

1. 專案實作環境：

1. 作業系統：Windows 10
2. 程式語言與版本：Python 3.9.5
3. OpenCV 版本：4.9.0.80

2. 實作方法流程與參數使用說明：

1. 定義函式 `readImage()` 進行讀取圖片，利用 `cv2.imread()` 讀入圖片。

```
def readImage(pic):  
    image = cv2.imread("hw1_picture/hw1_picture/" + pic)  
    return image
```

2. 定義函式 `toGray()` 進行轉成灰階圖片，利用 `cv2.cvtColor( img , cv2.COLOR_BGR2GRAY )`。

```
def toGray(img, pic):  
    image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    cv2.imwrite("output_picture/gray_" + pic, image)  
    return image
```

3. 定義函式 `toBinFilter()` 將圖片進行雙邊模糊，利用 `cv2.bilateralFilter()`。  
直徑為 50，混和顏色範圍 50，影響範圍為 1000。

```
def toBinFilter(img, pic):  
    image = cv2.bilateralFilter(img, 50, 50, 1000)  
    cv2.imwrite("output_picture/filter_" + pic, image)  
    return image
```

4. 定義函式 `toThreshold()` 將圖片進行二值化，利用 `cv2.threshold()`，使用 `type = cv2.THRESH_OTSU` 自訂義門檻值。

```
def toThreshold(img, pic, maxval):  
    ret, thresh = cv2.threshold(img, 100, maxval, cv2.THRESH_OTSU)  
    cv2.imwrite("output_picture/binarization_" + pic, thresh)  
    return thresh
```

5. 定義函式 `toMorphology()` 將圖片進行形態學開運算，kernal 採用 5\*5，迭代 3 次。

```
def toMorphology(img, pic):  
    kernal = numpy.ones((5, 5), numpy.uint8)  
    image = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernal, iterations=3)  
    cv2.imwrite("output_picture/morphology_" + pic, image)  
    # return image
```

6. 定義函式 `drawRectanglePutText()` 將 people 圖片進行畫 rectangle 與 putText，利用 `cv2.rectangle()` 與 `cv2.putText()` 完成。

rectangle 範圍為 ( 150, 200 ) 至 ( 600, 600 )，BGR 為 ( 0, 120, 120 )，粗細為 5。

text 起始點為 (150, 190)，正常尺寸，BGR 為 (0, 255, 0)，粗細為 2。

```
def drawRectanglePutText(img, pic):  
    image = cv2.rectangle(img, (150, 200), (600, 600), (0, 120, 120), 5)  
    image2 = cv2.putText(image, "Face", (150, 190),  
                          cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)  
    cv2.imwrite("output_picture/label_" + pic, image2)
```

7. 建立主程式 main()，依序執行程式。

```
def main():  
    print("Start reading pictures")  
    img1 = readImage('pic1.jpg')  
    img2 = readImage('pic2.jpg')  
    img3 = readImage('pic3.jpg')  
    print("Finish reading")  
    print("Start To Gary")  
    G_img1 = toGray(img1, 'pic1.jpg')  
    G_img2 = toGray(img2, 'pic2.jpg')  
    G_img3 = toGray(img3, 'pic3.jpg')  
    print("Finish to Gray")  
    print("Start to Filter")  
    B_img1 = toBinFilter(G_img1, 'pic1.jpg')  
    B_img2 = toBinFilter(G_img2, 'pic2.jpg')  
    B_img3 = toBinFilter(G_img3, 'pic3.jpg')  
    print("Finish to Filter")  
    print("Start to Threshold")  
    T_img1 = toThreshold(B_img1, 'pic1.jpg', 300)  
    T_img2 = toThreshold(B_img2, 'pic2.jpg', 300)  
    T_img3 = toThreshold(B_img3, 'pic3.jpg', 300)  
    print("Finish to Threshold")  
    print("Start to Morphology")  
    toMorphology(T_img1, 'pic1.jpg')  
    toMorphology(T_img2, 'pic2.jpg')  
    toMorphology(T_img3, 'pic3.jpg')  
    print("Finish to Morphology")  
    print("Start to Draw Rectangle")  
    People_img = readImage('people.jpg')  
    drawRectanglePutText(People_img, 'people.jpg')  
    print("Finish drawing Rectangle and putting Text")
```

3. 遇到困難與解決方法：

1. 濾波：使用各種濾波模式對於圖片清晰化效果不佳。

解決：選擇使用雙邊模糊對於圖片效果並重複更改參數確認。

2. 形態學：使用 Open 運算以及 Close 運算效果不佳。

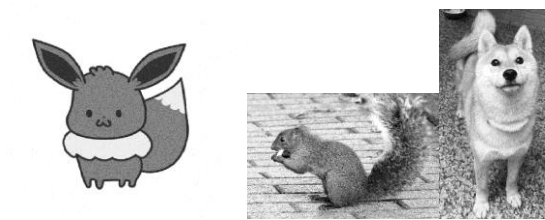
解決：重複改變迭代次數並觀察效果是否更佳，經測試 iterator = 3 時效果比較好。

#### 4. 個人所學與心得：

本次作業為 OpenCV 影像處理應用的作業，目的為熟悉 OpenCV 各項影像處理的工具與函式，我認為比較大的困難是判定處理後效果是否有更好，以及重複改變參數使之能有更佳の影像結果，需要重複調整，對於本次作業，使得我再次體會到訓練資料預處理の困難。

#### 5. 各項結果圖片：

灰階：



濾波：



二值化：



形態化：



繪製圖形：

