

# 高等電腦視覺

## 作業#(01)

姓名： 吳佩霖

學號： 111c52016

指導老師： 張陽郎

## 程式執行說明

1.

首先選擇要執行 C++ 版本或 openCV 版本

```
C:\Users\wup4416\Desktop\ACV\HW1\x64\Debug\HW1.exe
1: C++ version
2: OpenCV version
0: exit
Choose C++ version or OpenCV version: _
```

2.

選擇要執行的題號

```
C:\Users\wup4416\Desktop\ACV\HW1\x64\Debug\HW1.exe
1: C++ version
2: OpenCV version
0: exit
Choose C++ version or OpenCV version: 1
choose which process want to do
1. Write(.bmp)
2. Generate a color negative image.
3. Split and reassemble into new image.
4. Zoom Lena 512*512 to 1024*1024.
5. Shrink Lena 512*512 to 128*128.
6. Zoom color negative image 512*512 to 1024*1024.
7. Shrink split and reassemble image from 512*512 to 256*256.
8. Exit
_
```

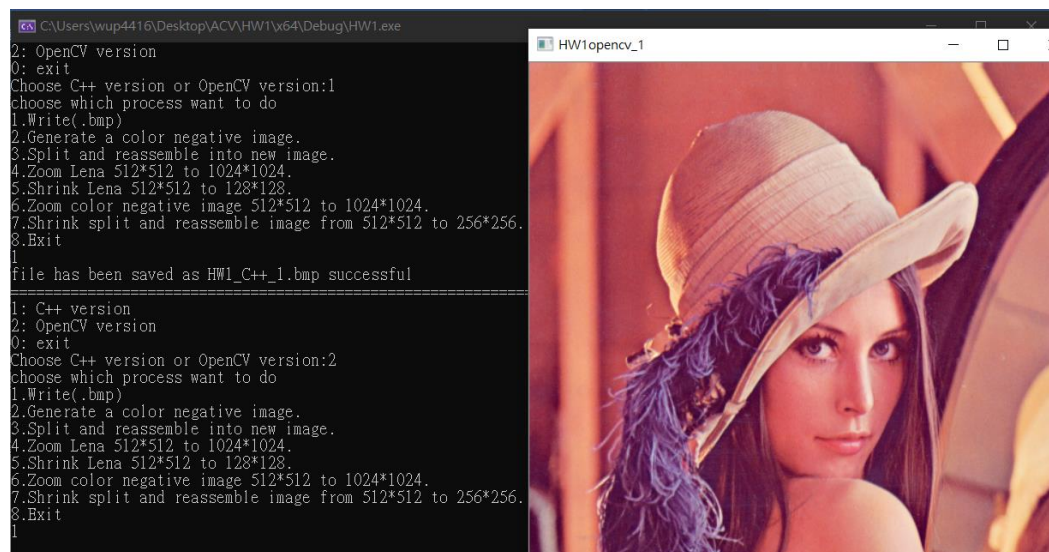
3.



執行完成後會看到“file has been saved 檔名.bmp successful!”的提示，檔案則儲存於 HW1 的資料夾中。



```
C:\Users\wup4416\Desktop\ACV\HW1\x64\Debug\HW1.exe
1: C++ version
2: OpenCV version
0: exit
Choose C++ version or OpenCV version:1
choose which process want to do
1.Write(.bmp)
2.Generate a color negative image.
3.Split and reassemble into new image.
4.Zoom Lena 512*512 to 1024*1024.
5.Shrink Lena 512*512 to 128*128.
6.Zoom color negative image 512*512 to 1024*1024.
7.Shrink split and reassemble image from 512*512 to 256*256.
8.Exit
1
file has been saved as HW1_C++_1.bmp successful
=====
1: C++ version
2: OpenCV version
0: exit
Choose C++ version or OpenCV version:_
```

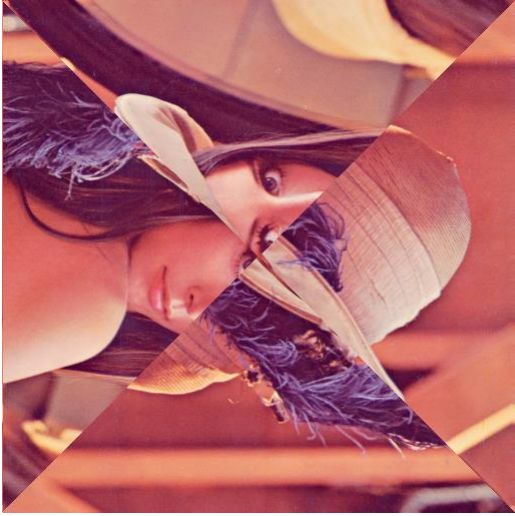
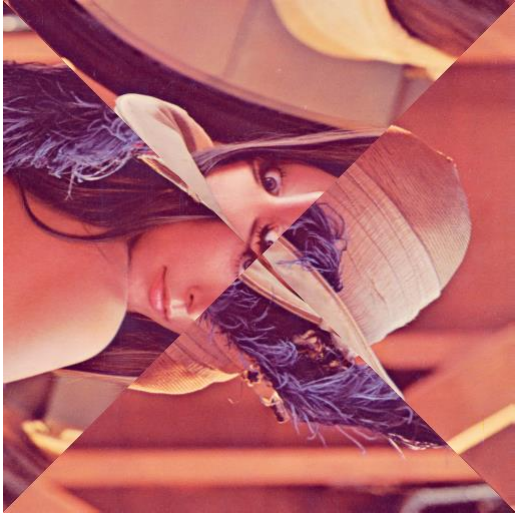
4.

若是執行 openCV 的版本則是會跳出結果的視窗，並如同 C++版本一樣有“file has been saved 檔名.bmp successful!”的提示，即為儲存完成。




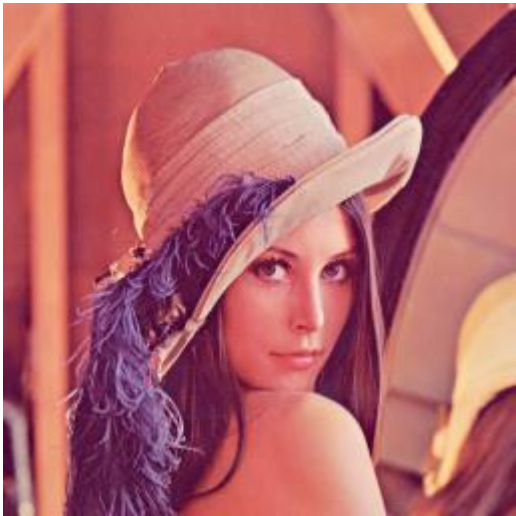





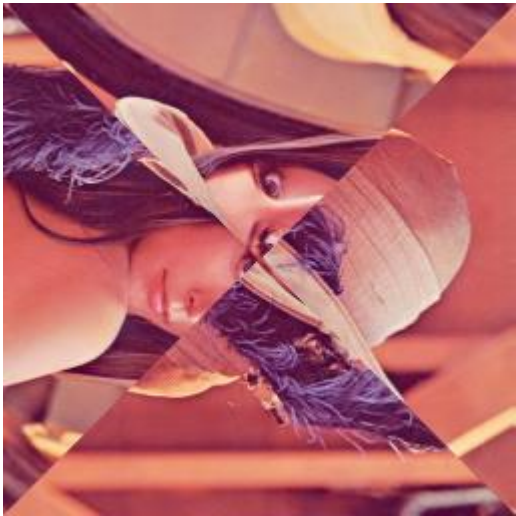
1	
C++	OpenCV
HW1_C++_1.bmp	HW1opencv_1.bmp
	
Discussion_1	
<p>由於 bmp 有標頭檔在讀取上會不如 raw 的直觀簡單，因此讀取的時候需要特別留意標頭與三通道的的部分。</p> <p>在 openCv 則沒有這個問題，直接使用 imread 跟 imwrite 就可以直接讀寫。</p>	

2	
C++	OpenCV
HW1_C++_2.bmp	HW1opencv_2.bmp
	
Discussion_2	
<p>負片的做法基本上就是由 255 扣掉數值，在這邊 C++ 版本有三通道的關係，因此需要扣除 3 次。</p> <p>OpenCV 的部分直接使用 bitwise_not 進行運算，即可得出結果。</p>	

3	
C++	OpenCV
HW1_C++_3.bmp	HW1opencv_3.bmp
	
Discussion_3	
<p>C++版本的旋轉需要經過計算，在取分割內容的部分則跟 openCV 版本無太大差異。</p> <p>openCV 版本的旋轉可由 cv::rotate 完成，之後按照想要的分割內容格是進行取值即完成。</p>	



Bonus1&2	
C++	OpenCV
HW1_C++_bonus1.bmp (1024*1024)	HW1opencv_bonus1.bmp (1024*1024)
	
HW1_C++_bonus2.bmp (256*256)	HW1opencv_bonus2.bmp (256*256)
	
Discussion_3	
<p>C++使用了 nearest_neighbor 進行了放大縮小，在並排上以肉眼觀察 openCV 是使用線性插值，輸出的圖片多少有點不同，但差異並無太大。</p>	

Bonus3&4	
C++	OpenCV
HW1_C++_bonus3.bmp (1024*1024)	HW1opencv_bonus3.bmp (1024*1024)
	
HW1_C++_bonus4.bmp (256*256)	HW1opencv_bonus4.bmp (256*256)
	
Discussion_3	
一樣使用前面題目所得到的結果進行放大與縮小。	