LAB # 04: <u>Transformation Operations</u>

<u>Lab Objective:</u>

The objective of this lab is to implement thresholding on images to convert them to binary, perform different transformation operations on images.

<u>Lab Description:</u>

Transformation operations help enhance the quality of the image by applying operations like log, inverse log and power on the entire image. Different type of transformation yields different results.

Gray level transformation

There are three basic gray level transformation.

    I.   Linear
   II.   Logarithmic
  III.   Power – law

The overall graph of these transitions has been shown below.
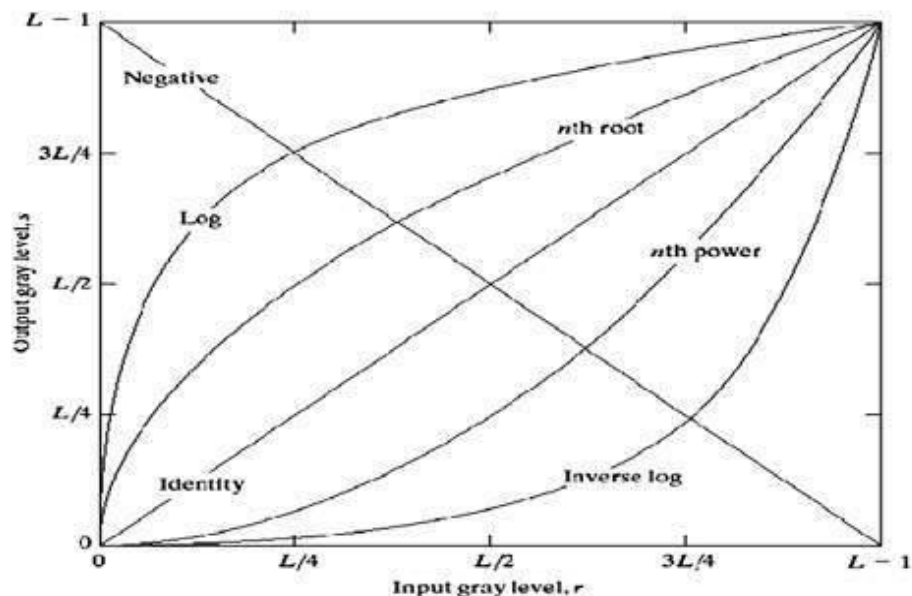
Transformation curves

Power law transformations: It includes nth power and nth root transformation. These transformations can be given by the expression:

s=cr^γ

This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.

Variation in the value of γ varies the enhancement of the images. Different display devices / monitors have their own gamma correction, that's why they display their image at different intensity.
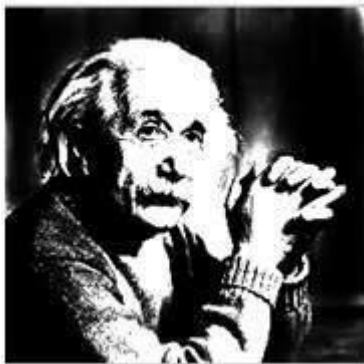
This type of transformation is used for enhancing images for different type of display devices. The gamma of different display devices is different. For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

Correcting gamma.

$$s=cr^\gamma$$

$$s=255*[(r/255)^\wedge \gamma]$$

The same image but with different gamma values has been shown here.

Gamma 8	Gamma 6	Gamma 10

Log transformations

The log transformations can be defined by this formula

$$s = c \log(r + 1).$$

$$c=255/\log(1+\text{max of image pixel})$$

$$s=c*\log(img+1)$$

Where s and r are the pixel values of the output and the input image and c is a constant. The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation.

| Input image | output of log transformation |

.

Negative Transformation

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image. This transformation is done by this formula

$$s = (L - 1) - r$$

So each value is subtracted by 255,so the lighter pixels become dark and the darker picture becomes light. And it results in image negative.

\

Negative transformation

Thresholding is the operation through which an image can be converted into a binary image/black & white i.e. having only two distinct levels. Threshold value can be the mean or median etc. of the image.

Gray level Slicing is used to highlight a specific range of gray levels in an image

Some Useful Commands:

1. To calculate the mean of 2D array using NumPy: my_mean = **numpy.mean (my_array)**

2. To calculate min (or max) of an array: **my_min = numpy.amin(my_array)**

3. To calculate the power of an array using NumPy: array_power = **numpy.power (my_array, power)**

4. To obtain percentile value. **percentile_array = numpy.percentile(my_array, percentile)**

5. To change data type of array. **my_array = my_array.astype(numpy.uint16)**

Lab Tasks:

1: Apply following transformation techniques on image provided and observe the output image of each transformation.

I.   Negative transformation
II. Log transformation

2: Apply the following transformation on an image.

	Pixel values range	Output pixel value
a)	Less than mean	0
	Greater than mean	255
b)	Less than mean	255
	Greater than mean	0
c)	±20 mean	0
	Otherwise	255

3: Apply Power Law transformation for the following values of γ (0.2, 0.5, 1.2 and 1.8) . Make sure to adjust data types accordingly.

4: Apply Gray level slicing using lower limit 100 and upper limit 200. Set all these values to 210.
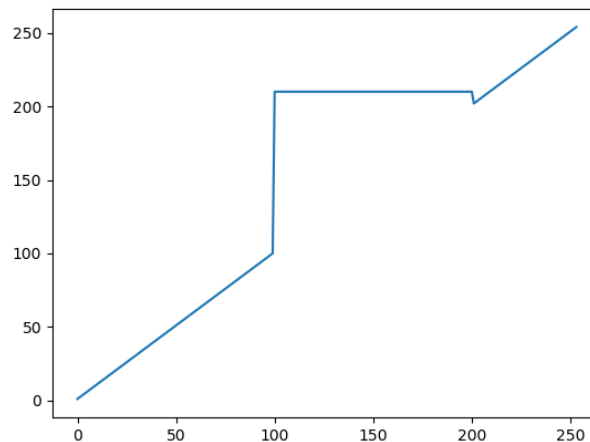
5: Calculate histogram of gray scale mage.(Don't use build in function) .Make your own algorithm**.**

Applications:

i.   Enhancement of image for different type of display devices.
ii.   Power-law transformation for enhanced recognition of born-digital word images
iii. Image Enhancement by Adaptive Power-Law Transformations

<u>References:</u>

I.   http://www.cs.uregina.ca/Links/class-info/425/Lab3/
II. https://www.tutorialspoint.com/dip/gray_level_transformations.html