**40847041S朱自宇 hw1**

**Problem statement:**

1. Input a color image *C*(*R*,*G*,*B*)

2. Output the color image *C*

3. Transform the color image *C* into a

grayscale image *I* by *I* = (*R+G+B*)/3

4. Show the grayscale image *I.*

**Experimental results:**

* **Input/Output images:**

**Input color image C**



**Output color image C** 

**Output grayscale image I**



* **Source code:**

from tkinter import W

import numpy as np

import cv2

# read the image

img = cv2.imread("takagisan.png", cv2.IMREAD\_COLOR)

# print out the original picture size

# print(img.shape)

"""

The original version of the picture

"""

# show the picture

cv2.imshow("Original", img)

"""

now do the grayscale

"""

# set the gray image variable

grayimg = np.zeros( (img.shape[0],img.shape[1]) , np.uint8 )

# divide the BGR of the original image by 3

for i in range(img.shape[0]):

for j in range(img.shape[1]):

(b,g,r) = img[i,j]

gray = (int(b) + int(g) + int(r))/3

grayimg[i,j] = np.uint8(gray)

# show the grayscale version of the picture

cv2.imshow("Gray", grayimg)

# save the picture

cv2.imwrite("gray\_takagisan.png",grayimg)

# wait until the user press any button to close the img window

cv2.waitKey(0)

cv2.destroyAllWindows()

* **Comments:**

I never use OpenCV before, so I have to learn to use it. After Google it, I found out that this is a nice & easy tool for us to learn image processing. And by searching the bgr2gray function, I learn that normally we don’t just use R:G:B = 1:1:1 to gray a picture; is the common rule when we transform a picture into a gray one.