Due on 28 February 2020

M.Sc. Data Science

Computer vision - Assignment 3

In this assignment we will learn how to generate and work with histograms.

1. Write a program to implement histogram equalization on an image. You may compare your results with OpenCV's cv2.equalizeHist(). Include a code to display the intensity transformation. Use it on the two images 'chestxray' and 'HawkesBay' uploaded in the Google folder labeled *'Histogram assignment'*.
2. 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.
3. Work with the image 'crayons.jpg'.
   (a) Apply histogram equalization to the 3 color channels (R, G, B) separately and recombine. Save the result.
   (b) Convert the image to HSV colour space. Extract the V (value) channel and apply histogram equalization to this channel only. Combine with the unchanged H and S channels.
   (c) What is the difference between the image obtained this way and the image obtained in part (a)?
4. Write a program to implement adaptive (i.e. local) histogram equalization. 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 class) to carry out equalization. Demonstrate your code on image1 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.

   Display all results for comparison.
6. Perform bit-plane slicing on the same grayscale image used in Exercise 4. Display all the bit planes, labeling them as 'bitplane0'(corresponding to the least significant bit) to 'bitplane7'(corresponding to the most significant bit). Try to reconstruct the image with fewer bit planes so as to visually appear as resembling the original.