Name: Socheath Sok

ID: 014470701

Class: EE 483

Professor: Kip Haggerty

Summary Report for Project 4

The objective of this project is to explore image segmentation which is a process of dividing an image into different parts for both grayscale and color images. For the thresholding method, the first two questions deal with superimposing a text image onto another image and determining the appropriate threshold to isolate the text. Using the imtool() function to see the intensity at each pixel was helpful in finding the right threshold. When the text is close to white, a single threshold was sufficient to remove the background. However, when the text is in black, it overlapped with some parts of the background, so double thresholding was needed to get the same result. As for the adaptive thresholding method, the image used has different shades of grays so it was not possible to completely separate all of the circles from the background. Using a 256x32 block, the majority of the circle was kept while a 256x8 block manages to perform better by saving a portion of the circle in the upper left corner that was in the darkest shade of gray.

The rest of the remaining questions deals with edge detection methods. Other than the Laplacian method which requires filtering and binarizing the image, all of the other techniques were done using the edge() function with the parameters mostly the same for better comparisons. With the cameraman image, all of the methods used were able to detect the foreground of the images, but only a few can fully detect the edges in the background. Overall, the Canny method seems to perform best at detecting all parts of the images, but too much detection like on the field of grass can make it hard to make out the details. When noise was added to the Arch image, all edge detection techniques perform poorly. With the salt and pepper noise, the Robert technique performs the worst showing mostly noise and barely any edges. The Prewitt method seems to be the best because the shape of the arch can be seen better than in other images. With the Gaussian noise, edge detection works slightly better with the Marr-Hildreth technique because it was able to show the complete structure of the Arch despite being in the presence of a lot of noise. The Canny method, on the other hand, performs the worst by showing only the center the Arch surrounded by a lot of uncomprehensible details.

To perform edge detection on the intensity component, the color image needs to be converted to grayscale first. When applying all the techniques, the Canny method detected the most details, but to get it close to Figure 13.23 which is the result of applying edge detection on the RGB components, a few additional parameters are needed. After testing many different values, the closest result to the figure was achieved using the threshold from 0.1 to 0.2 and a sigma of 0.5. Adjusting the sigma value, which is the standard deviation of the filter, by slightly can have a big impact on the amount of edge detection. The value 0.5 seems to be a good number for detecting a bit more details than the wanted result. Modifying the threshold helps remove small details from different parts of the image. The final result is very similar to the figure with only a bit of missing details in the lower center of the image.