COMP9517

Lab 1, T3, 2020

The lab files should be submitted online.

Instructions for submission will be posted closer to the deadline.

Deadline for submission is Week 3, Monday 28 Sept 2020, 23:59:59.

Objectives: This lab presents a revision of important concepts from week 1 and 2 lectures. Most questions require you to use OpenCV, an open source software package that is widely used in this field.

Materials: The sample images to be used in all the questions of this lab are available in WebCMS3. You are required to use OpenCV 3+ with Python 3.

Submission: Question 5 is assessable **after the lab** and is **worth 2.5% of the total course marks**. Submit your code and results as a Jupyter notebook in a zip file via WebCMS3 by the deadline.

1 Contrast Stretching

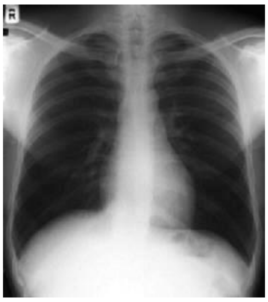
Contrast in an image is a measure of the range of intensity values within an image and is the difference between the maximum and minimum pixel values. The full contrast of an 8-bit image is 255(max)-0(min)=255, and anything less than that results in a lower contrast image. Contrast stretching attempts to improve the contrast of an image by stretching (linear scaling) the range of intensity values**.**

Assume that ***Or*** is the original image and ***Tr*** is the transformed image. Let **a** and **b** are the min and max pixel values allowed in an image (8-bit image, a=0 and b=255), and let **c** and **d** be the min and max pixel values in a given image, then the contrast stretched image is given by the function:

$$\boldsymbol{Tr} = (\boldsymbol{Or} - \boldsymbol{c})\left(\frac{\boldsymbol{b} - \boldsymbol{a}}{\boldsymbol{d} - \boldsymbol{c}}\right) + \boldsymbol{a}$$

QUESTION 1: Read a grey scale image (imageQ1.jpg) and perform contrast stretching to improve the quality of the image. Example shown below is a poor contrast X ray image and its contrast stretched version.

Original Image



Contrast Stretched image

2 Histogram

Histogram of an image shows the frequency of pixel intensity values. It only gives statistical information and nothing about the location of the pixels. For a digital image with grey levels from 0 to L-1, the histogram is a discrete function $\boldsymbol{h(Or_k) = n_k}$, where $\boldsymbol{Or_k}$ is the kth grey level and $\boldsymbol{n_k}$ is the number of pixels with a grey level $\boldsymbol{r_k}$.

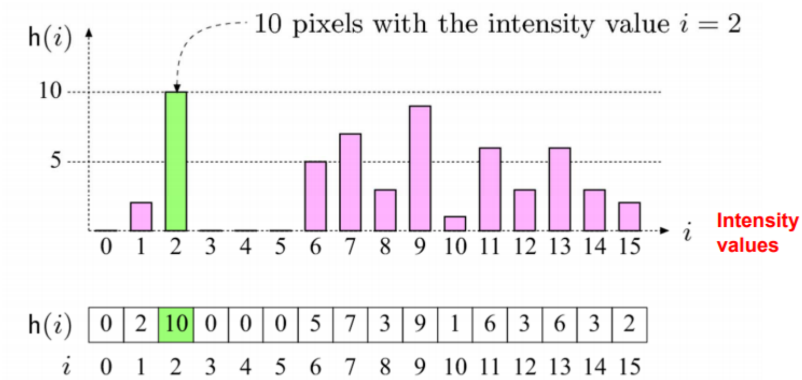
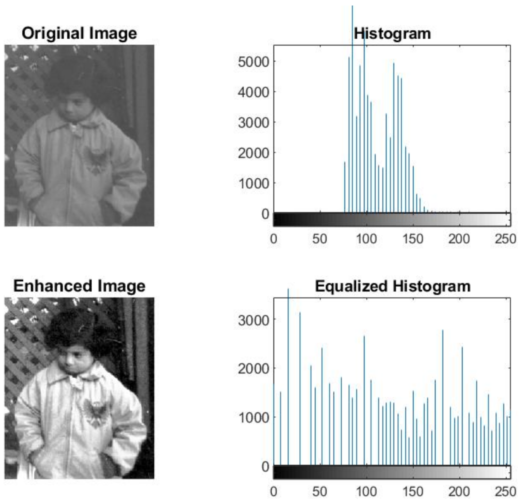


Figure 1: Histogram (Picture from [1]).

2.1 Histogram Equalization

It is an image processing technique used to improve the contrast in images. This is accomplished by spreading out the most frequent intensity values. This allows regions having a lower local contrast to gain a higher contrast. Example shown below is an image and its enhanced version with their respective histograms.



QUESTION 2.1 Write a function that computes the histogram of a given grey scale image (imageQ21.jpg) and displays a plot. Better not use in-built OpenCV functions.

QUESTION 2.2 For a given image (imageQ22.tif), enhance it by applying histogram equalization.

3 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 brightness, colour, and texture. A grey scale image can be thought of as a 2D representation of heights and areas of different brightness levels at different heights. A transition between different areas of brightness in an image ***I***, means there must be a steep slope which we formalise as the gradient of

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

of the Image. Now our image ***I*** is discrete, so we approximate the continuous quantities $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ by finite difference kernels. A simple example of a finite difference kernel is the Sobel filter (F_x and F_y), which is the subject of the following question.

QUESTION 3: With the given image (imageQ3.jpg), use the Sobel operator to compute the image gradients at x and y directions. To do this, first define the 2D filters (F_x and F_y). Then perform convolution between the image and F_x to obtain the gradients at x direction, and similarly perform convolution between the image and F_y to obtain the gradients at y direction.

$$F\_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad F\_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Notes:

- The OpenCV built-in Sobel functions can be applied to achieve the result. This can be a way of verifying the gradient outputs.

4 IMAGE FILTERING

Image filtering is a neighbourhood operation for modifying or enhancing an image, that can emphasise certain features or remove certain unwanted features. Image filtering is usually implemented as a convolution. Convolution is a mathematical operation which takes as input two functions (say ***f*** and ***g***) and produces an output function (say ***h***) which typically is a modified version of one of the original functions. In our case, ***f*** is the input image, ***g*** is the convolution mask/kernel, ***h*** is the filtered version of ***f*** and convolution creates weighted sums of the image pixels.

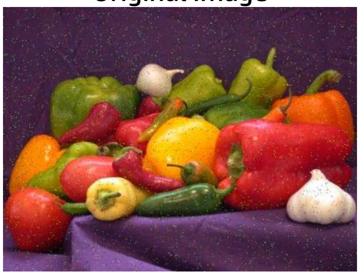
4.1 Smoothing Filters (Low-pass Filters)

Noise (random variations in intensity) removal from images is implemented using Image filtering. Salt and pepper noise, impulse noise and gaussian noise are some of the commonly observed types of noise. Different types of filters are suited for different types of noise. A smoothing filter is often called a low-pass filter as it allows low frequency components of the image to pass though (regions with similar intensity values) but stops high frequency components from passing through (edges or noise). Shown below are the results of applying different filters on an image corrupted by salt and pepper noise. It can be observed that the median filter produces the best result in this case.


Original Image


Mean filter (size:3*3)


Mean filter (size:5*5)


Noisy Image

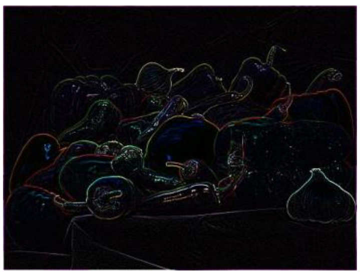

Median filter (size: 3*3)


Gaussian filter (size:3*3, σ =1.5)

4.2 Image Sharpening (High-pass filters)

Image sharpening is used to highlight fine details in an image or to enhance features that are blurred. High pass filters allow high frequency components such as edges and noise to pass through, while blocking low frequency components. Shown below are the results of applying a high-pass filter and the resulting sharpened image.


Original Image


High-pass filtered Image

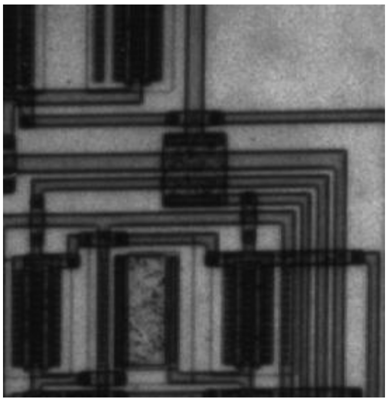

Sharpened Image

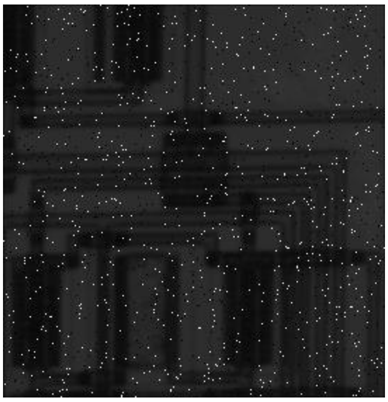
QUESTION 4.1: Implement a mean filter, median filter and a gaussian filter. Perform noise removal on the given image (imageQ41.png). Try with different filters and kernel sizes, observe the variations, and find the best filter and kernel size for the noisy images.

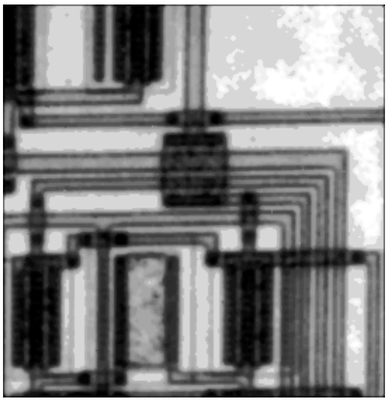
QUESTION 4.2: Implement a high-pass filter and perform Image sharpening on a given image (imageQ42.png). (A high pass filtered image can also be obtained by subtracting the low pass filtered image from the original image).

5 Restoring Images

Image restoration is the task of taking a corrupted image and trying to estimate the original image. Corruption can be caused due to motion blur, noise or an out of focus camera. The restoration task tries to restore the image information that was lost due to the corruption. Shown below is an example of a corrupted image restored by applying noise removal and other techniques discussed in the sections above.


Original Image


Corrupted Image


Restored Image

Assessment Question (2.5 marks)

QUESTION 5: Given a corrupted image (imageQ5.png), perform image restoration using techniques discussed in sections 1 – 4 to estimate the original image from its corrupted version.

6 REFERENCES

[1]. http://web.cs.wpi.edu/~emmanuel/courses/cs545/S14/slides/lecture02.pdf