Darren Tung
ID: 170049054

Assignment 1: Ground Region Detection

I. Introduction

Farms across the world are becoming increasingly automated, with robots being deployed to harvest fruits and vegetables in greenhouses. However, there are many questions that must be answered before deployment can occur, among them being deciphering ground and non-ground textures for robots to move. To solve this problem, three images along with three corresponding ground truth labels were provided, with the goal being to train a model to disseminate between ground texture and non-ground texture.

II. Methods

A Python file was provided containing starter code that would be used to read the images and overlay either the ground truth or the predicted ground over the greenhouse images. The starter code requires NumPy, Matplotlib, and OpenCV to operate. From there, one of the three images was selected as a training image, with the other two being testing images. In the following section, each image was given a chance at being the training image to see which one produces an optimal result. Regardless, every pixel in the training image was iterated over to determine both the color in RGB format, represented as x, and the ground label of “not ground” or “ground”, represented as y=0 and y=1 respectively. Iteration would be repeated for the testing images, first to determine the color in RGB format and then to predict the label of “ground” or “not ground” using a Bayesian Classifier.

The Bayesian Classifier is created by first using the results of iterating through the training image pixels to calculate Pr(y=1), Pr(y=0), Pr(x|y=1), and Pr(x|y=0). Then, Pr(y=1|x) and Pr(y=0|x) can be calculated through a simplified form of Bayes’ Theorem that does not divide by Pr(x). The simplified form can be used because the goal is to compare Pr(y=1|x) and Pr(y=0|x), meaning Pr(x) can be canceled out in the denominator. If Pr(y=1|x) > Pr(y=0|x), the pixel is labeled as a ground texture. Else, it is labeled as a non-ground texture. Note that to optimize computation time and results, the RGB pixels were binned into 8, 16, and 32 bins based on value, with another goal being to compare the bins to see which produces an optimal result. This is finally accomplished by calculating the precision, recall, and f-score for the predicted labels vs. the actual labels.

III. Results

Note: Results will be rounded to the nearest 3 significant figures for readability.

Figure 1a: Training Image 1

Figure 1b. Testing Image 2 with 8 bins

Figure 1c. Testing Image 2 with 16 bins

Figure 1d. Testing Image 2 with 32 bins

Figure 1e. Testing Image 3 with 8 bins

Figure 1f. Testing Image 3 with 16 bins

Figure 1g. Testing Image 3 with 32 bins

Training Image 1

Metric	Image 2, 8 Bins	Image 2, 16 Bins	Image 2, 32 Bins	Image 3, 8 Bins	Image 3, 16 Bins	Image 3, 32 Bins
Precision	0.685	0.628	0.638	0.746	0.630	0.625
Recall	0.542	0.254	0.173	0.430	0.178	0.124
F-Score	0.605	0.362	0.272	0.546	0.277	0.207

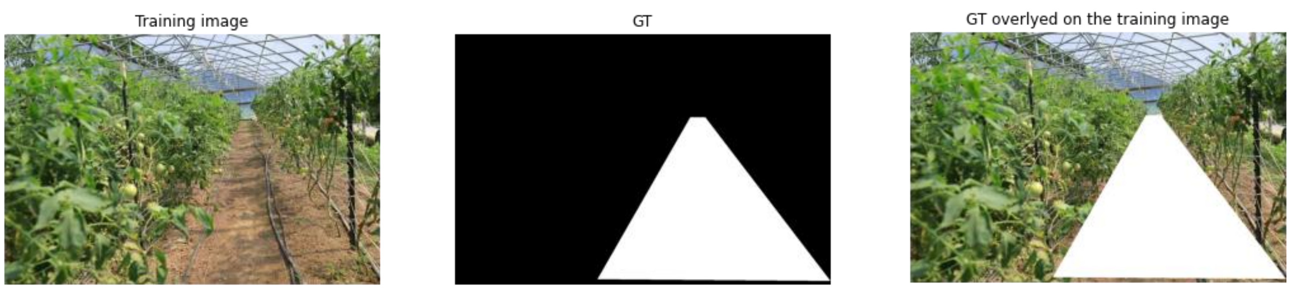

Figure 2a. Training Image 2

Figure 2b. Testing Image 1 with 8 bins

Figure 2c. Testing Image 1 with 16 bins

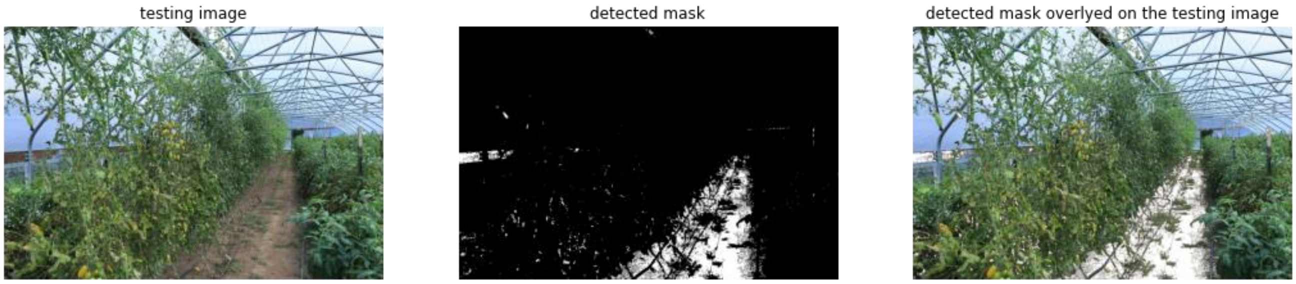

Figure 2d. Testing Image 1 with 32 bins

Figure 2e. Testing Image 3 with 8 bins

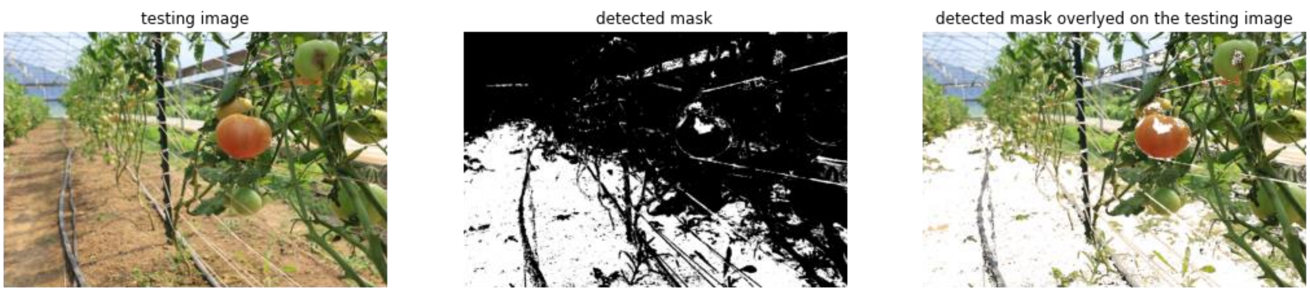

Figure 2f. Testing Image 3 with 16 bins

Figure 2g. Testing Image 3 with 32 bins

Training Image 2

Metric	Image 1, 8 Bins	Image 1, 16 Bins	Image 1, 32 Bins	Image 3, 8 Bins	Image 3, 16 Bins	Image 3, 32 Bins
Precision	0.821	0.836	0.838	0.840	0.866	0.882
Recall	0.264	0.452	0.501	0.641	0.708	0.709
F-Score	0.399	0.586	0.627	0.727	0.779	0.786

Figure 3a. Training Image 3

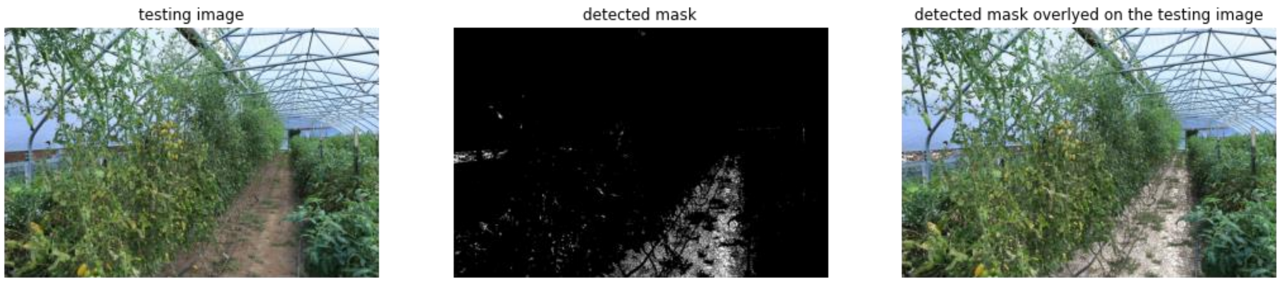

Figure 3b. Testing Image 1 with 8 bins

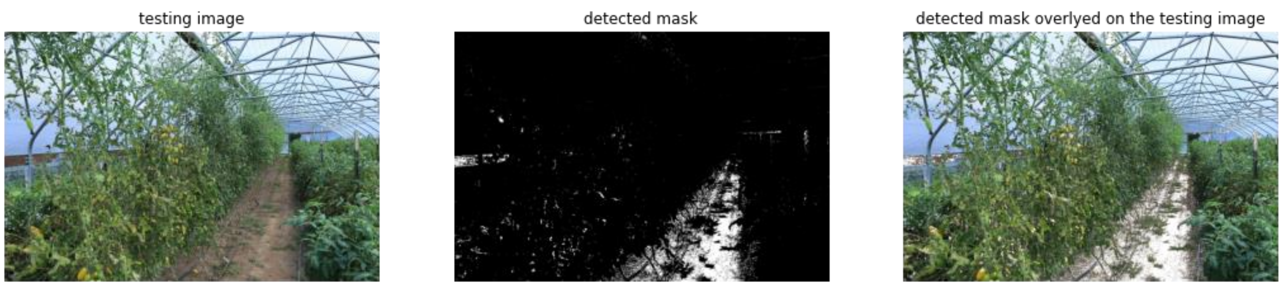

Figure 3c. Testing Image 1 with 16 bins

Figure 3d. Testing Image 1 with 32 bins

Figure 3e. Testing Image 2 with 8 bins

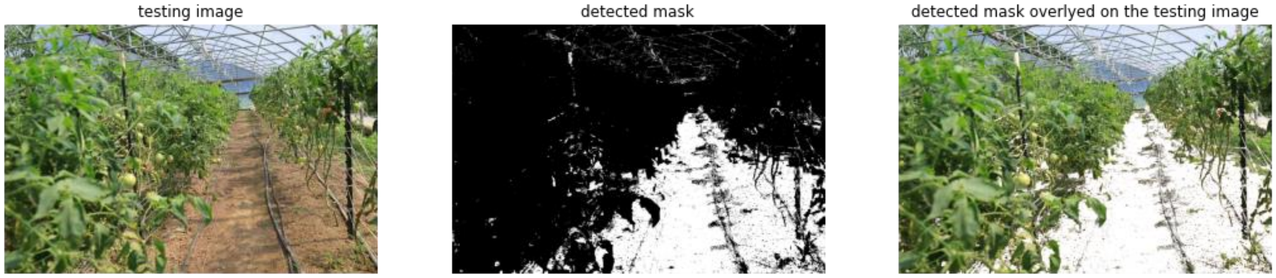

Figure 3f. Testing Image 2 with 16 bins

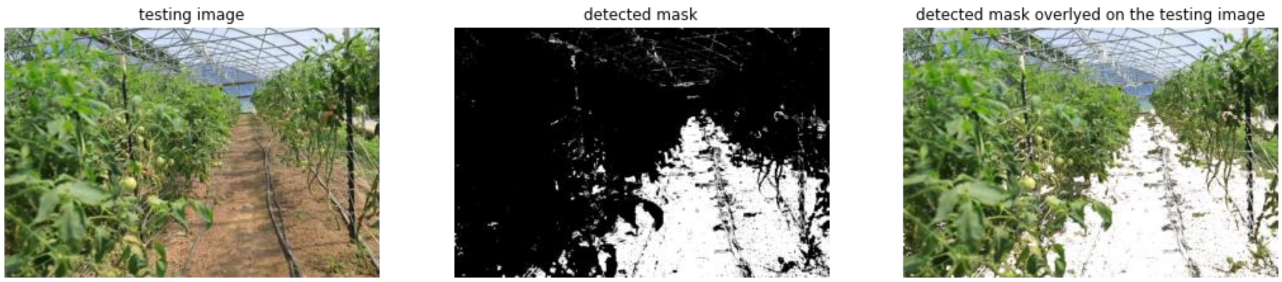

Figure 3g. Testing Image 2 with 32 bins

Training Image 3

Metric	Image 1, 8 Bins	Image 1, 16 Bins	Image 1, 32 Bins	Image 2, 8 Bins	Image 2, 16 Bins	Image 2, 32 Bins
Precision	0.831	0.798	0.700	0.765	0.728	0.723
Recall	0.265	0.405	0.348	0.566	0.684	0.662
F-Score	0.401	0.537	0.465	0.650	0.705	0.691

IV. Discussion

Regardless of the number of bins or the image that was trained, the resulting Bayesian Classifier will have a high precision, as none of the precisions found were below 0.6. However, recall varies more significantly depending on both the image that was trained, and the number of bins that were used. Specifically, in image 1, 8 bins was optimal as any value greater resulted in a sharp decline in recall, while in image 2, an increasing number of bins led to an increase in recall, and results peak at 16 bins for image 3. It should be noted that the ground texture in image 1 is noticeably darker than those in images 2 and 3, and as a result, when the number of bins increased, the model increasingly treated the darker texture as ground and the lighter texture as not ground, leading to more false negatives and a lower recall. Conversely, image 2 features both lighter and more varied ground, meaning that more bins would account for the minute differences in color, leading to fewer false negatives and a higher recall. As for image 3, it is possible that too few bins resulted in not enough discrepancy to account for features like the dark cable, but too many bins meant the model would skew towards the dark cable as ground texture, leading to false negatives when it comes to testing.

Objectively speaking, the best image to train is image 2 with 32 bins, as it produces the highest average F-Score of (0.627 + 0.786)/2 = 0.7065, but in reality, more images would be needed on both the training and testing side of the project. Specifically, one would probably train two images and test on one because more training data helps to spot differences in ground texture like those seen between the images used here. Nonetheless, the Bayesian Classifier has an overall mediocre performance with regards to predicting ground texture.