Due on 25 February 2024

Computer vision - Assignment 4

1. Download the images 'image1' and 'image2' from the Google folder labeled *'Histogram assignment'*.
   (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. Use cv2.equalizeHist().
   (c) Reconstruct the new image1 in colour.
   (d) Compare the result with the original images.

2. Study this OpenCV documentation on adaptive (i.e. local) histogram equalization (AHE). The following exercises demonstrate some AHE techniques. Use the image "Chestxray1.png" included in the above-mentioned folder for the following exercises.
   (a) (CLAHE) Use the function cv2.createCLAHE on the above image to apply contrast limited AHE. Experiment with different clip limits.
   (b) (SWAHE) Write a code to implement Sliding Window AHE. Choose neighbourhood size $k \times k$, where k is a positive odd integer. Slide the neighbourhood horizontally over the center pixels and use the updating method as mentioned in the notes. Alternatively, you may refer to the original paper. 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) Compare the results obtained by the above methods. You may also experiment with other images of your choice. Write your observations about when each method preferable?

3. 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.

4. Perform bit-plane slicing on the same grayscale image used in Exercise 3.
   (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?