

COMP9517: Computer Vision

2020 T2 Lab 1 Specification

Maximum Marks Achievable: 2.5

This lab is **worth 2.5% of the total course marks.**

The lab files should be submitted online.

Instructions for submission will be posted closer to the deadline.

Deadline for submission is Week 3, Wednesday 17 June 2020, 23:59:59.

Objective: This lab revisits important concepts covered in the lectures of Weeks 1 and 2 and aims to make you familiar with implementing specific algorithms.

Materials: The sample image to be used in all the questions of this lab is available in WebCMS3. You are required to use OpenCV 3+ with Python 3+. Jupyter notebook files are preferred for submitting your code.

Submission: The last question (Question 4) is assessable **after the lab**. Submit your code and results in a zip file via WebCMS3 by the above deadline.

Contrast Stretching

Contrast in an image is a measure of the range of intensity values and is the difference between the maximum and minimum pixel values occurring within the image. The full contrast of an 8-bit image is $255 (\text{max}) - 0 (\text{min}) = 255$. Anything less than that means the image has lower contrast than possible. Contrast stretching attempts to improve the contrast of the image by stretching the range of intensity values using linear scaling.

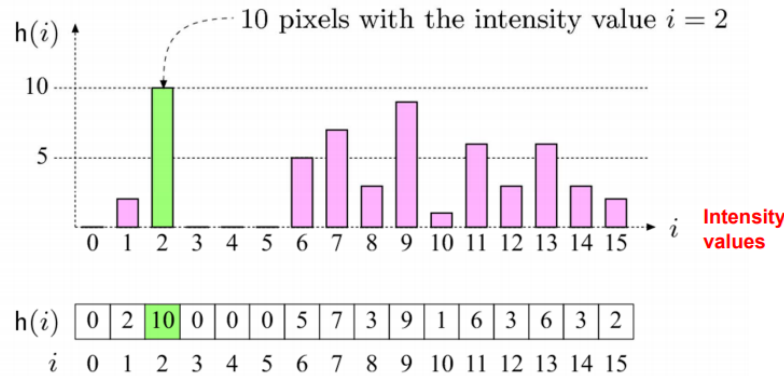
Assume that I is the original input image and O is the transformed output image. Let a and b be the minimum and maximum pixel values allowed (for an 8-bit image that means $a = 0$ and $b = 255$). And let c and d be the minimum and maximum pixel values occurring in I . Then the contrast stretched image O is given by the function:

$$O = (I - c) \left(\frac{b - a}{d - c} \right) + a$$

Question 1: Write a function that performs contrast stretching. Read the given gray-scale image and apply your function to test whether it indeed improves the image quality.

Intensity Histogram

The histogram of an image shows the frequency of pixel intensity values. It gives statistical information and removes the location information of the pixels. For a digital image with gray levels from 0 to $L - 1$, the histogram is a discrete function $h(r_k) = n_k$ where $r_k \in [0, L - 1]$ is the k th gray level and n_k is the number of pixels with gray level r_k .



Question 2: Write a function that computes and plots the histogram of the given image.

Image Edges

Edges are an important source of semantic information in images and they occur in human visual perception at divisions between different areas of intensity, colour, and texture. A gray-scale image can be thought of as a 2D landscape with areas of different intensity living at different heights. A transition between areas of different intensity in an image I means there must be a steep slope which we formalise as the gradient:

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

Since the image I is discrete, we need to approximate the continuous derivatives $\partial I / \partial x$ and $\partial I / \partial y$ by finite differences. Simple examples of convolution kernels that perform finite differencing are the Sobel filters defined as follows:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

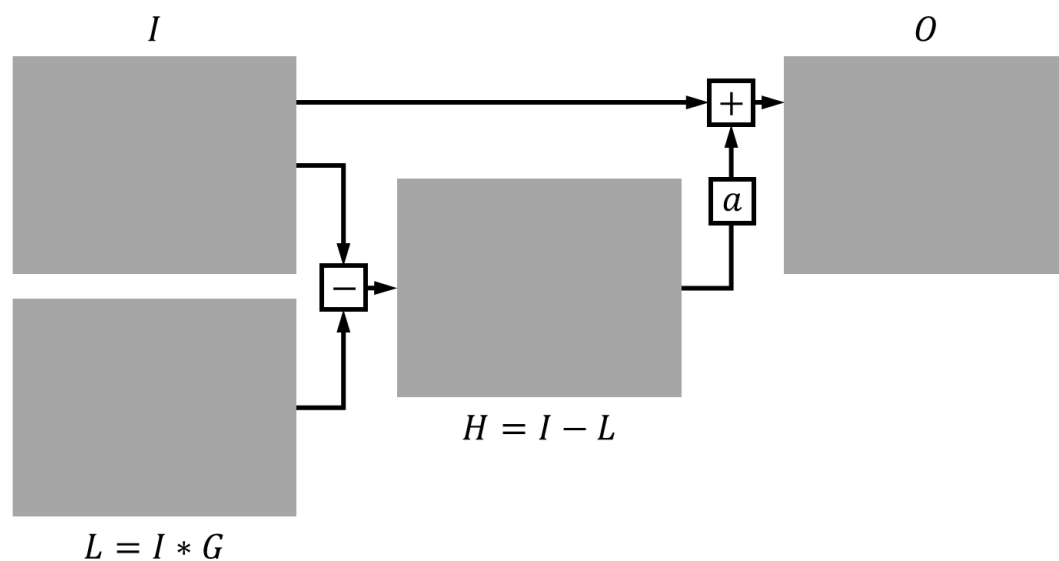
Question 3: Write a function that computes the two gradient images $\partial I / \partial x \approx I * S_x$ and $\partial I / \partial y \approx I * S_y$ from the given image.

Notice that the OpenCV built-in Sobel functions can also be applied to achieve this result. But the challenge is to implement your own functions. You can verify the result of your own functions by comparing with the result of the built-in functions.

Image Sharpening

Sometimes images are not as sharp as they could be, and fine details look a bit blurred. An image can be sharpened by using specific filters that enhance the high-frequency content in the image. One possible technique to accomplish this is called unsharp masking.

The figure below gives a schematic overview of this technique. In words, it takes the input image I and convolves it with a Gaussian kernel G with standard deviation or scale σ , resulting in a slightly blurred image L . Next, the blurred image L is subtracted from the input image I , resulting in image H , in which the high frequencies are enhanced. Then, each pixel in H is multiplied by a constant factor a , and the resulting image is added pixel-wise to the input image I , resulting in output image O .



Question 4 (2.5 marks): Write a function that implements the above technique. Apply it to the given image using, for example, $\sigma = 1.0$ pixels and $a = 1.25$. The differences between I and O are subtle but the latter should be visibly sharper.

Notice that this technique may produce output pixel values outside the range $[0, 255]$. Thus, make sure you use the right data types for the computations. Also, you can apply contrast stretching (which you implemented in answer to Question 1) to force the contrast in I and O to be in the same range $[0, 255]$, so you can properly compare the two images visually.

Copyright: UNSW CSE COMP9517 Team

Released: 11 June 2020