1132 Digital Image Processing Assignment #5 書面報告

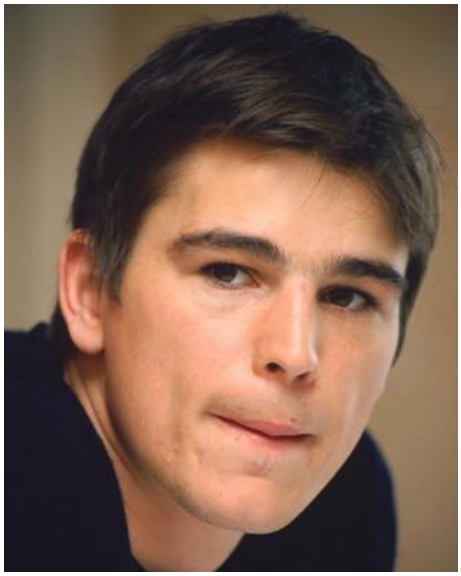
學號：1110927　　姓名：陳柏翰

主題: 膚色偵測 Skin Color Detection

利用在影像處理概論所學習顏色(Color)的知識與技術，撰寫一個程式來偵測照片中的皮膚顏色區域並將其標示。

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)。

以作業所附之 pic2.jpg 為例：



開發環境：Microsoft Windows 11, Visual Studio Code, OpenCV 4.11.0, NumPy 2.2.5, Python 3.10.6, Matplotlib 3.10.3

演算法與程式碼說明：

1. **色彩學範疇膚色偵測範圍**

   基於參考論文中的膚色判斷條件，調整參數最佳化六張圖像的偵測結果，包含：RGB 判斷 (rgb_condition)、HSV 判斷 (hsv_condition)以及 YCbCr 判斷 (ycbcr_condition)。

```python
def is_skin(r, g, b, h, s, y, cr, cb):
    """
    判斷一個像素是否為膚色。
    """
    rgb_condition = (
        r > 100 and g > 45 and b > 30 and r > g and r > b and abs(r - g) > 15
    )
    hsv_condition = s > 35
    ycbcr_condition = (
        cr > 145
        and cb > 75
        and y > 80
        and cr <= (1.9 * cb) + 40
        and cr >= (0.2448 * cb) + 7.209
        and cr >= (-4.5652 * cb) + 234.5652
        and cr <= (-1.25 * cb) + 301.75
        and cr <= (-2.2857 * cb) + 432.85
    )

    return rgb_condition and hsv_condition and ycbcr_condition
```

2. 膚色偵測主流程

- 圖片讀取

  讀取輸入圖與對應的 Ground Truth 掩膜圖。
- 高斯模糊

  降噪平滑圖像，減少雜訊影響膚色判斷。
- 建立空白遮罩

  建立與輸入圖同尺寸的全黑遮罩以標記膚色。
- 逐像素檢查

  逐像素轉換為 HSV 與 YCbCr 色彩空間並判斷是否為膚色。
- 膚色標記

  若通過 is_skin 判斷，該像素遮罩設為白色（255）。
- 形態學開運算

  去除遮罩中小型雜點，強化主體輪廓。
- 形態學閉運算

  填補遮罩內細小空洞，形成完整膚色區塊。
- 再次高斯模糊

  平滑處理後遮罩邊緣，減少銳利雜訊。

```python
def skin_detect(image_path, ground_truth_path):
    image = cv2.imread(image_path)
    ground_truth = cv2.imread(ground_truth_path, cv2.IMREAD_GRAYSCALE)

    image = cv2.GaussianBlur(image, (3, 3), 0)

    height, width, channels = image.shape

    skin_mask = np.zeros((height, width), dtype=np.uint8)

    for y in range(height):
        for x in range(width):
            b, g, r = image[y, x]

            hsv = cv2.cvtColor(np.uint8([[[b, g, r]]]), cv2.COLOR_BGR2HSV)[0][0]
            h, s, v = hsv[0], hsv[1], hsv[2]

            ycbcr = cv2.cvtColor(np.uint8([[[b, g, r]]]), cv2.COLOR_BGR2YCrCb)[0][0]
            yc, cr, cb = ycbcr[0], ycbcr[1], ycbcr[2]

            if is_skin(r, g, b, h, s, yc, cr, cb):
                skin_mask[y, x] = 255
            else:
                skin_mask[y, x] = 0

    # 後處理
    kernel = np.ones((3, 3), np.uint8)
    skin_mask = cv2.morphologyEx(skin_mask, cv2.MORPH_OPEN, kernel, iterations=2) # 開運算
    skin_mask = cv2.morphologyEx(skin_mask, cv2.MORPH_CLOSE, kernel, iterations=2) # 閉運算

    skin_mask = cv2.GaussianBlur(skin_mask, (3, 3), 0)

    iou = calculate_iou(skin_mask, ground_truth)

    return skin_mask, iou
```

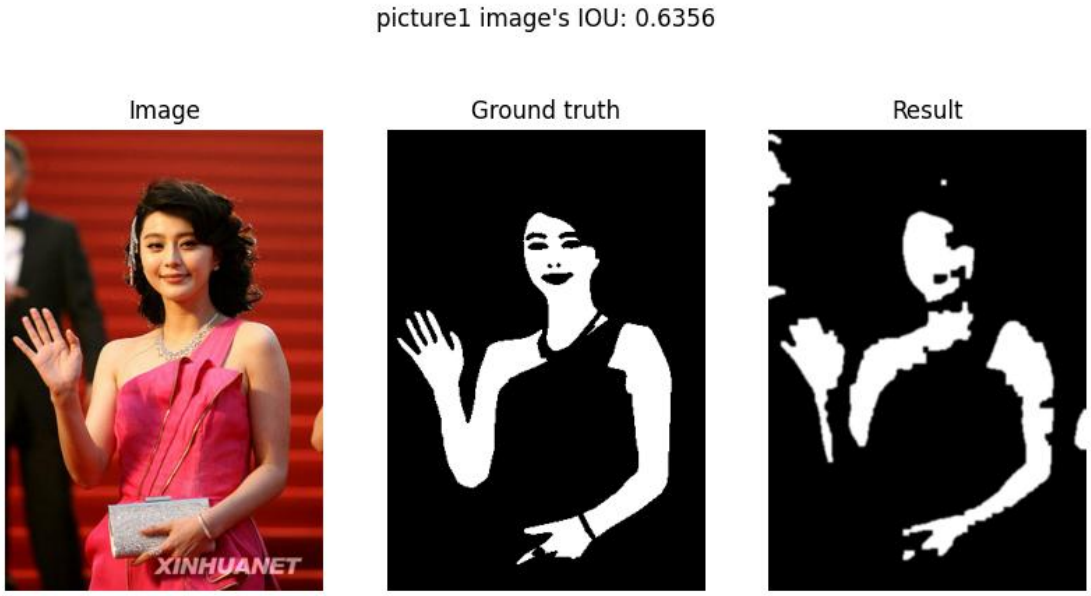
3. 計算 IOU

使用與 Ground Truth 的交集與聯集計算膚色偵測準確度。

```
def calculate_iou(mask1, mask2):
    """計算兩個二值化遮罩的 IOU"""
    intersection = np.logical_and(mask1, mask2)
    union = np.logical_or(mask1, mask2)
    iou = np.sum(intersection) / np.sum(union)

    return iou
```

執行成果：



picture2 image's IOU: 0.9101

picture3 image's IOU: 0.8822

picture4 image's IOU: 0.7276

Image

Ground truth

Result

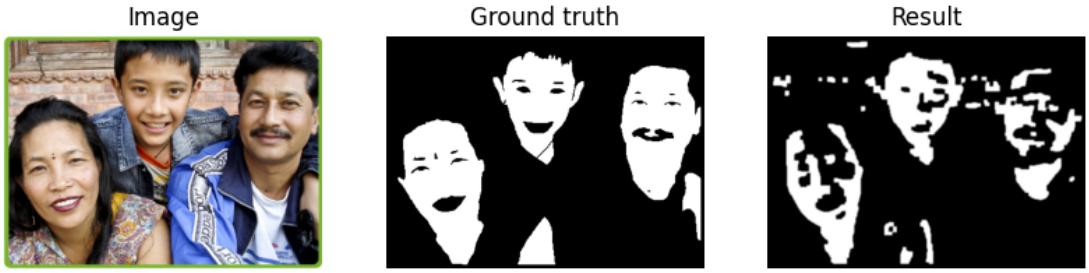
picture5 image's IOU: 0.6009

Image

Ground truth

Result

picture6 image's IOU: 0.5971



心得與討論：

最初，我根據作業論文的方法，在處理偵測膚色十，融合多種顏色空間並引入高斯模型。膚色在不同的顏色空間中具有特定的範圍，因此，我同時在 RGB, HSV 和 YCbCr 三個顏色空間中設定、調整閾值，並取其交集，希望藉此提高準確性（基於論文的數值設定進行改動）。但很明顯的，顏色資訊具有明顯的局限性。當背景與膚色高度相似時（如 pic05.jpg），單純依靠顏色很難區分，且在不同的光照條件、人種差異，都會導致膚色超出預設的範圍，造成誤判。

所以我嘗試了顏色空間以外的檢測方法：包含空間資訊、紋理特徵、連通分析，去除噪聲等方法，試圖使整體圖像都獲得良好的成果。不幸的是，大多數的方法僅在特定圖片上具有漂亮的成績，但是對於其他圖片卻並未如此，整體平均分數呈現下降的趨勢。我亦嘗試將多種方法合併處理，但皆未取得良好的結果。甚至嘗試不再執著於固定的顏色範圍，而是嘗試根據影像內容（亮度、色域等），動態地調整顏色閾值，但仍未取得較亮眼的分數。

沒有萬能的方法，只有最適合特定場景的方案。最終，我將顏色空間分析、統計方法與簡單的空間資訊結合起來，但仍未取得整體多組圖像檢測分數的進步。透過最佳閥值調整以及後續圖像處理（開閉運算、高斯模糊）將 iou 分數提高至 0.7。

Google Site 網址：https://sites.google.com/view/swsekai-dip/assignment-5