

國立清華大學資訊工程系 _112_學年度_1_學期專題報告

專題名稱	基於桌球落點判定之轉播視角桌球影片動作區間偵測				
參加競賽或計畫	<input type="checkbox"/> 參加對外競賽	<input type="checkbox"/> 參與其他計畫	<input checked="" type="checkbox"/> 無參加對外競賽或任何計畫		
學號	109062321				
姓名	余明璠				

摘要

桌球落點判定之轉播視角桌球影片動作區間偵測，旨在接收到全新、完整的轉播視角桌球比賽影片時，自動化的切割出球員動作的起點集中點的模組，提供給自動化分析桌球球員動作的模型，藉此提升準確率。我們開發的自動化切割球員動作演算法，是基於桌球的軌跡、落點去切割球員動作區間，此作法難點在於桌球在比賽畫面中變化幅度大，體積小，誤判及漏判機率高，因此此專題透過兩個 Yolo 模型以及多層條件判斷，降低誤判、漏判狀況，達成自動化、快速化生產訓練用的資料。

中華民國 112 年 11 月

1. 介紹

(1)專題動機：桌球落點判定之轉播視角桌球影片動作區間偵測，是為了解決做自動化分析桌球動作時，逐個標註影片中動作時間資料作為訓練用的資料非常耗時的問題。根據實驗室中的實驗需求，在訓練桌球影片動作分析模型時，比起隨意地將桌球影片切成短片後(noisy data)提供給模型，如果能提供大量依照桌球動作定義切割的動作片段(clean data)，在分析上是否會更加準確。

因此，為自動、快速化生產 clean data 的流程，故設計自動化切割動作的時間模組。

(2)專題目標：解決桌球影片中，球難以正確偵測的問題，利用兩個 YOLO 模型偵測桌球影片中球的位置來減少誤判，再以球員、球桌、球的 label 進一步截錄出桌球選手的動作影片

2. 現行相關研究概況與比較

(1) YOLOR :

特色: 可以保存所有輸入資料的特徵，善於利用模型中的隱性知識 (implicit knowledge)，讓原本專精單一任務的模型，透過保存起來的特徵，來學習其他任務，而不必針對其他新任務，一一重新訓練。

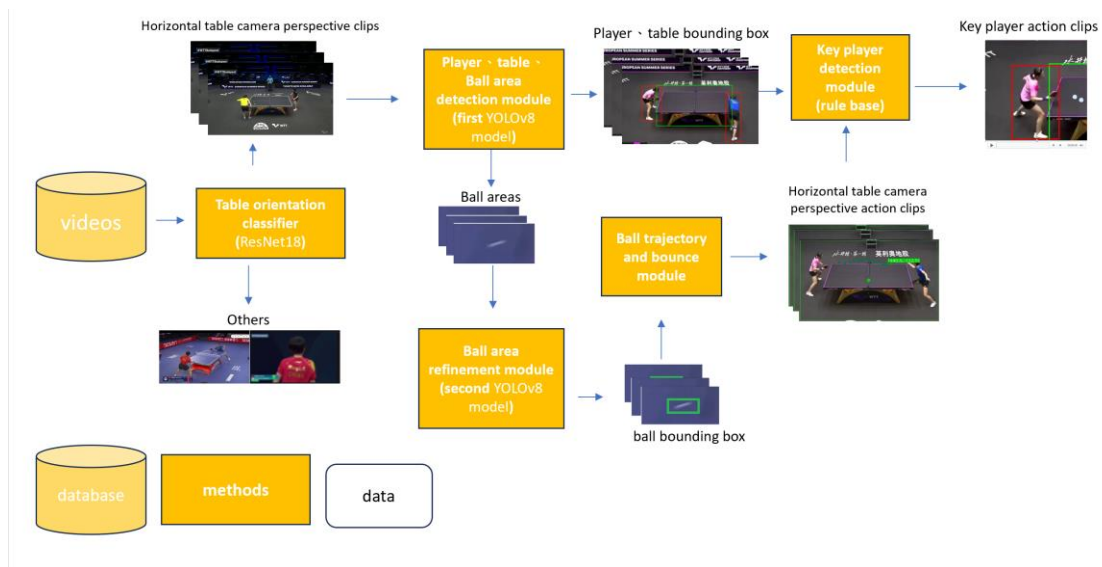
比較: 我們原先使用 YOLOR 模型進行物件偵測，雖然在偵測球的階段效果還不錯，但是在偵測桌球球員的效果上差強人意。

(2) YOLOv8:

特色: YOLOv5 的進化，專注於增強模型偵測的準確率以及效率

比較: 使用 YOLOv8 在球以及人物都偵測上效果都較好，我認為是因為我們使用的桌球比賽畫面並不複雜，所以沒有發揮到 YOLOR 的長處，故後來換成使用 YOLOv8 做偵測

3. 研究方法與步驟:



流程圖

研究方法: 我們依照流程圖中的 methods 照順序進行介紹

(1)Table orientation classifier:

ResNet model :

利用 ResNet model 判斷每個 frame 中球桌的狀況，分為三種狀況 0:水平(圖 0) 1:垂直(圖 1) 2:無球桌(圖 2)，在此專題中，只需要用到球桌為橫向的部分



圖 0

圖 1

圖 3

用此方法的動機：

利用 ResNet model 直接判斷每個 frame 中球桌的狀況後就送入 yolo model 做偵測，此作法花費的時間較短(省去從影片截出短片存進資料夾再從資料夾提取短片步驟)，同時 timeline 跟 ground truth 的比對上也沒有問題，

(2)Player、table、Ball area detection module & Ball area refinement module

在偵測球的步驟上，我們參考了這篇論文

TTNet: Real-time temporal and spatial video analysis of table tennis (如圖 4) 的做法去提高對球的正確偵測。相較於直接利用一個 yolo model 精確框出各個物體的 bounding box，我們選擇利用兩個 yolo model，並在第一個 model 的 training 時，特意拉大球的 bounding box，利用多框出來的球

附近的背景，增加 model 辨別球的能力。

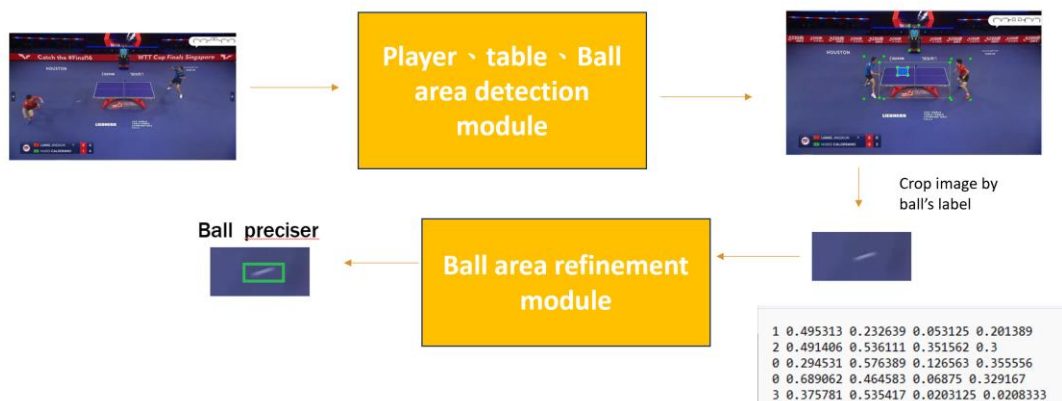


圖 4

Yolov8 Player、table、Ball area detection model training data :

準備幾千張桌球比賽中的畫面照片，標記上畫面中的球員、球桌、球的位置。特別將桌球的 bounding box 放大，給 yolo model 做 training set

Yolov8 Ball area refinement model training data :

利用上述的幾千張畫面以及 first model 預測出來的結果，crop 出幾千張球的照片，然後在此照片中標記上球的精確位置，給 Ball area refinement model 做 training

使用步驟:

Player、table、Ball area detection model 預測出 ball area 的位置

再利用 Ball area refinement model 找出 ball 在 cropped area 的 label，

然後通

過計算得出球在原始圖片中的 label 即為所求。

用此方法的動機：

Original thought：因為在桌球比賽中，球的變形程度很大，直接把標準的球的 bounding box 框出來去給一個 yolor model 做 training(如圖 5)，此 model 在進行預測時，時常出現誤判或漏判，例如將照片場地中出現的小白點誤判做桌球(如圖 7)，並且 confidence 可能高達 0.67，難以靠提高 confidence 預防(再提高 confidence 會大幅減少偵測到的桌球數量)

Modified：特意拉大球的 bounding box(如圖 6)，再利用第二個 model 找出球的精確位置並 return label 到原始圖片，此作法可以有效減少球的誤判(如圖 8)



圖 5



圖 6



圖 7

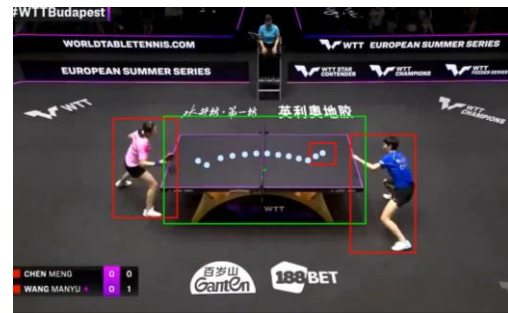


圖 8

(3) Ball trajectory and bounce module :

首先，然後從軌跡中判

斷 bounce 是否產生，並依此截出球員動作片段

在看這部分前，要先了解對於桌球選手動作的定義

桌球選手動作定義:

發球:

- ◆ 時間起始點：選手拿球預備開始(若準備時間太長，則從拋球點往前取 10 個 frame)
- ◆ 時間結束點：擊桌面第二次到對方桌面之落點。
- ◆ 總共撞擊桌面**兩次**。(參考下圖 9、圖 10、圖 11)



圖 9



圖 10



圖 11

其他動作:

- 時間起始點：上一次對方桌面之落點。
- 時間結束點：下一次對方桌面之落點。
- 總共**三個落點**之間為一個動作時間片段(參考下圖 13、14、15)



圖 13




圖 14



圖 15

Ball trajectory and bounce module 實作方式主要是將前一個步驟找出的 ball 的位置處存起來形成軌跡，如果球的 y 座標出現 高,低,高 的情形則計為一次 bounce (如圖 16)。並依照上述隊動作的定義，依靠 bounce 數去切出動作短片。因為同時要截錄的有兩個選手，所以同時要 maintain 兩個 ball_position 的 list(開始紀錄的時間不同)，來分別截錄出兩位選手的比賽動作短片

問題：因為錯誤的偵測導致誤判 bounce，需透過額外判斷條件加強偵測的準確性

- 處理連續偵測到同個 x 座標的球
- 因為差距過小而出現 y 座標相同的情況 e.g. 
- 角度問題，彈擊桌面時並未形成高低高的彈跳(如圖 9)

解決方法:

- 若下個偵測到的球 x 座標一樣則不收錄
- 若有出現高低低高的情形也要偵測為 bounce
- 先找出桌子的 center，在球桌的一側偵測到 bounce 之後，若在同側再次偵測到 bounce 則忽略
- 當球已經在回擊狀態但是此側球桌上未偵測到 bounce，代表遺漏了一個 bounce(如圖 17)，故補上一次 bounce



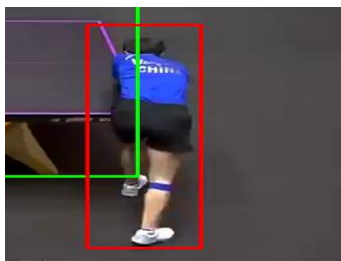
(圖 16)



(圖 17)

(4) Key player detection module

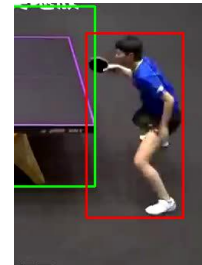
利用 yolov5 first model 偵測出來的 player label 和 bounce detection module 截出的動作短片，將球員在動作移動的 x、y 座標的最大最小記錄下來，以此做為球員動作畫面的 bounding box(如下圖 18、19、20)。因為球員有兩個，故需要有兩個不同的 list 分別記錄兩位球員，並做好標記從 bounce detection module 截錄出來的動作短片是屬於哪一位球員的。



(圖 18)



(圖 19)



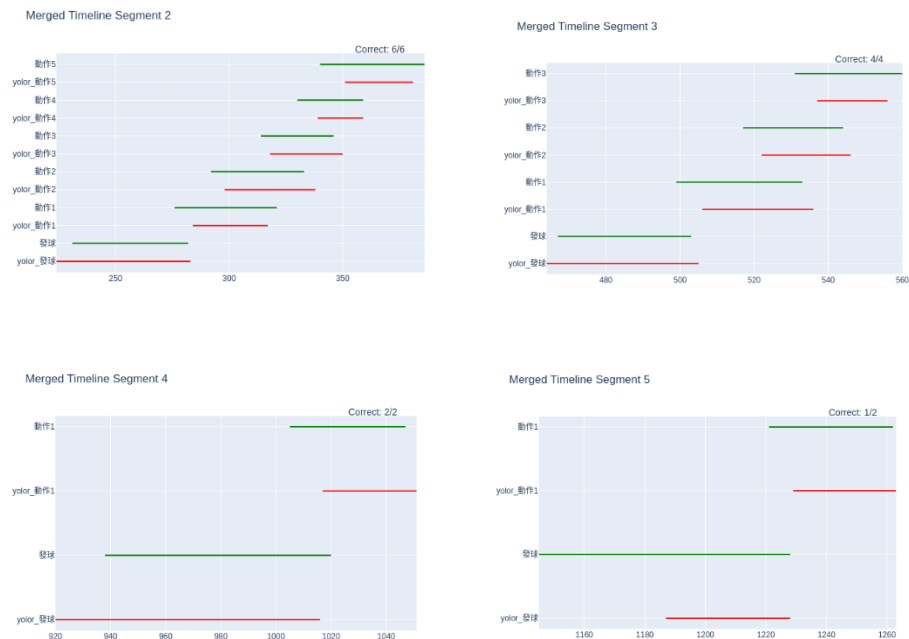
(圖 20)

因此在讀進 video 的 frame 時，首先利用 ResNet model 做 classification，只將球桌為橫向的 frame 留下，再來將這些 frame 送進 first model 做偵測，找出球桌、球員、球的座標。利用球的(big bounding box)座標將畫面上球的位置 crop 出來，再利用 second model 做偵測，找出球的精確位置，然後將偵測出來的 label return 回原始圖片。接下來依靠 bounce detection module 利用軌跡判斷 bounce，並依 bounce 數量截出上面提到的發球、動作定義，去截錄比賽影片

利用 Key player detection module 截錄出 key player action clip

4. 正確率評估

下 4 圖是我們預測出的動作時間線 跟 請桌球隊成員依照動作定義標註出的動作時間線所做的比對的例子



跑完一段影片後的成果，**total action** 是桌球隊成員標記出的動作數量，**total_pred_action** 是我們預測出動作數量，**total_correct_pred** 是 timeline 重疊程度有達到 50%以上的正確偵測

```
total_action 392
total_pred_action 438
total_correct_pred 330
```

5. 未來展望

在垂直桌球桌面上完成桌球選手動作區間偵測：

目前因為垂直桌面時球的軌跡不像水平桌面一樣容易偵測出 **bounce**，並且容易出現球員擋住球的狀況，不容易實施球的路徑偵測。

6. 參考

[1]TTNet: Real-time temporal and spatial video analysis of table tennis

Roman Voeikov, Nikolay Falaleev, Ruslan Baikulov

[2]YOLOR: <https://github.com/WongKinYiu/yolor>

[3]YOLOv8: <https://github.com/ultralytics/ultralytics>