**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    圖二 |
| 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 3. 點開 ACV\_HW1.exe 4. 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 2. 使用fopen( )打開.bmp檔案。 3. 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][23][25]取出影像高度。 4. 使用fread( )讀取header之後的Bytes並用unsigned char\*將其儲存。 5. 寫.bmp檔案 6. 將header的相關資訊寫入對應位置，如下圖所示。      1. 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。   1. 寫.bmp檔案   使用cv::imwrite( )寫出影像。 |

|  |  |
| --- | --- |
| **2.** | |
| Figure | |
| img\_neg\_C.bmp    img\_neg\_onehalf\_C.bmp    img\_neg\_double\_C.bmp | lena\_neg\_opencv.bmp    lena\_neg\_onehalf\_opencv.bmp    lena\_neg\_double\_opencv.bmp |

|  |
| --- |
| Coding by C |
| 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。 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。 |
| Using OpenCV |
| 1. 使用bitwise\_not( )，即對每個pixel做NOT運算，以達成負片效果。 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)。 |

|  |  |
| --- | --- |
| **3** | |
| Figure | |
| img\_new\_C.bmp    img\_new\_onehalf\_C.bmp    img\_new\_double\_C.bmp | lena\_new\_opencv.bmp    img\_new\_onehalf\_opencv.bmp    img\_new\_double\_opencv.bmp |

|  |
| --- |
| Coding by C |
| 1. 用兩個unsigned char\*儲存順轉90度與逆轉90度的影像。 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。   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。 |
| Using OpenCV |
| 1. 用兩個Mat儲存順轉90度與逆轉90度的影像。 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。   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)。 |