

# 多媒體內容分析 hw4

F74082141 資訊 112 王浩

---

## 0、介紹環境

使用 python 3.11.0, opencv 4.7.0, scikit-learn 1.2.2

## 1、演算法介紹

### 1.1 n\_component 選擇

本次使用 scikit-learn 的 Gaussian Mixture Model

```
GaussianMixture(n_components=n_components, covariance_type='full')
```

來作為套件使用，因為任務為區分「背景」與「非背景」，我們有兩種方向來設定 n\_component。

**1.1.1 直覺法：**直覺上會直接選擇 n\_component = 2 的情況，表示將輸入的圖片，直接找兩個 Gaussian Model 來模擬，一個代表背景、另一個則會代表前景，我們就可以根據哪一種 Gaussian 佔比比比較大，作為判斷是否為背景，反之則為前景，但我們本次並不是以此方式來判斷，而是使用 1.1.2 的分析法。

**1.1.2 分析法：**另外一種方式，表示將輸入的圖片找數個以上的 Gaussian Model 來擬合我們所提供的圖片，如此一來，我們可以解讀為將圖片的內容，做出更細緻的分層分類，每個 pixel 在不同 level (component) 都可以被分析成各種 Gaussian distribution 的佔比，接著我們在針對每一種的 distribution 分析，進而去找出「判斷背景與否」的標準，標準就可以利用 threshold 來判斷。而下方的選擇就是根據不同情況所做出的標準調整。

### 1.2 level 與 threshold 選擇

**1.2.1** 因為 n\_component 表示有很多種 Gaussian Distribution，所以我們沒辦法直接從 component 直接判斷出哪一個 level 就是代表背景，因此，我們要分析「從哪一個 level 的 distribution 在大於哪一個 threshold 的情況」下，才能判斷出每一個 pixel 是否為背景。

### 1.2.2 根據scikit-learn的套件

```
probs = M1.predict_proba(pixel_values)
```

我們可以找到每一張pixel的機率分布狀況，進而去判斷要在機率值低於某值時，才被判斷為背景，而我們所選的範圍候選包含 0.1、0.2 至 0.9。

## 2、Scenario 1

以 soccer1.jpg 中的場地 pixel 建構 GMM，稱此模型為 M1，並以 soccer1.jpg 做測試

### 2.1 結果

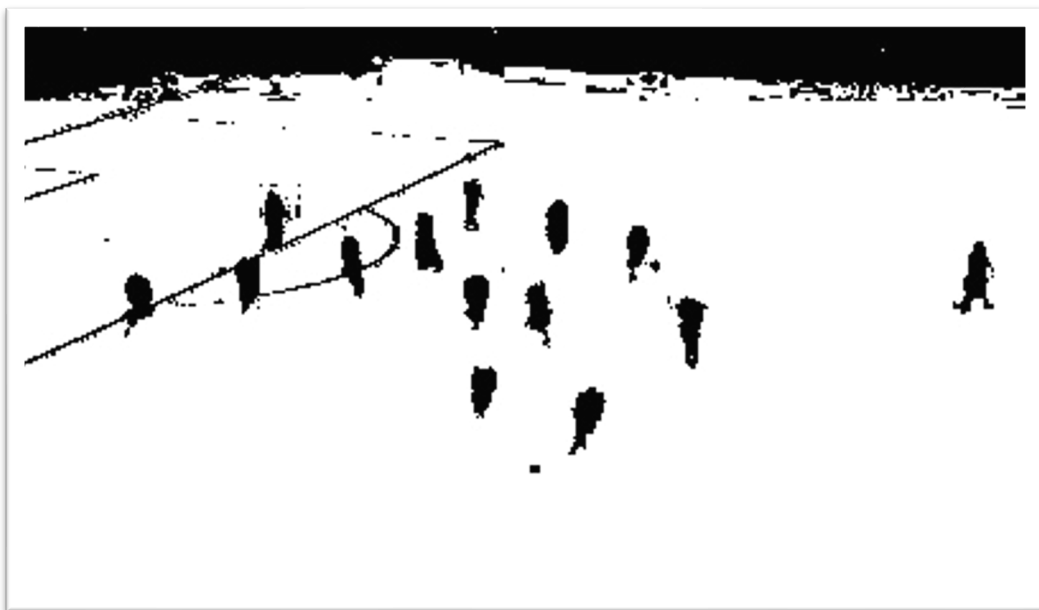
n_component	score	level (best)	threshold(best)
2	0.97	1	0.9
3	0.93	1	0.9
5	0.93	4	0.1

### 2.2 圖形

#### 2.2.0 ground truth



### 2.2.1 $n_{\text{component}}=2$



### 2.2.2 $n_{\text{component}}=3$



### 2.2.3 n\_component=5



## 3、Scenario 2:

以 M1 針對 soccer2.jpg 做測試

使用 Scenario 1 找到的最適合參數 (level、threshold) 測試 image2

### 3.1 結果

n_component	score
2	0.7
3	0.75
5	0.93

### 3.2 圖形

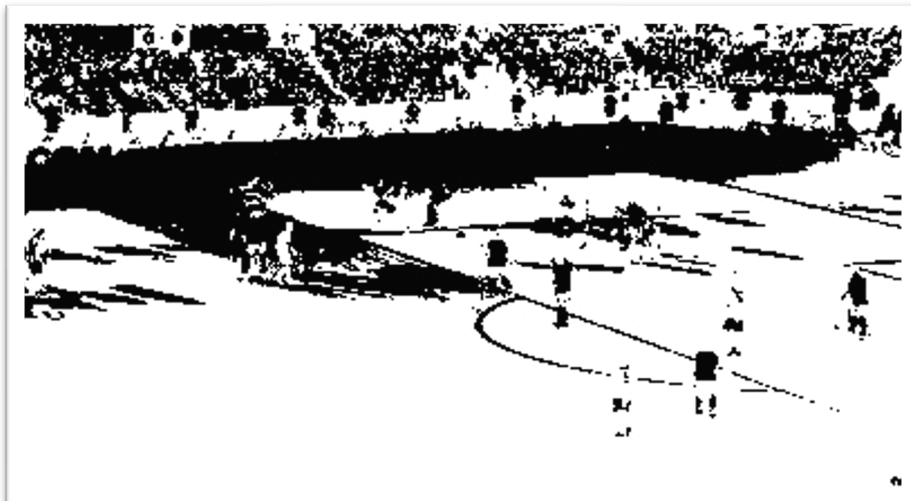
#### 3.2.0 ground truth



### 3.2.1 $n_{\text{component}}=2$



### 3.2.2 $n_{\text{component}}=3$



### 3.2.3 $n_{\text{component}}=5$



## 4、Scenario 3:

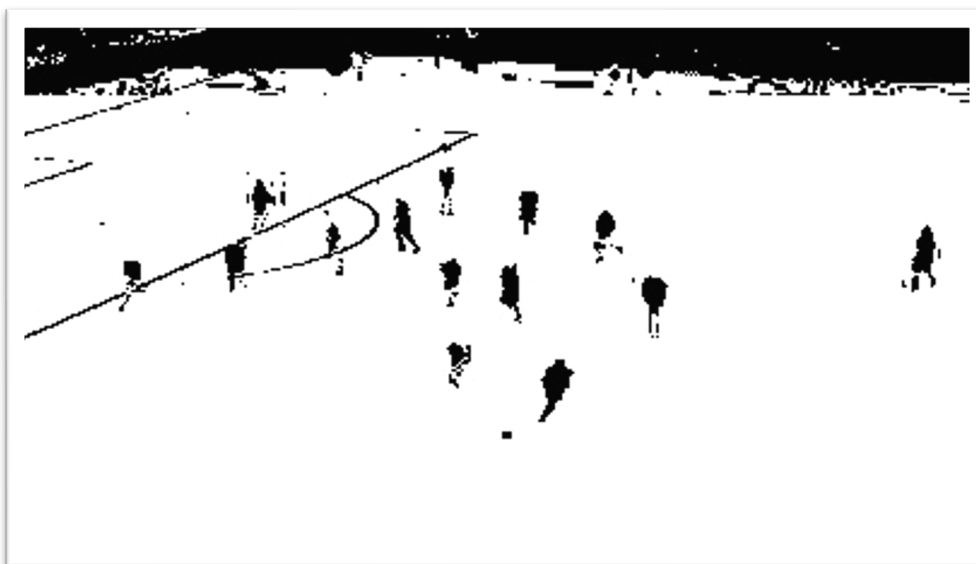
以 soccer1.jpg 以及 soccer2.jpg 中的場地 pixel 建構 GMM，稱此模型為 M2，並以 soccer1.jpg 以及 soccer2.jpg 做測試

### 4.1 分數

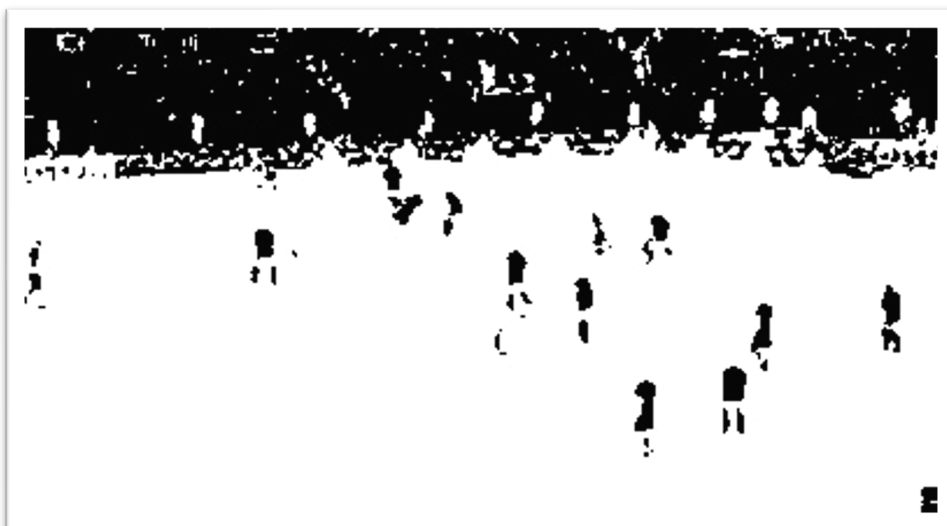
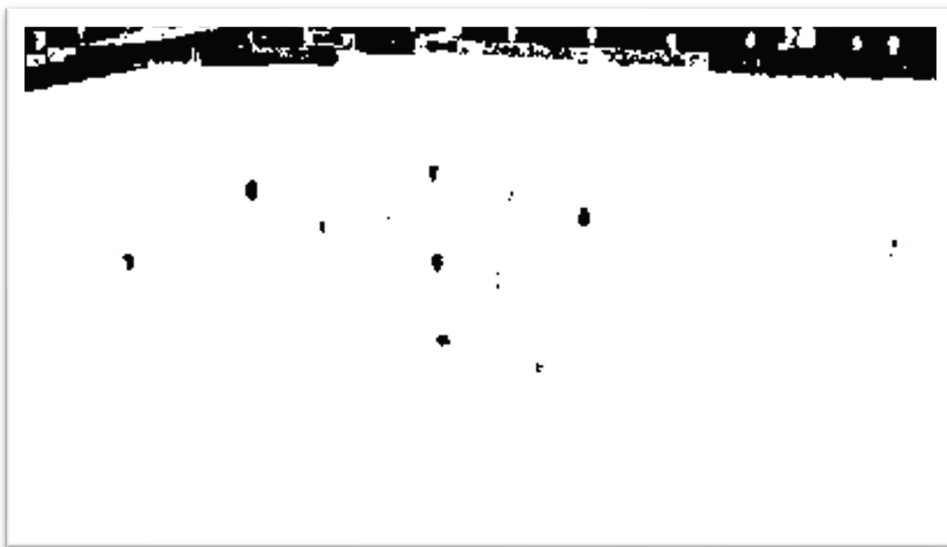
n_component	score	level (best)	threshold(best)
2	0.94	0	0.7
3	0.93	2	0.1
5	0.88	1	0.1

### 4.2 圖形

#### 4.2.1 n\_component=2



#### 4.2.2 $n_{\text{component}}=3$



#### 4.2.3 $n_{\text{component}}=5$





## 5、結論

根據 3 個 scenario 結果，我們可以得知，在 Scenario 1 的情況下， $n\_component$  選擇 2 的分數 0.97 為最高，而  $n\_component = 3$  和 5 都是 0.93。但如果分別利用各字的  $n\_component$  M1 模型，去預測不同資料的情況，也就是 Scenario 2 的情況，反而  $n\_component = 5$  的情況會得到非常高的 0.93，在 Scenario 1 最高的  $n\_component = 2$  時，分數是最低的 0.7，我們可以猜測， $n\_component = 2$  時，造成了 overfit，這才會造成訓練資料集分數極高、測試資料集的分數偏低。所以 Scenario 1, 2 的結果分數也就相反呈現。

在 Scenario 3 可以成功佐證，如果兩張圖片都拿來做訓練資料集，那就猶如 Scenario 1 的 M1 model 情況，分數的高低會是  $n\_compent = 2$  大於 3 大於 5，而結果也確實以 0.94、0.93、0.88 依次遞減。