

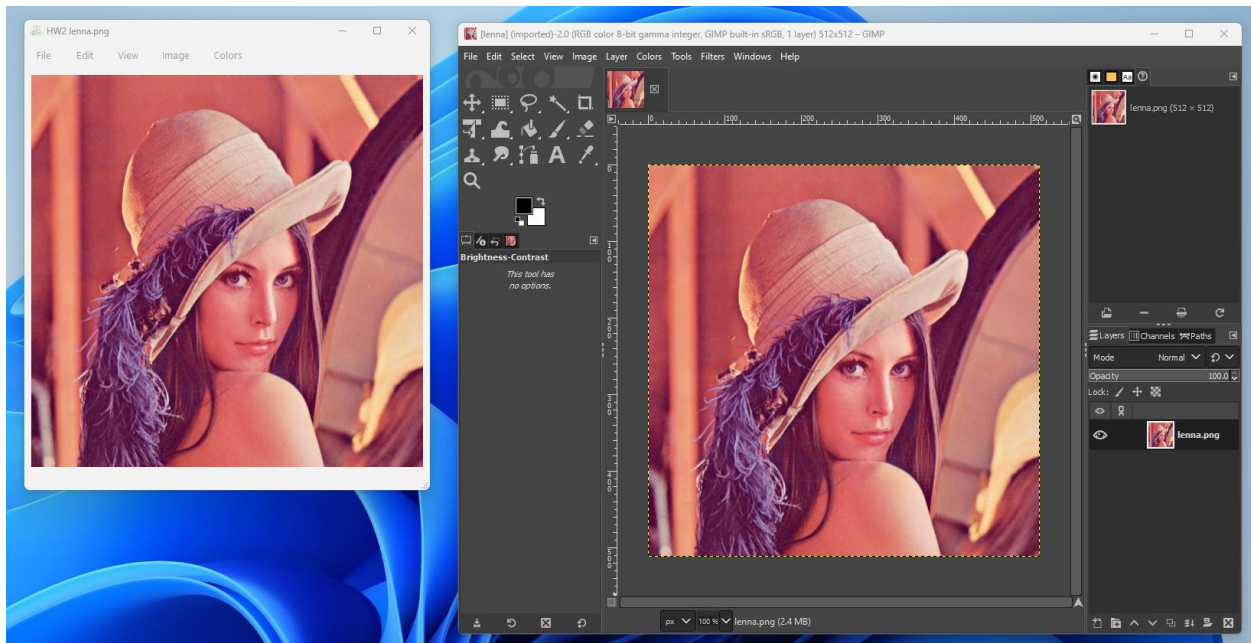
Principles and Applications of Digital Image Processing

Homework 2 report

這次的作業是參考 **GIMP(GNU Image Manipulation Program)**這個影像處理程式來進行 UI 設計的，在 main window 中需要呼叫 sub window 出來並且將 slider 的值傳送至 main window 中，這裡就必須使用到 Qt 一個核心的技術 **signals & slots** 來輔助，除此之外我另外將影像處理的演算法另外寫成 `ImageProcessing.{h, cpp}`進行模組化

1. Read a color BMP or JPEG image file and display it on the screen. You may use the functions provided by Qt, OpenCV, or MATLAB to read and display an image file. (10%)

使用 Qt 中的檔案系統 API 來讀取 `lenna.png` 進 QLabel 中並顯示之，在菜單欄的布局也是參考 GIMP，使用快捷鍵 `Ctrl+O` 開檔案、`Ctrl+Q` 結束程式、`Ctrl+Z/Y` 可以 undo/redo



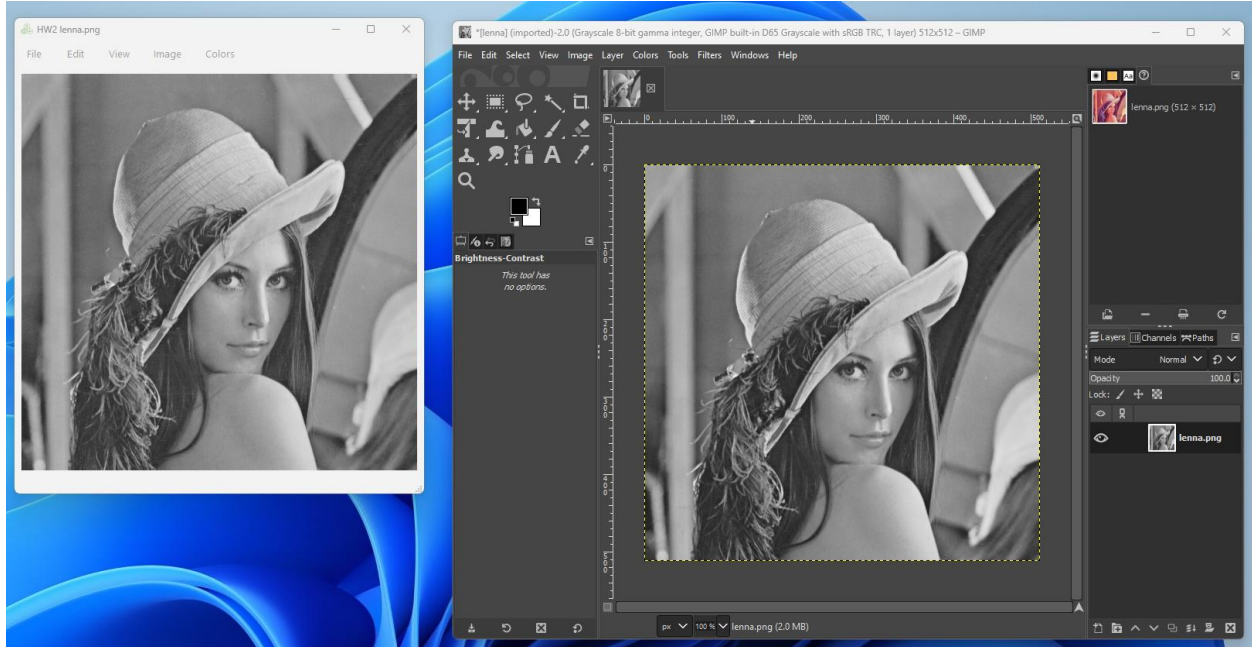
2. Convert a color image into a grayscale image using the following equations:

A. $GRAY = (R+G+B)/3.0$

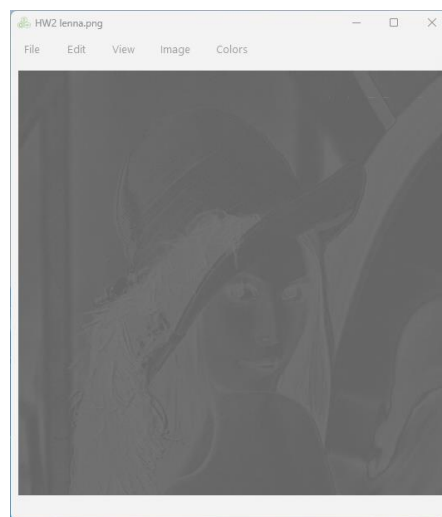
B. $GRAY = 0.299*R + 0.587*G + 0.114*B$

Compare the grayscale images obtained from the above equations. One way to compare the difference between two images is by image subtraction (5%)

Image->Mode 有兩個灰階模式可以選

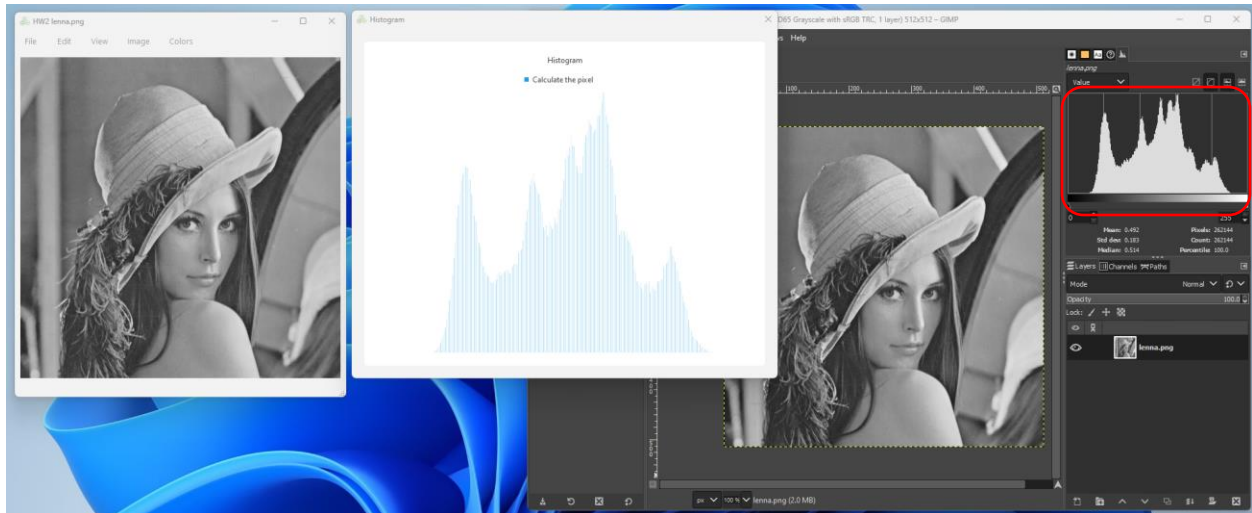


第三個選項可以看出兩種演算法的差異，因為差異不是很明顯，所以我用調整亮度的方式讓影像更清楚點



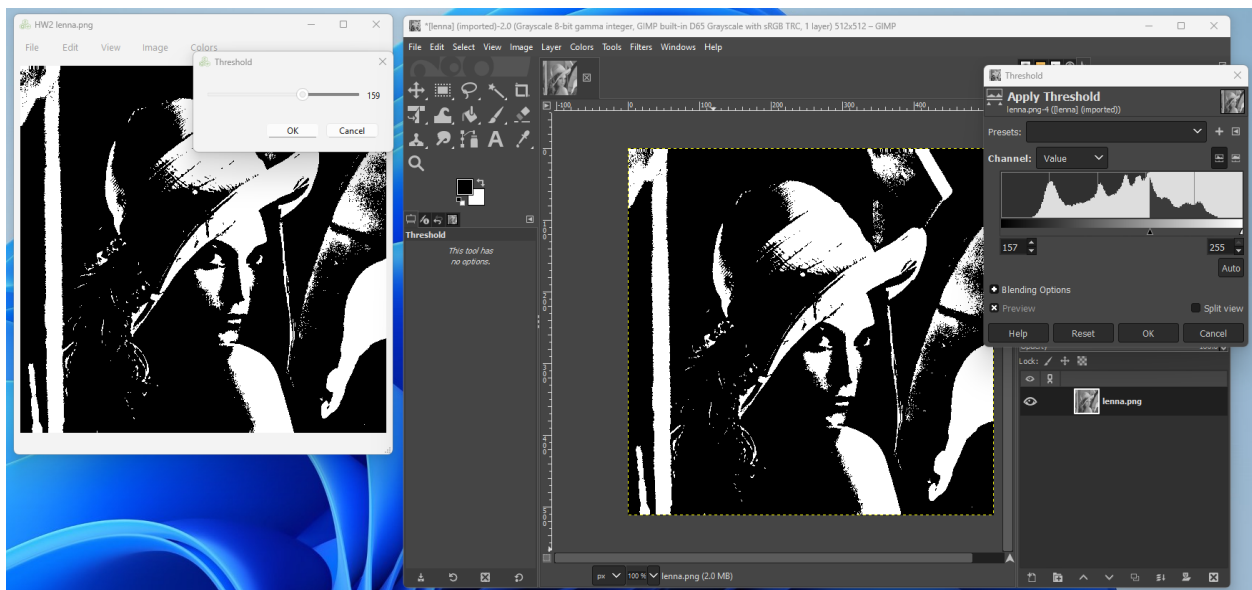
3. Determine and display the histogram of a grayscale image. (10%)

在上面的灰階模式按下後會彈出直方圖，與 GIMP 的直方圖(紅框)的分佈是一樣的



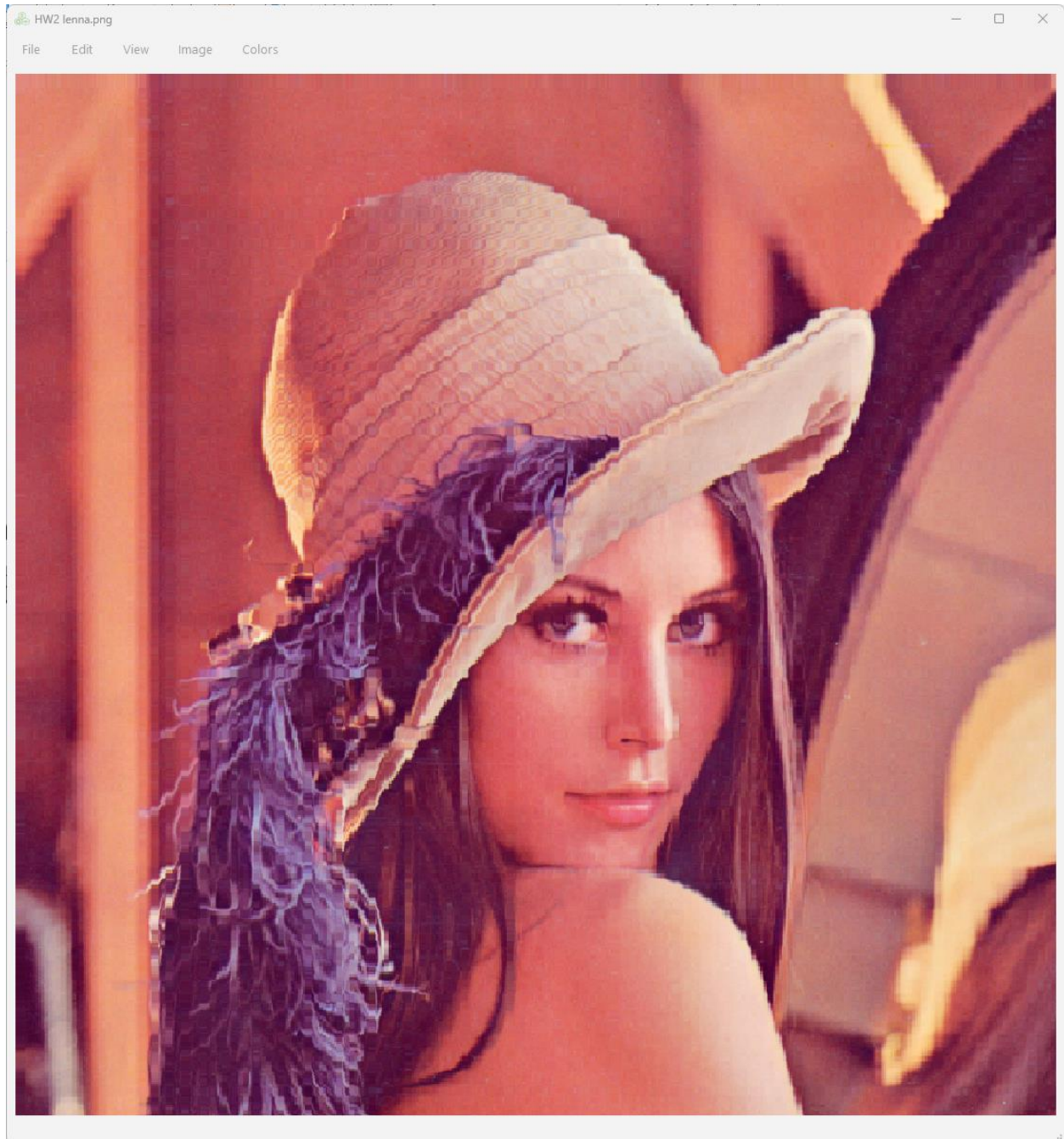
4. Implement a manual threshold function to convert a grayscale image into a binary image. (10%)

Colors->Threshold 中可以透過拉 slider 改變其 threshold



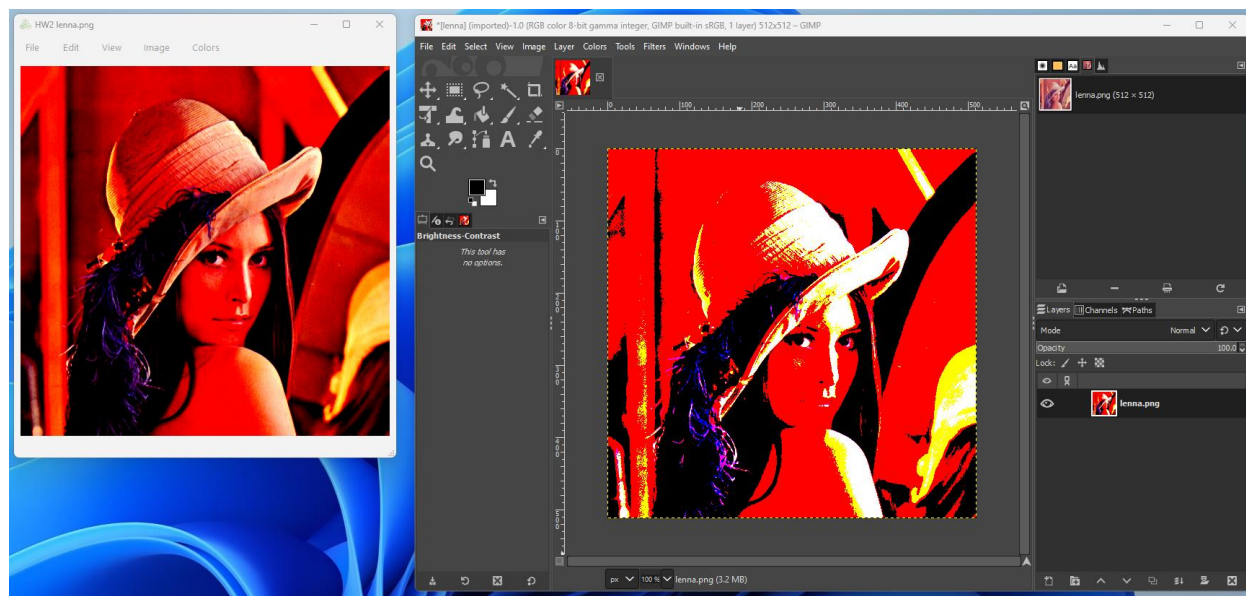
5. Implement a function to adjust the spatial resolution (enlarge or shrink) and grayscale levels of an image. Use an interpolation method on enlarging an image. (10%)

這裡為了實作上的簡單方便，我採取的演算法是 **nearest neighbor interpolation** 來達到縮放的效果，可以透過+(按下 shift 不放並按下=等於符號)/-快捷鍵來完成



6. Implement a function to adjust the brightness and contrast of an image. (10%)

應該是演算法的計算上有差異所以相同的輸入似乎有一點不太一樣



7. Implement a histogram equalization function for automatic contrast adjustment. (15%)

我的直方圖均化只有灰階的影像，而 GIMP 有提供彩色的影像，但大致上來看，直方圖所呈現出的像素分佈是一樣的

