

DIP FINAL PROJECT

杜家漢

前言

我們這組在傳統的方法跟深度學習的方法都有使用，目的是想了解哪種方法能更完好的切割水體，另外也想深入了解一下針對不同的圖片，這兩種方法會不會呈現不同程度的結果，以下會分別介紹 **conventional method** 和 **deep learning-based method** 兩個部分，最後會總結兩種方法的優劣之處以及還可以改善的方向。

一、Conventional Method

在 **conventional method** 的部分中，主要分為三個部份來進行水體的分割及優化，分別是特徵分割、**k-means segmentation** 和 **opening closing** 處理。

1.特徵分割

在特徵分割的部分，我將照片資訊轉到了 **HSV** 空間中進行處理，**HSV** 分別代表了色调 **Hue**、飽和度 **Saturation**、明度 **Value**，這種表達方式是相對較符合人眼且能針對圖像做更精細的分析，而除了這種分割以外我也有使用過 **RGB** 和紋理方式(**Gray level**

Co-occurrence Matrix GLCM)，其中 **RGB** 是以紅、綠、藍來記錄每個 **pixel** 的資訊，但無法將其他資訊分開處理(如飽和度 亮度)，且在實際執行後發現以 **HSV** 來辨別的效果較佳且能處理較多方面的資訊，故先把 **RGB** 的處理方式排除，而 **GLCM** 是通過考慮 **pixel** 間的共生關係，來測量不同的灰度值在一定距離和方向上的共生現象，而雖然這個方法能得到對比度、均勻度、同質性等性質，但他並非能針對每一個 **pixel** 做分析，而是需要針對一定的區塊來進行處理，在實際操作後發現對於這次的題目方向較難實施，故最後決定使用 **HSV** 空間的特徵值來操作，決定好使用 **HSV** 空間來當作特徵提取後，由以下步驟達成目標。

Step1.

撰寫 code 將訓練用的 60 張 image 和 mask data set 讀取進來，進行初步處理

Step2.

針對 mask 為 1 的部分套用在 image 上，只保留水體的部分，並計算出 60 張圖片分別的 HSV 值，並計算出相對應的平均值跟標準差

Step3.

分別使用 H S V 找出相對應的 mask，不斷調整參數直至找到最佳結果，我們最後使用的是以下範圍來當作相對應的 mask，唯有 pixel 符合以下三個條件(以平均值和標準差設置範圍)才會被保留下來，至此完成初步的分割。

```
hueRange = [mean(allHueValues) - 1.5*std(allHueValues), mean(allHueValues) + 1.5*std(allHueValues)];
```

```
satRange = [mean(allSatValues) - 1.6*std(allSatValues), mean(allSatValues) + 1.6*std(allSatValues)];
```

```
valRange = [mean(allValValues) - 1.5*std(allValValues), mean(allValValues) + 1.5*std(allValValues)];
```

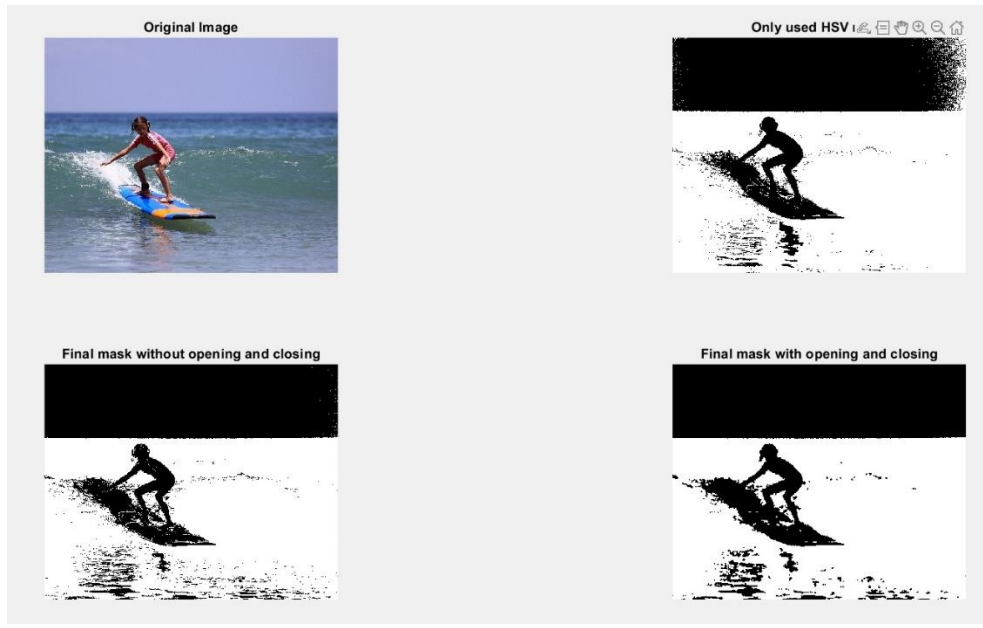
2. k-means segmentation

透過老師上課教的 k-means segmentation 將原圖分割為多個區塊，區塊的數量也是透過諸多嘗試才得到最後的成果，並將上述由 HSV 數值所得到的 mask 套用在每個區塊上，若此區塊內 **70%以上**的 pixel 是符合 mask 的條件，則將此區塊視為水體，而我們的想法是，單純透過 HSV 的 mask 可能會將一些零星出現的 pixel 也視為水體，為了盡量消除此情況，我們加入了這個限制條件，而成果也消除了許多錯誤的 pixel。

3. opening closing 的後處理

經過 k-means 和 HSV mask 的分割後已經有了初步雛形，最後經過 opening closing 的後處理讓圖像更加完整。

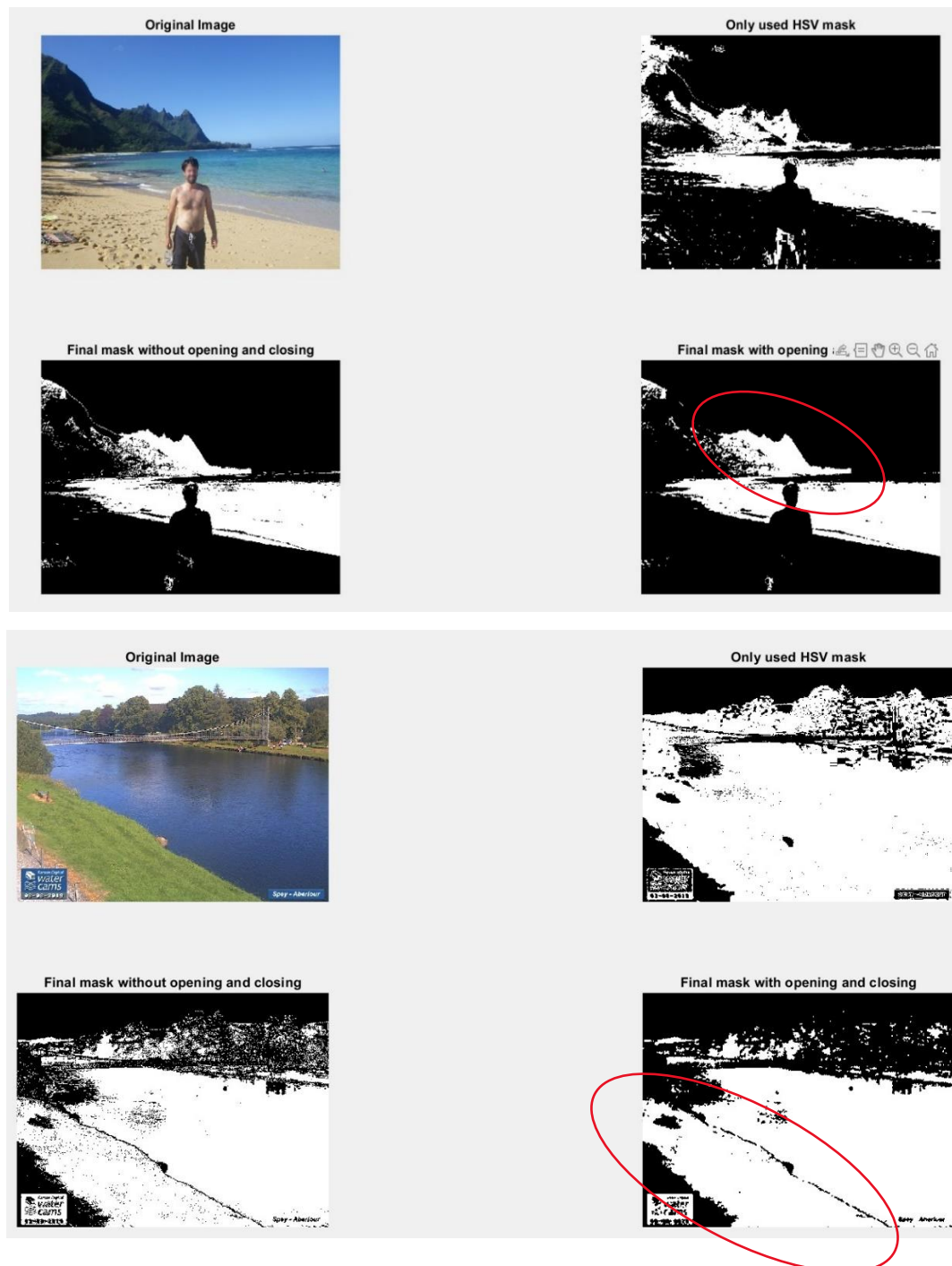
圖型處理結果

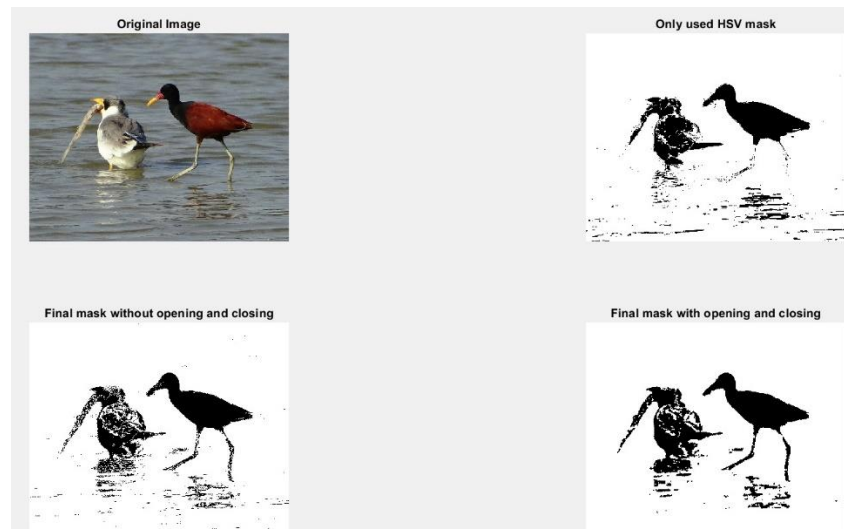


由上圖可以看出來各個步驟造成的結果，若只使用 HSV 的 mask，會造成一些非水體的 pixel 被誤認為水體，但當套用 k-means 分割區域後，那些白點因區塊未達到標準(未達 70%)所以就被判定為非水體了，最後使用 opening 和 closing 將一些小白點和小黑點補起來。



在上下兩張圖片中，指出了一個問題，因為 dataset 不足造成的一些誤差，在這次的 dataset 中有許多水體的顏色是偏土色和綠色，而統計下來導致一些相近顏色的物體(大多為樹林)也被歸納為水體，而我認為增加較多正常顏色的 dataset 便可解決此問題。





在圖片相對單純時，表現的較好

Deep Learning method

考量到傳統架構上可能會有出現錯誤的情況，我們決定也嘗試看看深度學習架構在影像分割上是不是會有更好的效果。

選擇模型:

深度學習架構針對這類的問題，主要有以下幾種模型可以處理這類的問題

U-Net：U-Net 是一種經典的卷積神經網路結構，特別適用於醫學圖像分割。它結合了編碼器和解碼器，有助於保留高解析度的特徵信息。

Mask R-CNN：基於 Faster R-CNN 的 Mask R-CNN 在目標檢測的基礎上增加了對每個目標實例的分割功能，是常用的實例分割模型。

FCN（Fully Convolutional Network）：FCN 是最早期的端對端全卷積神經網路，能夠直接在像素級別進行圖像分割。

考量到我們的資料集只有 60 張，在極小的樣張下，我們挑選模型上主要是選擇有 pretrained model 為主。因此選擇使用 FCN 為基礎，一直以來都很流行的 YOLO 來做圖像分割。

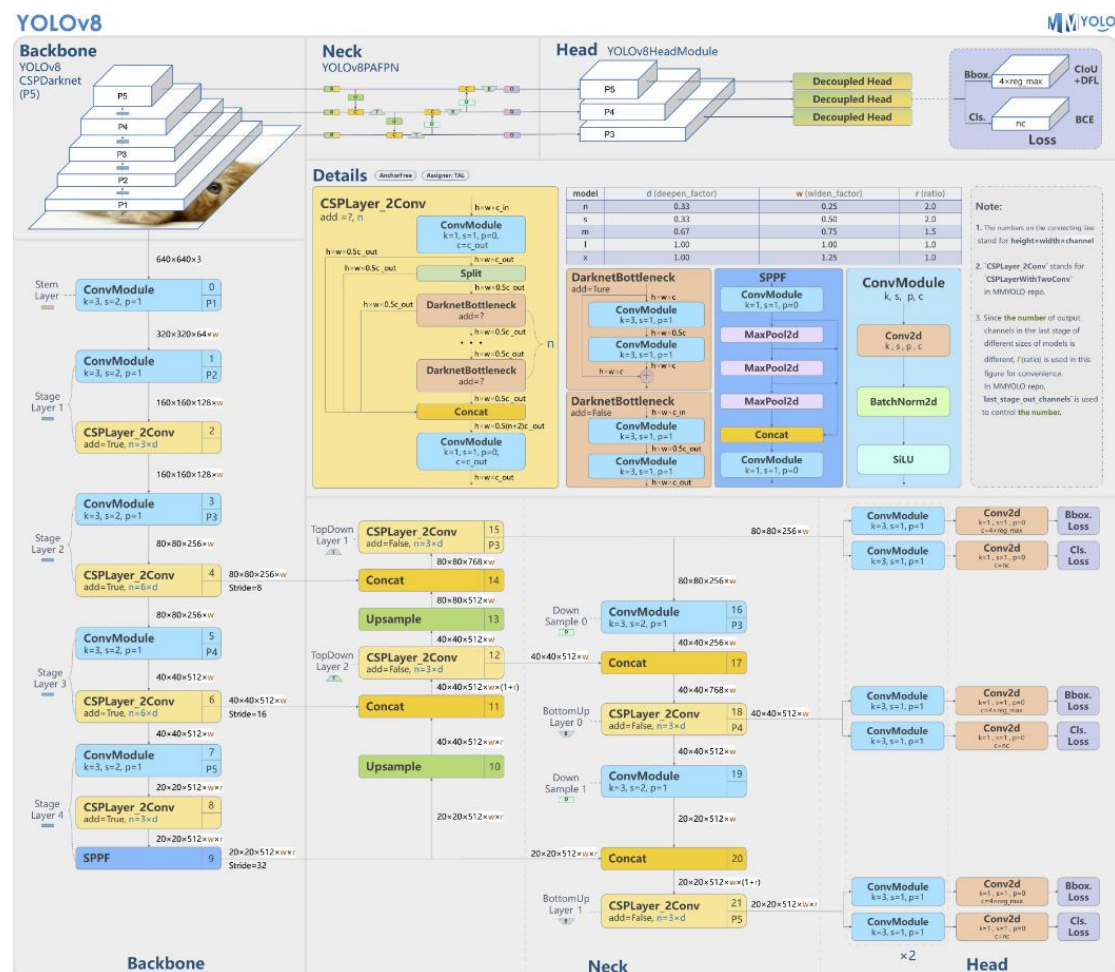
YOLO 架構

輸入端：調整圖片大小以供模型訓練使用。

- 主幹網路（Backbone）：該模型的主要網路，通過增加卷積層數量，提取不同感受野（P1-P5）的特徵圖，感受野逐漸增大。

- 頸部結構（Neck）：包括特徵金字塔網路（FPN）和自底向上的特徵金字塔（PAN）。FPN 利用多尺度對不同尺寸的目標進行檢測；PAN 則通過自底向上的方式建立特徵金字塔。這樣的結合方式中，FPN 層自頂向下傳遞強語義特徵，而特徵金字塔則自底向上傳遞強定位特徵，兩者共同作用，從不同的主幹層對不同的檢測層進行特徵整合。

- 預測層（Prediction）：位於模型/架構的頭部（Head），用於最終的預測輸出。在 P3 到 P5 的過程中，感受野不斷增大，因此逐步預測目標大小為小->中->大。



預訓練模型

這次我使用的預訓練模型主要是從 Roboflow 上面的資料集訓連過後得到的路徑權重檔案，訓練的圖片主要是河流和海域的圖片。

程式碼:

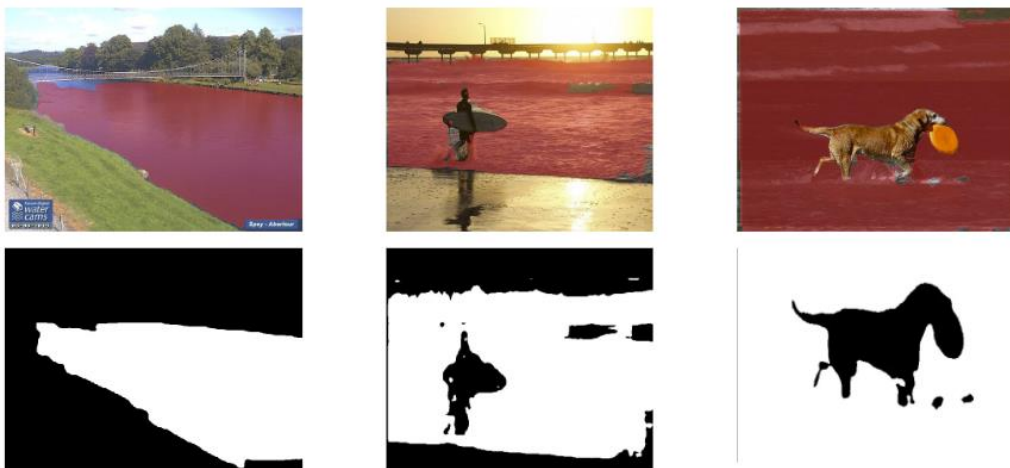
```
folder_path = './training_dataset/image/'

# 取得資料夾中的所有檔案
image_files = os.listdir(folder_path)
i=0
for image_file in image_files:
    print(image_file)
    # 組合完整的影像檔案路徑
    image_path = os.path.join(folder_path, image_file)
    source = Image.open(image_path)
    # 使用 YOLO 模型進行預測
    results=model.predict(image_path, save=False, conf=0.1)
# results = model.predict(source, save=False, conf=0.1, show_labels = False)
    H,W = source.size
    for result in results:
        im_array = result.plot(boxes = False, labels = False, probs = True)
        im = Image.fromarray(im_array[...::-1])
        im.save(f"./rgb_mask1/{image_file}")

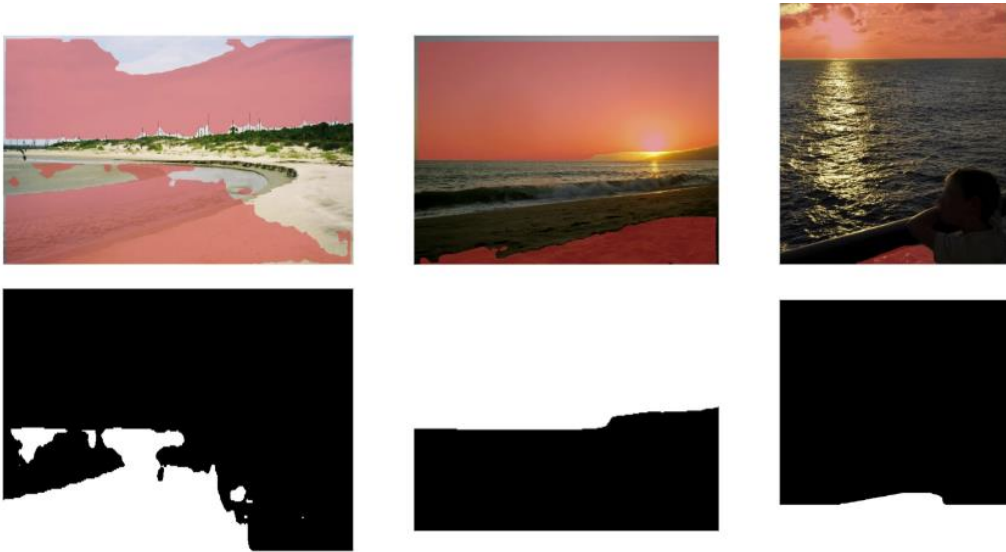
        if(results[0].masks != None):
            for j,mask in enumerate(result.masks.data):
                mask = mask.cpu().numpy()*255
                mask = cv2.resize(mask,(W,H))
                cv2.imwrite(f"./black_mask1/{image_file}",mask)
    i+=1
```

註:YOLOv8 輸出後的圖形會是紅色的 mask，在通過圖像處理得到 project 要求的 mask

輸出結果



辨別結果好的圖片



辨別結果不好的圖片

總結

Conventional method:

1. 對於顏色較為一般(沒有耀眼光、混著土的顏色、接近純白的海浪)，且分割界線明確的圖形效果較好，跳脫一般狀況太多的很難成功分割。
2. 相對於深度學習，能花費較少時間就完成分割
3. 雖然已經通過 HSV 來比較水體的顏色，也有對飽和度和明度進行處理(水體通常顏色較深)，但還是較難分辨接近於深藍色、深藍綠色的物體，如深藍的天空等等。
4. 未來若遇到相關議題，可以從水體的其他特徵下手，也可以嘗試 mean-shift 的方式先將圖型切割，或許會有更佳的效果，除此之外，先進行一些前處理也是可以嘗試的方向。

Deep learning method:

1. 訓練資料集若不夠多樣可能會出現錯誤的辨別內容
2. 針對顏色有變化的海水或河流同樣也無法有效判斷出來
3. 跟傳統方法做比較，撇除沒有偵測輸出的圖片，整體效果稍微好一些
4. 缺點就是訓練時間稍長，且須準備龐大資料集做訓練

反思:

總結來說，這次影像處理的期末專題我認為我們對預這方面的議題有著更深入的了解，在執行傳統方法時，會遇到圖片變顏色或特徵辨別的問題。轉換成深

度學習模型初期也會遇到這方面的挑戰，需要使用 STOA 的模型或是好的訓練資料才有機會得到好的驗證結果。這方面的問題我們認為要成功應用到現實生活中，可能還需一點點的時間。而使用到的方法很有可能還是以深度學習的方式去進行。

這學期感謝老師和助教們的指導，我們學到很多，你們辛苦了。

參考資料

- <https://chihshenghuang821.medium.com/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92%E7%89%A9%E4%BB%B6%E5%81%B5%E6%B8%AC-you-only-look-once-yolo-4fb9cf49453c>
- https://github.com/ultralytics/ultralytics?fbclid=IwAR0TNSMvMkM-GuPuEnxHt9KUjqZ_voSJGPkvWgpae9HghILkYTdyVs0vhUs
- <https://universe.roboflow.com/search?q=water%20river>