**PAPER – III         PRACTICAL NO. : 03**

**AIM : WRITE A PROGRAM TO IMPLEMENT POINT PIXEL INTENSITY TRANSFORMATION SUCH AS :**

1. **LOG AND POWER LOG TRANSFORMATION**
2. **CONTRAST ADJUSTMENT**
3. **HISTOGRAM EQUALIZATION**
4. **THRESHOLDING AND HALF TONING OPERATION .**

**ROLL NO. : 02                                BATCH : M.SC PART-I**

**DATE : 17/10/22**

**CODE :**

import numpy as np

from skimage import data,img\_as\_float,img\_as\_ubyte,exposure,io,color

from skimage.io import imread

from skimage.exposure import cumulative\_distribution

from skimage.restoration import denoise\_bilateral,denoise\_nl\_means,estimate\_sigma

#fromskimage.measureimportcompare\_psnr

from skimage.util import random\_noise

from skimage.color import rgb2gray

from PIL import Image,ImageEnhance,ImageFilter

from scipy import ndimage,misc

import matplotlib.pylab as pylab

def plot\_image(image,title=""):

pylab.title(title,size=20),pylab.imshow(image)

pylab.axis('off')

def plot\_hist(r,g,b,title=''):

r,g,b=img\_as\_ubyte(r),img\_as\_ubyte(g),img\_as\_ubyte(b)

pylab.hist(np.array(r).ravel(),bins=256,range=(0,256),color='r',alpha=0.5)

pylab.hist(np.array(g).ravel(),bins=256,range=(0,256),color='g',alpha=0.5)

pylab.hist(np.array(b).ravel(),bins=256,range=(0,256),color='b',alpha=0.5)

pylab.xlabel('PixelValues',size=20),pylab.ylabel('Frequency',size=20)

pylab.title(title,size=20)

im=Image.open('elephant2.jpeg')

im\_r,im\_g,im\_b=im.split()

pylab.style.use('ggplot')

pylab.figure(figsize=(15,5))

pylab.subplot(121),plot\_image(im,'OriginalImage')

pylab.subplot(122),plot\_hist(im\_r,im\_g,im\_b,'HistpgramofRGBImage')

pylab.show()

im=im.point(lambda i:255\*np.log(1+i/255))

im\_r,im\_g,im\_b=im.split()

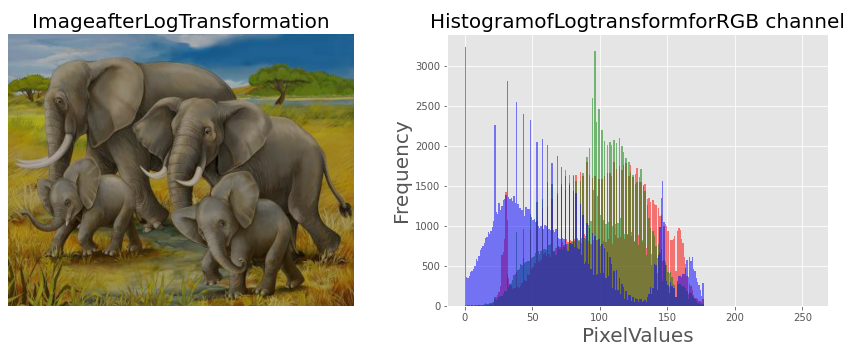
pylab.style.use('ggplot')

pylab.figure(figsize=(15,5))

pylab.subplot(121),plot\_image(im,'ImageafterLogTransformation')

pylab.subplot(122),plot\_hist(im\_r,im\_g,im\_b,'HistogramofLogtransformforRGB channel')

pylab.show()



im=img\_as\_float(imread('bird.jpg'))

#im\_r,im\_g,im\_b=im.split()

gamma=2.5

im1=im\*\*gamma

pylab.style.use('ggplot')

pylab.figure(figsize=(15,5))

pylab.subplot(121),plot\_hist(im[...,0],im[...,1],im[...,2],'HistogramforRGB channel(Input)')

pylab.subplot(122),plot\_hist(im1[...,0],im1[...,1],im1[...,2],'Histogramfor RGBOutput')

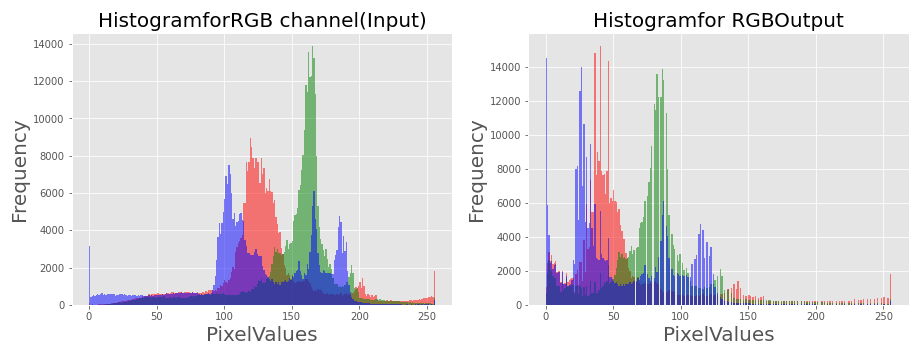
pylab.show()

pylab.figure(figsize=(15,5))

pylab.subplot(121),plot\_image(im,'Imageoriginal')

pylab.subplot(122),plot\_image(im1,'Output')

pylab.show()





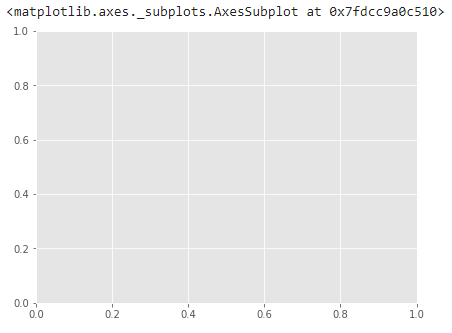
im=Image.open('bird.jpg')

im\_r,im\_g,im\_b=im.split()

pylab.style.use('ggplot')

pylab.figure(figsize=(15,5))

pylab.subplot(121)

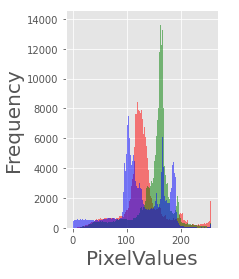


plot\_image(im)

pylab.subplot(122)

plot\_hist(im\_r,im\_g,im\_b)

pylab.show()



def contrast(c):

return 0 if c <50 else (255 if c >150 else (255\*c-22950)/48)

im1=im.point(contrast)

im\_r,im\_g,im\_b=im1.split()

pylab.style.use('ggplot')

pylab.figure(figsize=(15,5))

pylab.subplot(121)

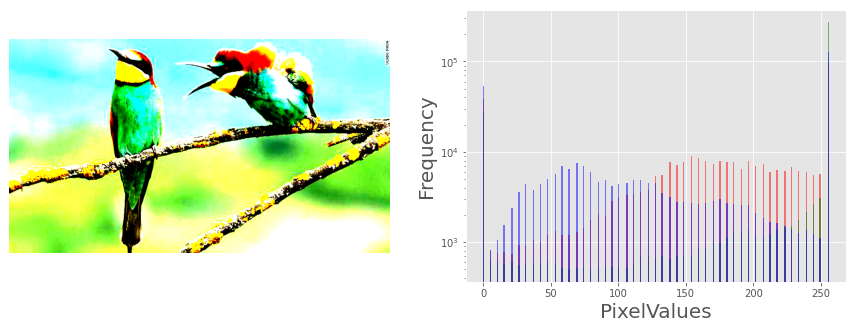
plot\_image(im1)

pylab.subplot(122)

plot\_hist(im\_r,im\_g,im\_b)

pylab.yscale('log',basey=10)

pylab.show()



im=Image.open('elephant.jpg').convert('L')

pylab.hist(np.array(im).ravel(),bins=256,range=(0,256),color='g')

pylab.xlabel('Pixelvalues'),pylab.ylabel('Frequency')

pylab.title('HistogramofPixelvalues')

pylab.show()

pylab.figure(figsize=(12,18))

pylab.gray()

pylab.subplot(221),plot\_image(im,'OriginalImage')

pylab.axis('off')

th=[0,50,100,150,200]

for i in range (2,5):

im1=im.point(lambda x:x > th[i])

pylab.subplot(2,2,i),plot\_image(im1,'binaryImagewith threshold='+str(th[i]))

pylab.show()

