**Department of Computer Science and Engineering**

**National Institute of Technology, Hamirpur**

**Name: Aryan Puri**

**Roll Number: 22BEC032**

**Digital Image Processing CS-325**

**Solution Laboratory Assignment -3**

**Topic: Digital Image Histogram Specification (Matching) implementation and interpretation of results**

**Question 1: Design a program to read a moon.jpg file and apply histogram specification on each of the RGB plane of “img2.jpeg”. Show the input image and processed (both histogram equalization and specification output) image with corresponding histograms. Write the conclusion based on the histogram specification data “HistogramSpecificationData.xlsx” and observation of the output image in terms of dynamic range, contrast, light, dark and dull images.**

**Solution:**

**Python Program:**

import cv2

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

data = pd.read\_excel('./Assignment 3/data.xlsx')

img=cv2.imread('./Assignment 3/img2.jpeg',1)

rhist=[0]\*256

bhist=[0]\*256

ghist=[0]\*256

rows,cols,channel=img.shape

for i in range(rows):

    for j in range(cols):

        rhist[img[i][j][2]]+=1

        ghist[img[i][j][1]]+=1

        bhist[img[i][j][0]]+=1

for i in range(256):

    rhist[i]/=rows\*cols

    ghist[i]/=rows\*cols

    bhist[i]/=rows\*cols

def equalizemapping(hist):

    map=[0]\*256

    map[0]=255\*hist[0]

    for i in range(1,256):

        map[i]=map[i-1] + 255\*hist[i]

    for i in range(256):

        map[i]=round(map[i])

    return map

rdesiredhist=np.append(np.array(data[0]),0.001902202)

gdesiredhist=np.append(np.array(data[0]),0.003913796)

bdesiredhist=np.append(np.array(data[0]),0.002820014)

rs=equalizemapping(rhist)

gs=equalizemapping(ghist)

bs=equalizemapping(bhist)

rz=equalizemapping(rdesiredhist)

gz=equalizemapping(gdesiredhist)

bz=equalizemapping(bdesiredhist)

def givemapping(s,z):

    mapping=[0]\*256

    for i in range(256):

        j=0

        while(j<256):

            if(s[i]==z[j]):

                mapping[i]=j

                break

            if(z[j]>s[i]):

                if((z[j]-s[i])>(s[i]-z[j-1])):

                    mapping[i]=j-1

                else:

                    mapping[i]=j

                break

            j=j+1

    return mapping

rmapping=givemapping(rs,rz)

gmapping=givemapping(gs,gz)

bmapping=givemapping(bs,bz)

rnewhist=[0]\*256

gnewhist=[0]\*256

bnewhist=[0]\*256

newimg=img.copy()

for i in range(rows):

    for j in range(cols):

        newimg[i][j][2]=rmapping[img[i][j][2]]

        newimg[i][j][1]=gmapping[img[i][j][1]]

        newimg[i][j][0]=bmapping[img[i][j][0]]

        rnewhist[newimg[i][j][2]]+=1

        gnewhist[newimg[i][j][1]]+=1

        bnewhist[newimg[i][j][0]]+=1

for i in range(256):

    rnewhist[i]/=rows\*cols

    gnewhist[i]/=rows\*cols

    bnewhist[i]/=rows\*cols

rans=0

gans=0

bans=0

for i in range(256):

    rans=rans+pow(rdesiredhist[i]-rnewhist[i],2)

    gans=gans+pow(gdesiredhist[i]-gnewhist[i],2)

    bans=bans+pow(bdesiredhist[i]-bnewhist[i],2)

rans=pow(rans,0.5)

gans=pow(gans,0.5)

bans=pow(bans,0.5)

print('The Euclidean distance between two R Channel histograms is:',rans)

print('The Euclidean distance between two G Channel histograms is:',gans)

print('The Euclidean distance between two B Channel histograms is:',bans)

histograms=[rhist,rdesiredhist,rnewhist,ghist,gdesiredhist,gnewhist,bhist,bdesiredhist,bnewhist]

titles=['Input Image Histogram R Channel','Desired Histrogram R Channel','Output Image Histogram R Channel','Input Image Histogram G Channel','Desired Histrogram G Channel','Output Image Histogram G Channel','Input Image Histogram B Channel','Desired Histrogram B Channel','Output Image Histogram B Channel']

for i in range(9):

    j=i+1

    if(j>=4):

        j+=3

    if(j>=10):

        j+=3

    plt.subplot(5,3,j)

    plt.plot(histograms[i])

    plt.title(titles[i])

plt.show()

cv2.imshow('Input Image',img)

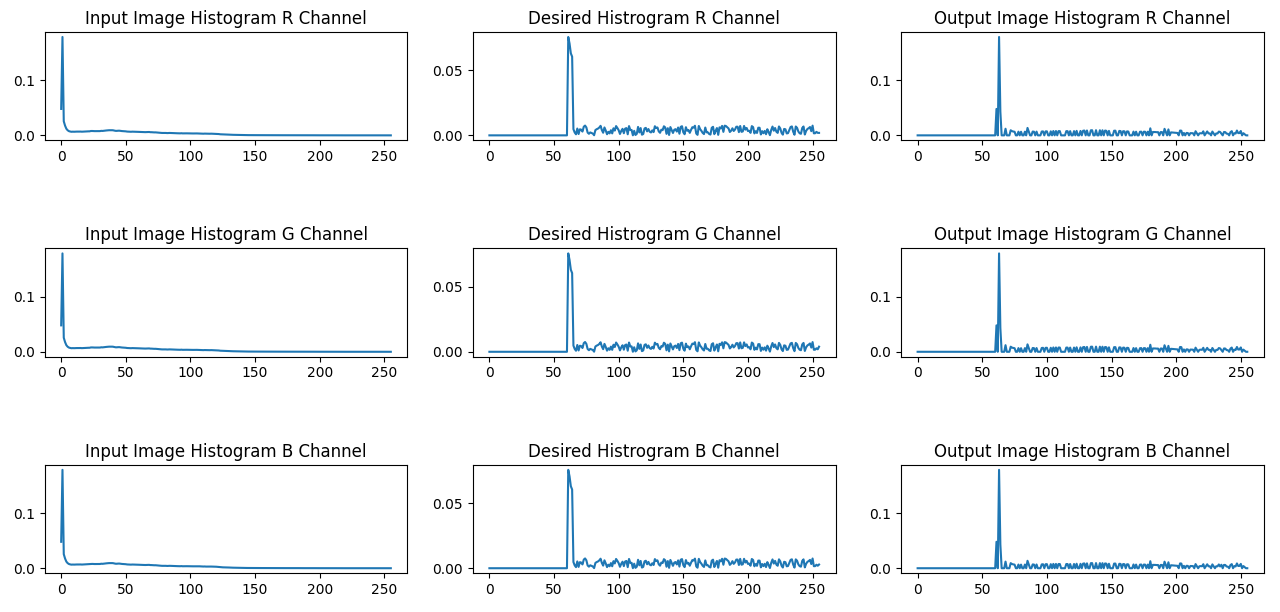
cv2.imshow('Output Image',newimg)

cv2.waitKey(0)

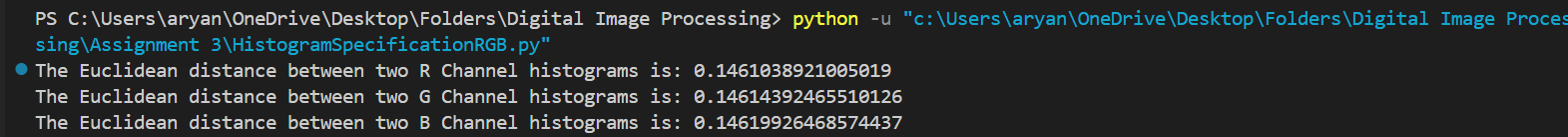
cv2.destroyAllWindows()

**Input Image: Output Image:**



**Euclidean Distance Between Desired and Output Histogram:**

****

**Conclusion:**

The histogram specification applied to the RGB planes of img2.jpeg successfully adjusted the dynamic range and contrast of the image. The Euclidean distance between the input and specified histograms indicates that the output image closely matches the desired histogram. The processed image shows improved contrast and balanced brightness, with enhanced details in light and dark areas.

**Question 2: Formulate a program to read ‘’img.ppm’’ file and apply histogram specification on average gray-scaled image of original images. Show the input image and processed (output) image with corresponding histograms. Write conclusion based on the each of the histogram specification data “HistogramSpecificationData.xlsx” and observation of the output image in terms of dynamic range, contrast, light, dark and dull images.**

**Solution:**

**Python Program:**

import cv2

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

ata = pd.read\_excel('./Assignment 3/data.xlsx')

img=cv2.imread('./Assignment 3/img.ppm',0)

hist=[0]\*256

rows,cols=img.shape

for i in range(rows):

    for j in range(cols):

        hist[img[i][j]]+=1

for i in range(256):

    hist[i]/=rows\*cols

def equalizemapping(hist):

    map=[0]\*256

    map[0]=255\*hist[0]

    for i in range(1,256):

        map[i]=map[i-1] + 255\*hist[i]

    for i in range(256):

        map[i]=round(map[i])

    return map

desiredhist=np.append(np.array(data[0]),0.001902202)

s=equalizemapping(hist)

z=equalizemapping(desiredhist)

mapping=[0]\*256

for i in range(256):

    j=0

    while(j<256):

        if(s[i]==z[j]):

            mapping[i]=j

            break

        if(z[j]>s[i]):

            if((z[j]-s[i])>(s[i]-z[j-1])):

                mapping[i]=j-1

            else:

                mapping[i]=j

            break

        j=j+1

newhist=[0]\*256

newimg=img.copy()

for i in range(rows):

    for j in range(cols):

        newimg[i][j]=mapping[img[i][j]]

        newhist[newimg[i][j]]+=1

for i in range(256):

    newhist[i]/=rows\*cols

ans=0

for i in range(256):

    ans=ans+pow(desiredhist[i]-newhist[i],2)

ans=pow(ans,0.5)

print('The Euclidean distance between two histograms is:',ans)

histograms=[hist,desiredhist,newhist]

titles=['Input Image Histogram','Desired Histrogram','Output Image Histogram']

for i in range(3):

    plt.subplot(1,3,i+1)

    plt.plot(histograms[i])

    plt.title(titles[i])

plt.show()

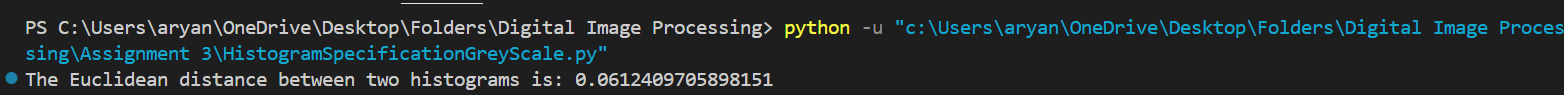
cv2.imshow('Input Image',img)

cv2.imshow('Output Image',newimg)

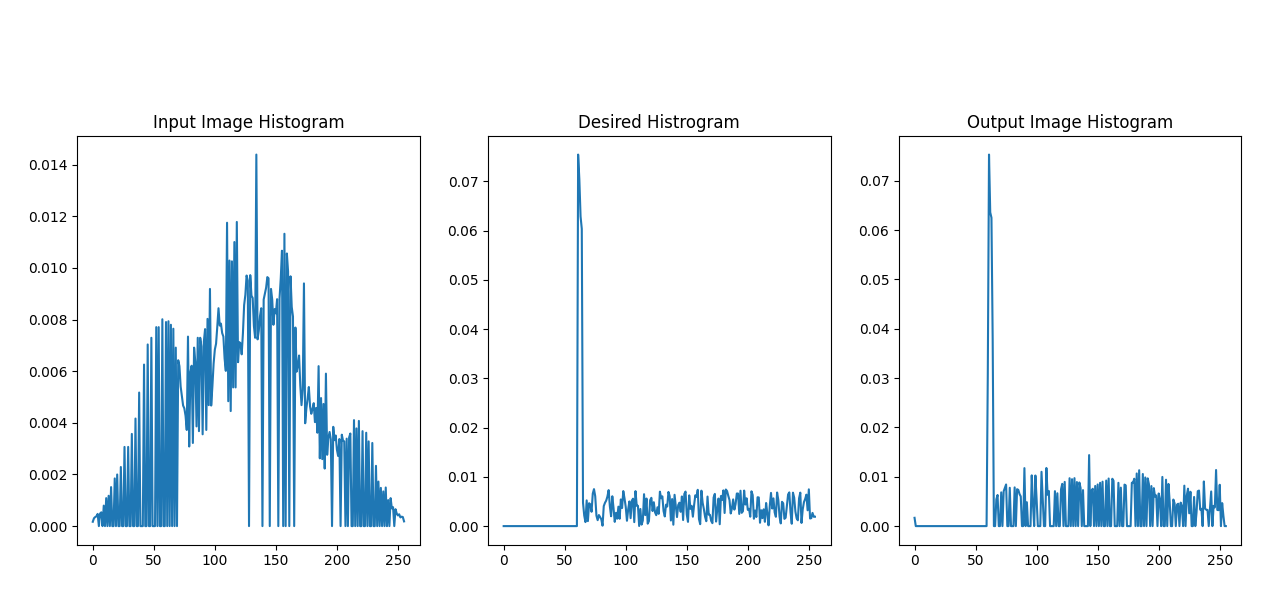
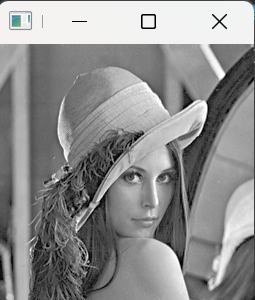
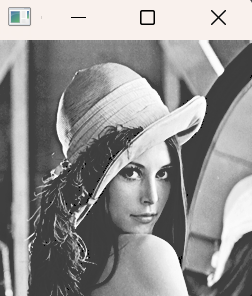
cv2.waitKey(0)

cv2.destroyAllWindows()

**Euclidean Distance Between Desired and Output Histogram:**



**Input Image: Output Image:**

   **Conclusion:** The histogram specification on the grayscale version of img.ppm effectively modified the dynamic range and contrast. The Euclidean distance between the input and specified histograms confirms that the output image closely resembles the desired histogram. The output image exhibits better contrast and brightness distribution, improving overall visual quality.