COMP9517

Lab 1, T1, 2021

The lab files should be submitted online.

Instructions for submission will be posted closer to the deadline.

Deadline for submission is Week 3, Monday 1st March 2021, 16:59:59 AEST

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 4 is assessable AFTER THE LAB** and is worth 2.5% of the total course marks. Please submit your Jupyter notebook with the solutions, intermediate steps and final output images via webCMS3 by **the deadline.** Submission link will be opened just prior to the lab session.

The sample image "cattus.jpg" is to be used for all four questions. The output image of some questions may be used as the input to other questions.

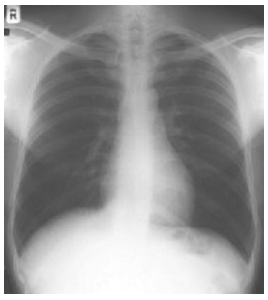
1 Contrast Stretching

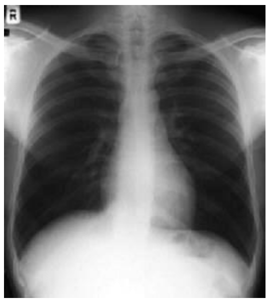
Contrast in an image is a measure of the range of intensity values within an image and is the difference between the maximum and minimum pixel values. The full contrast of an 8-bit image is 255(max)-0(min)=255, and anything less than that results in a lower contrast image. Contrast stretching attempts 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 (cattus.jpg) and perform contrast stretching to improve the quality of the image. Example shown below is a poor contrast X ray image (left) and its contrast stretched version (right)


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 $\boldsymbol{h}(\boldsymbol{Or}_{\boldsymbol{k}}) = \boldsymbol{n}_{\boldsymbol{k}}$, where $\boldsymbol{Or}_{\boldsymbol{k}}$ is the kth grey level and $\boldsymbol{n}_{\boldsymbol{k}}$ is the number of pixels with a grey level $\boldsymbol{r}_{\boldsymbol{k}}$.

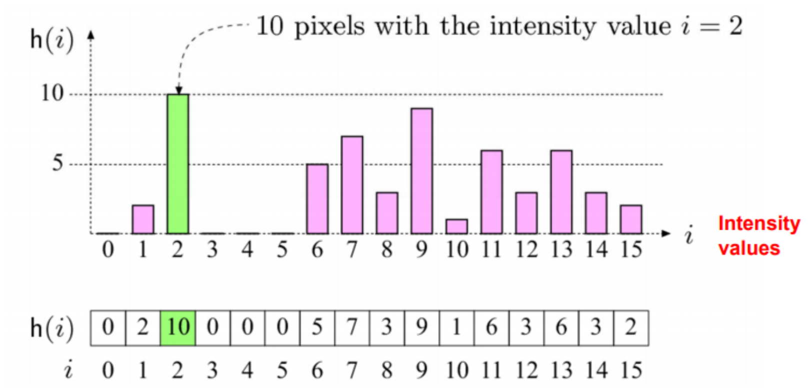
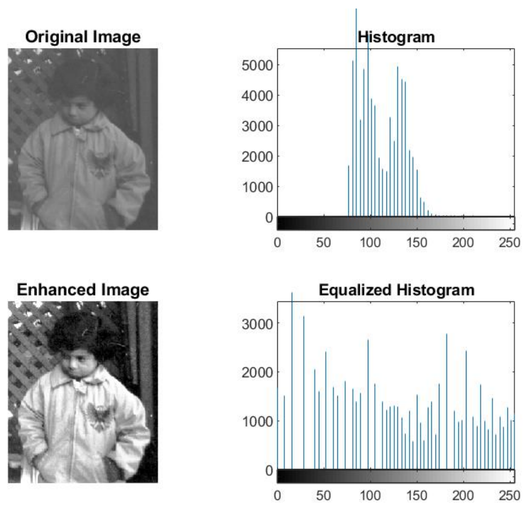


Figure 1: Histogram (Picture from [1]).

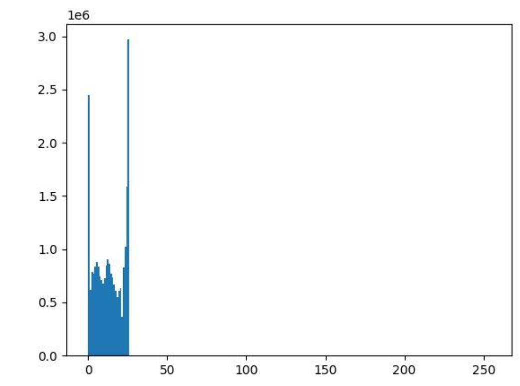
2.1 Histogram Equalization

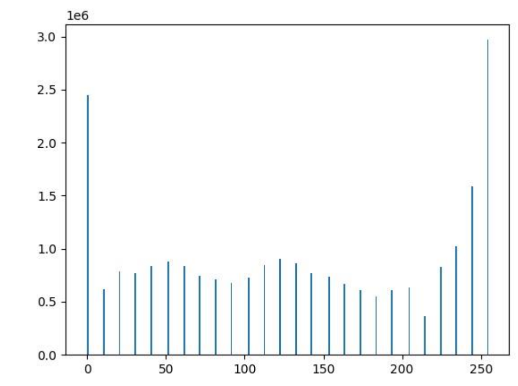
It is an image processing technique used to improve the contrast in images. This is accomplished by spreading out the most frequent intensity values. This allows regions having a lower local contrast to gain a higher contrast. Example shown below is an image and its enhanced version with their respective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

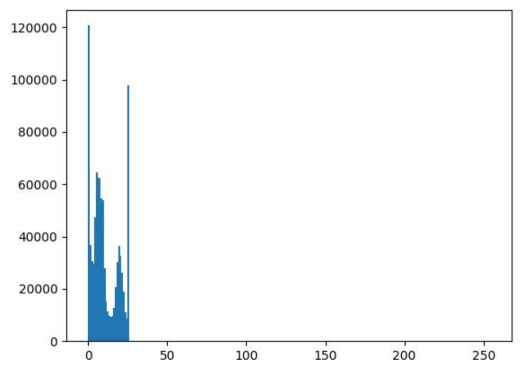


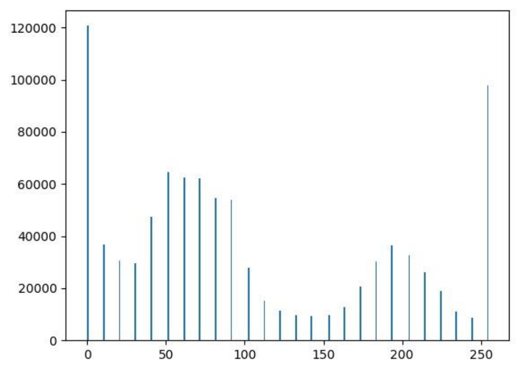
QUESTION 2.1 Write a function that computes the histogram of the given grey scale image and displays a plot. Do NOT use built-in OpenCV functions for this purpose.

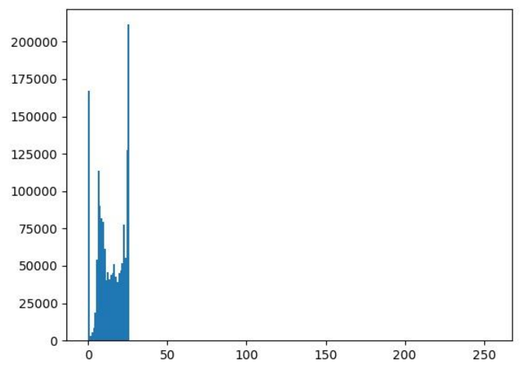
QUESTION 2.2 Use this function to now create and plot a histogram for the original image and the contrast stretched image (output of question 1). Match your calculated histogram for the contrast stretched image with the samples provided below. Which is the closest match?

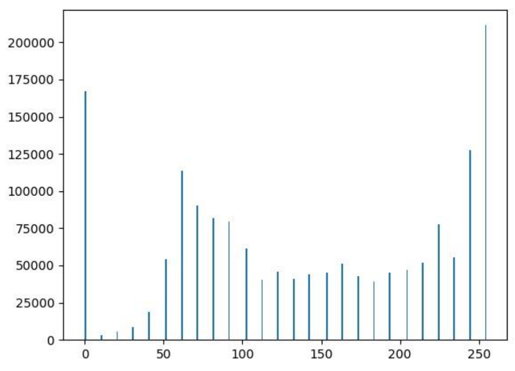










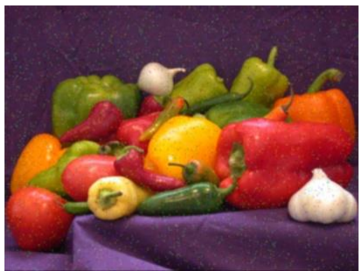


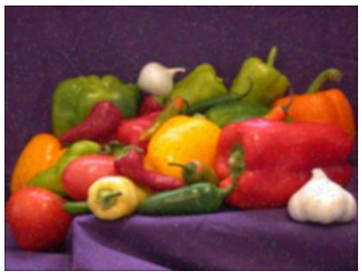
3 Smoothing Filters (Low-pass Filters)

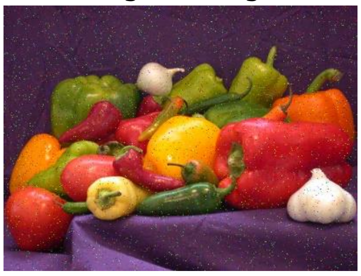
Noise (random variations in intensity) removal from images is implemented using Image filtering. Salt and pepper noise, impulse noise and gaussian noise are some of the commonly observed types of noise. Different types of filters are suited for different types of noise. A smoothing filter is often called 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).

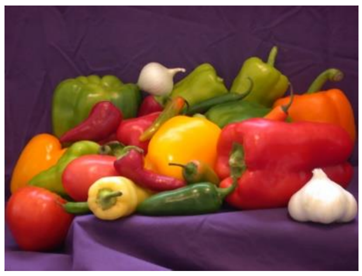
For example:

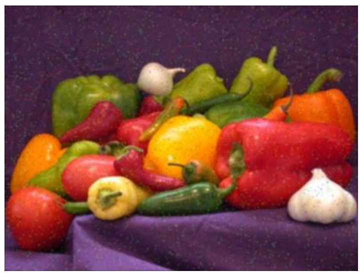
Original Image


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)

QUESTION 3: Implement a mean filter, median filter and a gaussian filter (without using OpenCV functions). Perform noise removal on the output image from question 1 using these filters. Then decide which filter performs best at removing the noise. Try with different filters and kernel sizes, observe the variations and find the best filter and kernel size for the noisy images.

4 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 live 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 above, 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.

Assessment Question (2.5 marks)

QUESTION 4: With the given image, use the Sobel operator to compute the image gradients at x and y directions. Note, that you should not use the built-in Sobel functions to do this, instead you should create the filters from scratch. 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. For this question, use the image that is the output of the median filter in question 3.

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

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

Note: The OpenCV built-in Sobel functions can be applied to achieve the same result. This is a way of verifying the gradient outputs computed by yourself.

5 REFERENCES

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