

Journal Report 2

9/16/19-9/20/19

Steven

Computer Systems Research Lab

Period 1, White

Daily Log

Monday September 16

Today, I started using the canny edge detection library. Played around with the arguments of method `cannyEdgeDetector()`. Was able to get a jpg output to a folder on my computer using openCV's `imwrite`.

Tuesday September 17

Decided on the parameters of the edge detection method. Changed sigma to 1.4, kernel size to 5, low threshold to 0.09 and high threshold to 0.17. With these parameters, I was able to output clean images that were processed through canny edge detection.

Thursday September 19

Worked on cleaning up these images further by drawing contours using openCV's `findContours()`, and getting rid of unnecessary edges, such as leaf textures. Saved these images to the same folder as the edge detection-processed images. When running, the program will output both the image and the contour counterpart in jpg form.

Timeline

Date	Goal	Met
Today minus 2 weeks	Get a start on the code in any form	Yes, able to start on a method
Today minus 1 weeks	Finish image processing and start edge detection code	Yes, found a canny edge detection library and started to use it
Today	Finish edge detection on all images gathered	Yes, was successful in creating clean images that were run through edge detection, and outputting them as jpgs.
Today plus 1 week	Gather height data from the edge detection and detect growth over all the images collected	
Today plus 2 weeks	Start on the web application design to implement the data processing and edge detection into	

Reflection

I made a great deal of progress this week as well. I managed to draw the actual lines for the canny edge detection, which will be critical for the data analysis and height detection that I am to perform next week. Not only was I able to find all the edges on my plant images, I did a little extra and cleaned up the image with contours so that there are less edges to deal with. After all, all I need is the height of the plant, and the height of the reference (which is a fertilizer bottle). What I plan to start next week is converting the output to python's pickle files, which helps in serializing and de-serializing output, and allowing data to be transferred across different code. This will make it much easier to store the data of all the edges for my data analysis instead of having my code re-decipher a jpg image.

The one issue I will have to fix before next week's goal is the same MemoryError I encounter when trying to run through all the training set images. I was able to get through seven before the machine stopped me. My solution that I devised but didn't have a chance to implement was only running a few images at a time, so that the memory doesn't run out while executing, and the output would eventually be the same as if I ran through all the images at the same time (I have about 40 images that I took).