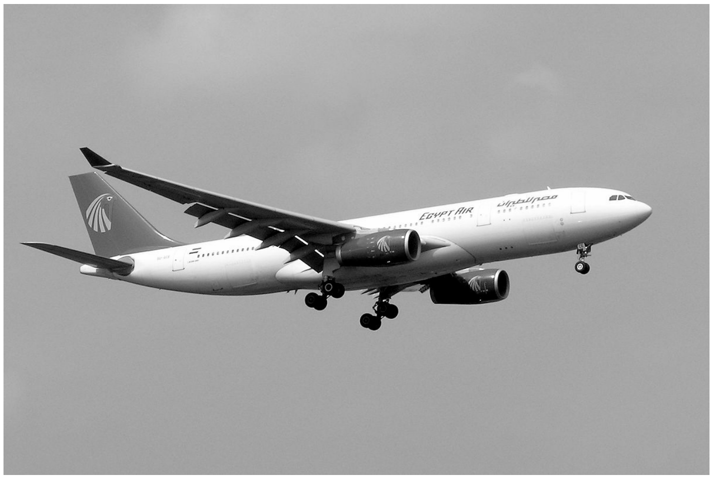
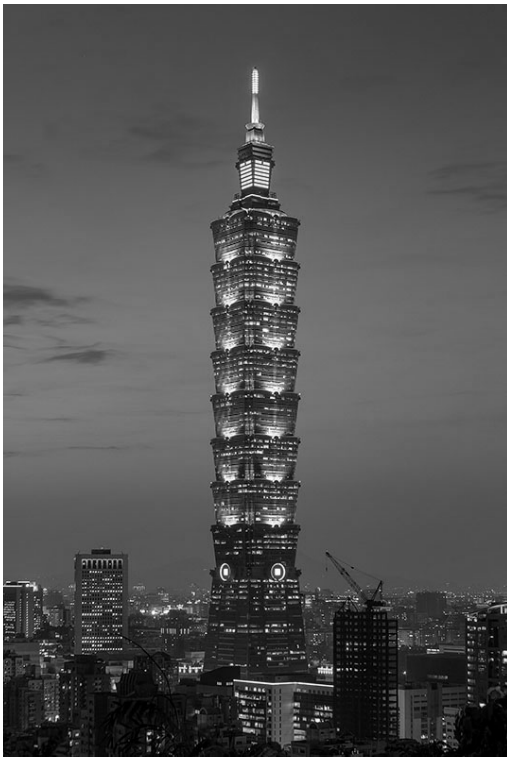
Computer Vision hw1

資工碩一 112598037 洪子翔

Q1

In Q1, it required us to change the input image into grayscale, `rgbToGray()` function was used.
The function use for loop to go through all pixel in the image and change the value in the formula: Pixel value = 0.2989 * red + 0.5870 * green + 0.1140 * blue.
The resulting image will have only one value in each pixel, instead of 3.

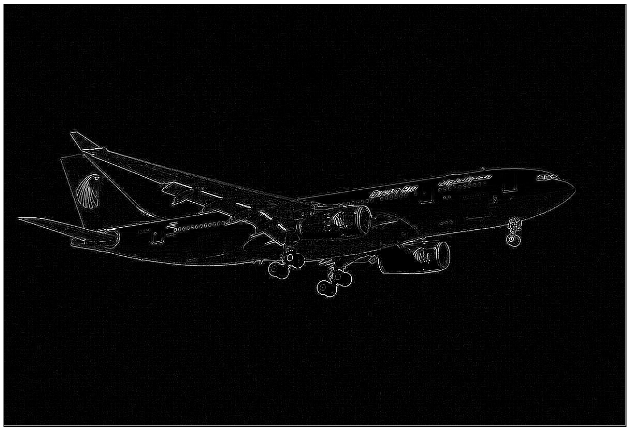


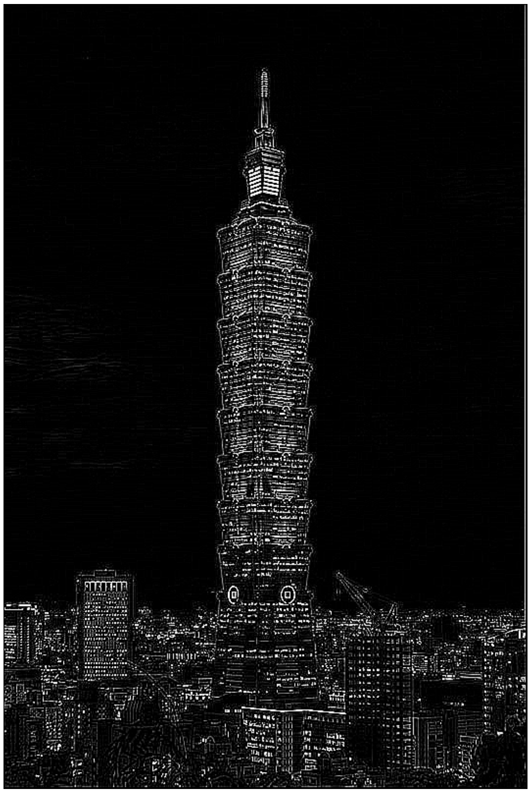


Q2

Question 2 required convolution on each image with edge detection kernel using zero padding and stride 1.
In the solution, function `convolution()`, `zeroPadding()`, `arrayMutipleAndAddAll()` were involved.
Function named convolution will call the zeroPadding function first, able to get the image after zero padding. Then it will traverse all the pixels and do the convolution math

by calling arrayMutipleAndAddAll function. It is important to be aware of the value after computing, that might overflow in the type `uint8`. If overflow occurs, just change the value to the minimum or maximum, depending on whether the value is too large or too small.





Q3

Question 3 asks us to do a pooling operation on both images using max pooling, 2x2 kernel, and stride 2.
In this case, `maxPooling()` was called first, traversing all the pixels by stride 2. Because of stride 2, handling the index became tricky, which I did by calculating it through the ` // ` operator. At every 2 pixels it traversed, `chooseMaxFromArray()` function would be called to do max pooling by choosing the max value from the input 2x2 array.

Q4

Last question, Q4, asked us to do a binarization operation on both images.
The solution is to traverse all pixels and compare it to the threshold, if it is higher or equal, set its value to 255. Otherwize, set it to 0.
In the result image, the threshold had been set to 128.