電機所控制組 110318051 張皓博

HW1\_opencv講解

我使用Visual Studio來寫這個C++作業，執行方式為把.exe檔跟InputImage.bmp放在同個資料夾底下後，執行.exe即可產生：

1. Rotated.bmp (順時針90度的圖片)
2. BlueImage.bmp (G→0,R→0的RGB圖片)
3. GreenImage.bmp (B→0,R→0的RGB圖片)
4. RedImage.bmp (B→0,G→0的RGB圖片)
5. doubleSize.bmp (放大兩倍的圖片)
6. oneHalfSize.bmp (縮小1/2倍的圖片)

我分別寫了6個function：

1. **void bmpRead():**

用以讀取InputImage.bmp，利用opencv的imread函數讀圖。

1. **void bmpWrite(char const\* pictureName, Mat OutputImage):**

用以輸出圖片，利用opencv的imwrite函數輸出，可以傳入圖片名稱及圖片的imageData。

1. **void ImageRotation():**

利用opencv的transpose函數,把圖片經過一次轉置，再經由opencv的flip函數,對轉置後的圖片做垂直鏡像,等同順時針旋轉90度。

1. **void channelSeparation():**

[1]先利用opencv的split函數，將旋轉後的圖片拆分成BGR三層,放到BGR\_channels[0],BGR\_channels[1],BGR\_channels[2]。

[2]利用Mat::zero函數製作一層純為0的channel,用來跟BGR單層合併。

[3]最後分別將(B,0,0)、(0,G,0)、(0,0,R)合併成三張只有單一色的RGB圖。

1. **void double\_Size():**

利用opencv的resize函數調整圖片的Height跟Width，將旋轉後的圖片放大兩倍。

1. **void one\_half\_Size():**

利用opencv的resize函數調整圖片的Height跟Width，將旋轉後的圖片放大兩倍。

最後在main()呼叫這些函數，以完成作業要求。

使用opencv跟自己手刻相比行數減少很多，許多函式都已經把複雜的操作流程打包起來了，寫起來也相對輕鬆。