**CEN 472**

**Digital Image Processing Lab**

**Journal - Lab 7**

### Name:

**Enrollment #:**

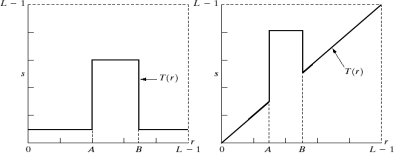
### Class:

**Objective**

The purpose of today’s lab is to have an insight into the gray intensity slicing, histogram equalization and contrast stretching.

# Gray-Level Slicing

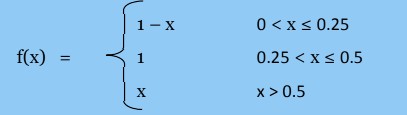
Highlighting a specific band (or range) of gray-intensities in an image is referred as gray-level slicing. The purpose of gray-level slicing is to assign more weight to certain details/information in an image for the purpose of analysis or to make them more visible in an image. Examples of typical gray level slicing can be seen in the following figure.



*Figure 1: Gray level slicing transformations*

The following two exercises will help you better understand the concept of gray level slicing.

# Exercise 1



Consider a function f(x) defined over an image with the intensities in

the range [0 1], as defined below:

1. Draw a graph for f(x) to show its influence on image intensities.
2. Write a Python program to implement f(x) on an input image. Also, show your result after transformation.

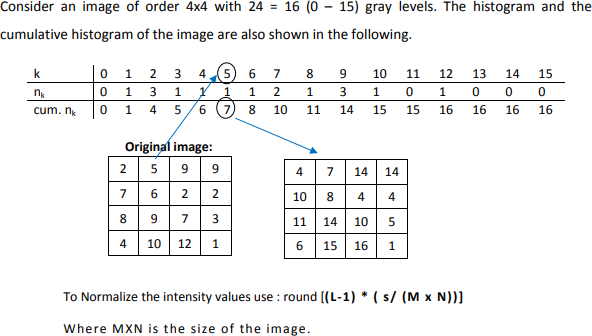
# Histogram Equalization

Histogram equalization is the process of re-allocating intensity values of the pixels in an image such that the output image contains uniform distribution of intensities defined by a monotonically increasing function T(r).

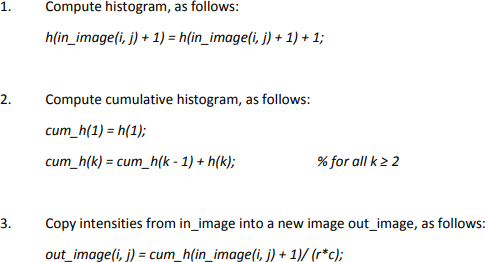
## Function in Python:

cv2.equalizeHist(gray): equalizes histogram of an input image in\_img for the specified gray levels L.

## How to Equalize a Histogram?



Algorithm to Equalize an Image Histogram



# Exercise 2



Write a function named ‘myhist\_equ’ to MANUALLY equalize histogram of an image

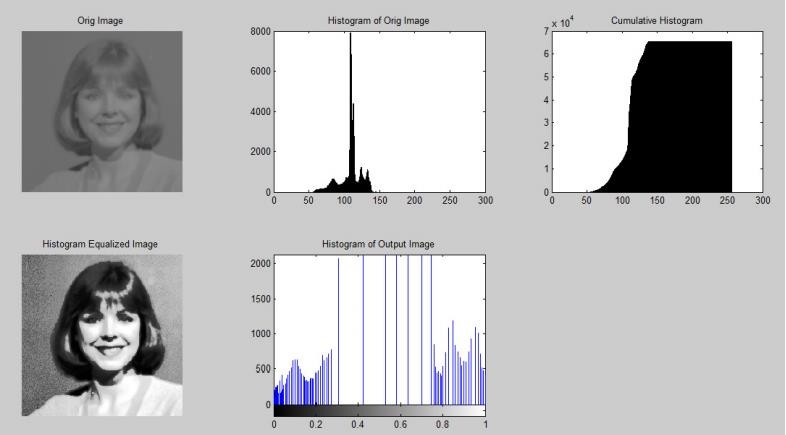
and map corresponding results:

INPUTS: image

OUTPUTS: Display the input image as well as equalized (resultant) image along with histogram before and after histogram equalization.

Note: Do not use the cv2.equalizeHist() function in Python Load the image

‘person.png’ and display the output as illustrated in Figure 2.



Contrast Stretching

*Figure 2:Histogram Equalization*

Contrast stretching attempts to improve an image by stretching the range of intensity values it contains to make full use of possible values. Unlike histogram equalization, contrast stretching is restricted to a linear mapping of input to output values. The result is less dramatic, but tends to avoid the sometimes artificial appearance of equalized images.

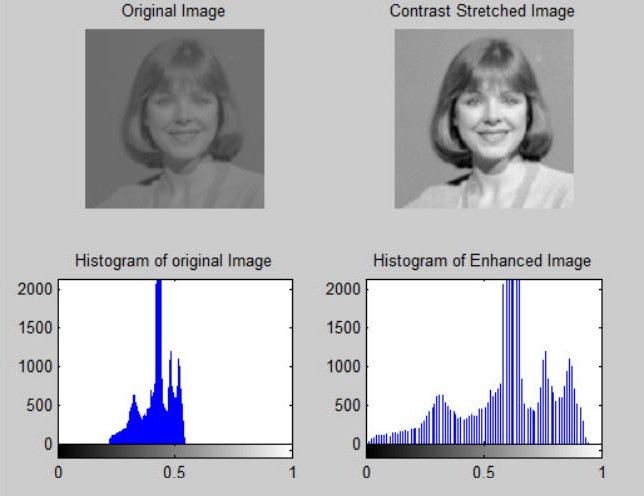
For each pixel, the original value r is mapped to output value s using the function:



Where rmin and rmax represent the minimum and the maximum gray values in the image, L-1 represents the maximum gray level in the image (for 8 bit images, L-1 = 255).

From the view point of implementation, convert the image to double, hence the factor L-1 = 1.

# Exercise 3





Write a function named ‘constrastStretch’ to stretch the contrast of an image.

INPUTS: image

OUTPUTS: Display the input image as well as the contrast stretched (resultant) image along with the histogram of each. Load the image ‘person.png’ and display the output as illustrated in Figure 3.

*Figure 3:Contrast Stretching*