國立中興大學資訊工程學系

資訊專題競賽報告

基於深度學習之排球球員追蹤與技戰術辨識系統之研究

Research on a Deep Learning-Based Volleyball Player Tracking and Tactical Recognition System

專題題目說明、價值與貢獻自評（限100字內）：

本專題將排球賽影片透過YOLOv8、ByteTrack進行排球、球員、動作的識別和球員追蹤，儲存執行戰術球員的位置後，使用深度學習方法辨識排球戰術，並將戰術片段剪輯、標註特徵。此專題可將比賽中的排球戰術視覺化，利於排球推廣。

專題隊員：

姓名	E-mail	負責項目說明	專題內貢獻度(%)
鄭皓	howardiamsorry@gmail.com	資料收集、各模型訓練	50%
范家馨	piyk7876@gmail.com	資料收集、各模型訓練	50%

【說明】上述表格之專題內貢獻度累計需等於100%。

指導教授簡述及簡評：

在此次專題中他們使用YOLOv8、ByteTrack進行球員追蹤，並結合深度學習方法進行排球技戰術分析。實驗結果顯示所提方法為可行，且表現不錯。過去他們曾遇到YOLOv8模型表現不如預期的問題，但在不斷研究、調整下，準確度提升不少，著實符合專題研究之精神。整體而言，此組學生表現不錯。 指導教授簽名：

中華民國 113年 5月15 日

目錄

一、摘要

現今影像識別已被廣泛應用在日常生活中，如：車牌辨識、人臉辨識，但影像識別在運動賽事上的應用，在近十年才剛起步。而在排球賽上的影像識別應用還停留在排球軌跡、動作識別，在針對排球球員跑位進行戰術分析之研究相當少見，可是在足球賽事上已有人透過此方法進行戰術分析，因此在本專題中將用於足球跑位戰術分析之技術轉用於排球賽上，並予以改進，實現排球戰術影像識別。

在本專題中需要進行球、球員、球員動作的物件辨識，在物件辨識領域中 You Only Look Once(YOLO)為常用的工具，YOLO 能夠處理實時影像，在發布後也發展出了多個版本，錯誤率低、應用廣泛度高、結構輕量，使 YOLO 成為物件識別中的首選，其中 YOLOv8 為較新的版本，在本專題中我們使用 YOLOv8 進行物件辨識。

物件辨識完後，本專題使用 ByteTrack 追蹤球員位置，ByteTrack 為 Tracking by detection 的追蹤方法，在使用 ByteTrack 前需要先進行物件辨識，因此本專題在用 YOLOv8 進行物件辨識後，再用 ByteTrack 進行物件追蹤，並記錄其位置。接著將位置資訊當作戰術分類模型的輸入，辨識出戰術類別。

關鍵字：排球、戰術、YOLOv8、ByteTrack

二、專題研究動機與目的

排球運動起源於 1895 年的美國，當時籃球風氣盛行，但激烈的籃球運動不太適合老弱婦孺、缺乏運動的人，因此發展出了排球，結合了籃球、棒球、網球和手球，排球不像籃球那樣激烈，但又充滿趣味、變化性，這也意即排球為闔家適合的運動。

隨著時間推移，排球戰術、規則逐漸完整，科技的進步使全球的排球賽都能實時轉播，場上一支排球隊伍會有六個人，能變化的戰術甚多，因此若沒有排球的先備知識，一般觀眾很難看懂、享受一場精采的排球賽，基於此原因本團隊想做一個排球戰術影像識別系統，透過此系統識別排球賽中執行的戰術，將執行戰術片段剪下，並在剪下的影片中標註戰術類別、執行戰術的球員。

目前已有透過足球球員跑位分析戰術的研究，但在排球的影像識別上相關的研究還停留在抓取排球軌跡、排球動作辨識，還未有透過球員跑位進行戰術分析的研究，因此本團隊想將用於足球戰術分析的概念轉用於排球上，並對相關技術進行改良。因為排球由多種運動結合發展而來，因此在排球中球員能執行的動作比足球多，如：托球、攻擊、扣球，而排球戰術執行時會伴隨著一連串不同的動作，所以在本專題中除了要辨識球員、抓取其位置外，還需要辨識球員的動作。

在參考文獻中的 Player Detection in Field Sports，他們所使用的方法就是將輸入的圖片以四種方式處理，分別為(a)原影像、(b)二元邊界、(c)形狀分析、(d)形態學運算，用以上四種處理分別運算。雖然該團隊的模型準確度很高，但缺點也很明顯，這樣的處理方式加上過於複雜的模型導致這樣的偵測方法難以用即時的方式運行，他們的專題中有提到平均每一部影片所需的計算時間超過 10 秒鐘，由此可知該專題的程式複雜度是很高的。另外，該專題也沒有提出對於鏡頭拍攝角度變化的解決方式，所以他們在面對場地變化時是沒有辦法很快地做出調整的。在此本團隊使用 YOLO 進行排球、球員的偵測，因為 YOLO 學習的特徵泛用度高，適用於不同場景，能夠解決先前提及的問題。

在追蹤球員的部分，文獻中使用 Spatiotemporal Attention Algorithm 和 Multicue RNN Algorithm，透過分割照片找出該幀目標物所在位置，計算物體可見區域後，帶入模型訓練，每個目標物皆會有一個捲積層分支。RNN 會將同時刻的的資訊融合在一起，並考慮追蹤目標軌跡，以解決球員被遮蔽問題。此兩個演算法的缺點為目標多、輸入多時模型大、計算成本高，考量到排球賽時長、球員數量，本團隊認為此方法不適用，在本專題中使用 ByteTrack 取代此兩個演算法。ByteTrack 模型尺寸較小，計算複雜度較低，較適合用於實時場景物件追蹤。

本專題選擇排球多人戰術中的 A、B 式小組時間差攻擊進行識別。小組時間差攻擊為排球戰術中常見的多人戰術，由三個人執行戰術，較單人戰術複雜，且小組時間差攻擊根據不同的快攻方式還可以再細分為 A 式、B 式，小組時間差攻擊戰術具有普遍性、多樣性，為理想的排球戰術代表，因此本專題先專注於辨識小組時間差攻擊 A、B 式戰術。

小組時間差攻擊戰術的概念為在攻擊時搭配誘餌騙敵隊的攔網起跳，使當真正攻擊時敵隊來不及回防。小組時間差攻擊戰術由三個球員執行，執行時間不超過五秒鐘，三個球員依序執行「托球」、「假攻擊」、「真攻擊」三個動作。小組時間差攻擊戰術根據不同的快攻方式能在細分為 A、B 式，在球員的跑位上會有所不同，如圖 1、圖 2。

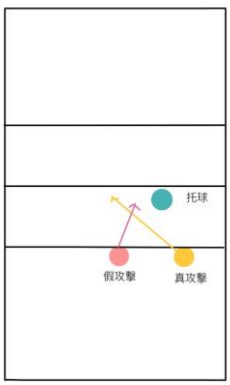


圖 1、小組時間差戰術 A

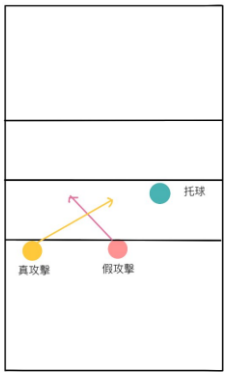


圖 2、小組時間差戰術 B

本專題預期透過 YOLOv8 獲得各幀球、球員、執行戰術球員位置，並用 ByteTrack 追蹤球員，將所有球員位置畫成跑位圖存下，使用 CNN 模型、前饋神經網路模型辨識小組時間差攻擊 A、B 式戰術，並將執行戰術片段剪下和標註戰術類型、執行戰術球員。

三、專題重要貢獻

本專題主要重要貢獻有：

1. 實現排球戰術影像識別
2. 自動化排球戰術精華影片剪輯、編輯
3. 排球運動推廣

四、團隊合作方式

表一為本團隊的工作項目分配表，本團隊將工作平分給每個成員，成員在執行各自任務時若有問題也會互相討論幫忙，每周都有共同作業時間，進行討論、協同工作、確認進度。

工作項目	參與人員
題目和架構設計與構想、資料查詢	全體組員
影片收集	全體組員
影片剪輯	范家馨
資料標註	全體組員
投影	鄭皓
YOLOv8-球模型訓練	范家馨
YOLOv8-人模型訓練	鄭皓
YOLOv8-動作識別模型訓練	范家馨
球員追蹤、位置資料儲存	范家馨
戰術分類模型建立	鄭皓
戰術分類模型訓練	全體組員
系統整合、測試	全體組員
報告撰寫	全體組員

表格 1、工作分配項目表

五、設計原理、研究方法與步驟

此專題中主要有八大步驟，蒐集影片、訓練 YOLO 模型、追蹤目標、偵測戰術發生、投影、統整資料、訓練戰術識別模型、輸出結果，各個步驟的技術內容與設計原理會在以下詳細說明。

1. 蒐集影片：

在此專題中本團隊使用 Asian Volleyball Confederation 此 YouTube 頻道上的排球賽影片(已獲授權)。影片尺寸為1280 * 720，影格率為 30fps。

圖 3、Asian Volleyball Confederation in YouTube 頻道

2. 訓練 YOLO 模型：

YOLO 是一種物件偵測演算法，它的特色與以往物件追蹤模型的最大差距就是 YOLO 會將輸入照片分為更小的網格，而網格就可以組成不同組合大小的邊界框，然後不同邊界框就會依照邊界框基於某種物件的信心值(Confidence)來決定是否輸出結果，模型的預設為 30%。自從 YOLO 在 2015 年發行以來，YOLO 還在以飛快的速度繼續發展，直到 2024 年 2 月，YOLO 已經發展到第九代了，也就是 YOLOv9；而本專題所使用的為 2023 年五月推出的 YOLOv8。

YOLOv8 是它的製作團隊先前作品 YOLOv5 的改良版本，YOLOv8 相較於先前版本最大的區別就是模型結構內用了 c2f 模組來取代 c3 模組，c2f 是參考了 YOLOv7 的 ELAN 模組來改良 c3 模組後的結果。可以看到 Bottleneck 模組的變化，c2f 的 Bottleneck 模型運用了梯度流分支來避免梯度流失的現象，讓 YOLOv8 在相同的複雜度下有更好的精準度。

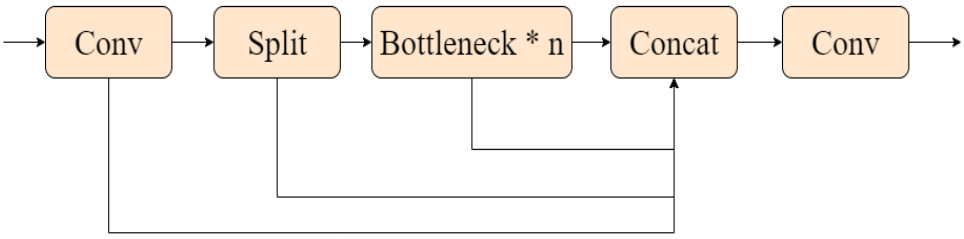


圖 4、c2f 模組改進了 c3 模組的缺點，將瓶頸層分流，避免梯度流失的

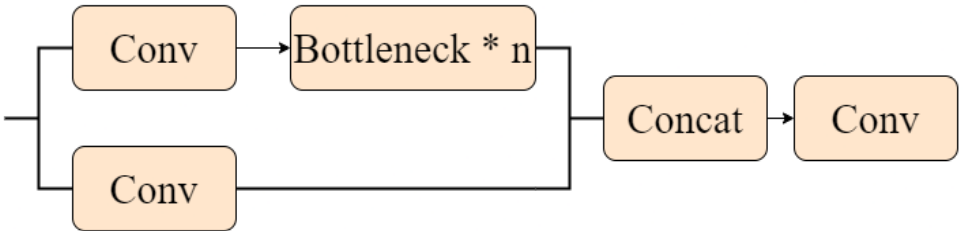


圖 5、c3 模組的建構方式可能會造成梯度流失

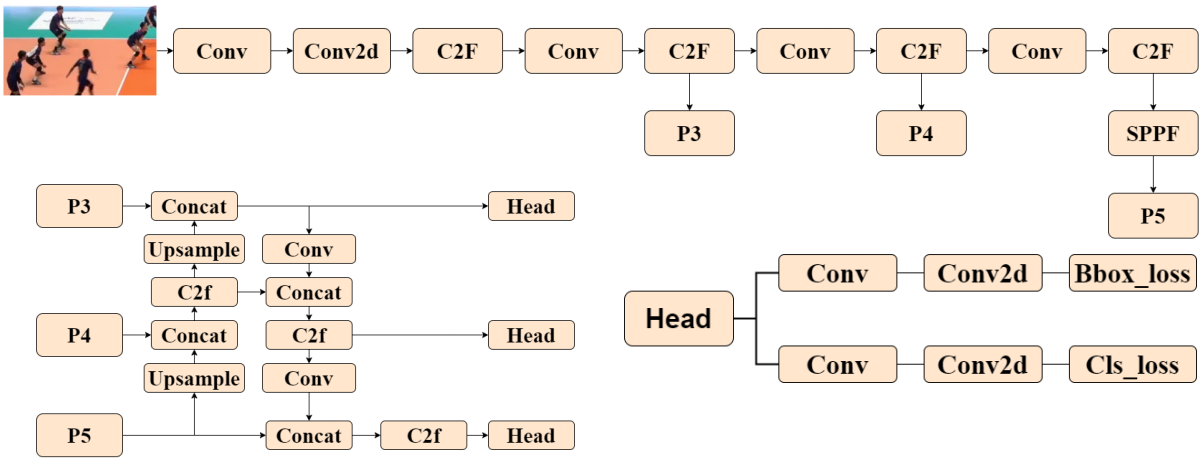


圖 6、YOLOv8 的基本模型結構

此專題所用到的 YOLO 模型總共有三種，分別為排球偵測、球員偵測、和動作偵測。在球員與動作偵測模型的訓練，本團隊在利用 Roboflow 上其他用戶分享的資料集來訓練，Roboflow 上有許多針對特定項目的公開資料集可以給其他用戶下載，因此有專門標註排球選手以及排球動作的資料集可以使用，用這樣的方法訓練出的模型表現相較於用模型預設的原始權重有顯著的提升。

3. 追蹤目標：

有了模型權重以後，就可以開始在影片上面執行看看效果如何並且開始做些應用。

YOLO 有很多種輸出結果，本團隊將輸出結果用 Data Frame 的方式做資料處理；如此一來，就有了球員站位、排球軌跡和動作發生的位置，這些資料可以成為之後戰術識別模型的輸入。

有了抓取資料的方法以後，現在要做的就是將整部影片當作輸入，就可以得到連續性的資料，還可以把整回合的位置圖視覺化，將球員的跑位圖畫出來。

4. 偵測戰術發生：

相較於球員站位，動作發生的時間非常短暫，因此本專題在動作偵測的資料儲存與站位會有所差異。在動作偵測方面對本專題來說最重要的兩個動作是「托球」和「攻擊」，當影片中在五秒的時間內出依序出現「托球」、「攻擊(假動作)」、「攻擊」，程式會自動將這回合所蒐集到的資料儲存下來，並交給之後的戰術辨別模型。

5. 投影：

攝影機在拍攝球場時，一定會有拍攝角度不同的問題產生；同時，直接以球場正上方拍攝的方法也較困難，所以要得到位置的平面圖的話就需要依靠投影的方式來將不同角度拍攝的照片轉換為俯視圖。

投影的基礎是基於透視變換來完成，透視變換的應用就是先標記影片中球場邊緣的四個點，再將其對應到投影平面上設定的四個點([120,80], [120,440], [840,80], [840,440])，對應完後會產生一個$3*3$的轉換矩陣；有了轉換矩陣以後，影片中所有的座標就可以依照這個轉換矩陣投影到平面上。

投影變換的公式為:

$$\begin{bmatrix} t_i x' \\ t_i y' \\ t_i \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

投影變換利用了投影中心、原像素點、目標像素點三點共線的原理來達成投影的效果；(x,y)為輸入點、輸出為$(t*x', t*y')$且ti為大小倍率，a表示旋轉、縮放的線性變換，b為平移向量，c為投影向量。

6. 統整資料：

在程式運行的過程中，我們會蒐集以下資料，「球員跑位圖」、「排球路徑圖」、「動作發生位置」 來當作本專題戰術辨別模型的輸入，此外球員站位以及排球運動路徑還可以畫成每個回合的跑位圖；到最後，模型辨別完戰術後，會將結果顯示在正在播放的影片當中。

由於辨識模型的輸入為一個 Data Frame，因此動作發生位置在絕大多數情況都會是空值，本團隊直接用 0 來代表空值；此外，當球類或位置出現空值時，本團隊則會利用空值前後的值來做線性插值，用前後兩點的中間點來盡可能還原空值。

7. 訓練戰術辨識模型：

本團隊的戰術辨識模型是一個前饋類神經網路，輸入為「球員跑位圖」、「排球路徑圖」、「動作發生位置」，輸出則是[A 戰術的機率, B 戰術的機率]；本專題目前準備了 22 部影片來訓練，訓練完以後模型的表現是 17/22。

8. 輸出結果：

在辨識模型輸出完結果以後，本團隊會將輸出結果呈現在正在播放的影片上面，並且將該次戰術執行的片段剪取下來然後存取到電腦裡面。

六、系統實現與實驗

準備資料集的平台利用了 Roboflow 網站上提供的介面來做資料處理，Roboflow 上提供了方便的工具讓用戶輕鬆的準備訓練 YOLO 所需的資料集。

1. 訓練 YOLO 模型：

此專題在模型訓練時所遇到最大的困難在於排球偵測模型的訓練，排球在比賽過程中常會以很快的速度在球場中移動，在這種時候球在每一幀都會有動態模糊的效果，導致本專題原本使用的 YOLOv5 追蹤效果極差，只有在排球影像清晰不模糊的時候才偵測的到，只要排球模糊模型就追蹤不到；本團隊後來靠增加資料集以及將模型升級為 YOLOv8 才達到了滿意的效果。

除了利用 Roboflow 以外，本團隊還有利用 Google Colab 來協助訓練模型；Google Colab 是 Google 公司免費提供的線上 Python 編輯以及執行平台，除此之外，Google Colab 最好的服務是提供用戶免費的 GPU 來幫助模型訓練得更快，因此本團隊可以先在 Google Colab 訓練好模型權重再將模型下載到電腦上面直接使用。

在標註完資料集後，本團隊進行資料增強，透過調整亮度、飽和度和對比度增加資料集中的影像資料，擴充資料集。

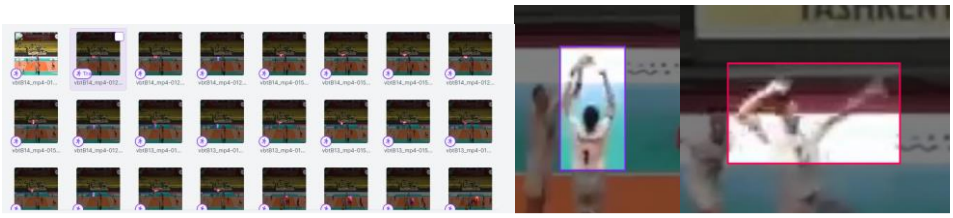


圖 7、Roboflow 動作資料集

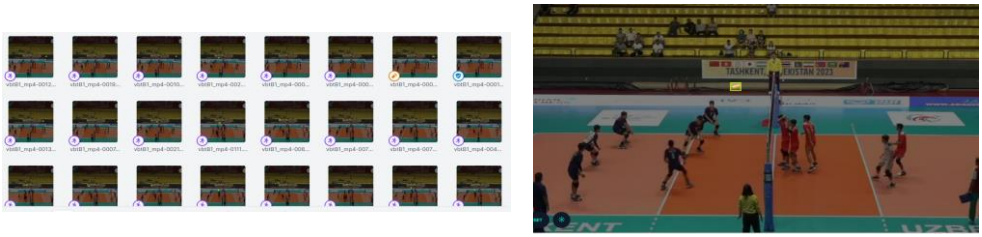


圖 8、Roboflow 排球資料集

圖 9、YOLOv8 訓練後動作辨識結果圖

原先 YOLOv8 並沒有排球動作識別的功能，透過訓練後 YOLOv8 能夠成功辨識出托球、攻擊。

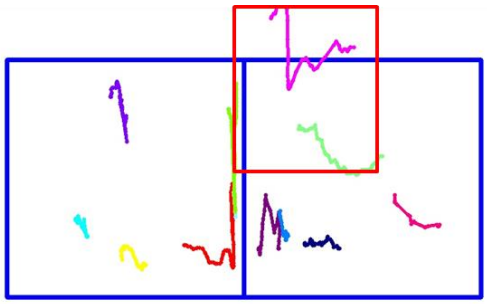
2. 追蹤目標：

圖 10、ByteTrack 追蹤球員的效果

透過 ByteTrack 追蹤球員，ByteTrack 會給辨識出的球員編號，本團隊將球員編號當作 Data Frame 的索引，將球員每幀資料存進 Data Frame 對應的位置。

3. 偵測戰術發生：

YOLOv8 模型、追蹤模型建立完後，接著要進行戰術發生的偵測，若球員執行托球動作，則開始將球員位置資料存進 Data Frame，托球動作發生後的五秒內沒有攻擊動作發生，則將 Data Frame 清空，若有攻擊動作發生就將球員跑位資料畫成跑位圖，此外 Data Frame 的索引值即為追蹤球員時的球員編號，可依此將不同球員軌跡用不同顏色表示。同理，可將球的軌跡、執行戰術的球員位置存進 CSV 檔中。



set_x	set_y	spike_x	spike_y	ball_x	ball_y
544	113	0	0	723	434
546	96	0	0	721	438
547	73	0	0	720	429
0	0	0	0	0	0
0	0	0	0	721	404
0	0	533	146	721	384
0	0	528	124	720	368
0	0	528	116	720	362
0	0	528	99	0	0
0	0	528	82	0	0
0	0	528	73	717	329
0	0	528	73	716	315
0	0	525	76	715	307
0	0	523	79	715	305
0	0	520	110	717	294
0	0	0	0	715	287
0	0	0	0	715	285
0	0	0	0	0	0
0	0	0	0	716	275
0	0	0	0	715	273
0	0	0	0	715	270
0	0	0	0	713	268
0	0	0	0	713	267
0	0	0	0	713	267

圖 11、資料儲存範例

4. 投影：

在進行球場平面投影前，本團隊先計算了球場四點投影後對應的點座標，透過將實體球場等比例縮小，並繪製於尺寸為1280 * 720的照片正中心，求出投影後對應的座標。

接著透過線性代數運算，將變換矩陣M求出後，使用變換矩陣將球員位置投影至俯視圖上，求出球員在球場上的正確相對位置。

在 OpenCV 資料庫中，有一些函式的功用就是在執行投影變換的，本專題運用到的就是其中的 getPerspectiveTransform 以及 warpPerspective，實體操作如下：

首先要點出影片中球場邊緣的四個點，將這四個點分別對應到 2D 圖上的四個預設座標 ([120,80], [120,440], [840,80], [840,440])，將這兩套座標載入 getPerspectiveTransform 就可以得到投影要使用到的轉換矩陣了。有了轉換矩陣後，影片中的每一個點都可以經過 warpPerspective 這個函式依照轉換矩陣進行投影，如此一來就可以達到以下的效果。

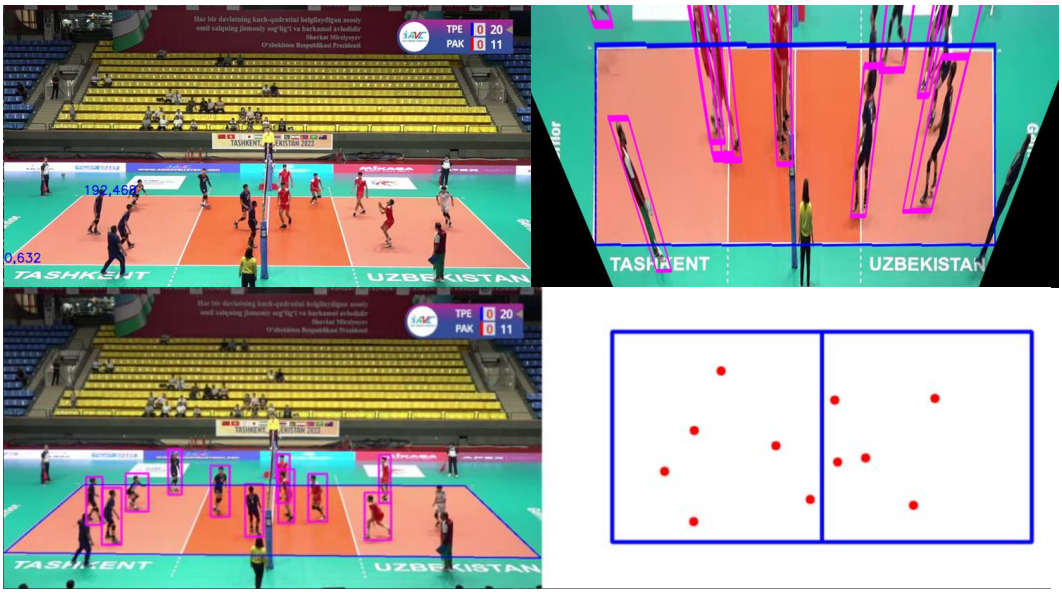


圖 12、透過轉換矩陣所達到平面投影的效果

5. 統整資料：

本團隊在抓完資料以後會將所有資料存入 Data Frame 當作辨識模型的輸入。

6. 訓練戰術辨識模型：

原先本團隊建立了一個 CNN 模型，並以跑位圖作為輸入，希望能透過跑位圖辨識出戰術，但因為小組時間差攻擊 A、B 式戰術的跑位圖特徵並不是很明顯，且跑位圖上有其他非執行戰術球員的移動軌跡，會造成混淆，導致訓練效果不好；後來本團隊改為使用存有球軌跡、執行戰術球員軌跡之 CSV 檔作為前饋類神經網路模型之輸入，準確度大幅提升。

七、效能評估與成果

在此專題中訓練了 YOLOv8-球、YOLOv8-球員、YOLOv8-動作識別、戰術辨識模型四個模型，本團隊分別為此四模型計算了準確率。

1. YOLOv8-球模型：

資料集：22 部影片，每部影片平均時長 3~6 秒，一秒 30 幀，共有 2641 張有球的照片。

正確判斷	誤判、未抓取到
1405	1236

準確率=1405/2641=0.53

誤判率=1236/2641=0.47

在此專題中所選的影片資料，由於背景顏色與排球顏色相似，再加上排球在運動時影像會有動態模糊的問題，本專題訓練此模型時資料量太少，以致只能抓到一半的排球位置資料，但只要還能辨識排球運動軌道，此部分的誤差不至於影響專題結果。爾後能再多加一些資料量訓練此模型，提高準確率。

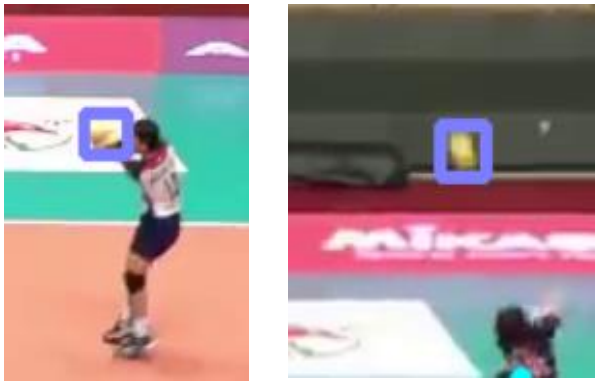


圖 13、YOLO 訓練排球權重的效果

2. YOLOv8-球員模型：

資料集：22 部影片，影片時長 3~6 秒，一秒 30 幀，場上有 12 個球員，共有 35784 張球員照片。

正確判斷	誤判、未抓取到
31603	4181

準確率 = 31603/35784 = 0.88

誤判率 = 4181/35784 = 0.12

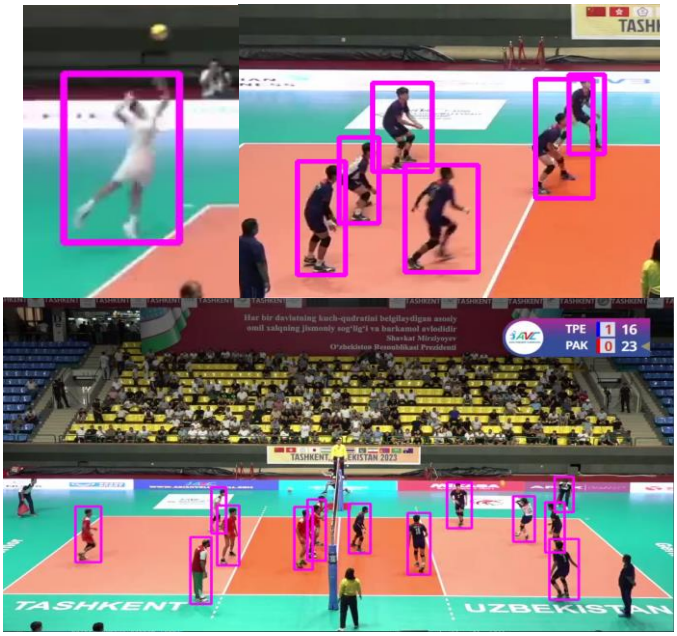


圖 14、YOLO 訓練球員權重後運行的效果

3. YOLOv8-動作識別模型：

資料集：22 部影片，影片時長 3~6 秒，一秒 30 幀。

(1) 托球(Set)：共有 162 張執行托球的照片

正確判斷	誤判、未抓取到
137	25

準確率 = 137/162 = 0.85

誤判率 = 25/162 = 0.15

圖 15、YOLO 偵測球員動作(舉球)

(2)攻擊(Spike)：共有 408 張執行攻擊的照片(含假攻擊)

正確判斷	誤判、未抓取到
347	61

準確率：347/408 = 0.85

誤判率：61/408 = 0.15

圖 16、YOLO 偵測球員動作(攻擊)

4.戰術識別模型：

資料集：小組時間差攻擊 A、B 式戰術分別有 9 部、13 部，共 22 部影片

辨識正確	辨識錯誤
17	5

準確率：17/22 = 0.77

誤判率：5/22 = 0.23

本專題中 YOLOv8-球員、動作辨識模型準確度表現不錯，戰術識別模型準確率有 77%，表示此專題計畫是可行的，日後可再訓練各部分模型，加強此專題。

在此專題中，辨識出戰術後會剪輯排球戰術片段及標註戰術類別、執行戰術之球員，結果如下：

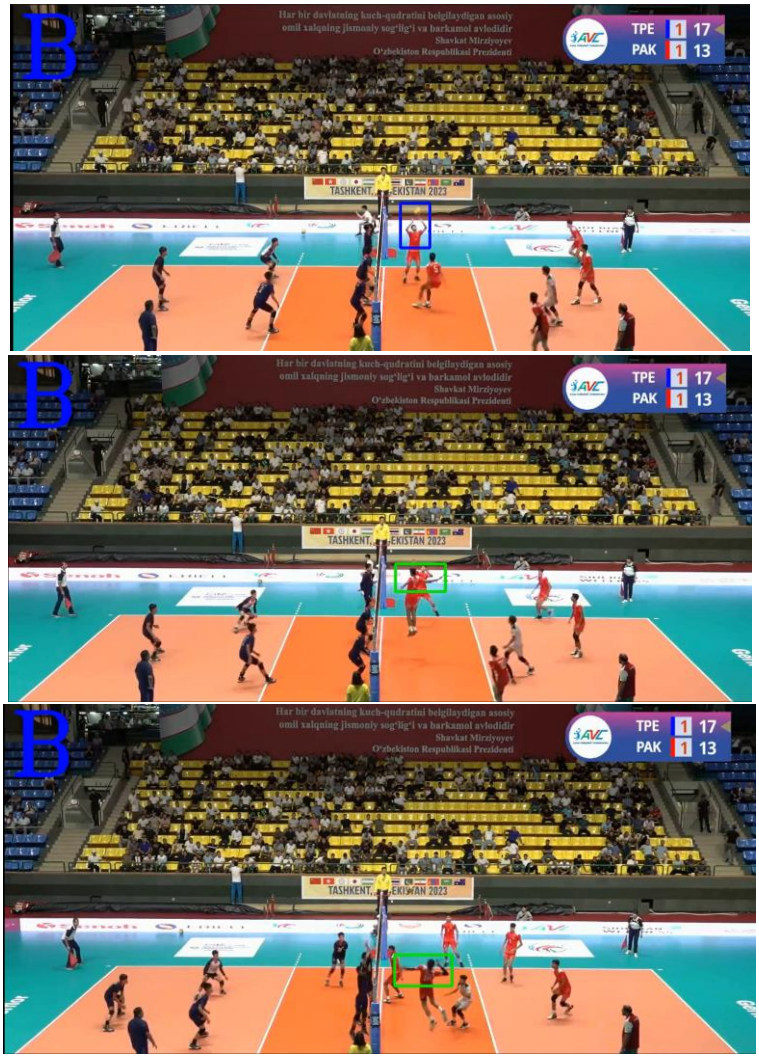


圖 17、將戰術類型顯示在影片中

八、結論

在此專題中使用了 YOLOv8 辨識球、球員、球員動作，並使用 ByteTrack 追蹤球員，獲得球員位置資訊，再將球員跑位資訊轉成跑位圖、存入陣列中，作為戰術識別模型的輸入，最後輸出戰術分類。YOLOv8-球員、動作識別的模型準確度皆達 85%以上，戰術識別模型準確度為 77%，能將約八成的小組時間差攻擊 A、B 式戰術分辨正確。日後可多加一些資料，訓練各模型，增加準確度，也能多增加其他戰術進行識別，完善此系統。

在指導教授專業且耐心的技術教導、組員間的合作、溝通，我們不斷學習，最後設計、實作出此效果不錯的排球戰術識別系統。

九、參考文獻

[1] Zhe Wang, Petar Veličković, “TacticAI: an AI assistant for football tactics”, 2024.

[2] Xin Zuo, “Visualization of Football Tactics with Deep Learning Models”, Article ID 9259328, 2022.

[3] Yunjun Xu, “A Sports Training Video Classification Model Based on Deep Learning”, Article ID 7252896, https://doi.org/10.1155/2021/7252896, 2021.

[4] Shen, L., Tan, Z., Li, Z. et al. “Tactics analysis and evaluation of women football team based on convolutional neural network”, https://doi.org/10.1038/s41598-023-50056-w, 2024.

[5] Tianyi Wang, Tongyan Li, “Deep Learning-Based Football Player Detection in Videos”, Article ID 3540642, https://doi.org/10.1155/2022/3540642, 2022

[6] Burić, M., Pobar, M., Ivašić-Kos, M., “Adapting YOLO Network for Ball and Player Detection”, http://doi.org/10.4108/eai.23-11-2023.2343216, 2019

[7] Cem Direkoglu, Melike Sah, Noel E. O’Connor, “Player Detection in Field Sports”, https://doi.org/10.1007/s00138-017-0893-8, 2017

[8] V, Shankara., Ahmed, Syed., M, Sneha., Jayabalasamy, Guruprakash., “Object Detection and Tracking for Football Data Analytics”, 10.4108/eai.23-11-2023.2343216, 2024

卡內基「溝通與人際關係」（ncu.edu.tw）

【排球人生】戰術球！ - 排球 | 運動視界 Sports Vision

Player and Ball Detection using Yolov8 + BotSORT tracking on a custom Dataset | by Nikhil Chapre | Medium

Track - Ultralytics YOLOv8 Docs

deep-learning-with-keras-notebooks/3.0-yolo-algorithm-introduction.ipynb at master · erhwenkuo/deep-learning-with-keras-notebooks · GitHub

深度學習-物件偵測:You Only Look Once (YOLO) | by Tommy Huang | Medium