Grayscale Image Histogram Equalization

Source code:

```
# -*- coding: utf-8 -*-
"""
Created on Wed Feb  6 10:38:21 2019

@author: eagle
"""

import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('orig_gray_img.jpg',0)  # use this to change the grayscale image to manipulate

# Get CDF of original image
hist,bins = np.histogram(img.flatten(),256,[0,256])
cdf = hist.cumsum()
cdf_normalized = cdf * hist.max()/ cdf.max()

# Performing Histogram Equalization Equation with the use of Numpy masked array
cdf_m = np.ma.masked_equal(cdf,0)
cdf_m = (cdf_m - cdf_m.min())*255/(cdf_m.max()-cdf_m.min())
cdf = np.ma.filled(cdf_m,0).astype('uint8')

# Now we have the look-up table that gives us the information on what is the
# output pixel value for every input pixel value
# we apply the transform here
img2 = cdf[img]

# Plot of Before and After Histogram of the Image
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.hist(img2.flatten(),256,[0,256], color = 'b')
plt.legend(('original','equalized'), loc = 'upper left')
plt.title('Original and Equalized Histogram Plots')
plt.show()

# Resize the Before and After Images to fit on laptop screen
height, width = img.shape[:2]
height2, width2 = img2.shape[:2]

if height > 800 or width > 800:
    img = cv2.resize(img, (int(width/1.5), int(height/1.5)), interpolation = cv2.INTER_CUBIC)

if height2 > 800 or width2 > 800:
```

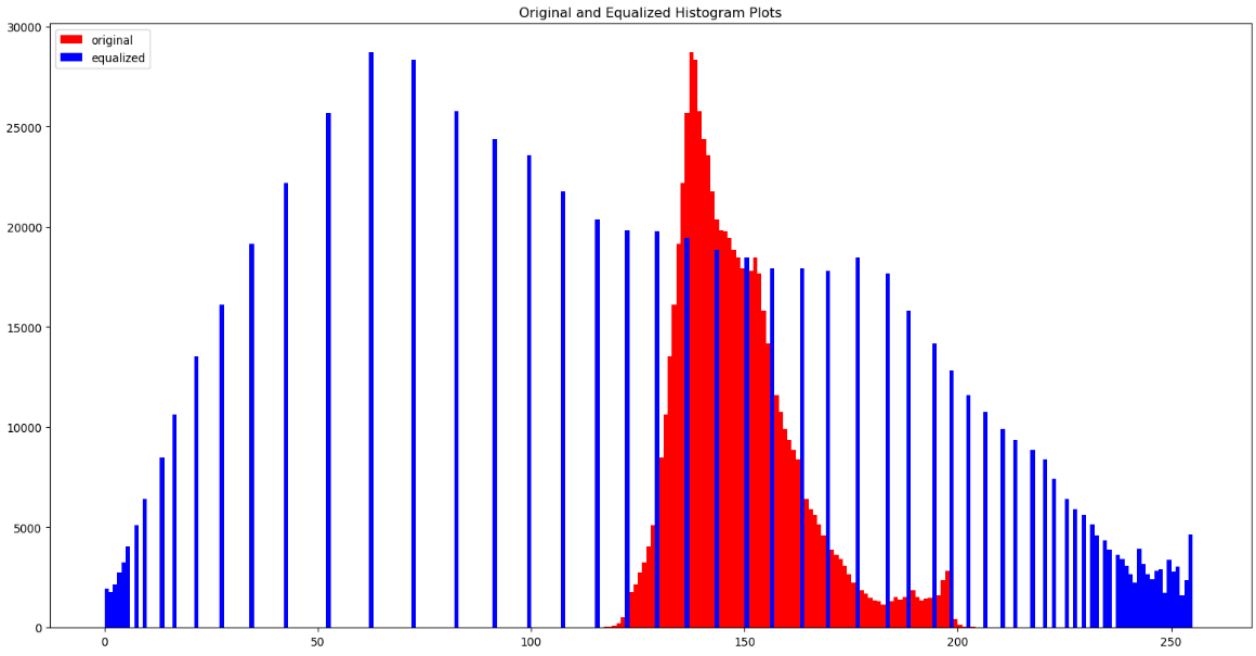
```
    img2 = cv2.resize(img2, (int(width/1.5), int(height/1.5)), interpolation = cv2.INTER_CUBIC)

# Show Before and After Image
res = np.hstack((img, img2))
cv2.imshow('Before and After', res)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

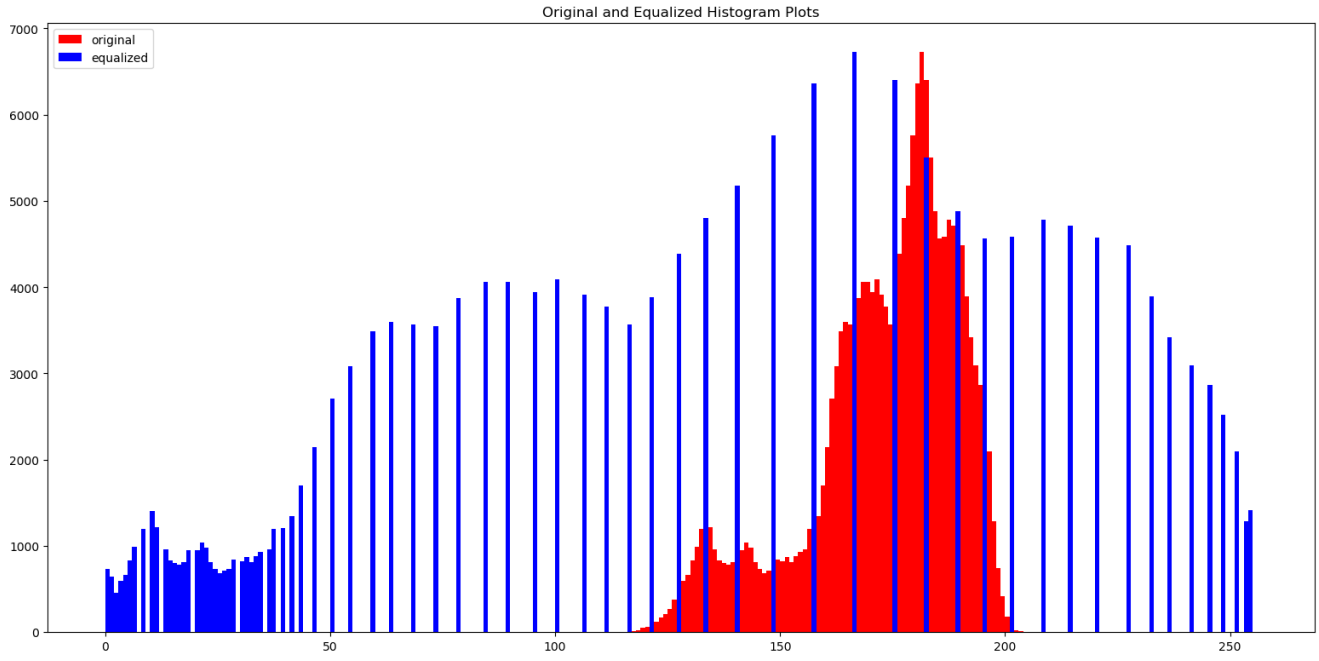
Outputs:

Example 1:



Example 2:


Before and After


Original and Equalized Histogram Plots
original
equalized
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

Color Image Histogram Equalization

Source code:

```
# -*- coding: utf-8 -*-
"""
Created on Wed Feb  6 11:35:04 2019

@author: eagle
source: opencv.org
"""
import cv2
import numpy as np
import argparse

# run this code using the command prompt
parser = argparse.ArgumentParser(description='Code for Histogram Equalization tutorial.')
parser.add_argument('--input', help='Path to input image.', default='orig_color_img.jpg')  # loading the image
                                                    # when running this code in command, put --input orig_gray_img_2.jpg
                                                    # after python HistoEqOpenCV.py to run it on that image
args = parser.parse_args()
src = cv2.imread(args.input)

# if no image is loaded
if src is None:
   print('Could not open or find the image:', args.input)
   exit(0)

img_yuv = cv2.cvtColor(src, cv2.COLOR_BGR2YUV)

# equalize the histogram of the Y channel
img_yuv[:,:,0] = cv2.equalizeHist(img_yuv[:,:,0])

# convert the YUV image back to RGB format
img_output = cv2.cvtColor(img_yuv, cv2.COLOR_YUV2BGR)

height, width = src.shape[:2]
height2, width2 = img_output.shape[:2]

if height > 800 or width > 800:
```

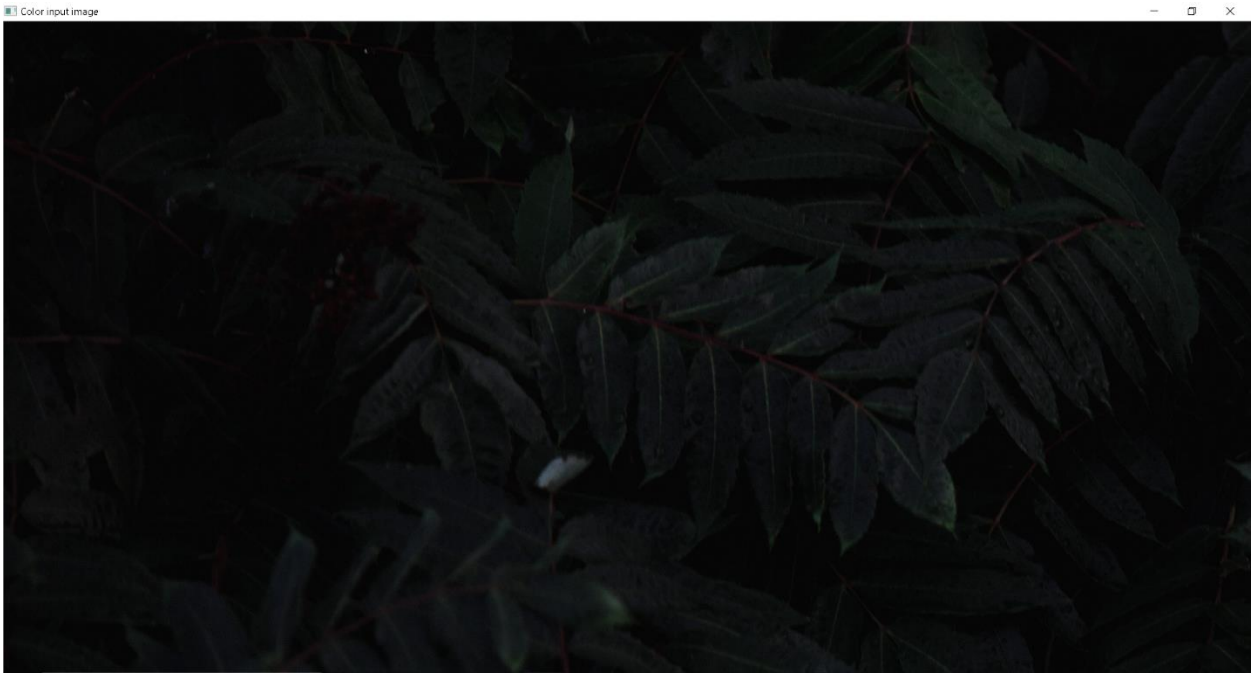
```
    src = cv2.resize(src, (int(width/1.5), int(height/1.5)), interpolation = cv2.INTER_CUBIC)

if height2 > 800 or width2 > 800:
    img_output = cv2.resize(img_output, (int(width/1.5), int(height/1.5)), interpolation =
cv2.INTER_CUBIC)

cv2.imshow('Color input image', src)
cv2.imshow('Histogram equalized', img_output)
cv2.waitKey()
```

Outputs:

Example 1:



Example 2:

Color input image

Histogram equalized