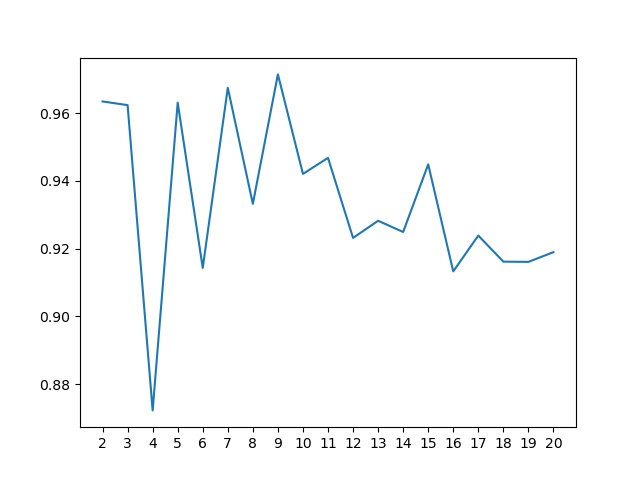
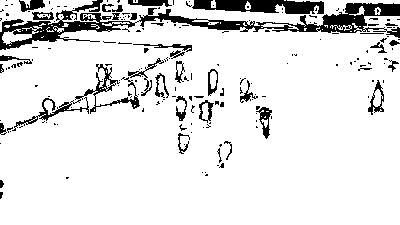
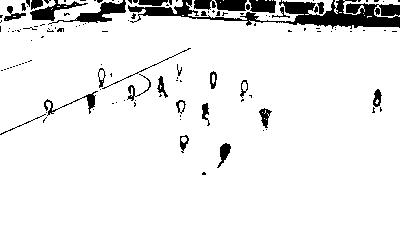
1. 程式執行環境及說明
   * Python = 3.8.8
   * Numpy = 1.19.2
   * scikit-image = 0.17.2
   * Matplotlib = 3.3.4
   * 執行方式 python ./src/main.py
   * 執行後會將結果 (segment圖片、準確率) 存至./result
2. 三個scenarios的實驗報告
   * 由於A New Method to Segment Playfield and Its Applications

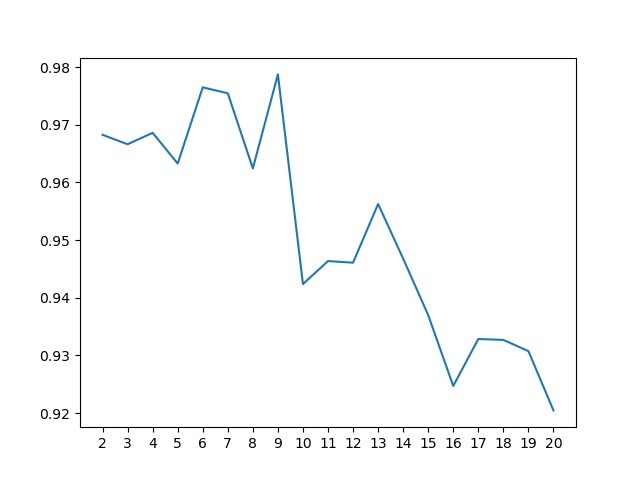
in Match Analysis in Sports Video這篇paper中提到可以將RGB轉換至HSL/HSV色域以獲得更好的準確度，故以下分成RGB / HSV的結果

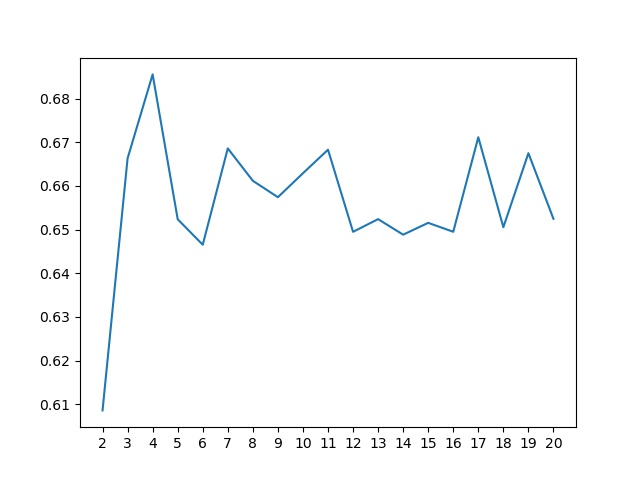
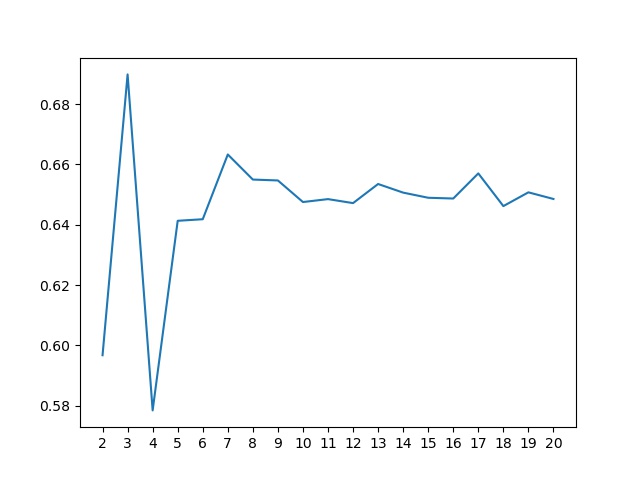
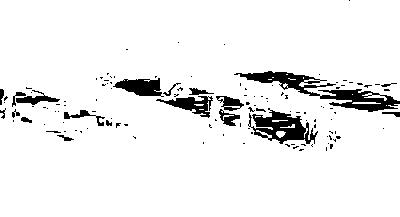
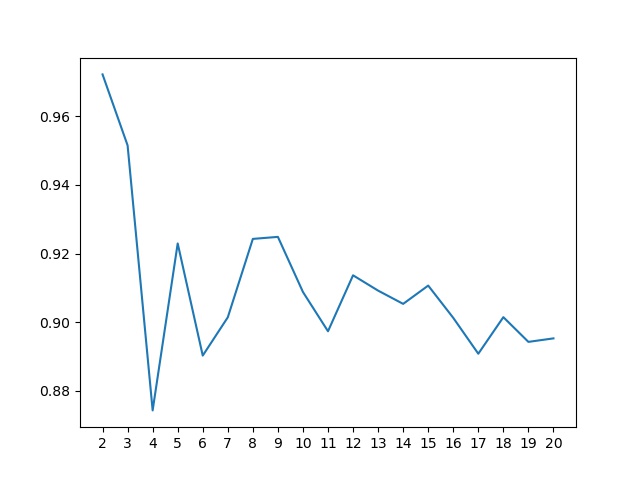
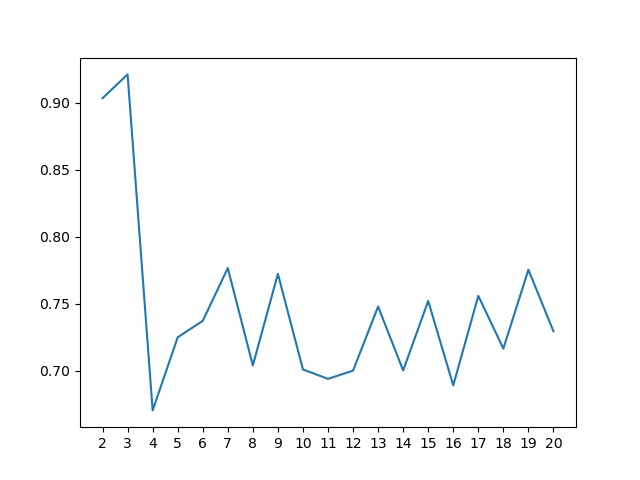
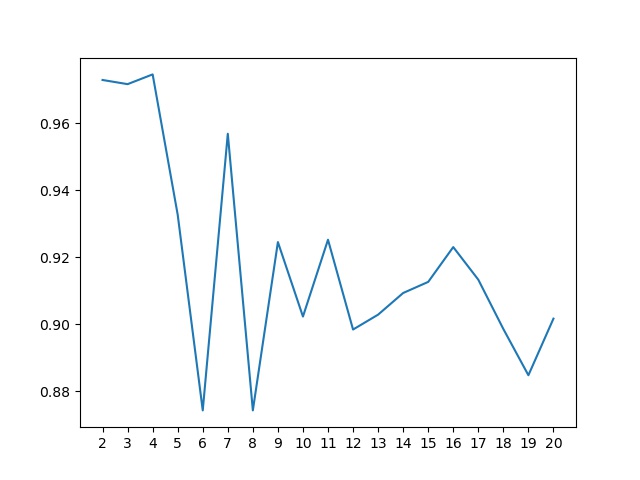
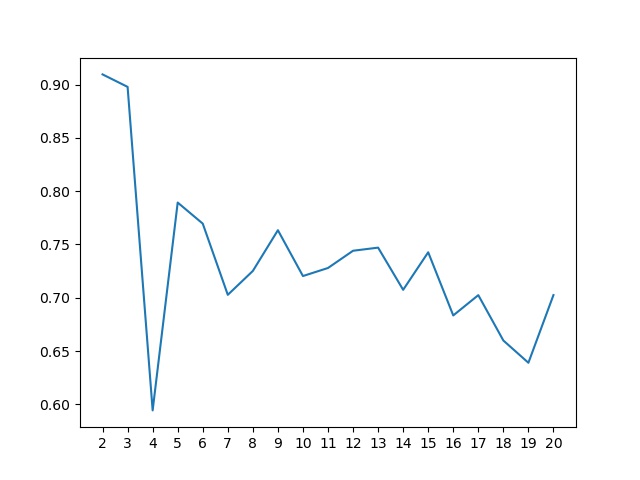
* + 將圖片的pixel分成n(n=2~20)群Gaussian後，將n群依照pixel總數排序，因為足球場地的pixel占比較多，故根據觀察，取前n/2群label成足球場地(1)，其餘n/2群標記為其它(0)
  + Scenarios 1的情況下，RGB平均表現比起HSV更為良好且穩定，尤其在Gaussian=2~9的時候，根據觀察，這部分可能是因為只有利用單張圖片去訓練GMM，造成在HSV色碼下會將人體部分的pixel跟場地的pixel分成同一類。而切割準確度在9群後逐漸下降，有可能是因為分群太細的情況下，會將球場這種大範圍面積的物件切割得更細，以至於更容易跟其它非球場的部分歸類為相同一群
  + Scenarios 2的情況下，HSV平均表現比起RGB更為良好，尤其RGB在不同Gaussian分割下的準確率與Scenarios 1下使用HSV的情況雷同，可以驗證應該是因為單張圖片去訓練GMM，使得GMM預測圖片的分群時，過度符合某張圖片的特性。
  + Scenarios 3的情況下，RGB與HSV的就各有千秋，如果今天將訓練GMM的圖片資料增加，可能就會得出paper中提到的HSV平均效能較好的結果

1. 切割效能
   * 所有切割出來的分群圖片都有存至相對應的result中
   * HSV Q1



* + 1. 平均準確率：0.934
    2. 其中在gaussian = 4, 6, 8的時候突然降低切割效能，為前面提到的HSV色碼的問題
    3. Gaussian = 4 
    4. Gaussian = 6 
    5. Gaussian = 8 
  + RGB Q1



* + 1. 平均準確率：0.951
  + HSV Q2 
    1. 準確率：0.656
  + RGB Q2
    1. 準確率：0.645
    2. 其中Gaussian = 2, 4效能突然下降，為前面提到RGB在不同scenario下GMM過度overfitting的問題
    3. Gaussian = 2 
    4. Gaussian = 4
    5. 
  + HSV Q3 – image 1 
    1. 準確率：0.909
  + HSV Q3 – image 2 
    1. 準確率：0.745
  + RGB Q3 – image 1 
    1. 準確率：0.918
  + RGB Q3 – image 2 
    1. 準確率：0.733
  + 根據以上結果，可觀察出在Gaussian = 2~4的時候，撇開因為GMM模型overfitting的關係，平均的切割準確率較高，故應該此類足球場地圈選問題，Gaussian 2~4可以得到較好的結果，這也符合paper中提到的結果