

電腦視覺 HW1 611415001 康愷辰

Give **two** images (liberty.png and car.png) for testing:

1. Read a RGB image and write a function to convert the image to **grayscale image**.

```
img[:, :, 2]
img[:, :, 1]
img[:, :, 0]
```

首先先讀取影像後將 `img` 的 `bgr` 給讀取出來

將三個通道加權平均，得到灰階圖像

`gray_img = 0.2989 * red + 0.5870 * green + 0.1140 * blue`



2. Write a convolution operation with **edge detection kernel**

Then input to **activation function** (ReLU: Rectified Linear Unit)

定義了一個 3×3 的邊緣檢測核，然後對影像進行卷積運算。我們通過在影像周圍填充 0 來處理邊緣像素，然後進行卷積計算，得到一個新的影像。最後，我們將輸出影像限制在 0 到 255 之間，以便在顯示之前進行正確的縮放和顯示。當進行卷積運算後，通常會進一步進行激活函數的運算，這是為了讓網路能夠將線性轉換後的輸出轉換為非線性。其中，ReLU (Rectified Linear Unit) 是最常用的激活函數之一，它的定義是 $f(x) = \max(0, x)$ 將卷積運算和 ReLU 函數的運算結合起來



3. Write a pooling operation with using **Max pooling, 2x2 kernel, and stride 2**

將 ReLU 函數的結果傳遞給 Max Pooling 函數，通過選擇每個區域中的最大值來實現並且以 stride 為 2 將圖像縮小



4. Write a **binarization operation** (threshold = 128). (≥ 128) set 255 (< 128) set 0

C/C++ and Python are allowed to use (Choose one to finish homework)

Note: Output Q1 -> Input Q2-> Input Q3 -> Input Q4

設定一個閾值設為 128 跑過所有陣列將大於 125 的值都設定成 255

