Arifullah Jan - 186943 - BSCS 6A

LAB07 - DIP

```
'''
python task1.py path/to/input
'''




import PIL
from PIL import Image
import itertools
import numpy as np
import sys
from PIL import ImageDraw
import matplotlib.pyplot as plt
image = Image.open(sys.argv[1])
# image = image.convert("L")  # convert to signle channeled image



width, height = image.size
totalPixels = width* height

freq = [0] * 256  # fill
cdf = [0] * 256  # fill zeros


pixels = image.load()  # allows pixel values to be edited



freq = image.histogram() # get frequencies

# input histogram
a = np.array(image)
plt.hist(a.ravel(),  bins=256)
plt.savefig('inputhist.png')
```

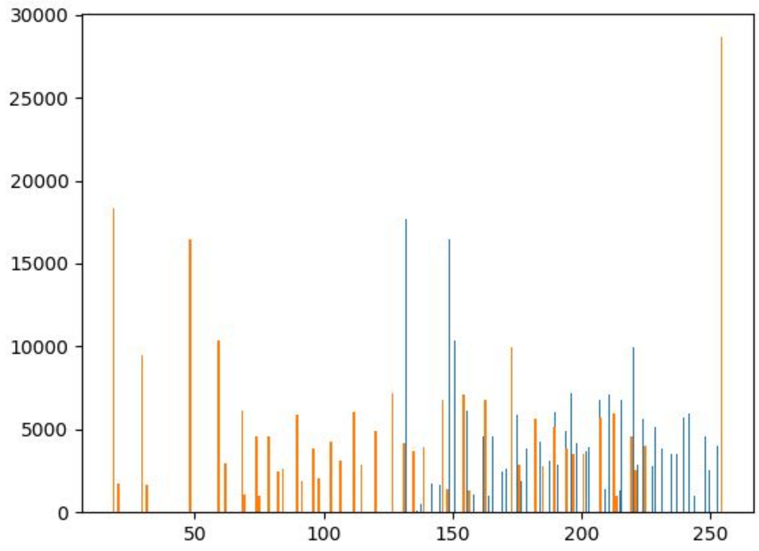
```
############# EQUALIZATION

s = 0
for i in range(256):
    s += freq[i]*1.0/totalPixels
    cdf[i] = s

print(cdf[255])
for x, y in itertools.product(range(width), range(height)):
    pixels[x,y] = int((255 * cdf[pixels[x,y]]))


image.save('output.tif')

# output histogram
a = np.array(image)
plt.hist(a.ravel(),  bins=256)
plt.savefig('outputhist.png')
```



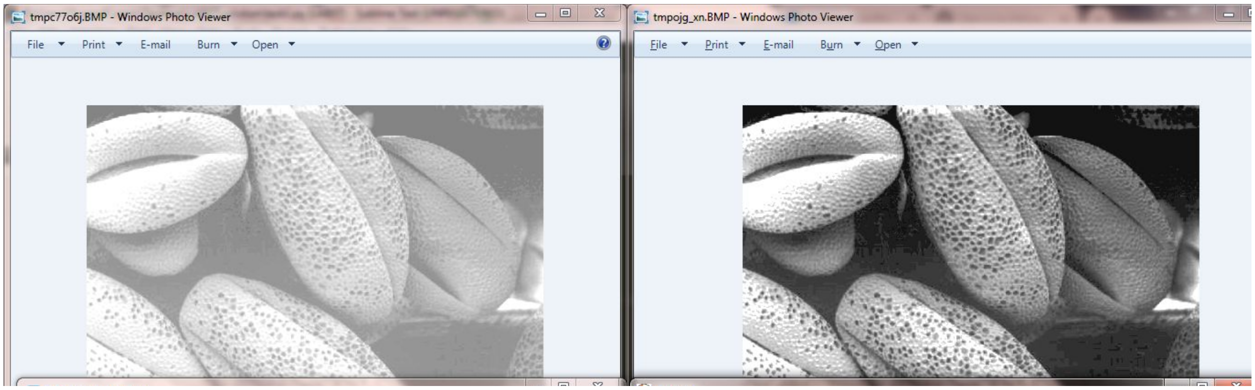
Input hist: blue

Ouptut hist: orange

Does the resultant image's histogram has uniform distribution?

No the resultant image's histogram is does not have exactly a uniform distribution. The reason is that the concept of histogram equalization is perfect only for continuous domain where's image is in discrete domain.

Still the histogram becomes dispersed which causes increase in contrast as visible in image below


tmpc77o6j.BMP - Windows Photo Viewer
File
Print
E-mail
Burn
Open
tmpojg_xn.BMP - Windows Photo Viewer
File
Print
E-mail
Burn
Open