

```

import numpy as np
def calculate_probability(image,hist):
    total_pixels = image.shape[0]
    print(total_pixels)
    for i in range(len(hist)):
        hist[i]=hist[i]/total_pixels
    return hist
def histogram(image,l):
    image = image.reshape(image.shape[0] * image.shape[1])
    histogram = np.zeros(l)
    for i in image:
        histogram[i] = histogram[i]+1
    histogram = calculate_probability(image,histogram)
    return histogram
def round_of_values(hist,l):
    #running_sum
    running_sum = np.zeros_like(hist)
    sum = 0
    for i in range(len(running_sum)):
        sum = sum + hist[i]
        running_sum[i] = sum*l
    round_of_values = np.round(running_sum)
    return round_of_values
def histogram_eualization(image,l):
    hist = histogram(image,l)
    round_of = round_of_values(hist,l)
    image_2 = np.zeros_like(image)
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            value = image[i][j]
            image_2[i][j] = round_of[value]
            #print(round_of)
    return image_2
import cv2 as cv
import matplotlib.pyplot as plt
image= cv.imread("image.png")
img= cv.cvtColor(image,cv.COLOR_BGR2GRAY)
plt.figure(figsize=(20,5))
plt.subplot(1,2,1)
plt.title('input')
plt.imshow(img, cmap='gray')
plt.axis('off')
plt.subplot(1,2,2)
plt.title('output')
img2 = histogram_eualization(img,256)
plt.imshow(img2, cmap='gray',vmin=0,vmax=255)
plt.axis('off')
plt.show()

```

input



output

