

國立成功大學  
資訊工程學系  
多媒體內容分析

作業二

基於高斯混合模型的彩色圖像分割  
*GMM-based Color Image Segmentation*

學生：P76094169 黃仁鴻

授課老師：朱威達 教授

中華民國 110 年 4 月

# 一、執行環境與說明

表 1 環境配置

處理器	Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz
記憶體	16.00GB
作業系統	Windows 10 64 位元作業系統
使用語言	Rust 1.51.0
函式庫	Image-rs、rusty-machine

表 1 是本次作業的環境配置。使用 rusty-machine 中的 gmm 來進行此次作業的實驗，不過 rusty-machine 缺乏自訂初始參數的功能，因此將原版的 rusty-machine fork 下來做些微調整，使其可以載入預訓練的參數。

## 執行方法

- 訓練

- hw2.exe train -h

查看訓練模式的使用說明

- hw2.exe train -e <最大更新次數，可選，預設為 10>  
-k <kernel 數目，可選，預設為 2>  
-w <儲存權重檔名，可選，  
預設為 gmm\_file>  
-r [<場景分類參考用的輸入檔>  
<場景分類參考用的標記檔>] (至少一組)  
-f <訓練用檔案，必要，可多個>

- 評估

- hw2.exe eval -h

查看訓練模式的使用說明

- hw2.exe eval -i <輸入檔案，必要>  
-w <儲存權重檔名，必要>  
-o <輸出分割圖檔名，可選，預設不儲存>  
-t <標記圖檔名，可選，  
有的話會計算 Dice Coefficient>

# 關於場景分類參考

雖然使用高斯混和模型可以依據色彩將圖片分割成多個區塊，但這些區塊並沒有特定的類別。在沒有標記解答參照的情況下是無法分辨哪些 kernel 被歸類在場景的，因此在訓練時會多使用一組(或多組)輸入、標記，藉由 GMM 將「參照用輸入圖檔」分割的結果來分類每個 kernel 是否屬於場景類別。

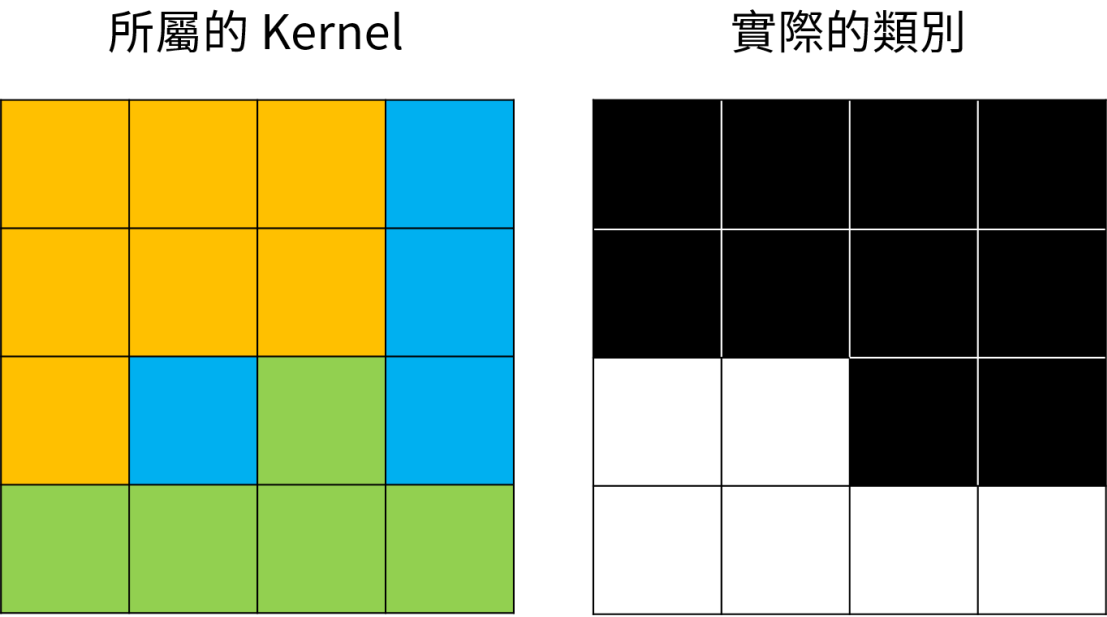


圖 1 對參考圖片使用 GMM 分割

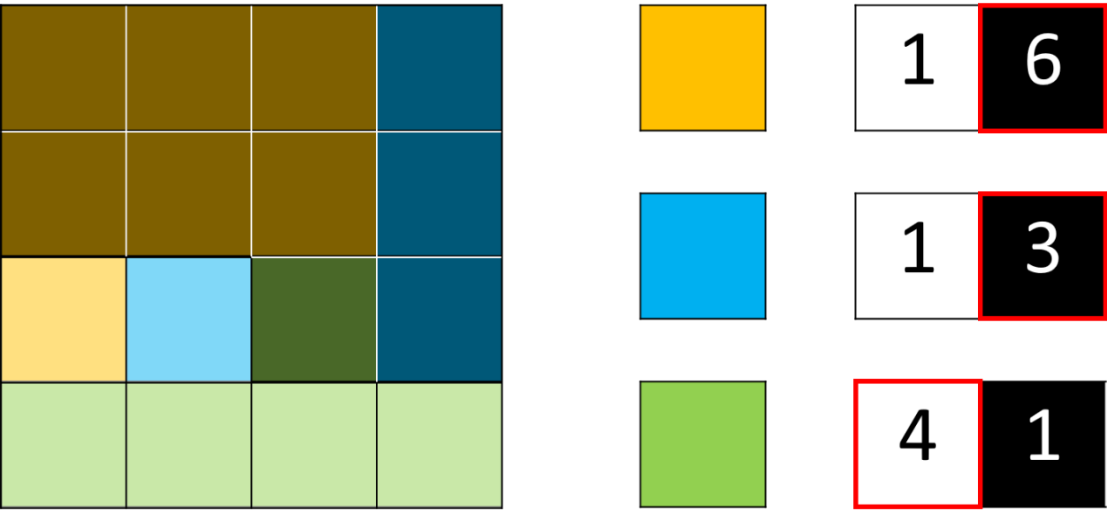


圖 2 將分割結果與標記答案進行比對，依照每個 kernel 對應到的標記數量進行分類。

## 二、效能分析

由圖 3 可以發現，用較多資料訓練的 model 2 在不同 kernel 數量時都有相近的正確率，其表現相對穩定。而在只使用 soccer 1 訓練的 model 1，在 kernel 數量增加時會有較明顯的性能提升。

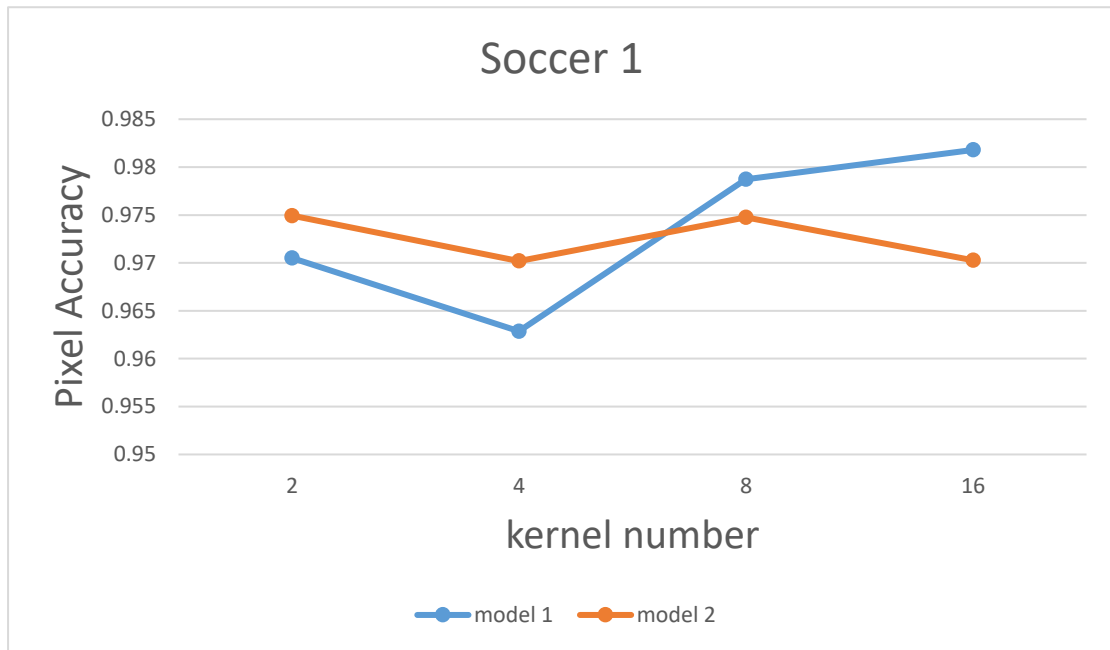


圖 3 比較使用於 soccer 1 時，model 1 與 model 2 的效能差異。

因為 soccer 1 與 soccer 2 的明暗差異較大，用 soccer 1 訓練出來的模型難以兼容 soccer 2，即使提增加 kernel 的數量，能提升的效果也有限(如圖 4)。

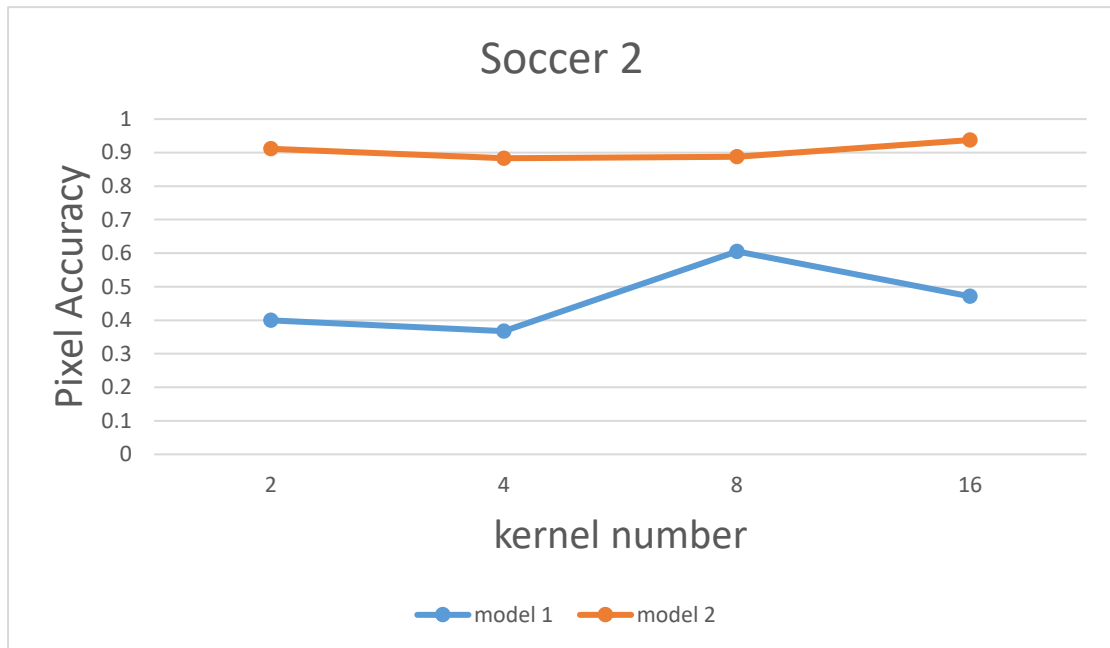


圖 4 比較使用於 soccer 2 時，model 1 與 model 2 的效能差異

Model 2 使用了兩筆資料進行訓練，然而因為 soccer 2 的場景因明暗度較為複雜，導致切割能力較低。但是當 kernel 增加時，就能讓 GMM 有更多能力擬合 soccer 2 使其效能有所提升(如圖 5)。

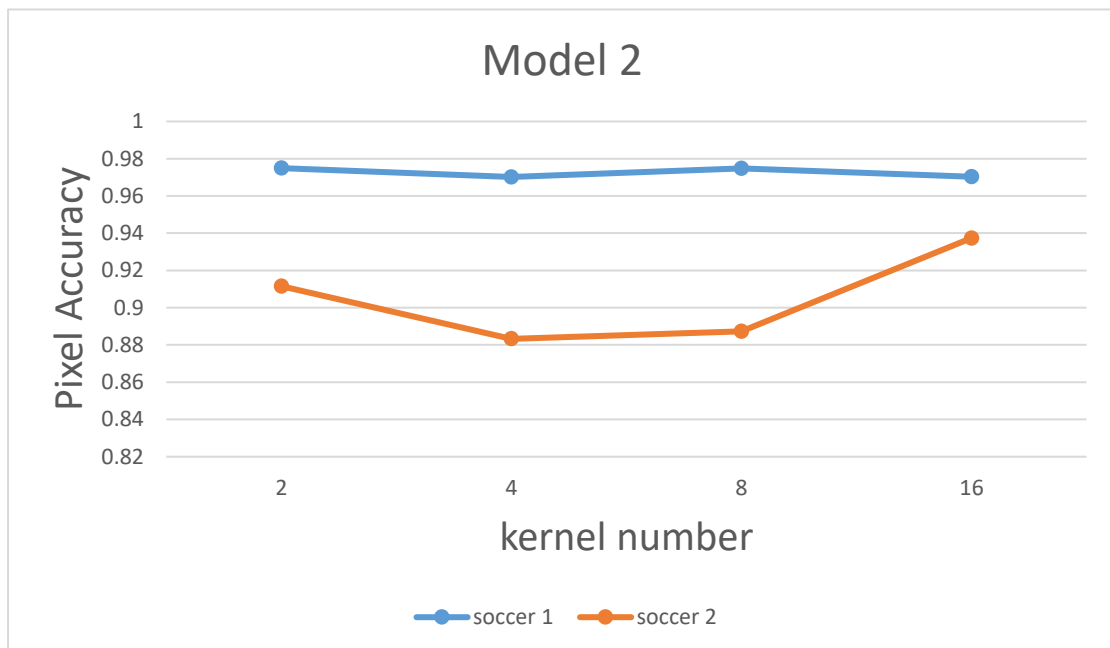


圖 5 比較 model 2 對 soccer 1 與 soccer 2 的效能差異



圖 6 model 1 對 soccer 2 的切分結果，上圖是  $\text{kernel}=4$ ，中圖是  $\text{kernel}=8$ ，下圖是  $\text{kernel}=16$ 。soccer 1 與 soccer 2 的顏色分布差異較大，由 soccer 1 訓練的模型難以兼容至 soccer 2。

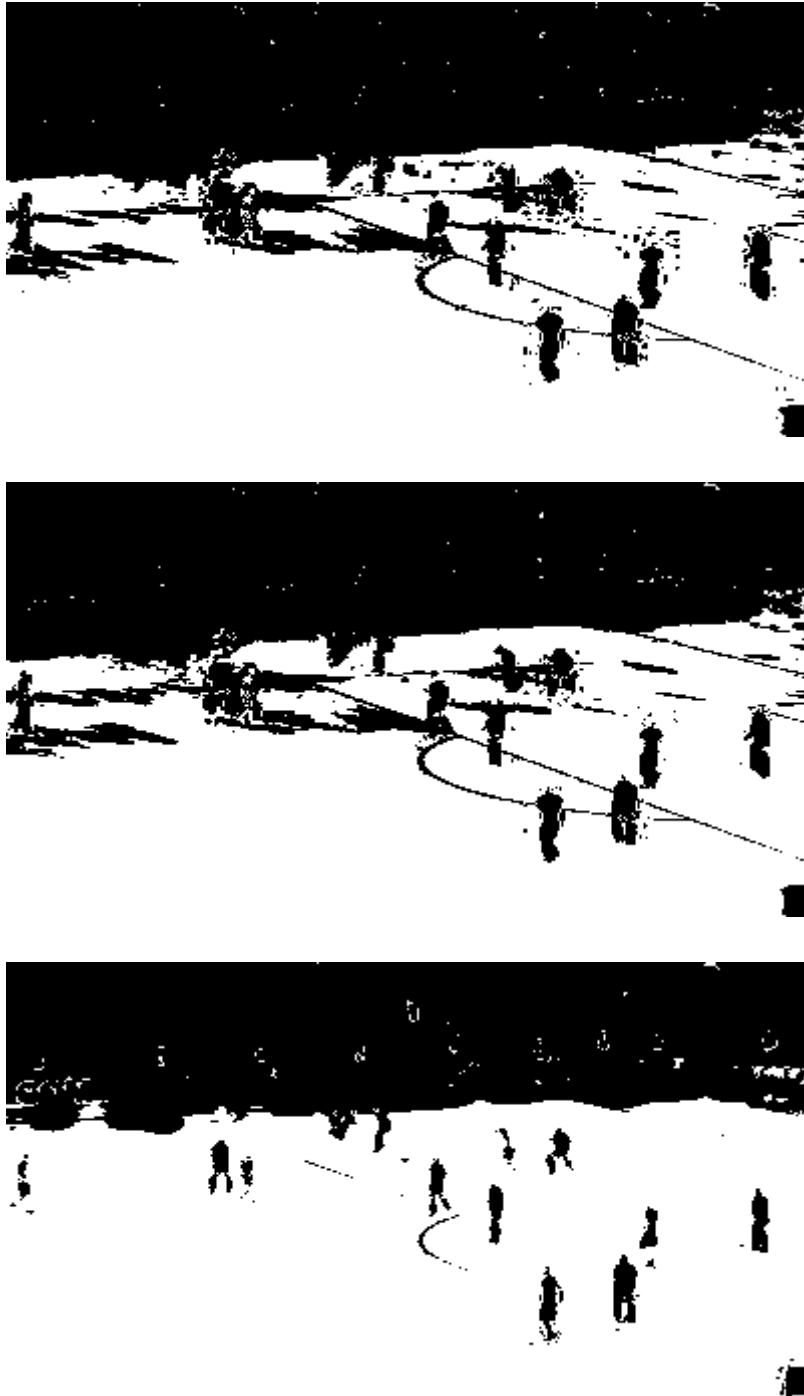


圖 7 model 2 對 soccer 2 的切分結果，上圖是 kernel=4，中圖是 kernel=8，下圖是 kernel=16。可看出在較複雜的圖片上，增加 kernel 的數量有助於提升效能。