

## Lab Assignment 7

### Intensity Transformations and Histogram Processing

- With each problem, required python libraries, functions are provided for your help while writing codes in Jupyter notebook. You can use any other libraries or functions as well.
- The theoretical details and concepts of all the problems have been covered in lectures which are uploaded on LMS

#### 1. Point/Intensity transformation: Negative of an image.

First print the maximum intensity value of input image and then subtract the intensity values from max value.

Jupyter Notebook:

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
im = Image.open("../parrot.png")
'?'          # print max value of the image here
im_t = ?      # You can use 'point' function to take negative of an image
im_t.show()
```

#### 2. Point/Intensity transformation: Log transformation

Jupyter notebook:

```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
im_g = im.convert('L')  # rgb to gray
im_tt = ?               # log transformation
# The brighter pixels from the input image have become darker, and the darker
pixels have become brighter
im_tt.show()
```

### 3. Histogram Processing

- a) **To plot the histogram of R,G,B channels of an image.** First split your RGB input image into three channels and then plot the histograms of these three R,G,B channels in same graph. In graph, x-axis: Pixel value, y-axis: Frequency of those pixels

Jupyter Notebook:

```
# To plot histogram
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
im = Image.open("path")      # RGB input image
im_r, im_g, im_b = im.split() # To split the RGB image into 3 channels
plt.style.use('ggplot')
plt.figure(figsize=(15,5))
plt.subplot(121)
plt.imshow(im, cmap='gray')
plt.title('original image', size=20)
plt.axis('off')
plt.subplot(122)
# Use the function here to plot histogram of im_r
# Use the function here to plot histogram of im_g
# Use the function here to plot histogram of im_b
plt.xlabel('pixel value', size=20)
plt.ylabel('frequency', size=20)
plt.title('histogram for RGB channels', size=20)
plt.show()
```

#### b) Histogram Equalization

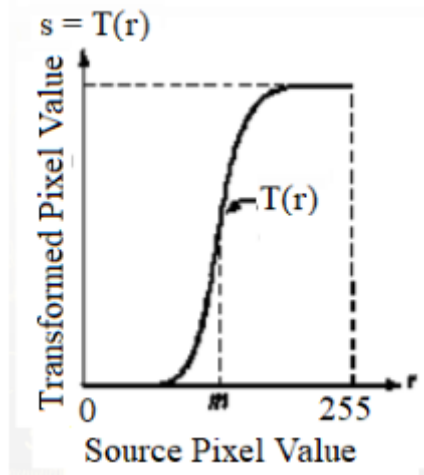
For input image, show 4 subplots: original image, its histogram, new image after equalization and its histogram plot.

Jupyter Notebook:

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from skimage import data, img_as_float
from skimage import exposure, io
# 'exposure.equalize_hist' function can be used for histogram equalization
```

#### 4. Contrast Stretching:

Write a program for contrast stretching of an RGB image and then draw the histograms of R,G,B channels of output image. The following section describes how to implement contrast-stretching using the PIL library.



[Jupyter Notebook:](#)

# contrast stretching

from PIL import Image, ImageEnhance

[‘ImageEnhance.Contrast’ function can be used here](#)