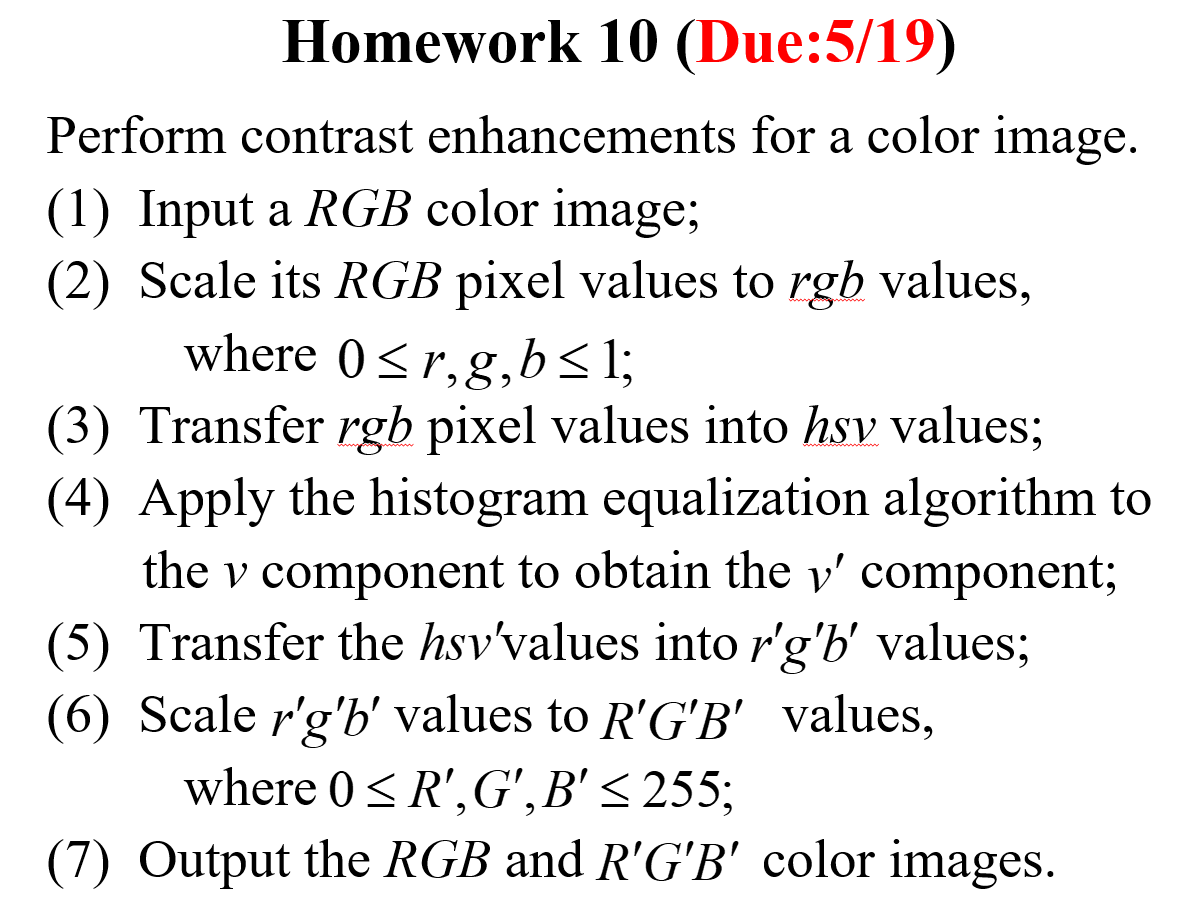
影像處理 Homework 10

資工112 40847015S 紀軒宇

* Statement



* Code
* #import the library
* import cv2
* import numpy as np
* import colorsys
* def histo\_equalization\_HSV(img):
* H, S, V = cv2.split(img)
* n\_img = np.clip(np.round(V\*255), 0, 255).astype('uint8')
* h\_img = n\_img.copy()
* y, x = n\_img.shape
* level = list(range(256))
* img\_1dim = n\_img.flatten()
* his\_data = np.bincount(img\_1dim, minlength=256)
* acc\_data = np.cumsum(his\_data)
* acc\_prob = acc\_data/(x\*y)
* roundvalue = np.rint(acc\_prob\*255).astype('uint8')
* for i in range(y):
* for j in range(x):
* h\_img[i][j] = roundvalue[h\_img[i][j]]
* output = n\_img\*(np.divide(h\_img, n\_img, where=n\_img!=0))
* output = np.clip(output, 0, 255)/255
* output = cv2.merge([H, S, output])
* return output
* img = cv2.imread("C.jpg")
* cv2.imwrite('input.jpg', img)
* img\_norm = img/255
* y, x, c = img.shape
* for i in range(y):
* for j in range(x):
* B, G, R = img\_norm[i][j]
* img\_norm[i][j] = colorsys.rgb\_to\_hsv(R, G, B)
* hsv\_ = histo\_equalization\_HSV(img\_norm)
* new = np.zeros(img.shape, dtype='uint8')
* for i in range(y):
* for j in range(x):
* H, S, V = hsv\_[i][j]
* r, g, b = colorsys.hsv\_to\_rgb(H, S, V)
* new[i][j] = b\*255, g\*255, r\*255
* cv2.imshow('output', new)
* cv2.waitKey(0)
* cv2.imwrite('output.jpg', new)
* 輸入/輸出圖片

輸入



輸出



RGB



R’G’B’

* 心得

這個方法相比作業三（圖如下）直接用灰階來轉換的相比過曝光的感覺少了一點，做到了更自然的提升對比的作用

