I. Histogram Equalization

In this report, I will first explain what is Histogram Equalization; Then give the implement in Python; Finally discuss the relationship between Histogram Equaliztion and Histogram Matching.

A. What is Histogram Equalization

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. Since digital images are represent by pixel array, for a grayscale image it can be described by a 2D array, each pixel's value range from 0 to 255, the brightness will grow over the value increase . For an unprocessed Image, If we count the each value's occurrence and make a histogram, it would be like the left picture in Fig.1

Our goal is to let the occurrence of each value distributed more equally on the axis 0-255, like the right picture in Fig.1; For the intuition this will make the image shows more details in where used to be too drak or bright.

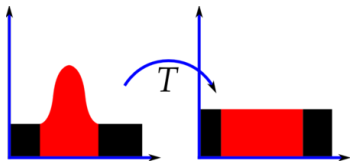

Fig. 1. shows the basic concept of Histogram Equalization

B. Implement in Python

The shortcut is to use the Opencv package, which include the Function *qualizeHist()*.To use this function just input your source image then it will return the processed image:

reuslt = cv2.equalizeHist(source).

Also, there is an easy way to implement it step by step using Numpy:

First,we flatten the image array and count the ocurrence of each value using Numpy *bincount()*:

occurence=np.bincount(flaten,minlength=256)

Second, calculate the PDF and CDF of the pixels, getting a projection list sk[]:

nk=occurence/(MN),MN is the total size of pixels.

*sk[k]=255*nk[0:k].sum()*,calculate CDF and get the sk[].

Finally, apply the projection to original image:

img[img==pix]=sk[pix]

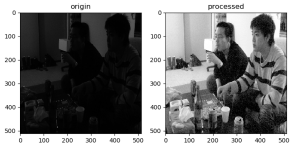

Fig. 2. shows left is original image, right is processed

C. relationship with Histogram Matching

In short, Histogram euqalization is a special case of Histogram Matching; In Histogram euqalization we supposed the after processed histogram is uniformly distributed. But in Histogram Matching, first we need another image as an tartget distribution, second we need to find the distribution projection between the original and target histogram; finally we can apply this projection on the origial image.