Programming Assignment-2: CSL536

Note: 15 *Marks.* Any Five can be done

Q1. Experiment with using either top hat or bottom hat as a preprocessing before applying global thresholding to the image morf_test.png. This image contains letters on a very noisy background with varying illumination. Try to compute a good estimate of the background, and subtract the estimated background from the original. Then try to use global thresholding on the resulting image. Do you also need a noise filter? Can you segment out the numbers fairly well? Also consider if there are gaps or connected symbols that you can improve by applying simple binary morphological operations.

Q2. Segment the given images (white-flower.png, and butterfly.jpg) using Kmeans, mean shift clustering, and Otsu's thresholding approach and compare the results. Don't use built in functions apply hand coded function.

Q3. Detect corners in the given chessboard image using Harris Corner detector and SIFT Feature detector.

Q4. Develop a rgb2hsi function to transform red-green-blue (RGB) color values into hue, saturation, and intensity (HSI). The function prototype should be as follows:

function [H, S, I] = rgb2hsi(OCI)

where OCI represents the original color image, while H, S, and I denote the normalized hue, saturation, and intensity values within the HSI color space, respectively. It is essential that H, S, and I fall within the range of [0, 1]. Proceed to load and convert the image camel.png into the HSI color space by invoking the rgb2hsi function. Present the three resulting images in figures 1 to 3, ensuring that each figure is appropriately titled. Additionally, utilize a suitable OpenCV/Matlab function to perform a similar conversion and display the resulting Hue, Saturation, and Intensity images in figures 4 to 6, complete with appropriate titles. Furthermore, illustrate the difference images between your outcomes and those produced by OpenCV/Matlab in figures 7 to 9, again with suitable titles. Provide an explanation for the observed differences and the visual discrepancies between your results and those generated by OpenCV/Matlab.

Q5. Develop a program to match an object in two specified images utilizing the Feature Matching technique. Use the built-in SIFT algorithm/descriptor for feature detection and employ a brute-force method for feature matching. Test the program using the images image1.png and image2.png. (if you are using OpenCV, please ensure to install "pip install opencv-contrib-python" to access the built-in SIFT descriptor).

Q6. Develop a program that identifies moving vehicles through the application of the median differencing background subtraction method, and provide an analysis of your findings based on the results obtained. Utilize the traffic.mp4 video clip to evaluate your code.