import numpy as np  
import cv2  
import copy  
  
  
def CountPixel(original\_image, ArrayB, ArrayG, ArrayR, imageH, imageW):  
 B=0  
 G=1  
 R=2  
 for i in range(imageH): #use i as index of image to determine which pixels need to be filtered  
 for j in range(imageW): #use j as index of image to determine which pixels need to be filtered  
 ArrayB[original\_image[i][j][B]] += 1 #count the corresponding pixel number in the image from 0 to 255  
 ArrayG[original\_image[i][j][G]] += 1  
 ArrayR[original\_image[i][j][R]] += 1  
 return ArrayB, ArrayG, ArrayR  
  
  
def Accumulate(Array):  
 AccumArray = [0]\*256  
 j = 0  
 for i in range(256):  
 AccumArray[i] = Array[i] + j #accumulate the appear pixel  
 j = AccumArray[i]  
 return AccumArray  
  
  
def ImageHeq(original\_image, ACPAB, ACPAG, ACPAR, imageH, imageW):  
 img\_tmp = original\_image  
 R = 2  
 G = 1  
 B = 0  
 for i in range(imageH):  
 for j in range(imageW):  
 img\_tmp[i][j][B] = ACPAB[img\_tmp[i][j][B]] \* 255 #do the histogram equalization  
 img\_tmp[i][j][G] = ACPAG[img\_tmp[i][j][G]] \* 255  
 img\_tmp[i][j][R] = ACPAR[img\_tmp[i][j][R]] \* 255  
 return img\_tmp  
  
  
image = cv2.imread('lenna\_rgb.png’)  
image\_temp = copy.deepcopy(image)  
arr = np.array(image)  
imageSize = arr.shape  
  
H = imageSize[0]  
W = imageSize[1]  
IRed = [0]\*256 #initial the RGB value for counting the corresponding number of image  
IGreen = [0]\*256  
IBlue = [0]\*256  
AcPAR = [0]\*256  
AcPAG = [0]\*256  
AcPAB = [0]\*256  
  
CountPixel(image\_temp, IBlue, IGreen, IRed, H, W)  
  
EveryPAR = np.array(IRed)/(H\*W)  
EveryPAG = np.array(IGreen)/(H\*W)  
EveryPAB = np.array(IBlue)/(H\*W)  
  
AcPAR = Accumulate(EveryPAR) #accumulate the number of appear pixel for each channel  
AcPAG = Accumulate(EveryPAG)  
AcPAB = Accumulate(EveryPAB)  
  
image\_temp = ImageHeq(image\_temp, AcPAB, AcPAG, AcPAR, imageSize[0], imageSize[1]) #do the histogram equalization  
print('end')  
cv2.imshow('Before', image)  
cv2.imshow('After', image\_temp)  
cv2.waitKey(0)