

# 高等電腦視覺

## 作業#1

姓名：\_\_\_\_\_ 巫伯銘 \_\_\_\_\_

學號：\_\_\_\_\_ 111318096 \_\_\_\_\_

指導老師：\_\_\_\_\_ 張陽郎 教授 \_\_\_\_\_

## 作業說明

圖一

名稱	修改日期	類型	大小
input image	2022/10/13 下午 06:48	檔案資料夾	
output image	2022/10/13 下午 03:27	檔案資料夾	
Source Code	2022/10/13 下午 06:48	檔案資料夾	
ACV_HW1.exe	2022/10/13 下午 03:07	應用程式	167 KB
HW#1_111318096.docx	2022/10/16 下午 08:08	Microsoft Word ...	935 KB

圖二

```
D:\1_MASTER\ACV\HW\HW1\submit\ACV_HW1.exe
0:Exit
1:hw_C.cpp
2:hw_C_bonus.cpp
3:hw_opencv.cpp
4:hw_opencv_bonus.cpp
Enter the question number to select output result: _
```

### 說明

#### 1. 檔案說明(如圖一)

- input image 資料夾：存放輸入影像
- output image 資料夾：存放輸出影像
- Source Code 資料夾：存放所有程式檔案
- ACV\_HW1.exe：主程式執行檔案

#### 2. 執行方式

- 點開 ACV\_HW1.exe
- 輸入 1~4 選擇要執行的程式，其中：
  - hw\_C.cpp：使用純 C 撰寫的作業程式
  - hw\_C\_bonus.cpp：使用純 C 撰寫的作業加分題
  - hw\_opencv.cpp：使用 OpenCV 撰寫的作業程式
  - hw\_opencv\_bonus.cpp：使用 OpenCV 撰寫的作業加分題

## 1.

### Coding by C

#### 1. 讀取.bmp 檔案

- 使用 `fopen()` 打開.bmp 檔案。
- 使用 `fread()` 讀取前 54 個 Bytes 並用 `unsigned char*` 將其儲存，再從其中的 [18][19][20][21] 取出影像寬度、[22][23][24][25] 取出影像高度。
- 使用 `fread()` 讀取 header 之後的 Bytes 並用 `unsigned char*` 將其儲存。

#### 2. 寫.bmp 檔案

- 將 header 的相關資訊寫入對應位置，如下圖所示。

```
34 int bmp_write(unsigned char* image, int xsize, int ysize, const char* filename) {
35     unsigned char header[54] = {
36         0x42, 0x4d, 0, 0, 0, 0, 0, 0, 0, 0,
37         54, 0, 0, 0, 40, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 24, 0,
38         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
39         0, 0, 0, 0
40     };
41     long file_size = (long)xsize * (long)ysize * 3 + 54;
42     long width, height;
43     char fname_bmp[128];
44     FILE* fp;
45
46     header[2] = (unsigned char)(file_size & 0x000000ff);
47     header[3] = (file_size >> 8) & 0x000000ff;
48     header[4] = (file_size >> 16) & 0x000000ff;
49     header[5] = (file_size >> 24) & 0x000000ff;
50
51     width = xsize;
52     header[18] = width & 0x000000ff;
53     header[19] = (width >> 8) & 0x000000ff;
54     header[20] = (width >> 16) & 0x000000ff;
55     header[21] = (width >> 24) & 0x000000ff;
56
57     height = ysize;
58     header[22] = height & 0x000000ff;
59     header[23] = (height >> 8) & 0x000000ff;
60     header[24] = (height >> 16) & 0x000000ff;
61     header[25] = (height >> 24) & 0x000000ff;
```

- 使用 `fwrite()` 先將 header 寫入，再將 image 的陣列寫入。

### Using OpenCV

#### 1. 讀取.bmp 檔案

使用 `cv::imread()` 讀取影像。

#### 2. 寫.bmp 檔案

使用 `cv::imwrite()` 寫出影像。

2.

Figure

img\_neg\_C.bmp



lena\_neg\_opencv.bmp



img\_neg\_onehalf\_C.bmp



lena\_neg\_onehalf\_opencv.bmp



img\_neg\_double\_C.bmp



lena\_neg\_double\_opencv.bmp



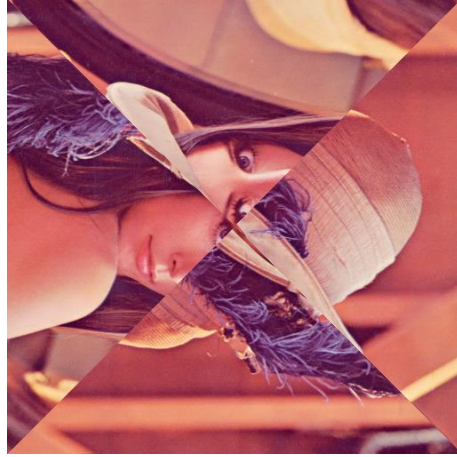
Coding by C
<ol style="list-style-type: none"><li>1. 使用 int x, y, c 的三層 for 迴圈以取得影像的 pixel 位置以及 RGB 三通道的 位置，並在迴圈中用 255 減去每個 pixel 值以達成負片效果。</li><li>2. Bonus： 本題使用 nearest neighbor interpolation 進行縮放，先計算縮放目標的 x, y 在 原圖的 x, y 座標的哪個位置，再取其四捨五入以得到最鄰近的 pixel 值。</li></ol>
Using OpenCV
<ol style="list-style-type: none"><li>1. 使用 bitwise_not()，即對每個 pixel 做 NOT 運算，以達成負片效果。</li><li>2. Bonus： 使用 OpenCV 套件提供的 resize( )進行縮放，縮放方法使用雙線性插值法 (INTER_LINEAR)。</li></ol>

Figure

img\_new\_C.bmp



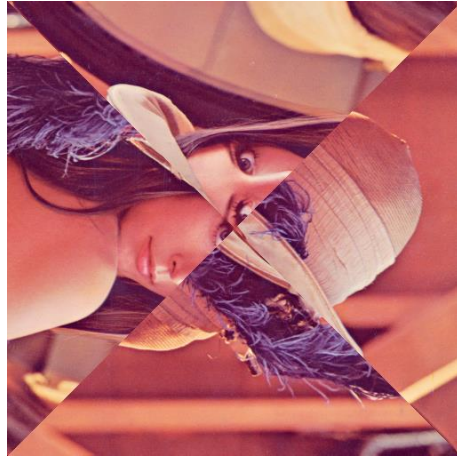
lena\_new\_opencv.bmp



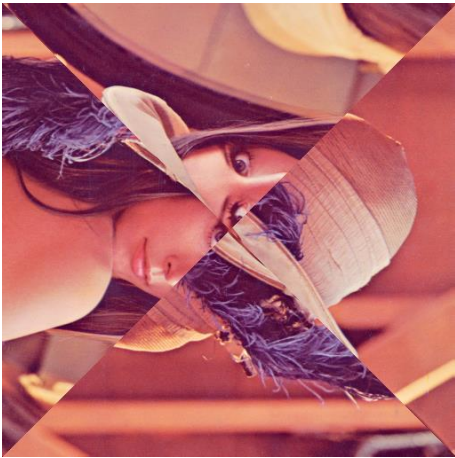
img\_new\_onehalf\_C.bmp



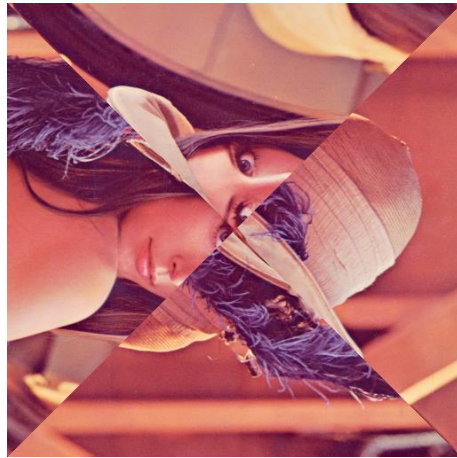
img\_new\_onehalf\_opencv.bmp



img\_new\_double\_C.bmp



img\_new\_double\_opencv.bmp



Coding by C
<ol style="list-style-type: none"> <li>1. 用兩個 unsigned char* 儲存順轉 90 度與逆轉 90 度的影像。</li> <li>2. 在 for 迴圈中利用四個 if 判斷式來切分兩條對角線所形成的四個區塊，第五個 if 來判斷影像的對角線上若有 pixel 值則直接將原圖寫入。</li> </ol> <p>Bonus：</p> <p>本題使用 nearest neighbor interpolation 進行縮放，先計算縮放目標的 x, y 在原圖的 x, y 座標的哪個位置，再取其四捨五入以得到最鄰近的 pixel 值。</p>
Using OpenCV
<ol style="list-style-type: none"> <li>1. 用兩個 Mat 儲存順轉 90 度與逆轉 90 度的影像。</li> <li>2. 在 for 迴圈中利用四個 if 判斷式來切分兩條對角線所形成的四個區塊，第五個 if 來判斷影像的對角線上若有 pixel 值則直接將原圖寫入。</li> </ol> <p>Bonus：</p> <p>使用 OpenCV 套件提供的 resize( ) 進行縮放，縮放方法使用雙線性插值法 (INTER_LINEAR)。</p>