**Electrical Engineering Report**

**Project 1: Machine Vision in Agricultural Robots**

**Task 1**

In the first task we were required to perform histogram equalization on each image. We first converted our RGB images to HSV with the rgb2hsv() command before performing histeq() on them. After performing histogram equalization on these HSV images, we converted them back into RGB images. We faced some issues of incorrect colors being displayed. We went back to our HSV images prior to implementing the histeq() command on them. We took only the “value” portion of our HSV image and performed histogram equalization on it. We took these HSV images and converted them back to RGB and this solved the issues we were facing.

**Task 2**

The second task requires preparation of a mask to remove the background of all images. First, we assigned the ‘R’, ‘G’, and ‘B’ channels of the processed images to individual variables, then we calculated all the vegetation indices from these channels. We used the “graythresh” function to find the best threshold for ExGR. After converting the processed images to grayscale, we subtract the “G” channel of the processed images by the grayscale images and convert the result to black and white with a value of 0.075. Then, we concatenate the black-and-white images to convert them into 3d images so that we can multiply them with the processed images of the first task. From this, we obtain the images with the background removed.

**Task 3**

The third task requires the determination of crop lines. First, we divide the image into two halves for line1 and line 2. We calculate the max in each interval which gave us the peaks in numerical form. We used the find function to get us the corresponding x values on which the peak occurs for each plant lane.

Since multiple peaks occurred, we used the max function to get the highest peak and summed it all up. We took the sum of all the highest peaks (greenest part of lanes) and divided it by the total number of peaks we got to get the average line for each plant lane. Lastly, we took these two lines and plotted them onto the im\_processed2 image and the result was that we got two lines that hovered smoothly over each line plot, with minimal deviation.

**Task 4**

The fourth task required us to calculate the total number of plants in all the image files. This was tricky since the same plants were being shown in the next image and some plants were jammed together. To tackle this, we created an algorithm to pinpoint plant locations and applied this algorithm for both lane 1 and lane 2 (left and right).

We built upon the ideas and results from task 3. Firstly, we divide the image in half from the middle to isolate lane 1 and lane 2 and perform column wise summation on it by using the sum ()’ function. After that we identify the max of each line by using the max () function. We have now obtained the maximum value of the columns of each lane which we put into two variables named plant\_line\_1 and plant\_line\_2. The idea was that as each image goes to the next, the plant line value changes. If no plant were in the next image, or the value itself decreased, it meant that there is no plant. If the value increases it mean that image identified a lot of values which were green. This meant that there was another plant.

Next, we averaged out how much greenery was in one pic, which we got to around 240. If plant lane value dropped below this, we knew that there was no additional plant in the next image, if it increased, it meant that there was a plant. The lines shown in each image pinpoint each plant rather accurately.

We took it one extra step further and counted all the plants which came out to be 50 +- 1. Our algorithm got it to be 52 plants.