

تمرین سری ۲ - مرضیه امیری

۱- الگوریتم histogram equalization را روی تصویر زیر پیاده سازی کرده (بدون استفاده از دستور histeq) و بر روی تصویر زیر اعمال کنید.

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

img = cv2.imread('1.jpg') # reading image
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) #convert to gray
plt.hist(gray.ravel(), bins=256) #showing histogram of original image
plt.show()
w , h = gray.shape
n = w * h #n is the number of pixels
My_dict = dict()
for i in range (0,256): #create a dictionary to keep every pixel intensity as a key
    My_dict[i] = 0

for item in gray.ravel(): #keep the number of each pixel intensity as dictionary values
    My_dict[item]+=1

for i in My_dict: #calculate cumulative normalized histogram
    if i > 0:
        My_dict[i] += My_dict[i-1]

for i in My_dict: #calculate cumulative normalized histogram and transformed intensity
    My_dict[i] = round(( My_dict[i] / n ) * 255)

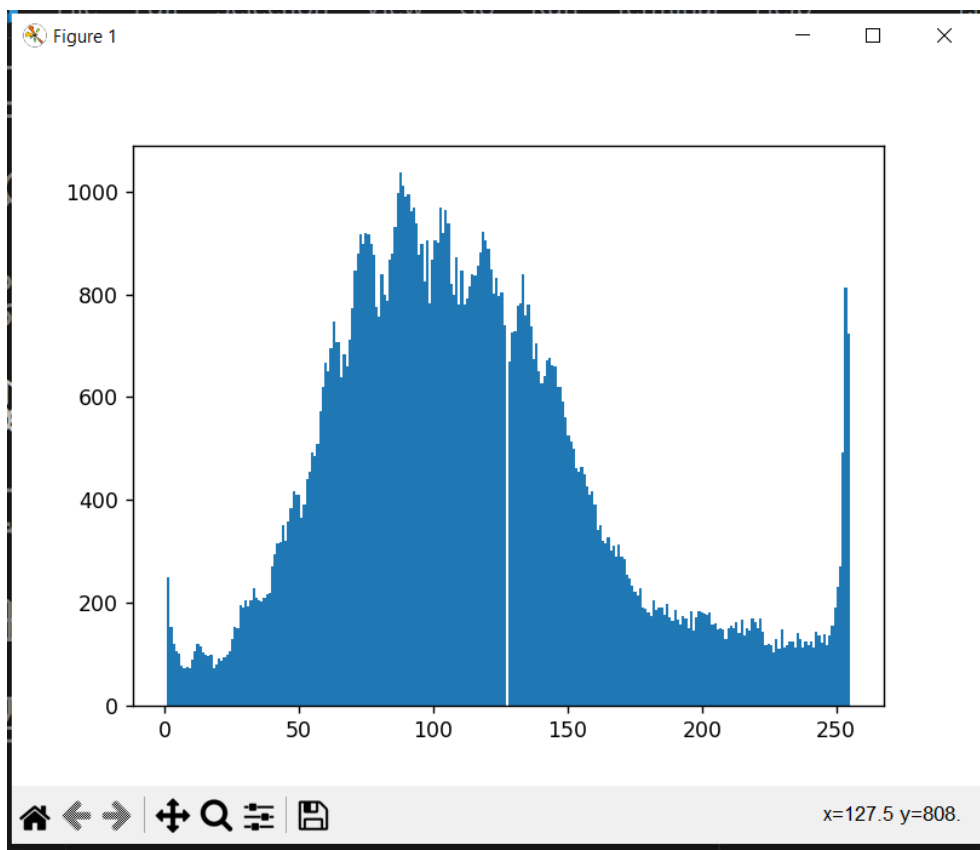
equalized=gray.copy()
for i in range(w): #apply the equalization to original image
    for j in range(h):
        k=equalized[i,j]
        equalized[i,j]=My_dict[k]

plt.hist(equalized.ravel(), bins=256) #showing equalized histogram
plt.show()

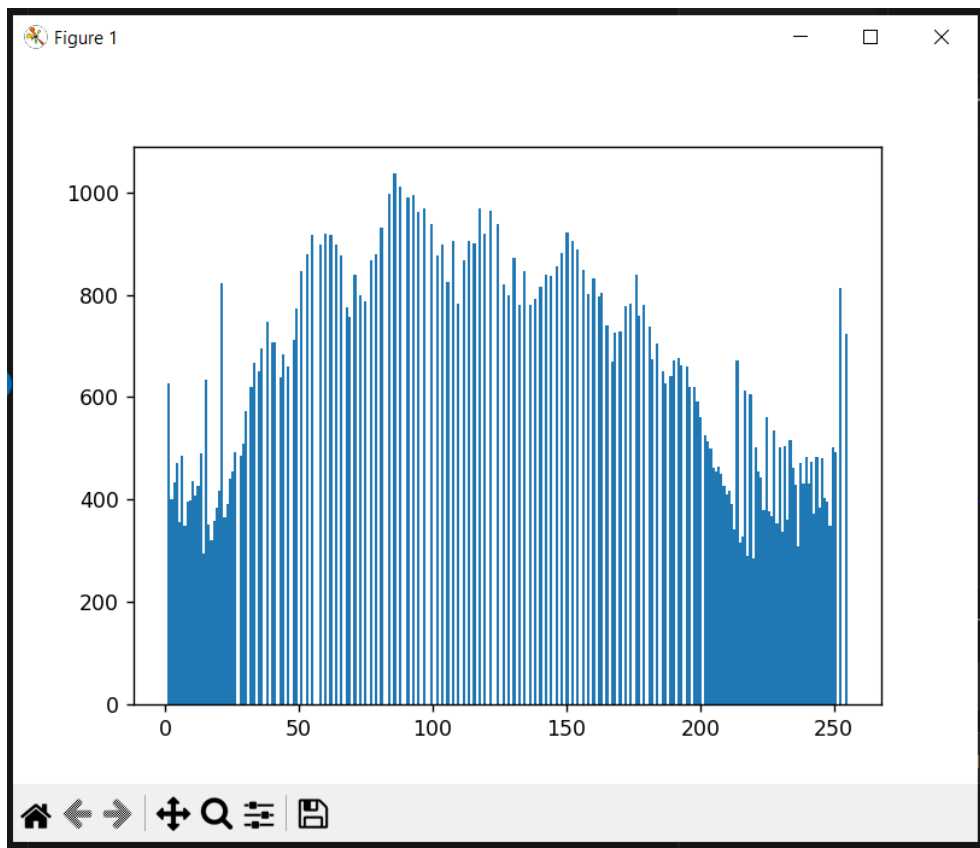
cv2.imshow('original',gray) #showing main image
cv2.imshow('equalized',equalized) #showing equalized image
cv2.waitKey(0)
cv2.destroyAllWindows()
```

خروجی :

هیستوگرام تصویر اولیه:



هیستوگرام equalize شده:



مرضیه امیری - تصویر پردازش



تصویر اصلی و
ایکوالایز شده

ب) نتیجه الگوریتم خود را با نتایج دستورات `equalizeHist` و `createCLAHE` مقایسه کنید. (تصویر را به `gray` تبدیل کنید)

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

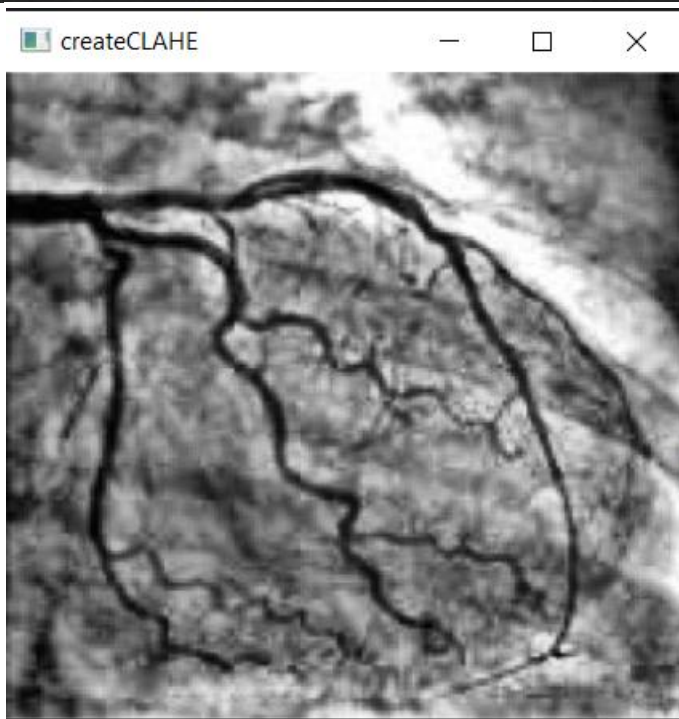
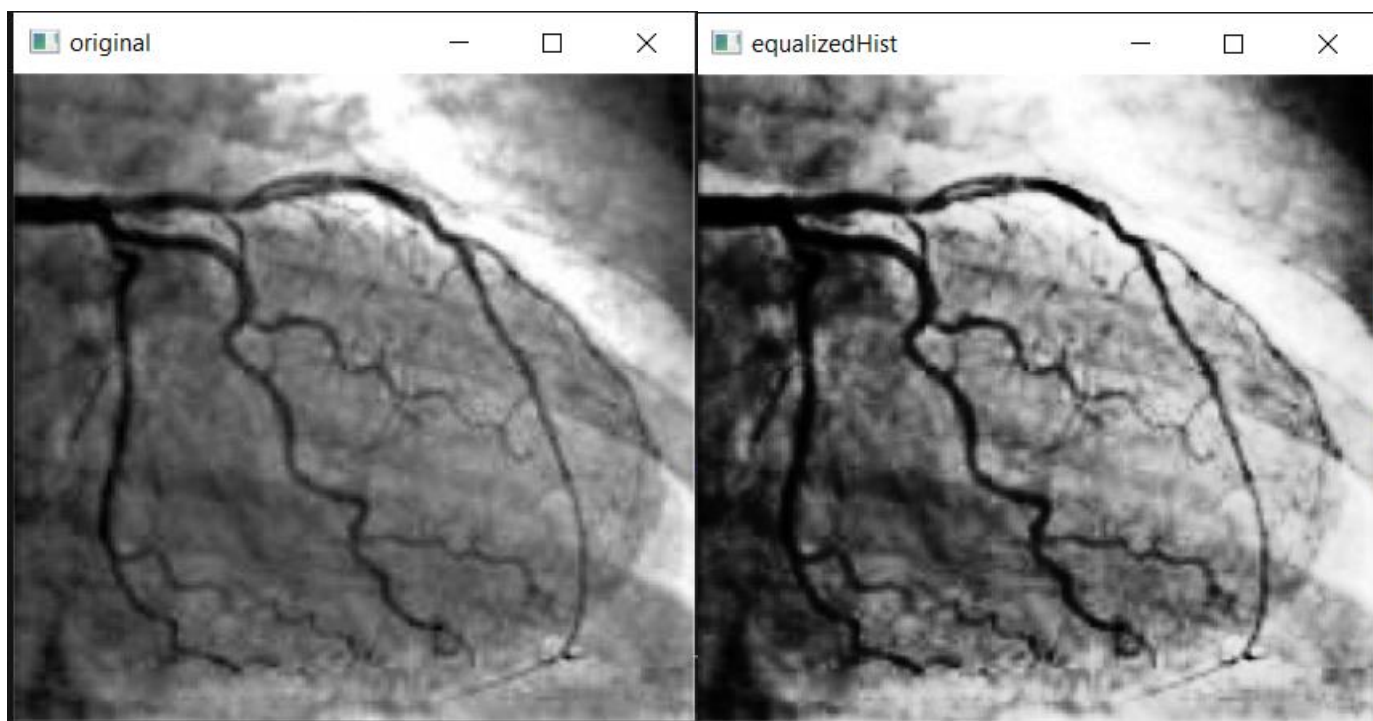
img = cv2.imread('1.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

img2 = cv2.equalizeHist(gray) #equalizeHist
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8)) #createCLAHE
img3 = clahe.apply(gray) #createCLAHE

cv2.imshow('original', gray)
cv2.imshow('equalizedHist', img2)
cv2.imshow('createCLAHE', img3)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Original:

equalizedHist:



createCLAHE

۲- تصویر زیر را خوانده و با منحنی زیر **contrast stretching** بر روی آن انجام دهید.

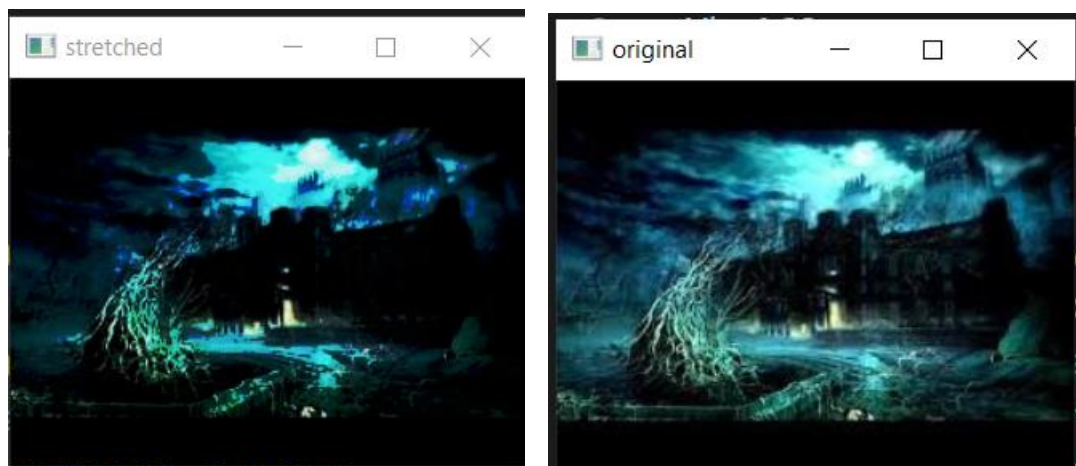
مقادیر پارامترها را بصورتی در نظر بگیرید که تصویر خروجی بهترین کیفیت را از نظر شما داشته باشد. (مقادیر پارامترها را ارسال کنید)

```
import cv2
a=50
b=150
alpha=0.1
beta=0.5
gama=0.8
Ya=20
Yb=160

img = cv2.imread('2.jpg')
h,w,z=img.shape
stretched=img.copy()
for i in range(h):
    for j in range(w):
        for c in range(z):
            if stretched[i,j,c]<a:
                stretched[i,j,c]*=alpha
            elif stretched[i,j,c]<b:
                stretched[i,j,c]=beta*(stretched[i,j,c]-a)+Ya
            else:
                stretched[i,j,c]=gama*(stretched[i,j,c]-b)+Yb

cv2.imshow('original', img)
cv2.imshow('stretched', stretched)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

خروجی:



مرضیه امیری – تصویر پردازش

۳- می‌خواهیم تصویر ورودی زیر را با تیز کردن لبه‌ها بهبود ببخشیم (۱) با اعمال تبدیل مستقیم لاپلاس جمع با کردن با تصویر اصلی (۲) با استفاده از unsharp فیلترینگ اینکار را انجام دهید. (بدون استفاده از دستورات پایتون و با استفاده از این دستور نتایج را نمایش داده و مقایسه کنید)

```
import cv2
import numpy as np

#laplacian function
def laplacian_addition_function(img):
    h,w=img.shape
    filtered_img=img.copy()

    #applying laplacian filter to original image
    # -1 -1 -1
    # -1  9 -1
    # -1 -1 -1
    for i in range(1,h-1):
        for j in range(1,w-1):
            filtered_img[i,j]=-img[i-1,j-1]-img[i-1,j]-img[i-1,j+1]-img[i,j-1]-img[i,j+1]-
img[i+1,j-1]-img[i+1,j]-img[i+1,j+1]+9*img[i,j]

    return filtered_img

#blurring function
def blurring_function(img):
    h,w=img.shape
    blured_img=img.copy()
    blured_img-=blured_img
    #average filter
    # 1/16 * 1 2 1
    #       2 4 2
    #       1 2 1
    for i in range(1,h-1):
        for j in range(1,w-1):
            blured_img[i,j]=1/16*(img[i-1,j-1]+2*img[i-1,j]+img[i-1,j+1]+2*img[i,j-
1]+2*img[i,j+1]+img[i+1,j-1]+2*img[i+1,j]+img[i+1,j+1]+4*img[i,j])

    return blured_img

def unsharp_masking(img):
    h,w=img.shape
    k=1
    masked_img=img.copy()
    g_mask=img-blurring_function(img)
    masked_img+=(k*g_mask)

    return masked_img
```

```
img = cv2.imread('3.jpg', 0)
filtered_img = laplacian_addition_function(img)
unsharp_masked = unsharp_masking(img)
g_mask=img-bluring_function(img)
blured=bluring_function(img)

cv2.imshow('Original', img)
cv2.imshow('laplacian',filtered_img)
cv2.imshow('unsharpemasking',unsharp_masked)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

خروجی :

تصویر اصلی



laplacian :



unsharped:



استفاده از دستورات پایتون برای unsharp کردن:

```
from PIL import Image, ImageFilter
import cv2
# Open an image file
with Image.open("3.jpg") as im:
    # Apply unsharp mask filter
    im_sharp = im.filter(ImageFilter.UnsharpMask(radius=2, percent=150, threshold=3))
    # Save the filtered image
    im_sharp.save("image_sharpened.jpg", "JPEG")
imgsh = cv2.imread("image_sharpened.jpg")
img = cv2.imread("3.jpg")
cv2.imshow('original',img)
cv2.imshow('mage_sharpened',imgsh)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

خروجی :

تصویر اصلی :



مرضیه امیری - تصویر پردازی

