CMSC426 - Project 1: Color Segmentation

Akash Guha

Choice of ColorSpace:

Used RGB color space for this project, but variation in environmental lighting created many difficulties viz. due to illumination/shadow the same color appeared in different shades resulting in false positives and true negatives. Another challenge is to experimentally determine the threshold values of the probability density function. This problem of determining the color threshold values increases with the number of colors to be segmented.
A good alternative to the RGB color space, which is more robust towards the changes in lighting conditions and other noises is the HSV color space. The lighting difference is handled by value and the colors by the saturation component.

GMM Initialization:

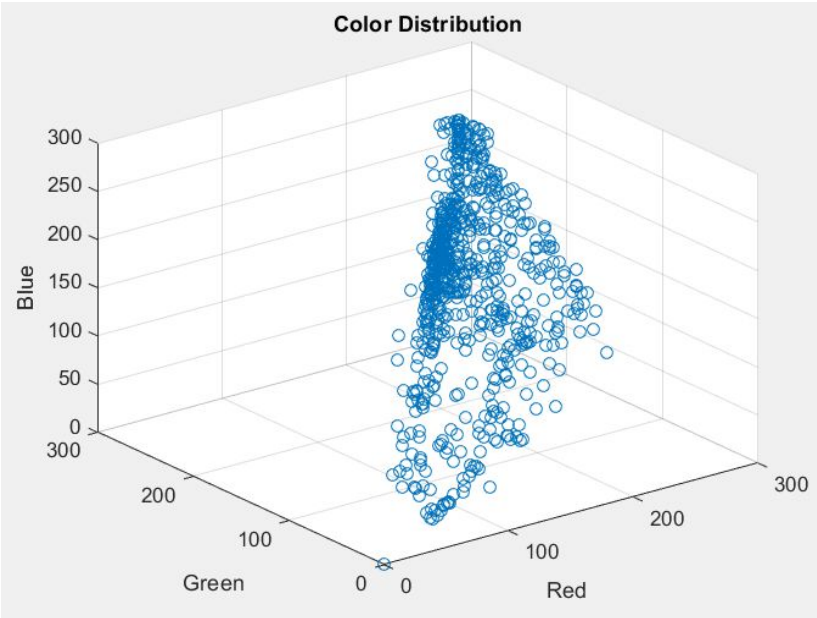
We randomly selected data points to use as the initial means and the covariance matrix for each gaussian is equal to the covariance of the full training set initially. Also each gaussian has equal prior probability i.e. same fraction of the dataset belongs to the gaussian.
It was observed, depending upon where the parameters are initialized there are chances of getting stuck in the local minima, hence the result of every run is not consistent. For faster processing in MATLAB save the workspace with optimal parameters.

Increasing the number of gaussians to fit the GMM model improved the output result significantly. But at the same time one should remember that a large value of number of gaussians increases the chances of overfitting the dataset. Thus an optimal number of gaussians need to be selected which appropriately fits the model and at the same time don't overfit.
Seven gaussian models are used in our GMM model.

Running 100 iterations of estimation and maximization steps with a mean difference threshold value of 0.001 helps in good convergence of the parameters (mean, covariance and weights).
Time required is more but not iterating for a significant number results in non-convergence.

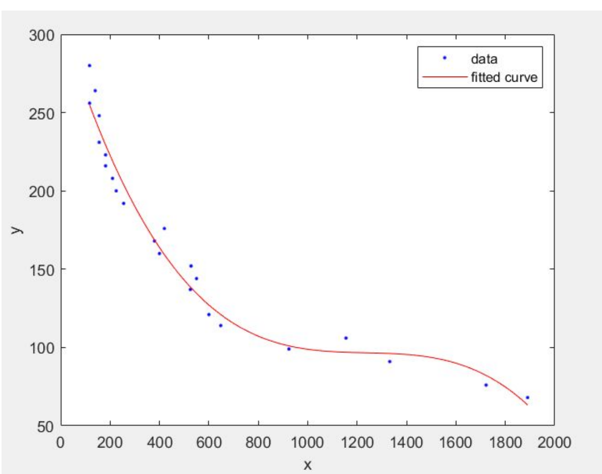
Advantages of GMM over Single Gaussian:

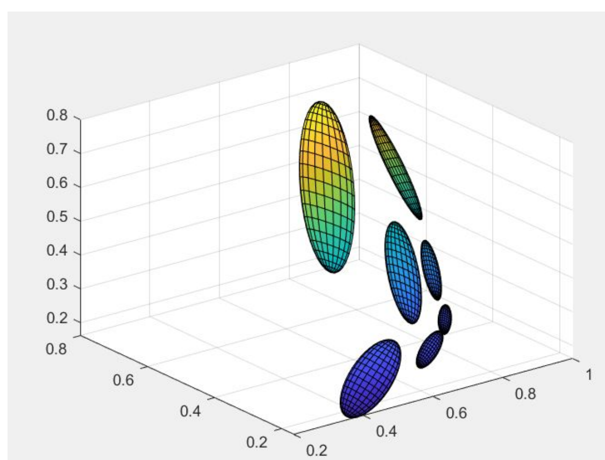
As with environments with different lighting conditions, it is not possible to fit a simple single gaussian (colors not well bounded by an ellipsoid) model, thus a sum of multiple simple gaussians is a very good approximation to model the complicated function. Each gaussian takes care of a specific set of data points which is not the case with a simplified single gaussian.



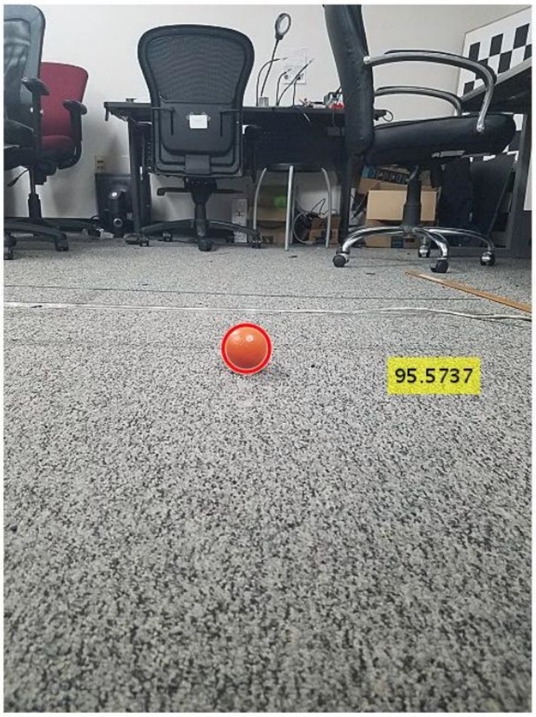
As seen from the color distribution of the orange ball, very difficult to fit a single ellipsoid to the dataset.

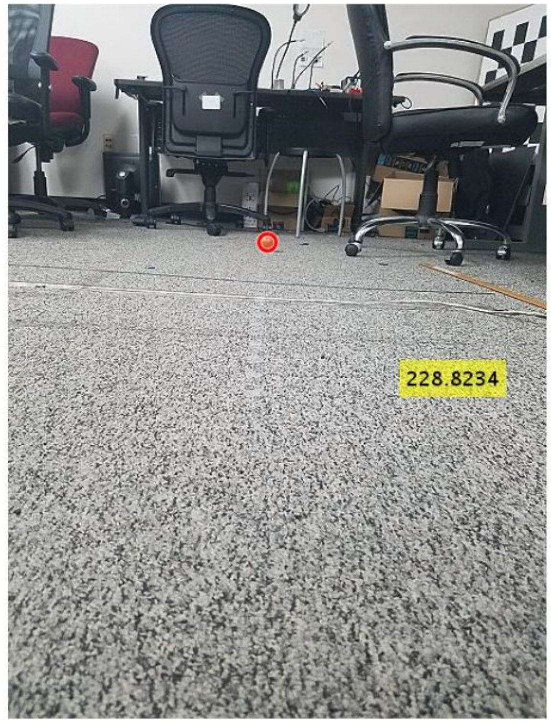
Distance Estimation and Cluster Segmentation on Test Set:
Depth/distance of the ball was calculated by training a regression model whose input feature is the area of the ball in pixels and the corresponding output distance is in centimeter. After training, the same model is used on the test images to predict the distance. A third order polynomial function is used to fit the training dataset.



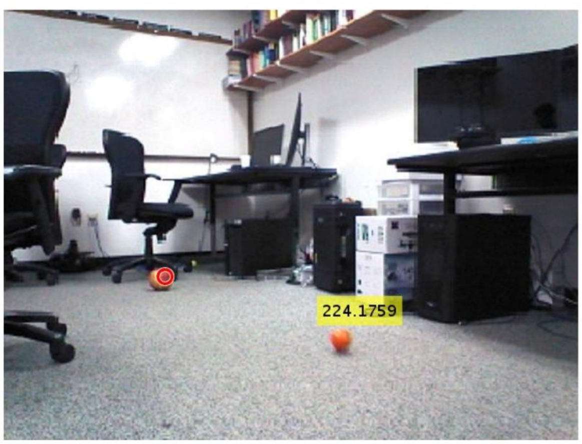


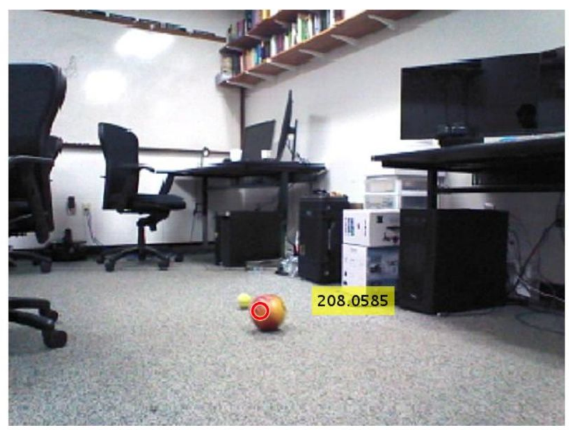
Sum of seven simple gaussians are used to fit the complicated dataset. Below images show the correctly identified ball along with the estimated distance in 'cm'.





Below images show the misclassifications of the orange ball.





Some issues which we encountered in the testing phase were that few of the images were blurry/not properly shot, the image sizes were different, background had shades of red and orange, ball with different shade of red(similar to orange) present in the image. These led to false positives.

An interesting observation was changing the prior from 0.5 to 0.2(as the probability of finding an orange pixel in the image is very less) drastically improved the result, thereby eliminating many false positives.

Vectorization of code helped in increasing the time efficiency compared to using for loops for implementation.
One of the biggest limitations is the consistency of the GMM during training, this might be occurring due to underfitting of the color distribution and the way the parameters are initialized.

<u>References:</u>

https://cmsc426.github.io/colorseg/#gaussian
http://cs229.stanford.edu/notes/cs229-notes7b.pdf
https://www.mathworks.com/help/
https://en.wikipedia.org/wiki/
http://mccormickml.com/2014/08/04/gaussian-mixture-models-tutorial-and-matlab-code/
https://www.mathworks.com/matlabcentral/fileexchange/16543-plot_gaussian_ellipsoid