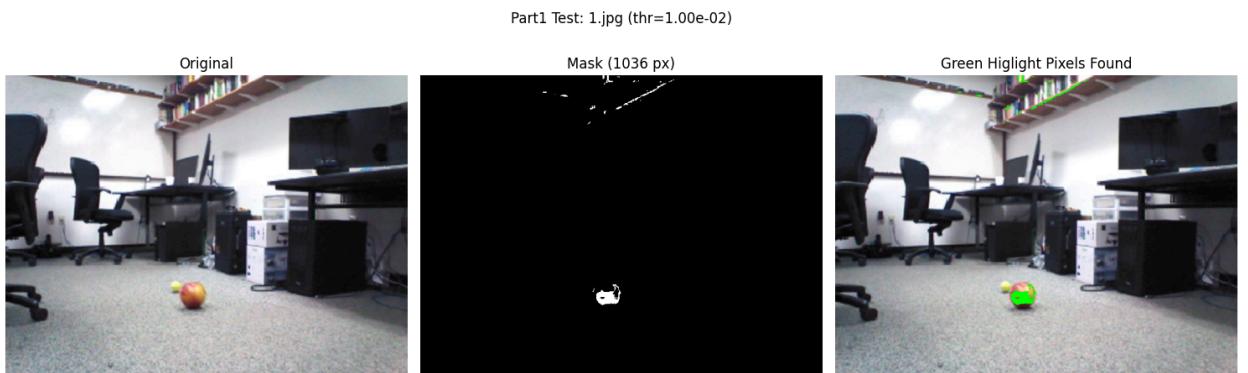
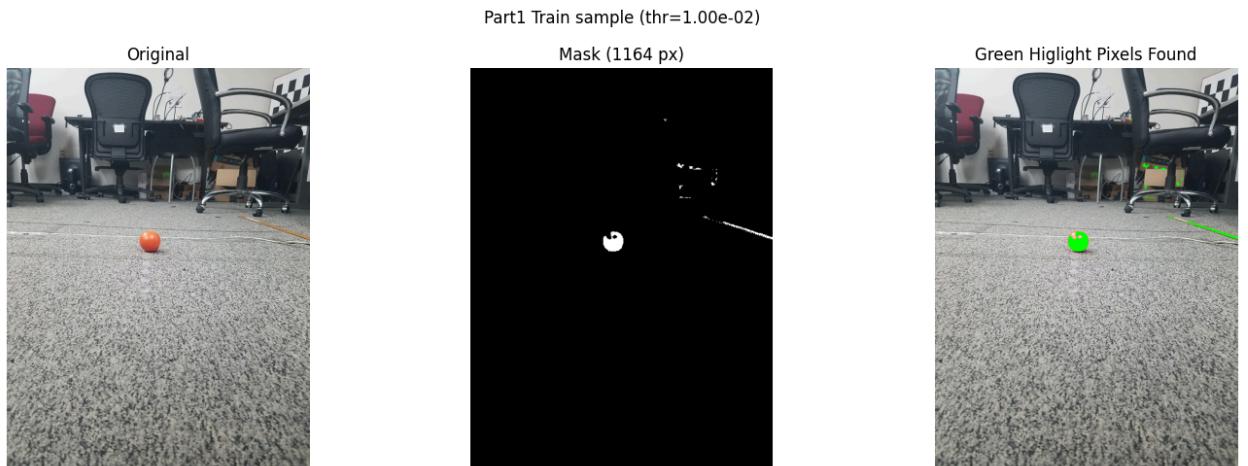
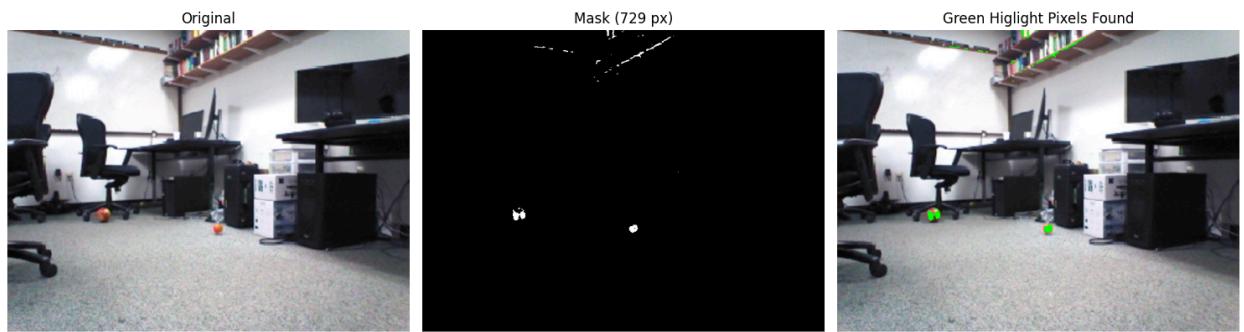


Report

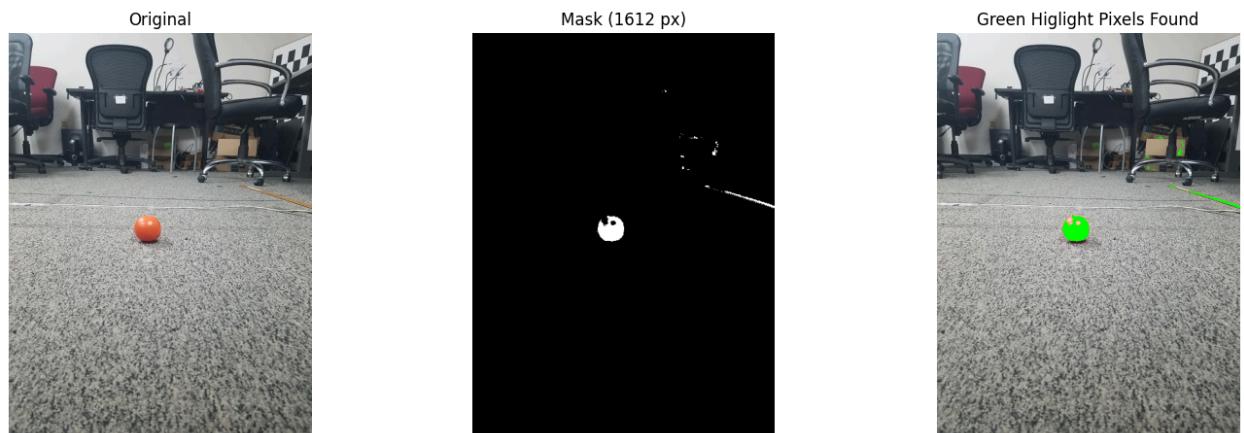
In the first part of this project, using a single Gaussian to detect orange. There was extracting training data images, extracting the orange off each training data image using masks of RGB and HSV, and then running a single Gaussian to detect orange in these images. In more detail, our orange pixel extraction process was done by first image normalization, RGB filtering to apply color constraints, ratio-based filtering to distinguish colors from each other, and brightness. This mask was done through trial and error with the goal of capturing as much of the ball as possible. One interesting problem faced in this masked through trial and error was that we were able to grid search to find the proper masks and threshold so that our result was the most optimal through testing on training images. After this, our single Gaussian was done after having the M step, the Gaussian PDF, and the classification step. These were textbooks, and a way to fully comprehend our algorithm was done in our last step, the visualization step, where our training images were being tested on.



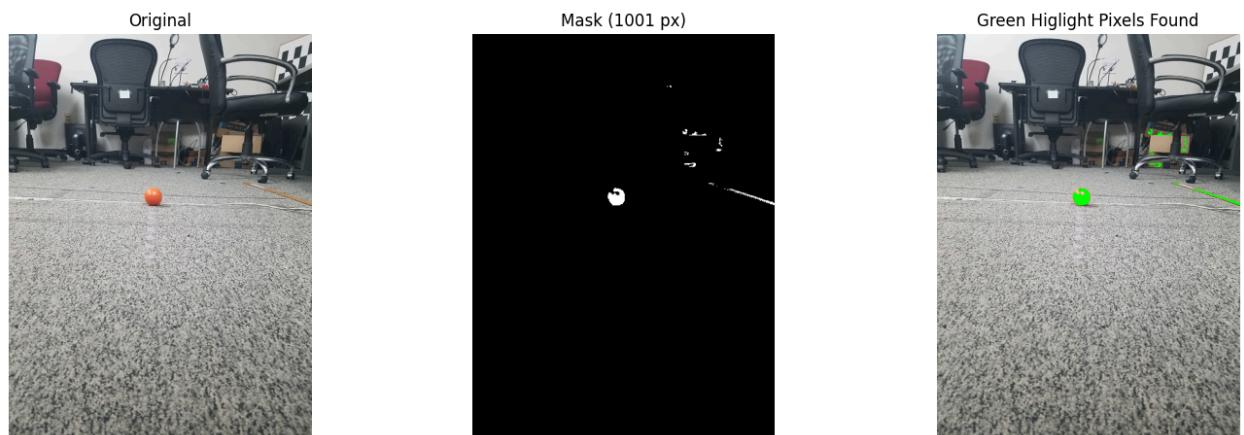
Part1 Test: 2.jpg (thr=1.00e-02)

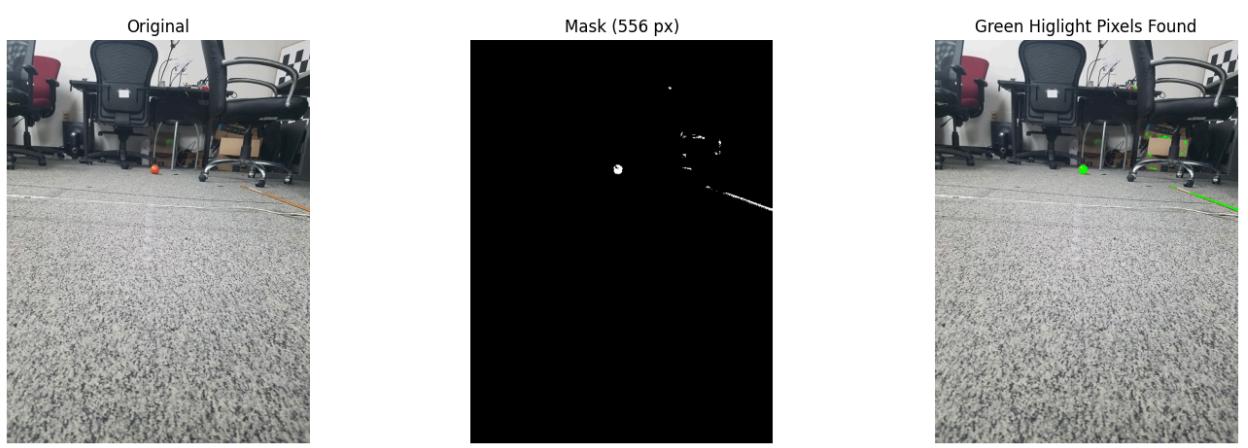
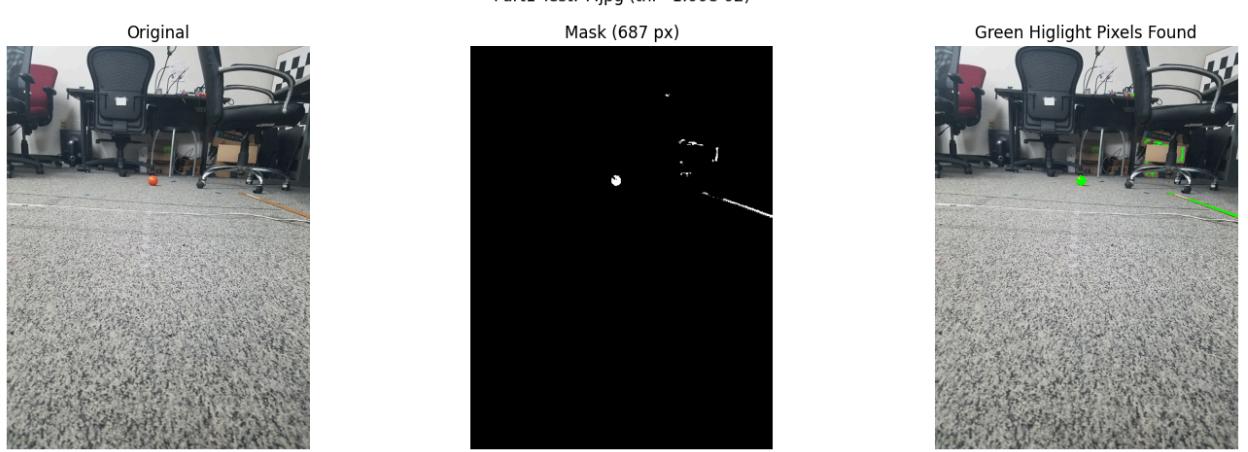
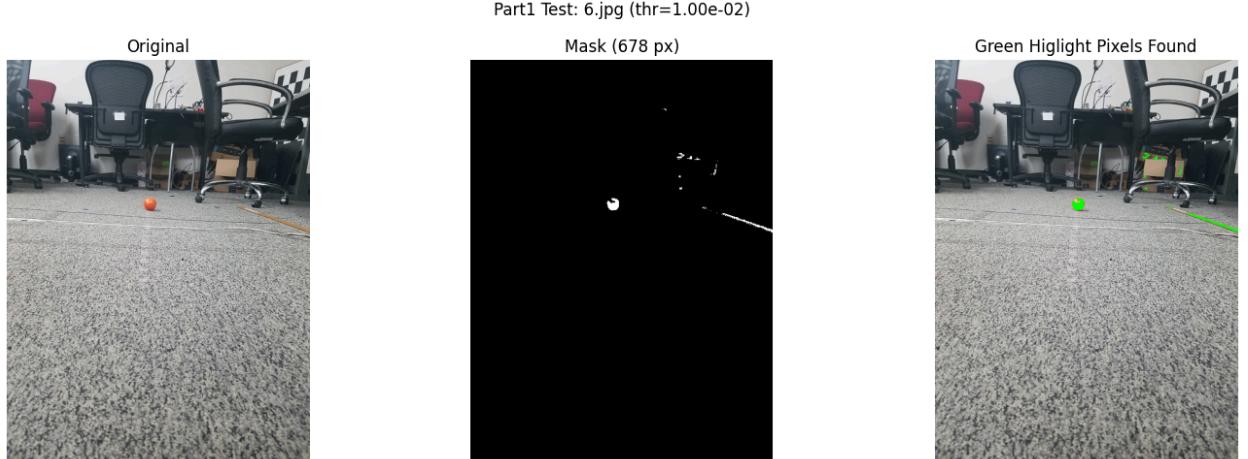


Part1 Test: 4.jpg (thr=1.00e-02)



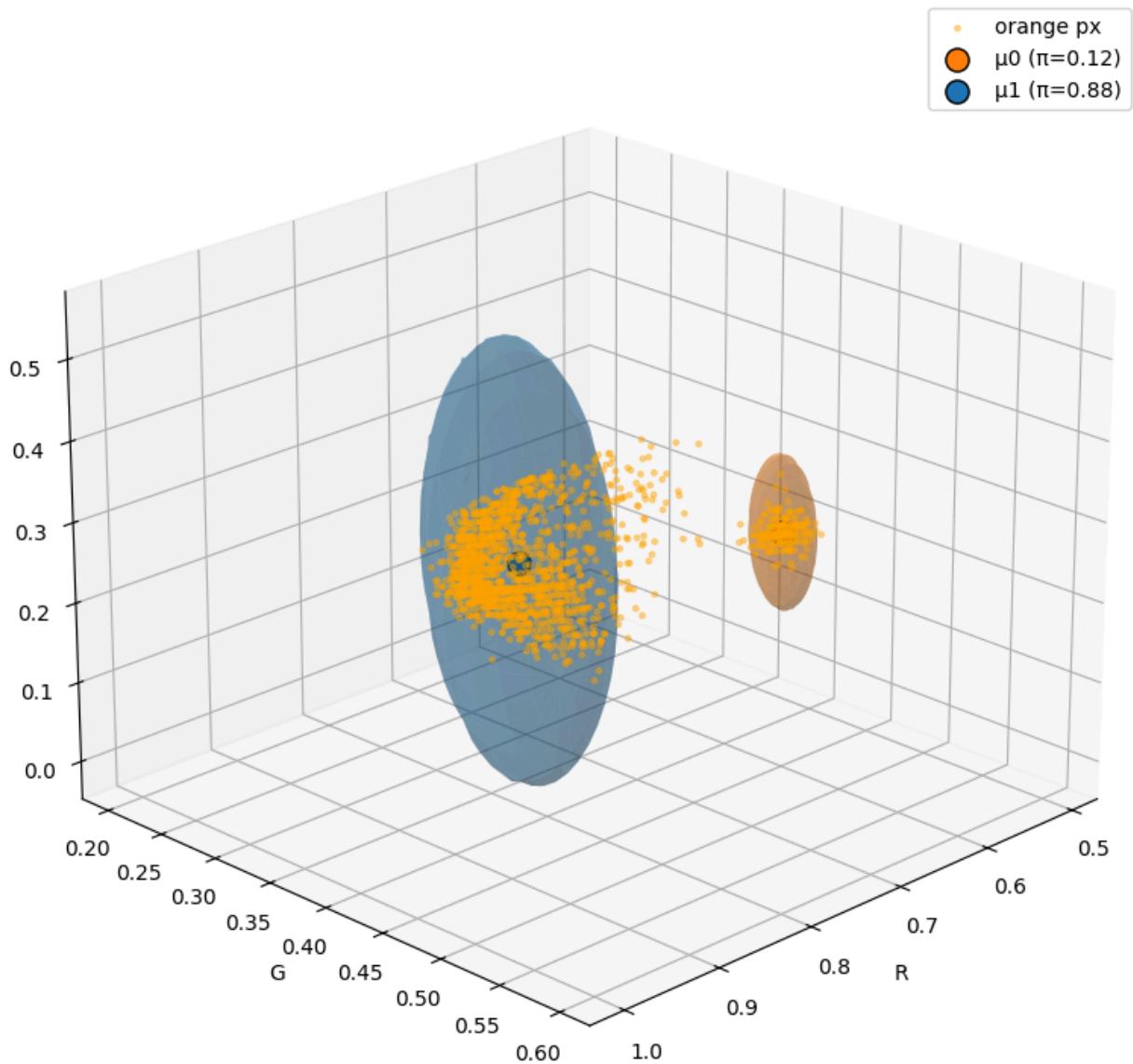
Part1 Test: 5.jpg (thr=1.00e-02)





In the second part of this project, we implemented a Gaussian Mixture Model to detect the orange ball and estimate its distance from the camera. The GMM approach built upon the single Gaussian foundation by using multiple Gaussian components to better capture the color variations in the orange ball. We began by implementing the EM algorithm from scratch, which alternated between the expectation step and maximization step to iteratively refine our model parameters. Our initialization strategy involved randomly selecting sample points from the extracted orange pixels to serve as initial means, while covariances were initialized using the global covariance of the training data with regularization added for numerical stability. We set K=2 components because this captured both the brighter and shadowed regions of the ball without overcomplicating the model. After training the GMM through the EM iterations, we applied the trained model to test images by computing posterior probabilities for each pixel and thresholding to create binary masks. For distance estimation, we leveraged the relationship between ball area and distance, fitting a parametric model of the form $\text{distance} = a/\sqrt{\text{area}} + b$ using least squares on the training data. The final visualization step involved plotting the GMM ellipsoids in 3D RGB space to show how the two components captured different aspects of the orange color distribution, as well as displaying the segmentation results and distance estimates for each test image.

GMM Ellipsoids in RGB



For this project, we used RGB color space as our primary representation. The RGB space was chosen because it directly corresponds to the camera sensor output and provides clear separation for orange detection through the red, green, and blue channels. Our initialization method for the GMM involved randomly selecting K sample points from the training data as initial means, with K=2 gaussians. The covariance matrices were initialized using the global covariance of all training data with regularization added to ensure numerical stability. This initialization approach was effective because it started with actual data points that represent real orange pixels,

rather than arbitrary values. The scaling factors (mixing coefficients) were initialized uniformly as $1/K$, giving each component equal weight at the start. We chose $K=2$ because the orange ball exhibits variation under different lighting conditions and viewing angles, which can be captured by two distinct color distributions better than a single Gaussian.

GMM is better than a single Gaussian in many aspects. In this project, where we report finding the color orange, GMM proves to be able to detect more orange. Fundamentally, a single Gaussian tries to group all oranges together into one. Due to our understanding of the color orange, it is actually groups of colors that make up a certain orange, where our GMM detects red-based orange and blue-based orange. In the case of using a single Gaussian, it generalizes all the oranges into one. Single Gaussian also tries to fit all into one; however, this solution is too general. In this case, GMM actually creates multimodal models, which allow for each component to have its own mean and variance, allowing for a better fit. Another benefit of using GMM is under lighting in computer vision, different lighting allows for the color orange to change, and can optimize each component under its own conditions. In the case of using a single, the Gaussian either becomes too broad(which includes non-orange) or too narrow(which misses valid orange). In this project, using both single and GMM, single seemed to be affected by outliers a lot more than GMM. The structure of GMM only affects a component, not the whole model. In this case, if one component is failing in GMM, others can still work.

The distance estimation was performed using an inverse square root relationship between the detected ball area and distance. From the training data, we observed that ball area decreases as distance increases, following an approximate inverse square relationship. We fit a linear model of the form: $\text{distance} = a/\sqrt{\text{area}} + b$ using least squares regression on the training images where filenames indicated the ground truth distance in centimeters. For the test images, our results showed: 1.jpg detected 1036 orange pixels with estimated distance 2.71 units, 2.jpg detected 729 pixels with distance 5.32 units, 3.jpg detected 1105 pixels with distance 2.71 units, 4.jpg detected 1612 pixels with distance 4.01 units, 5.jpg detected 1001 pixels with distance 4.01 units, 6.jpg detected 678 pixels with distance 4.01 units, 7.jpg detected 687 pixels with distance 5.32 units, and 8.jpg detected 556 pixels with distance 7.93 units. The cluster segmentation successfully identified the ball region in all test images, with centroids computed from the detected pixels. The segmentation masks show clean ball detection with minimal false positives, indicating that the GMM threshold of 0.1 was well-calibrated for separating orange ball pixels from background.

The primary strength of our algorithm is its robustness to light variation with the use of the two-component GMM in order to represent the bright and shadowed regions of the orange ball. Color masking during training data extraction allows the high-quality samples required to learn the color distribution. The EM algorithm is stable under proper initialization and regularization of covariance matrices. In addition, the distance estimation model is simple but effective with only area calculation required without the need for intensive geometric calculation. The algorithm, however, is subject to several drawbacks. It greatly relies on the initial color mask for data extraction during training, so if the mask parameters are not set properly, the consequent GMM learned will be suboptimal. Also, the algorithm depends on the ball being the principal orange object in the scene and would fail if there are other orange objects. The distance estimation assumes constant ball size and camera parameters and would hence fail for other ball sizes or camera configurations. Taking a look at the test output, the algorithm did perform very well in all test images, correctly identifying the ball in each case. The distance estimates also have an intuitive pattern where smaller areas detected signify larger distances. Failure would be under very unusual lighting conditions, partial occlusion of the ball, or in environments with orange objects that would provide false positives.