**python+opencv直方图均衡化**

**前言**

直方图均衡化的三种情况,分别是:  
- 灰度图像直方图均衡化  
- 彩色图像直方图均衡化  
- YUV 直方图均衡化  
插入原图:  


**灰度图像直方图均衡化**

对直方图均衡化主要使用opencv提供的一个equalizeHist()方法.

import cv2

import numpy as np

img = cv2.imread("image0.jpg", 1)

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

cv2.imshow("src", gray)

dst = cv2.equalizeHist(gray)

cv2.imshow("dst", dst)

cv2.waitKey(0)

将灰度图像作为参数传进equalizeHist()方法即可,效果如下:  


**彩色图像直方图均衡化**

彩色图像的直方图均衡化和灰度图像略有不同,需要将彩色图像先用split()方法,将三个通道拆分,然后分别进行均衡化.最后使用merge()方法将均衡化之后的三个通道进行合并.操作如下:

import cv2

import numpy as np

img = cv2.imread("image0.jpg", 1)

cv2.imshow("src", img)

*# 彩色图像均衡化,需要分解通道 对每一个通道均衡化*

(b, g, r) = cv2.split(img)

bH = cv2.equalizeHist(b)

gH = cv2.equalizeHist(g)

rH = cv2.equalizeHist(r)

*# 合并每一个通道*

result = cv2.merge((bH, gH, rH))

cv2.imshow("dst", result)

cv2.waitKey(0)

均衡化之后的效果如下:  


**YUV 直方图均衡化**

import cv2

import numpy as np

img = cv2.imread("image0.jpg", 1)

imgYUV = cv2.cvtColor(img, cv2.COLOR\_BGR2YCrCb)

cv2.imshow("src", img)

channelsYUV = cv2.split(imgYUV)

channelsYUV[0] = cv2.equalizeHist(channelsYUV[0])

channels = cv2.merge(channelsYUV)

result = cv2.cvtColor(channels, cv2.COLOR\_YCrCb2BGR)

cv2.imshow("dst", result)

cv2.waitKey(0)

均衡化之后效果如下:  
