**Department of Computing**

**Digital Image Processing**

**Class: BSCS-9ABC**

**Lab 6**

**Submitted By: Fatima Seemab**

**CMS ID: 291310**

**BSCS 9B**

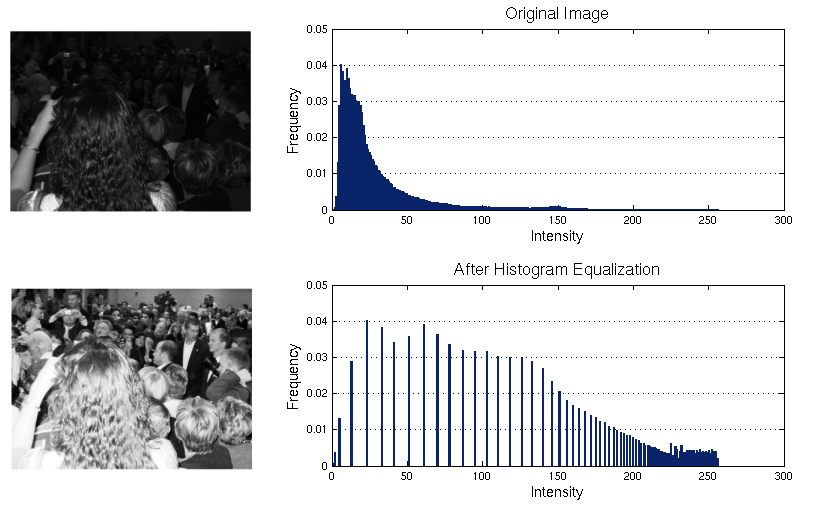
Lab 6

Histogram Equalization

**Introduction**

In this lab you will try to improve the contrast of an image by doing histogram equalization.

Consider the example below.

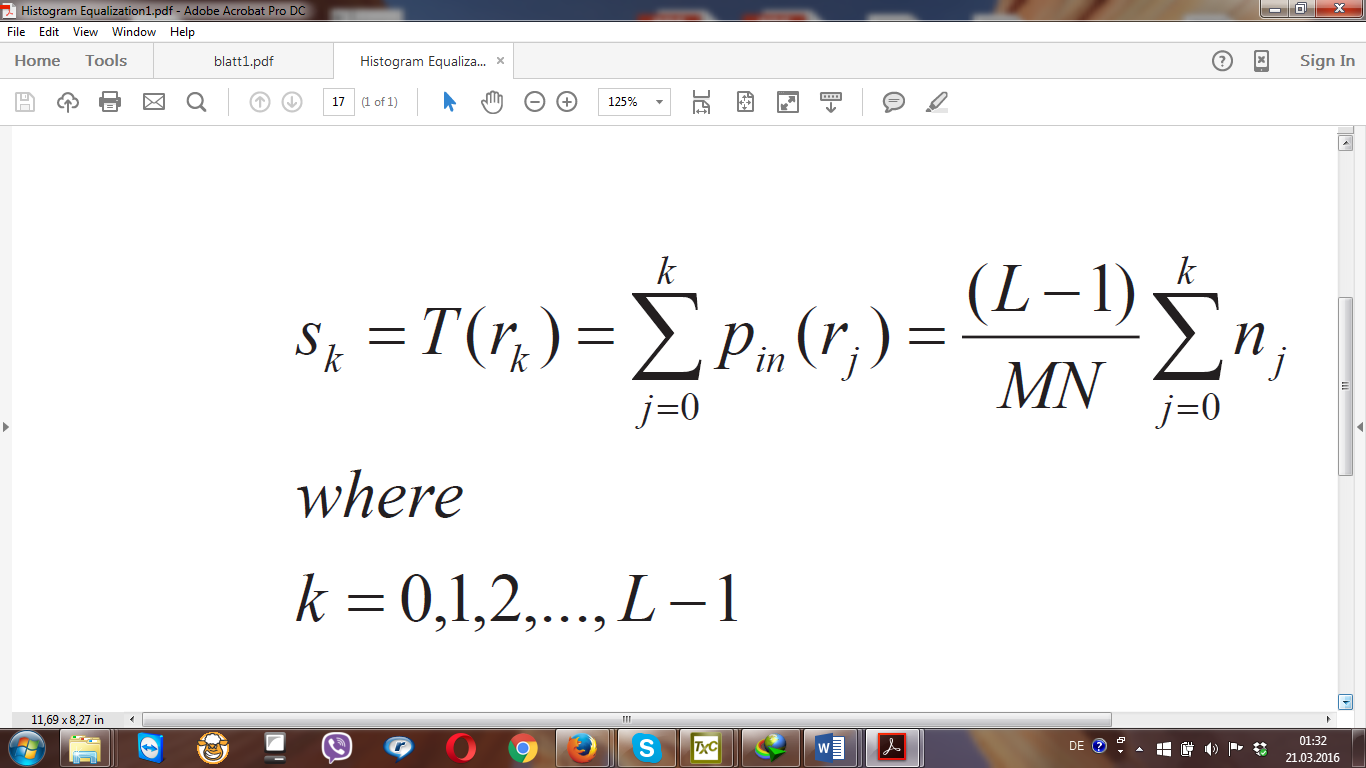
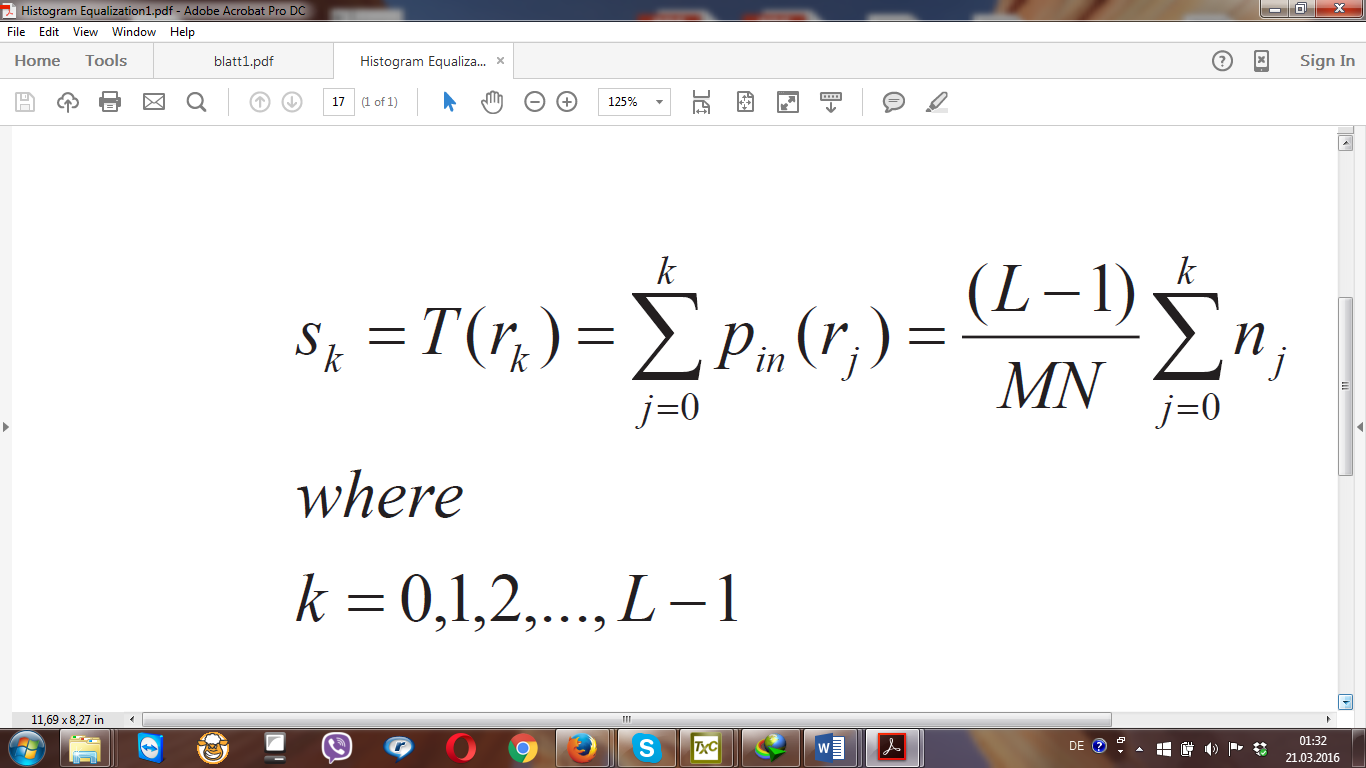


**Tools/Software Requirement**

Python 3.X

**Description**

1. Write a program that shows the histogram for a given image. For displaying the histogram, you may use Matplotlib. The following link might be helpful in this regard. <https://bespokeblog.wordpress.com/2011/07/11/basic-data-plotting-with-matplotlib-part-3-histograms/>
2. Write a program that equalizes the histogram of a given image. Consider the formula,



1. Show the histograms before and after equalization. Again consult Matplotlib. Opencv can also be used for some parts.

**CODE**

from PIL import Image,ImageOps

#For processing image class PIL is imported from library PIL

from matplotlib import pyplot as plt

#For plotting histogram class PIL is imported from library PIL

import numpy

# numpy library for dealing with array is imported

plt.style.use('fivethirtyeight')

#For opening the image

im=Image.open('image\_lab6.tif')

#This will return no of rows and no of cols in an image

x,y=im.size

#For M in equalization of equalization

size=x\*y

#8 bit image total levels =265

L=256

#arrays are initialized with zeros size equals no of levels in image

frequency=numpy.zeros((L))

pdf=numpy.zeros((L))

cpdf=numpy.zeros((L))

s=numpy.zeros((L))

count=0

#To iterate over all the rows

for i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current pixel

f=im.getpixel((cordinate))

#calculating frequency for that pixel intensity

frequency[f]=frequency[f]+1

#For calculating probability density function

for i in range(L):

pdf[i]=frequency[i]/size

#For calculating cumulative pdf

for j in range(i+1):

cpdf[i]=pdf[j]+cpdf[i]

#Last value of cumulative is approximately 1

if(cpdf[L-1]>0.95):

print("pdf is successfully calculated")

#For transformation multiply by L-1

for i in range(L):

cpdf[i]=cpdf[i]\*(L-1)

cpdf[i]=round(cpdf[i],0)

#For calculating s new equalized values

for i in range(L):

for j in range(L):

if(cpdf[j]==i):

s[i]=frequency[j]+s[i]

for i in range(x):

#To iterate over all the cols in a row

for j in range(y):

cordinate=i,j

# finding the intensity of current

# pixel

f=im.getpixel((cordinate))

im.putpixel((cordinate),(int(cpdf[f])))

im

Chart, histogram

Description automatically generated**Histogram before Equalization**

**Histogram after Equalization**

Chart, histogram

Description automatically generated

**Image before equalization:**

A close-up of several rolls of yarn

Description automatically generated with low confidence**Image after equalization:**

1. Does the equalized histogram has a uniform distribution?

Yes now the new histogram is equalized as it has distributed all of its peak on the whole range of intensities.

**Some important points about the exercise.**

1. You should apply histogram equalization on Greyscale image (given in lab folder).  
2. You should not use the builtin histogram equalization method available in matplotlib or opencv. You can use these packages only for displaying histograms.  
3. You should implement the formula mentioned in the exercise sheet to implement histogram equalization.

**Deliverable**

Please upload the report with code and screenshots of output.