



### 影像風格轉換之研究

指導老師：莊政宏老師

組員：林哲宏、賴可軒

本研究打算利用Python的OpenCV套件來進行圖片風格轉換，藉此來分析出最好的圖片轉換效果，並且設計一個GUI介面，可依照使用者的需求來調整轉換的幅度和特殊效果的使用。

另外，我們也嘗試進行圖像區域分割(Graph Based Segmentation)，來分辨出圖片內的不同物件，並以顏色顯示。

```
def createAdjustableImg():  
    adjustable_img = cv.Canny(blur_img,  
        TrackBars_value[0], TrackBars_value[1])  
    if TrackBars_value[2]:  
        adjustable_img =  
            cv.bitwise_or(cv.dilate(adjustable_img,  
                (1, 1), anchor = (0, 0), iterations =  
                TrackBars_value[2]),  
                cv.dilate(adjustable_img, (1, 1),  
                    iterations = TrackBars_value[2]))  
    adjustable_img = cv.bitwise_not(adjustable_img)  
    adjustable_img = cv.cvtColor(adjustable_img,  
        cv.COLOR_GRAY2BGR)  
    adjustable_img = cv.bitwise_and(img,  
        adjustable_img)  
    cv.imshow('Adjustable Image', adjustable_img)
```



原始圖片



根據輸入的圖片  
產生相對應的邊框



使用者調整後  
可立即顯示最終  
結果在視窗中

一開始我們先用高斯模糊去除雜訊的圖片，這裡使用canny方法去尋找「邊緣」，在此需要兩個門檻值。

**TrackBars\_value**陣列記錄了調整的門檻值，若**TrackBars\_value[2]**不為0，那麼就會執行判斷式內的程式碼。

**TrackBars\_value[2]**值是dilate方法的iterations帶入值，也就是「疊代次數」，疊代的次數越多代表邊緣的線越粗，但因為dilate方法會根據anchor參數決定往下或往上而使得繪製出來的邊緣線變粗，因此再使用「OR運算」將分別往上與往下的線結合，避免繪製方向單一。

之後將處理好圖片用「NOT」運算反轉，這會使黑白調換，使得繪製出黑線而不是白線。

為了使得原始圖片可以與邊緣互相接合，須將圖片從灰階轉回BGR，也就是「24位元」圖片因為灰階圖只有「8位元」，直接結合會因為陣列的大小不同而出現錯誤。

最後使用「AND運算」轉成24位元的邊緣圖像陣列與原圖結合並顯示出使用者所要圖片。