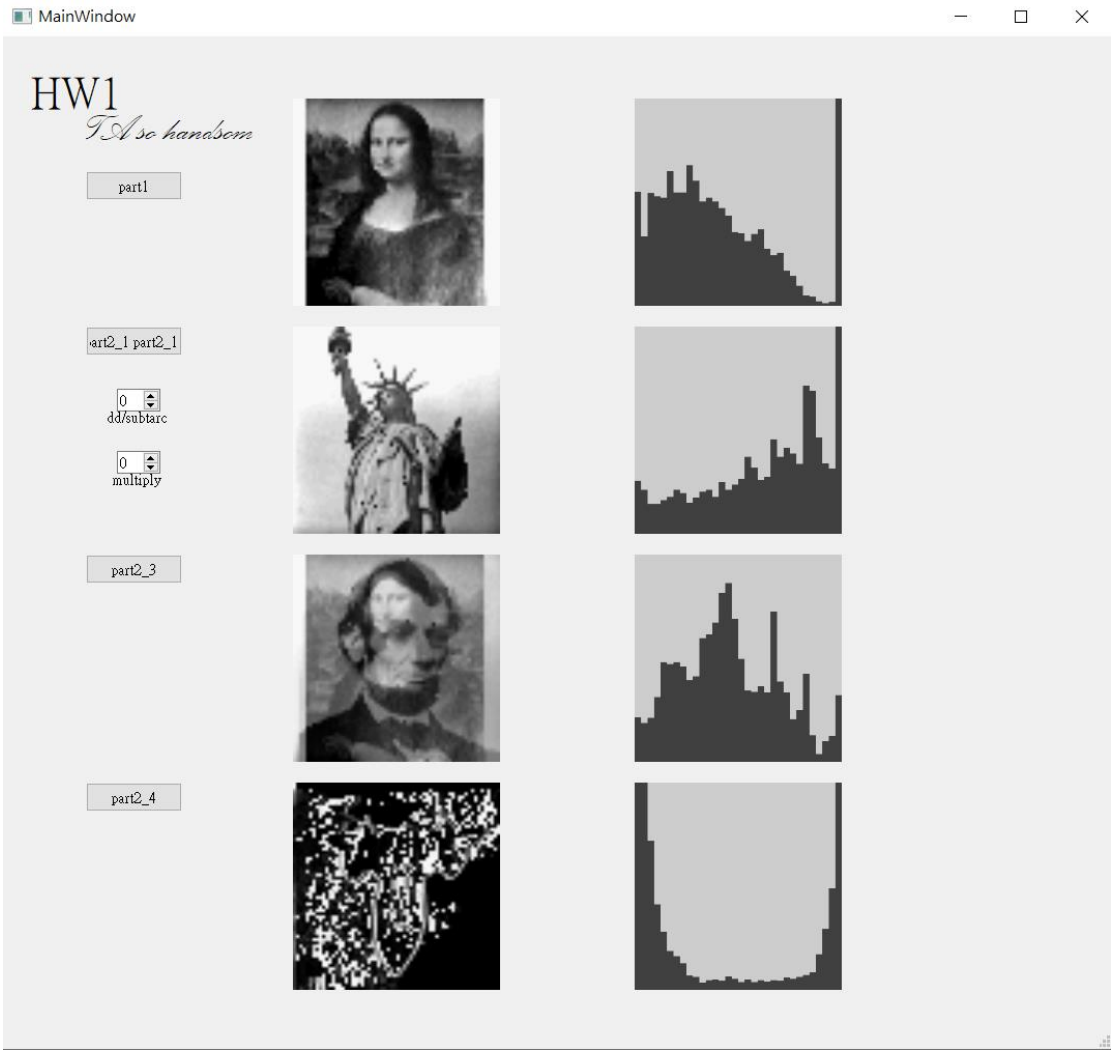
Image processing HW1

Name:武敬祥

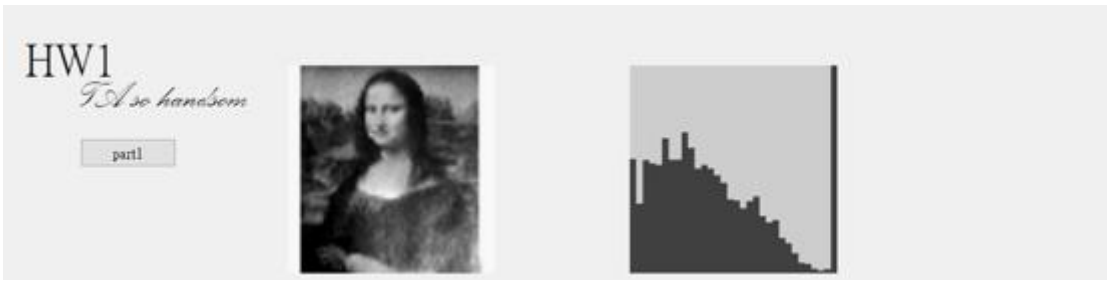
ID:b06611032

GUI



Press button labeled to choose the .64 file

Part 1: Histogram of an Image



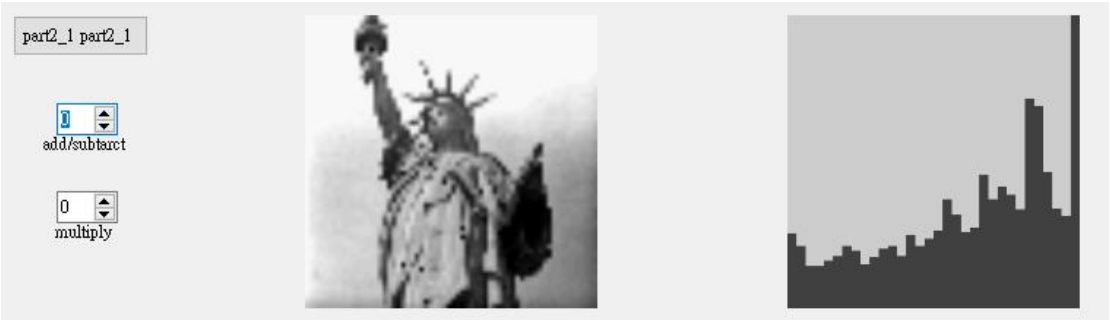
Using c++ package <fstream> to load the .64 file and encode 0~9 , A~Z into 0~31 in order to plot the histogram. Multiplying every pixels by 8 and

using cv::resize, cv::point, cv::line to plot the image on the ui.

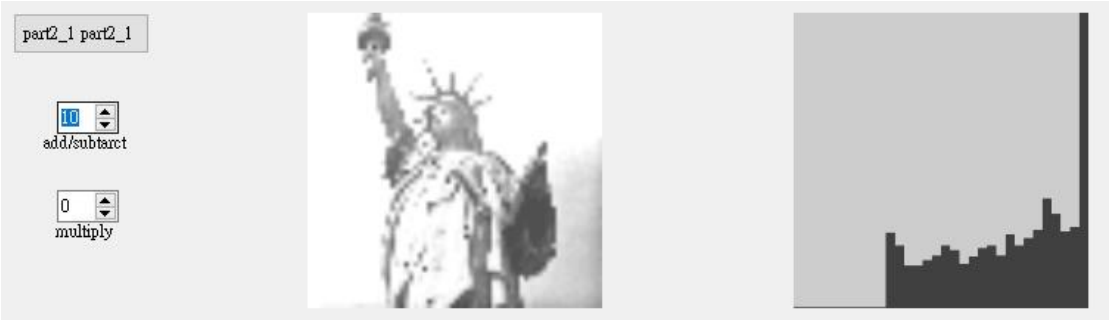
Part 2: Arithmetic Operations of an Image Array.

1. **Add or subtract a constant value to each pixel in the image.**

n=0



n=10



n=20

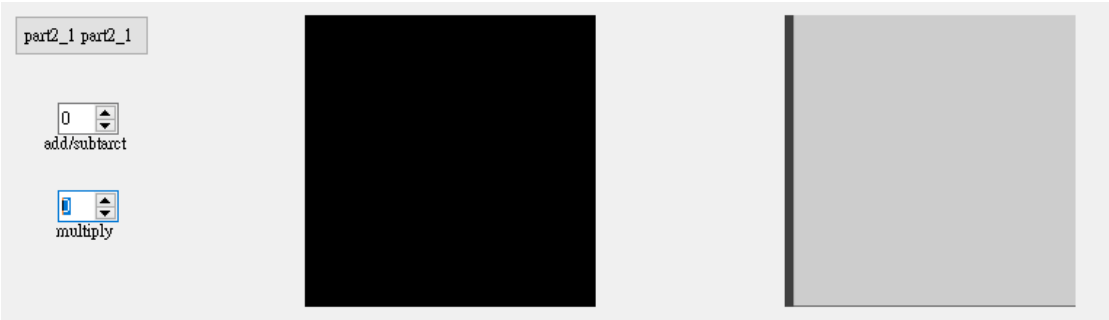
We can decide how much we want to add to every pixel by QSpinBox.

The above is a demonstration of Liberty to add every pixel by n=0, 10, 20, separately. The image is getting brighter when n is larger, and the

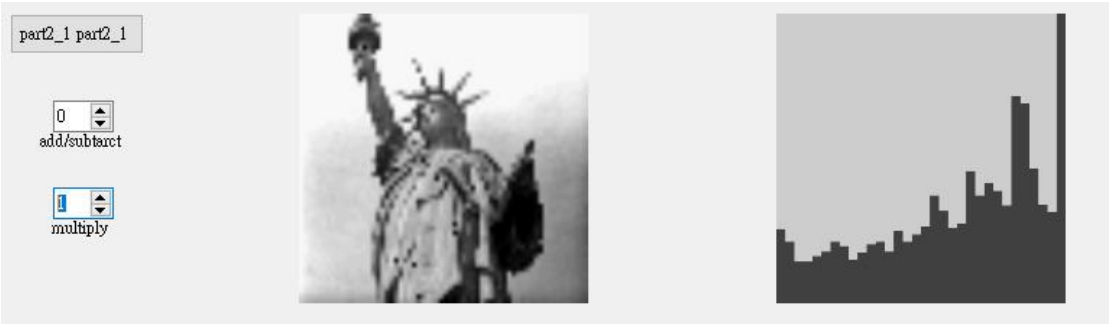
value in the histogram will also fall into 31 gradually. On the other hand, if we subtract a constant number, the image will get darker.

2. **Multiply a constant to each pixel in the image.**

n=0



n=1



n=5

We can get a similar result like add a constant number to every pixel when we multiply a constant number to every pixels. the above is a demonstration of Liberty to add every pixel by n=0, 1, 5, separately.

Notice that when the pixel value is 0. No matter how much we multiply the number to it, it will be 0 forever.

3. **Create a new image which is the average image of two input images.**

This two picture is a little transparent because their pixel values reduce by half.

4. **Create a new image $g(x,y)$ in which the value of each pixel is determined by calculating the pixel values of the input image $f(x,y)$ using the following equation:**

$$g(x,y) = f(x,y) - f(x-1,y)$$

We can detect edges of an object by calculating its gradient.

And this manipulation can be regarded as a subtraction of adjacent pixels. As a result, we can detect the 1D edges in this demonstration.