1. 輸入圖片
2. 輸出圖片
3. Source code:

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

import cv2

def rgb\_to\_gray(rgb):

# I = (R + G + B)/3

return np.dot(rgb[...,:3], [0.333, 0.333, 0.333])

print('Please enter the image name(ex. image.jpg): ')

# get image name

img\_name = input()

# read the image

img = mpimg.imread(img\_name)

print('\nshowing origin image...')

plt.imshow(img)

# disable axis

plt.axis('off')

# show the image

plt.show()

print('\nPocessing the image to grayscale image...\n ')

# calculating

gray\_img = rgb\_to\_gray(img)

print('done')

print(gray\_img)

plt.imshow(gray\_img, cmap = 'Greys\_r')

plt.axis('off')

# save imgage's snapshot

plt.savefig('output\_grayscale\_image.jpg')

# show the grayscale image

plt.show()

1. 心得

一開始簡單的一個作業，在查要如何灰階圖片時，看到了很多內建的函式。但因為作業的要求，還是自定義了一個函式來處理，把每個部分都乘上0.333（雖然查到的灰階函式都是乘上另外一組數字，不是RGB各自平分）。