Due on 7 March 2023

Computer vision - Assignment 4

1. Write a program to implement histogram equalization on an image. Write and implement a code that
   (a) displays the histogram of the input image,
   (b) implements histogram equalization as discussed in class,
   (c) displays the equalized histogram,
   (d) plots the transform function corresponding to the pair of input and output histograms and
   (e) displays the image corresponding to the equalized histogram.
2. Use the above code to apply histogram equalization on the images uploaded in the Google folder labeled *'Histogram assignment'*.
   (a) Compare your results with OpenCV's cv2.equalizeHist() to see if you get comparable results.
   (b) Is the intensity distribution of the equalized histogram a uniform distribution? Why or why not?
3. Download the image 'image1' and 'image2' from the Google folder
   (a) Extract the 3 color channels for each image and plot their respective histograms (so six in all).
   (b) For each channel, match the histograms of each colour channel of image1 with corresponding channel histograms of image2.
   (c) Reconstruct the new image1 in colour.
   (d) Compare the result with the original images.
4. Write a program to implement adaptive (i.e. local) histogram equalization (AHE). Your program should allow the user to choose a region for equalization and the neighbourhood size. Use a grayscale image form scikit's repository for this purpose.
   (a) Choose neighbourhood size $k \times k$, where k is a positive odd integer.
   (b) (SWAHE) Slide the neighbourhood horizontally over the center pixels and use the updating method (discussed in the lecture posted on Moodle) to carry out equalization. Demonstrate your code on the image above.
   (c) Break up the image into 'blocks' or 'tiles' of suitable size and apply histogram equalization on each block separately. Now combine these blocks to create the entire image.
   (d) (CLAHE) Use the function cv2.createCLAHE on the above image to apply contrast limited AHE. Experiment with different clip limits.
   (e) Compare the results obtained by the above methods.
5. Pick a grayscale image, either from the given ones or from scikit's image repository. Perform intensity slicing on the image in two ways:
   (a) highlight a particular range and set the rest to zero;
   (b) highlight a particular range and leave the rest unchanged.

   In each case, plot the intensity transformation being applied. Display all results for comparison.
6. Perform bit-plane slicing on the same grayscale image used in Exercise 5.
   (a) Display all the bit planes, labeling them as 'bitplane0'(corresponding to the least significant bit) to 'bitplane7'(corresponding to the most significant bit).
   (b) Plot the intensity transformation being used to extract bitplane0, bitplane3 and bitplane7.
   (c) Try to reconstruct the image with fewer bit planes so as to visually appear as resembling the original. What is the minimum number of bit planes the you can use without degrading the image quality too much?