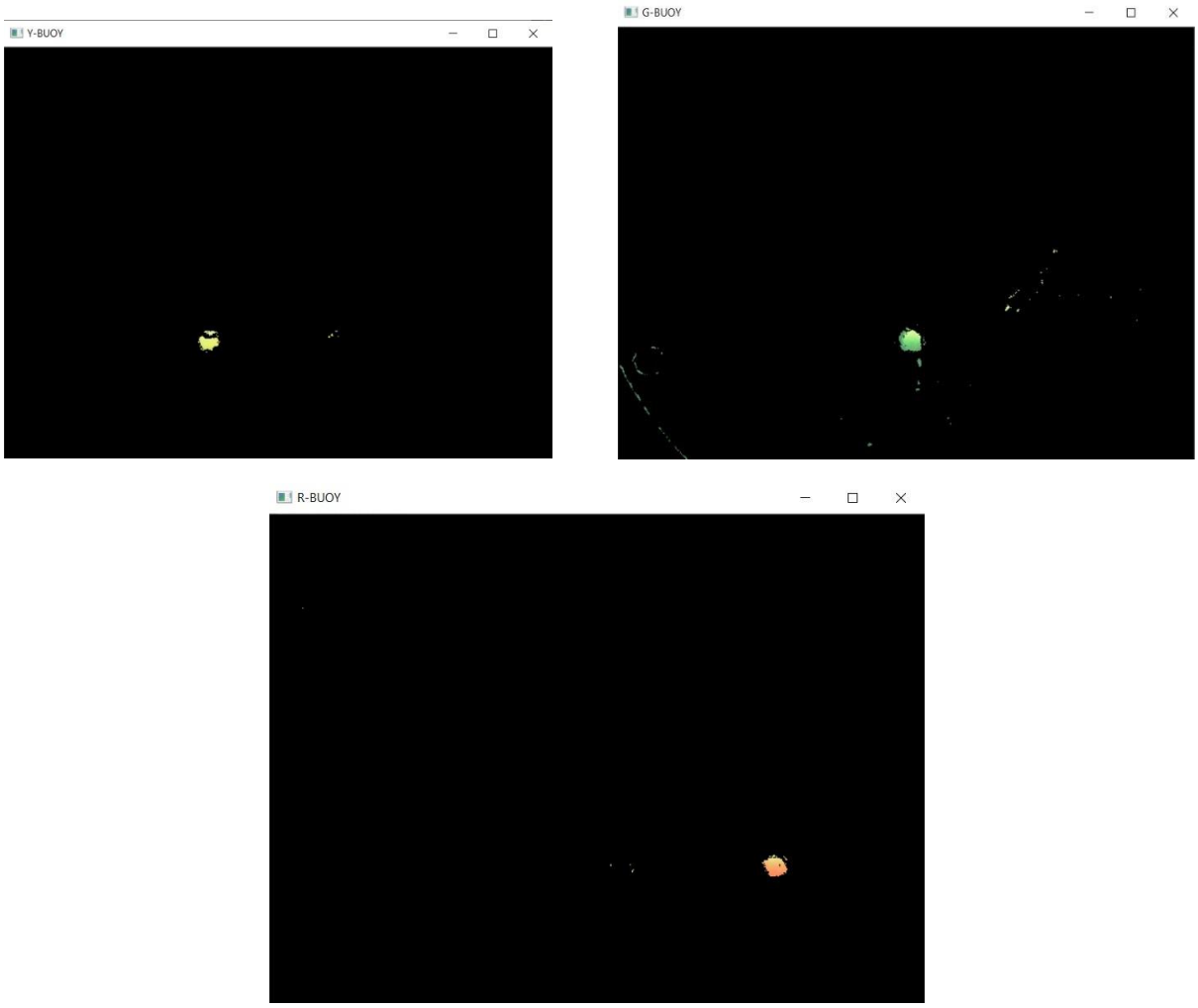
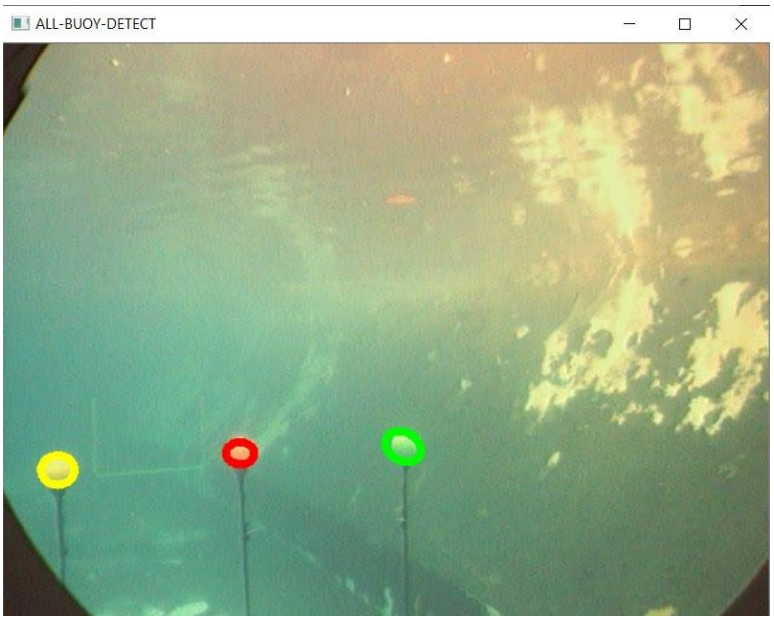


Fig. Masked and separated Yellow, Green and Red Buoys respectively

Final Fitting on the original frame-



References-

1) https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=fitellipse
2) https://docs.scipy.org/doc/scipy/reference/stats.html
3) https://brilliant.org/wiki/gaussian-mixture-model/