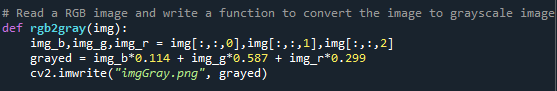
**Computer Vision**

NTUT 資訊工程所111598070 賴俊霖

**Homework 1**

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

In rgb2gray( img ), you can see because of python feature, we can easily to use [:,:,0]~[:,:,2] to get the three BGR channels.

And use the formula , we can get the **grayscale for** human eye.

Then, we use cv2.imwrite to save images.

There are the outputs of Questen1.

Car

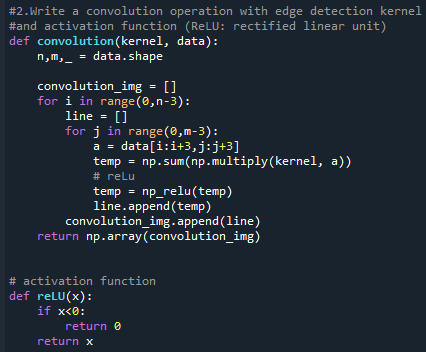
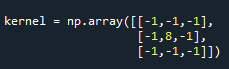


Liberty



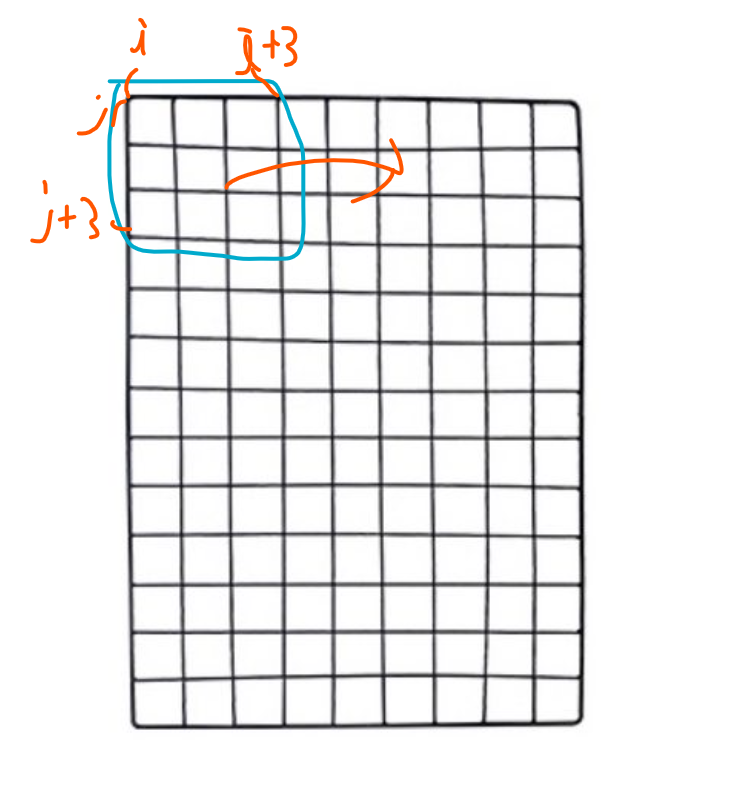
2. Write a convolution operation with **edge detection *kernel***

and **activation function** (ReLU: rectified linear unit)

In convolution(kernel,data),you can see we use n,m,\_ = data.shape to get length ,width and channel from the image.

And using nesting Loop to traversal the image, the a = data[i:i+3,j:j+3] as shown in the figure below.



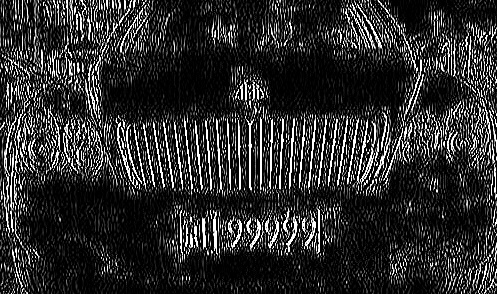
Then we use np.multiply(kernel,a) to do a \* kernel .

And sum the result of multiplication to get the convolution result.

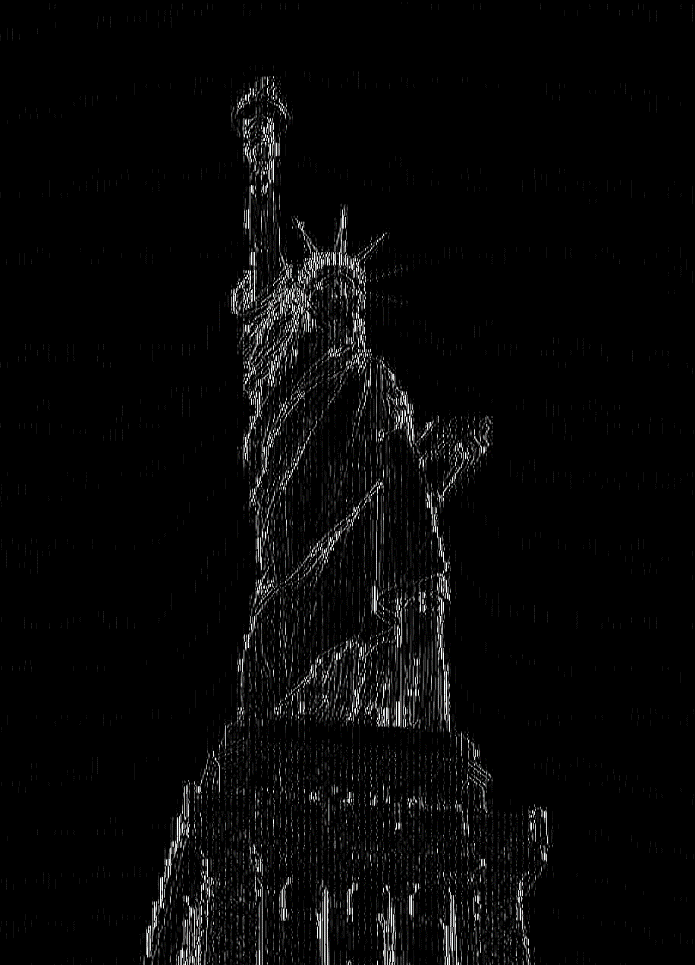
Before executing line.append(temp), we use relu to filter negative numbers.

There are the outputs of Questen2.

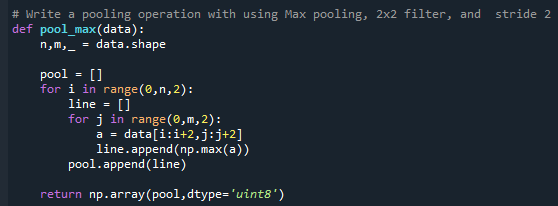
Car



Liberty



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

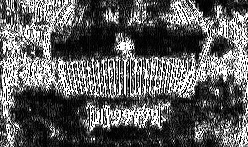


In pool\_max(data), we still use the same approach as mentioned above to traversal the image. But a little different because Q3’s said the stride is 2 which means we gonna add 2 every for loop. Like range(0,n,2). Then before appending , we use np.max() to get the maximum value in the 2x2 a np matrix.

Owing to MaxPooling , we get the small image then before.

There are the outputs of Questen3.

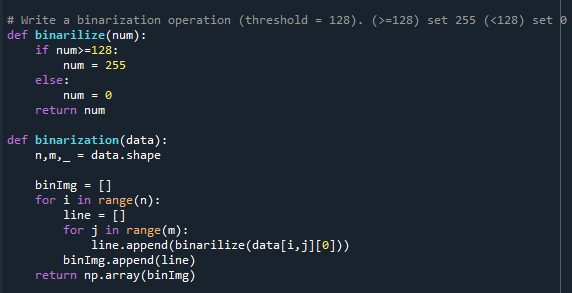
Car



Liberty



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



In binarilize() , it will make every num it read to be **binarization.**

In binarilization() ,there is still the old way I traversal the image.

And append every binarilized number.

There are the outputs of Questen4.

Car



Liberty

