

# 1122 Digital Image Processing Assignment #4

學號: 1093338 姓名: 簡毓臻

主題: 膚色偵測 **Skin Color Detection**

專案目標:

撰寫一個程式來偵測照片中的皮膚顏色區域並將其標示出來。使用附件中的 6 張照片及膚色標準答案 (Ground Truth) 做測試, 並呈現所設計膚色偵測方法執行結果的 IOU (Intersection Over Union) 數值 (每張圖片值及所有 6 張圖片平均值)。

開發環境:

- 用的作業系統: Windows
- 開發環境: VScode
- 用的套件: OpenCV version: 4.9.0, numpy, matplotlib.pyplot
- 程式語言: Python 3.12.0

程式架構與功能說明:

- 先將圖片轉成 HSV, 這樣可以更準確提取出圖片的皮膚區域, 不會受光影影響。

```
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

- 選出 lower & upper 的皮膚閾值範圍。

```
lower_skin = np.array([0, 40, 120], dtype=np.uint8)
upper_skin = np.array([30, 200, 255], dtype=np.uint8)
```

- 將 hsv\_image 中位於 lower\_skin 和 upper\_skin 之間的像素值標示為白色 (255), 將其他像素標示為黑色。

```
skin_mask = cv2.inRange(hsv_image, lower_skin, upper_skin)
```

- 填充不連續區域, 得到更準確的皮膚範圍。

```
kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
skin_mask = cv2.morphologyEx(skin_mask, cv2.MORPH_CLOSE, kernel)
```

- 創建一個與原始圖片大小一樣的空白圖片 result\_image, 將前面標記為皮膚部分的地方設為白色。

```
result_image = np.zeros_like(image)
result_image[skin_mask != 0] = (255, 255, 255)
```

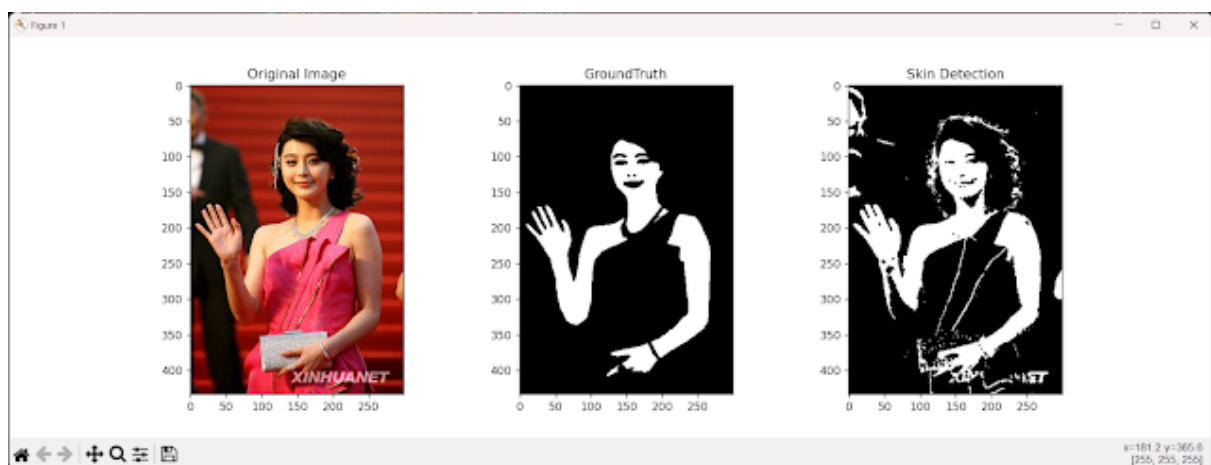
- 計算 IOU, IOU 越大代表程式偵測與標準答案 (Ground Truth) 越接近, 代表偵測結果越好。

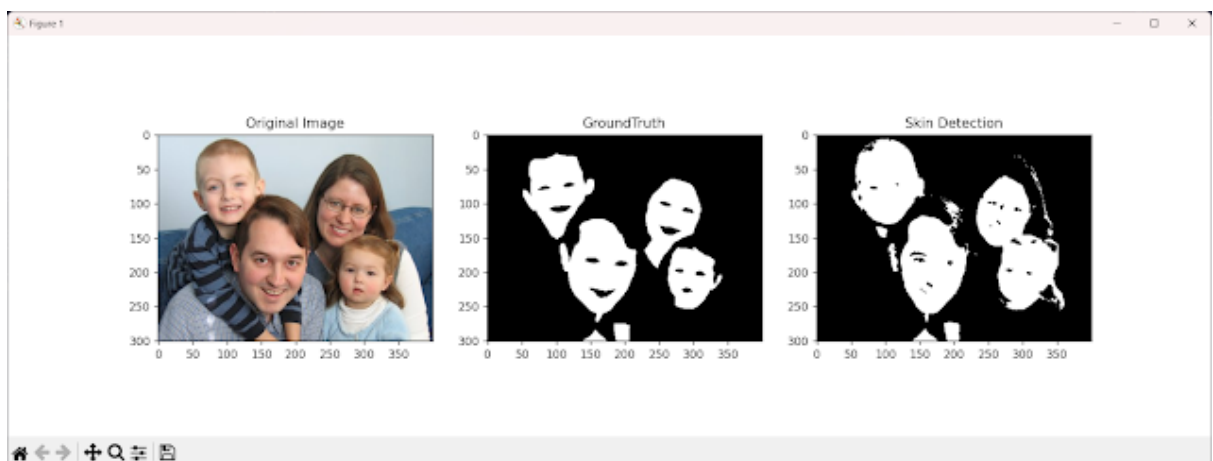
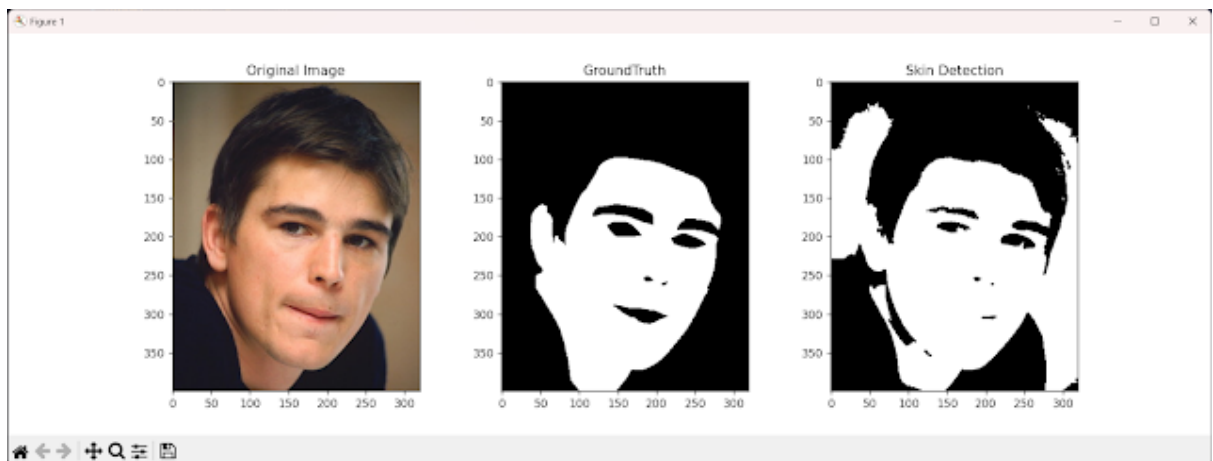
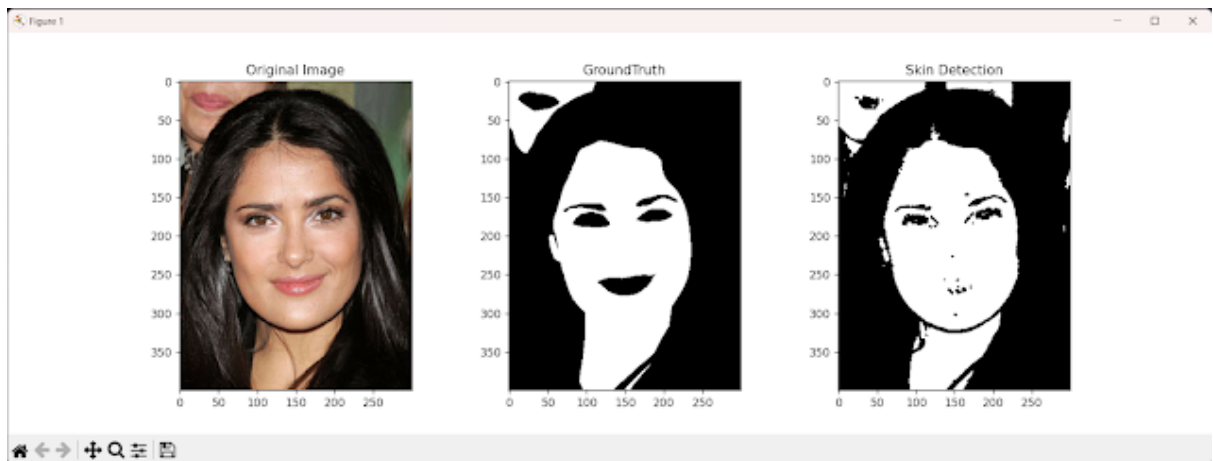
$$\text{IOU} = \frac{\text{你的方法所計算出的膚色區域像素} \cap \text{膚色標準答案區域像素}}{\text{你的方法所計算出的膚色區域像素} \cup \text{膚色標準答案區域像素}}$$

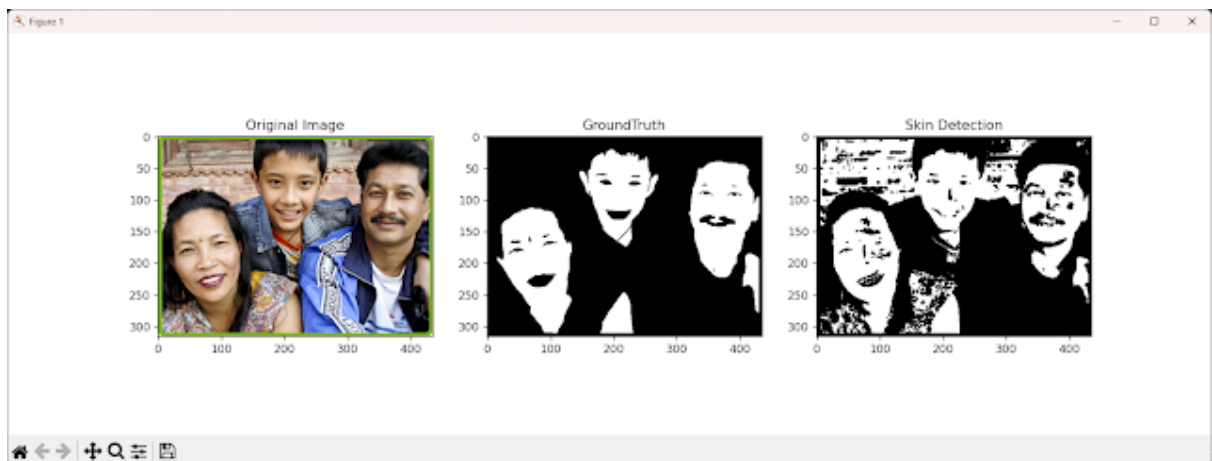
```
def intersection_over_union(mask1, mask2):  
  
    intersection = np.logical_and(mask1, mask2)  
  
    union = np.logical_or(mask1, mask2)  
  
    iou_score = np.sum(intersection) / np.sum(union)  
  
    return iou_score
```

\* 嘗試過結合 HSV 及 YCbCr, 發現單用 HSV 效果就不錯了。

成果展示與討論:







```
IoU for pic1.jpg : 0.6306701881809178
IoU for pic2.jpg : 0.8571793398454958
IoU for pic3.jpg : 0.6415887717703214
libpng warning: iCCP: known incorrect sRGB profile
IoU for pic4.jpg : 0.7338982576838101
IoU for pic5.jpg : 0.3704494016490071
IoU for pic6.jpg : 0.5826955746556103
Average IoU: 0.6360802556308605
```

\* warning 可能是因為圖片中包含的 sRGB 色彩配置文件不正確，但不影響程式運算