AI CUP 2023 春季賽
「教電腦看羽球」競賽報告

隊伍：TEAM_2956
隊員：洪偉倫 (隊長)
Private leaderboard：0.4449 / Rank 5

壹、環境

一、 作業系統：Windows 10

二、 程式語言：Python

三、 套件/函式庫 (僅列出重要部分)

1. PyTorch: 主要使用的深度學習框架
2. TorchVision: 用於影像類資料前處理
3. Tensorboard: 用以記錄訓練結果
4. TensorFlow: 為符合 TrackNetV2 需求而安裝
5. Keras: 為符合 TrackNetV2 需求而安裝
6. scikit-learn
7. scipy
8. Numpy
9. Pandas
10. OpenCV
11. Pillow
12. ffmpegcv
13. imageio
14. MMDetection: 物件辨識之框架
15. MMPose: 人體姿勢偵測之框架
16. mmcv-full: MM 系列的依存套件
17. mmengine: 同上，MM 系列的依存套件
18. tqdm

四、 預訓練模型

1. TrackNetV2: 用於偵測羽球位置之模型
   來源: https://nol.cs.nctu.edu.tw:234/open-source/TrackNetv2
2. Faster RCNN: 用於偵測人體之模型
   來源: https://github.com/open-mmlab/mmdetection/tree/main/configs/faster_rcnn
3. HRNet: 用於偵測人體關鍵點之模型
   來源: https://github.com/open-mmlab/mmpose/tree/main/configs/wholebody_2d_keypoint/topdown_heatmap/coco-wholebody

貳、演算方法與模型架構

為了更好的訓練效果，有在損失函數中引入權重，因此對於每個欲預測之欄位，必須訓練各自專屬的模型，但模型架構大致上相同。

關於輸入資料部分，每份輸入資料長 L 幀，包含各幀資料或片段屬性資料，以下為可能包含之子資料：

A. 各幀資料：選手姿勢圖片
   為兩選手姿勢之黑白照片，故 shape = (BS, L, 2, 64, 64)

B. 各幀資料：選手姿勢數值
   包含兩選手 133 個關鍵點的 X 與 Y，故 shape = (BS, L, 4, 133)

C. 各幀資料：羽球位置
   包含 X 與 Y，故 shape = (BS, L, 2)

D. 各幀資料：該幀的時間相對位置
   「該幀 index ÷ 影片總幀數」，shape = (BS, L, 1)

E. 片段屬性資料：影片隸屬之場地背景圖 ID
   one-hot 編碼，共 13 種場地背景圖，故 shape = (BS, 13)

F. 片段屬性資料：打擊者
   one-hot 編碼，故 shape = (BS, 2)

關於模型部分，主要分為兩類：M-to-M 以及 M-to-1。
除偵測打擊幀之任務使用 M-to-M 模型，其餘任務皆使用 M-to-1。

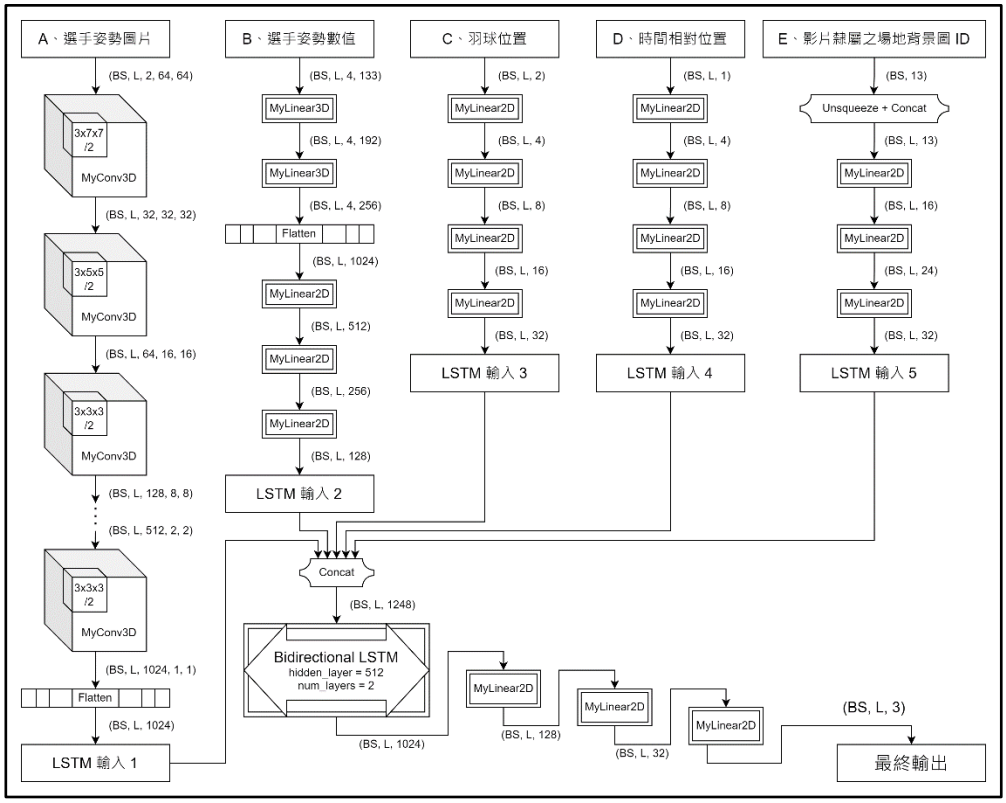
以下為各任務使用之輸入資料以及模型架構：

1. 打擊事件偵測
   由於使用傳統方法分析 TrackNetV2 對於羽球之偵測結果以辨識打擊事件之幀數與次數的效果不佳，故訓練此模型。
   此模型訓練目標為：辨識各幀中的兩位選手是否正在擊球。

   輸入資料包含代碼 F 以外的所有子資料。
   但由於此模型為 M-to-M 模型，輸入資料「E、影片隸屬之場地背景圖 ID」須轉變為可與多幀資料相容，故將其延展至長 L 幀之資料，也就是由 shape (BS, 13) 複製延展至 (BS, L, 13)。

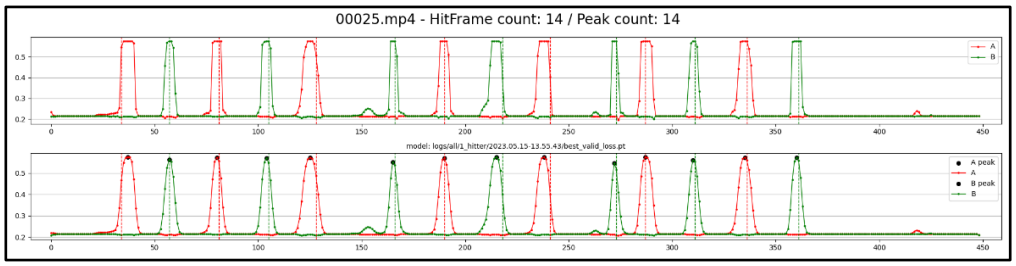
   輸出資料除各幀兩選手正在進行打擊的可能性，另有一欄為無人正在進行打擊的可能性，shape 為 (BS, L, 3)，並對最後一個 dimension 做 Softmax。

下圖為完整的模型架構，訓練時對深度與寬度做過微調：



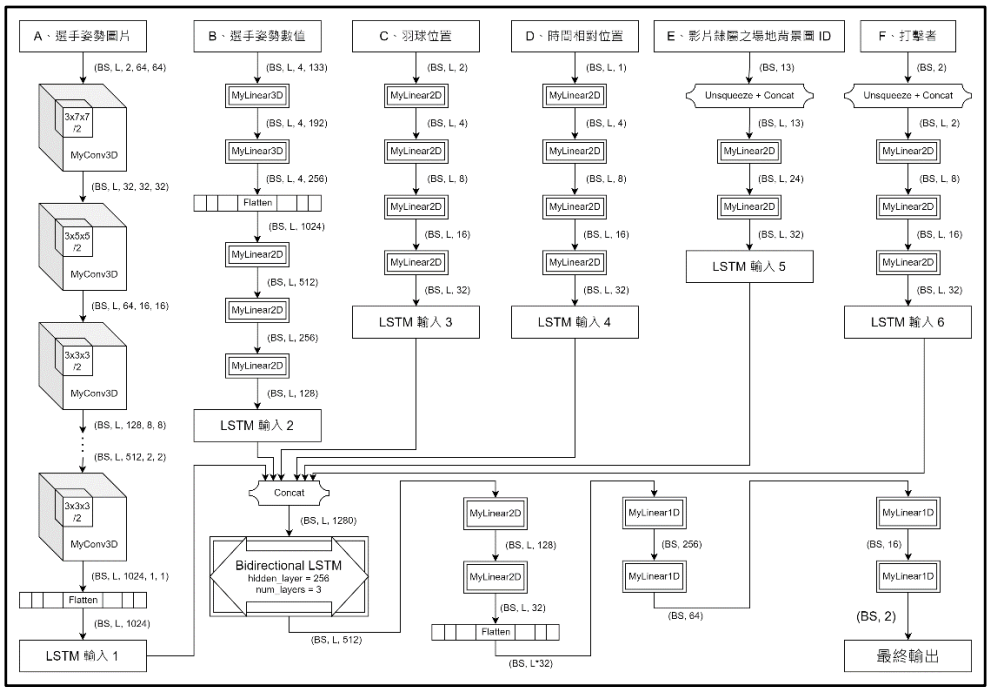
由於擔心將原影片切割成數個 L 幀片段投入模型辨識，對於各片段之首數幀與末數幀會有辨識效果不佳的問題，因此比起 stride = L，我選擇以 stride = 1 的方式擷取 L 幀片段，並將各幀在多次偵測中獲得的數值進行平均。

下圖為 train/00025.mp4 的範例輸出示意圖，上半部分為原始輸出，下半部分則是進行過高斯模糊後，取出波峰並去除多餘波峰的結果：



2. RoundHead、Backhand、BallHeight、BallType 欄位辨識
   輸入資料包含所有子資料。
   對於 RoundHead、Backhand、BallHeight 欄位，輸出資料為 1 與 2 的判斷信心，shape 為 (BS, 2)，並對最後一個 dimension 做 Softmax。
   對於預測 BallType 欄位的模型，輸出資料為 1～9 的判斷信心，shape 為 (BS, 9)，對最後一個 dimension 做 Softmax。

下圖為 RoundHead 最佳模型的完整架構，由於辨識各欄位的模型架構近乎相同，故不一一列出：

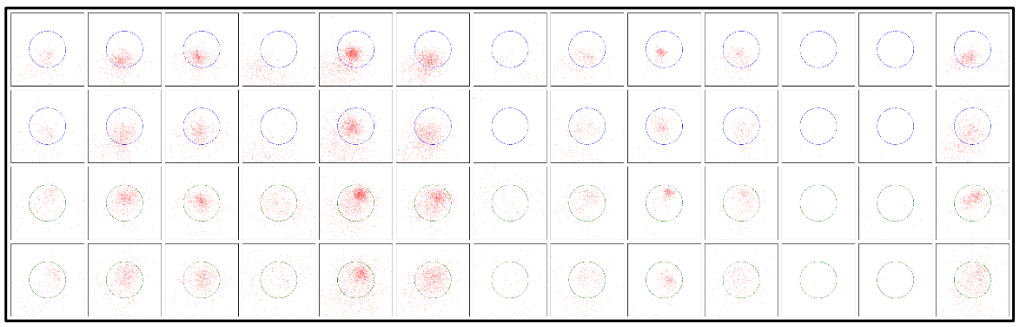


3. Landing、HitterLocation、DefenderLocation 欄位辨識

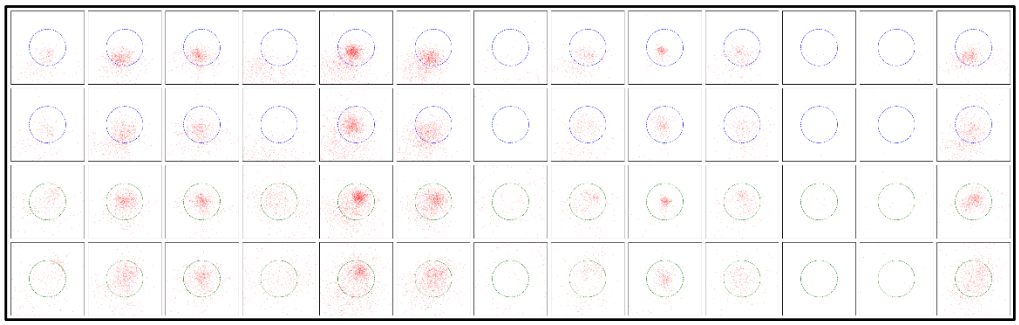
輸入資料包含所有子資料；輸出資料為 X 與 Y，shape 為 (BS, 2)。

模型架構與 2. 之內容相差不大，故不特別列出；但因正確率近乎為 0，最終不採用深度學習模型作為解方，而是採用統計方法。

下圖橫軸為各個不同的場景 (共 13 種)，縱軸為 MMPose 偵測到的兩位打擊者之右腳大拇指 (圖片正中央) 與正確打擊者位置 (紅點) 之相對位置。藍色虛線圓形之半徑為 10 像素、原點位於打擊者右腳大拇指，綠色虛線圓形之半徑為 10 像素、原點位於使用貪婪演算法找到的最佳偏移位置。



下圖與上圖相同，但綠色虛線圓形的原點改為去除極值後的平均數。



最終，我選擇以去除極值後的平均數，作為正式用以預測打者位置的偏移量。

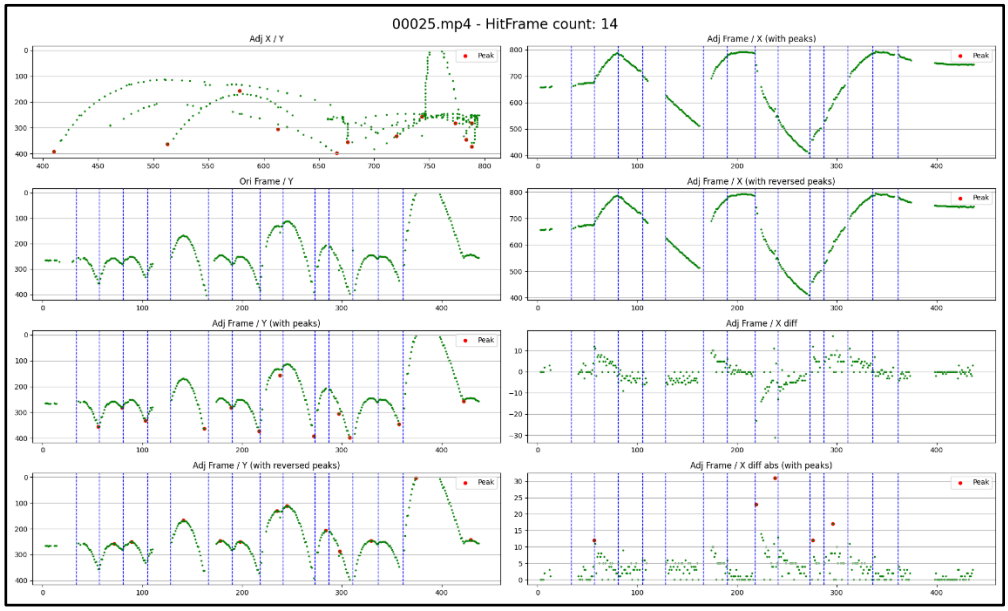
4. 獲勝者 (Winner 欄位) 辨識
   輸入資料包含所有子資料。
   由於需要最後一拍後的大量資訊才能判斷獲勝者，故輸入資料之長度部分為最後一拍之幀數前 15 幀至後 105 幀，共 121 幀。
   輸出資料為對於兩位選手獲勝的判斷信心，shape 為 (BS, 2)，對最後一個 dimension 做 Softmax。
   模型架構方面採用同 2. 之模型，故不特別列出。

參、創新性

我並未修改任何外部資料，也未使用、引入任何額外資料集。
論創新部分，我認為有以下三點：

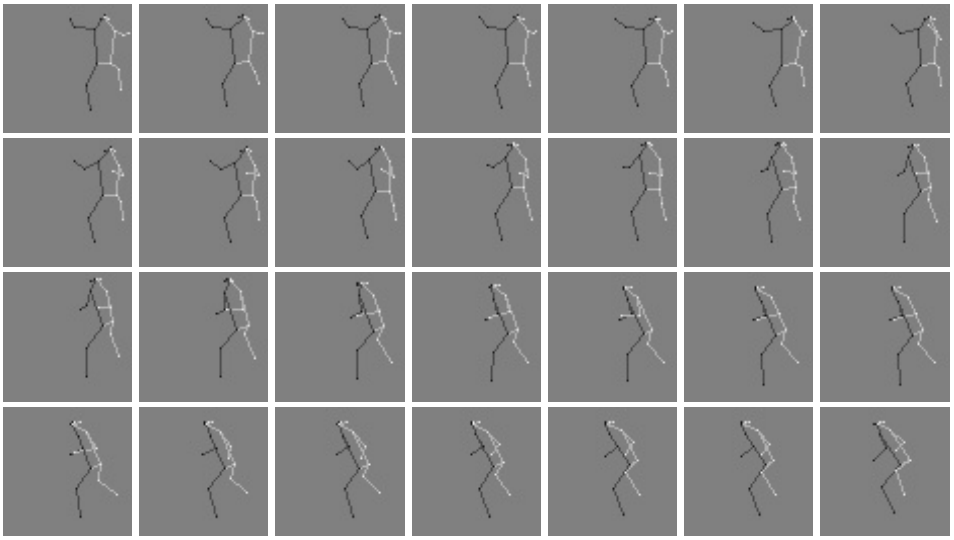
1. 我並未如 Baseline 所建議的方法，透過傳統方法分析 TrackNetV2 的辨識結果以辨識打擊事件，而是將 TrackNetV2 的辨識結果結合其他所有影片萃取出來的資訊，利用深度學習的方式辨識打擊事件。因此，我採用的方法非常看重 TrackNetV2 的辨識率。

   下圖為我嘗試透過分析位置和位移等數值以辨識打擊事件的示意圖：



   可以看到利用位置和位移等數值辨識打擊事件，無法對所有情況做出高品質的偵測。

2. 由於影像的讀取耗費大量時間與空間，我選擇將選手的原始影像透過姿勢關鍵點繪製成圖作為訓練資料。採取此方法能大幅減少訓練模型時耗費的 RAM，也去除了所有選手以外的雜訊；壞處則是如同上一點，非常看重預處理階段時，人像辨識與姿態辨識的模型成效。



圖中，底為 0.5、右半邊身體關鍵點為 0.9、右半邊身體關鍵點連線為 0.7、左半邊身體關鍵點為 0.1、左半邊身體關鍵點連線為 0.3。

3. 有別於傳統深度學習架構單一型態的輸入資料，在這次競賽中，我嘗試了多重型態的輸入資料，各個資料有著各自的意義，在模型內各自進行向前傳播後再匯流，最終被視為多幀連續資料餵入雙向 LSTM 中，再透過多層的 Fully Connection 逐漸萃取出最終答案。

肆、資料處理

一、 Background Extraction

透過人體偵測定位並去除影片中的兩位選手，然後計算各像素對各幀之平均，即可得出沒有兩位選手的場地背景圖。

下圖為 train/00001.mp4 的正常畫面與去除兩位選手後的場地背景圖：


然而對於某些影片，若選手常駐某個地點，或是因為影片過短導致選手幾乎沒有移動，則會得出不完整的場地背景圖。

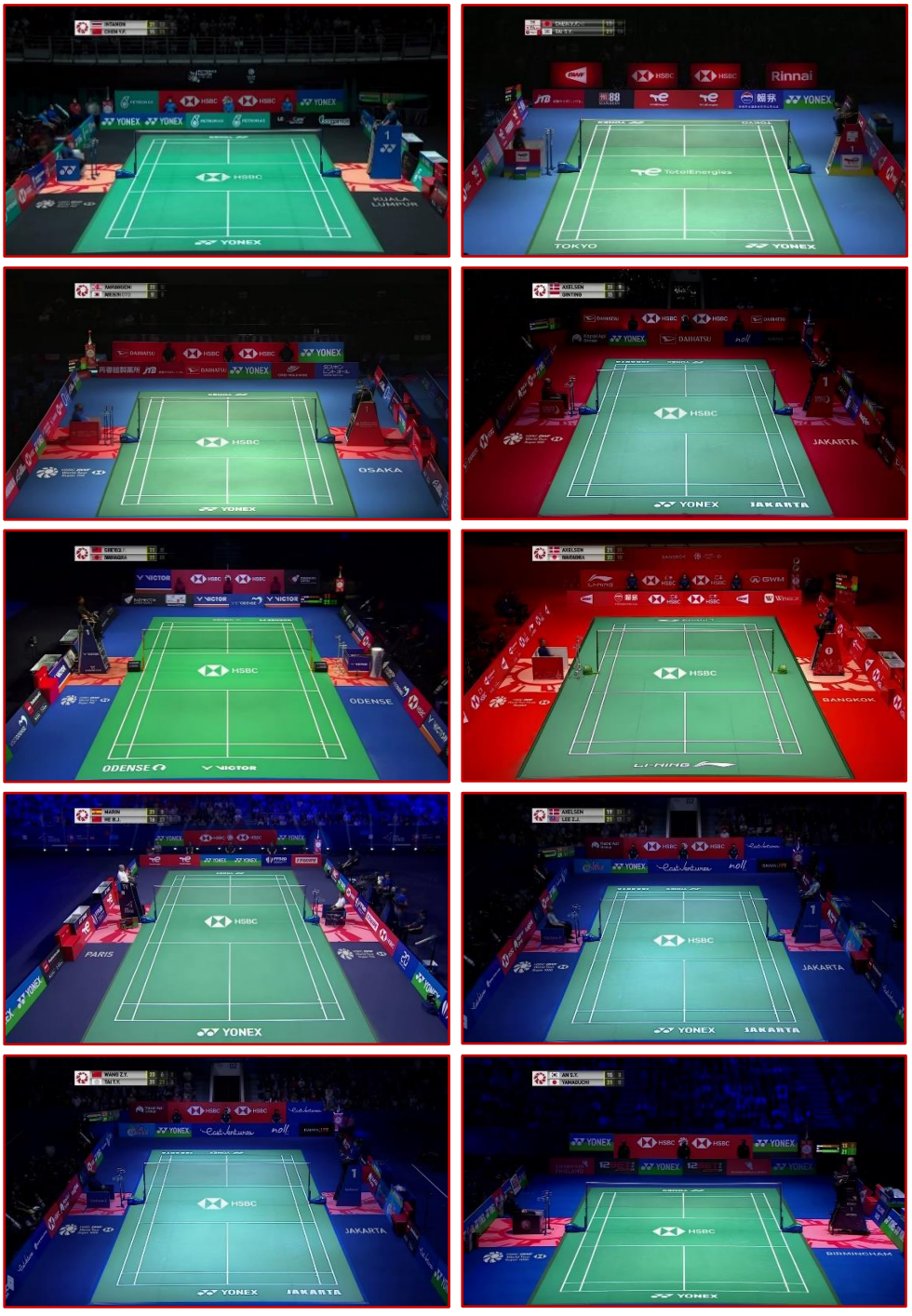
下圖為 valid/00033.mp4 去除兩位選手後的場地背景圖：

另外，由於我只去除兩位選手的人像本身，並外去除其陰影，故場地背景圖中的地上難免留下一些黑影。為了得到更好的場地背景圖，下一個步驟 Background Clustering 是必須的。

二、 Background Clustering

對於每部影片得出的場地背景圖，將相近者群聚起來並再次計算各像素對各幀之平均。

以下為 14 種資料集中出現過的場地背景圖：



得到所有背景圖後，就能透過手動標註場地邊界，更精準地偵測選手；並且也能透過各幀與場地背景圖的差異，區分並非進行比賽的片段如精彩回顧或選手介紹。

在前面部分有提及場地背景圖僅有 13 種，是因為第 14 種 (最後一張) 僅出現在測試資料集 (test) 中無法訓練，然而因其性質與第 12 種非常相近，故在 predict 時會將其歸類於第 12 種進行預測。

三、 Ball Detection

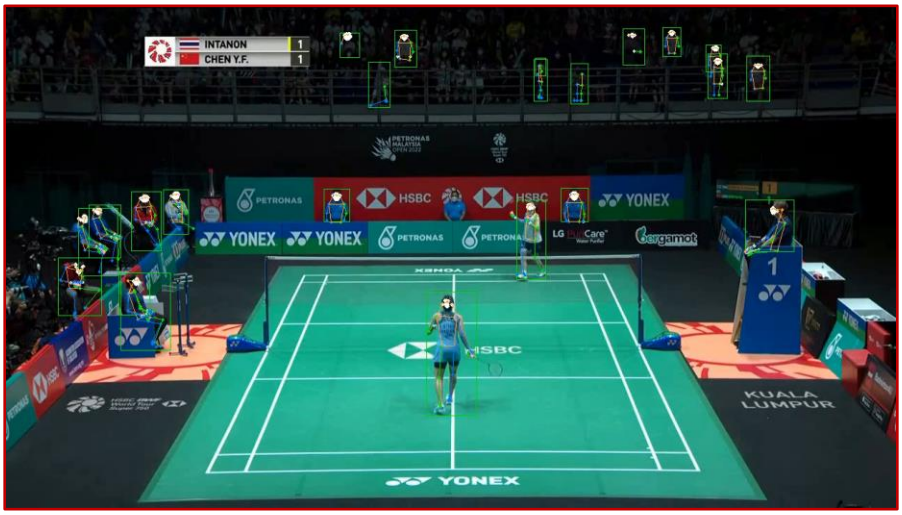
此部分使用官方提供的 TrackNetV2 進行羽球的定位，然後再透過個人撰寫的演算法去除明顯為誤差的偵測結果。

四、 Pose Estimation

此部分使用 Open-MMLab 提供之框架與預訓練模型。

首先使用 Faster RCNN 辨識人體並透過手動標註的場地邊界過濾出兩位選手。然後使用 HRNet 辨識兩位選手的姿勢，辨識出來的姿勢資料為各關鍵點的位置。

下圖為辨識效果示意圖：



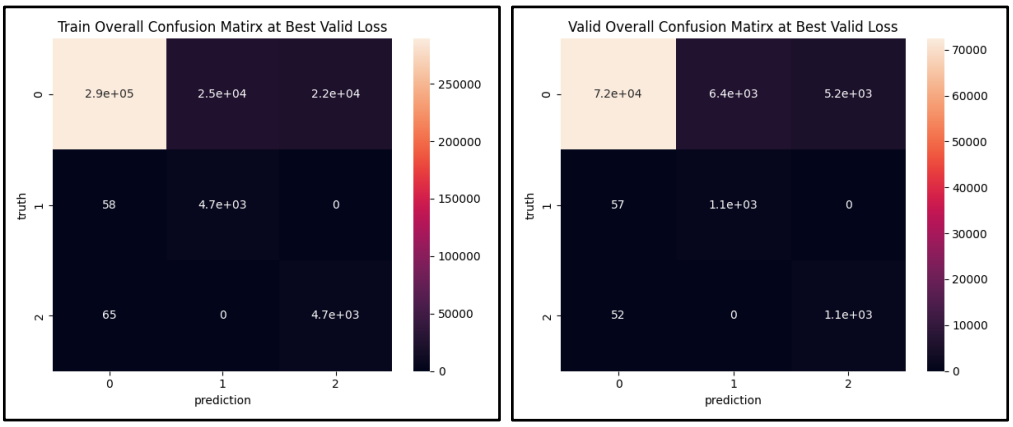
伍、訓練方式

整體而言，訓練方式採經典方法，為一次訓練搭配一次驗證，然後根據驗證成績是否最佳決定是否儲存模型。
其中，損失函數以及正確率的計算都有加入權重，以避免某些答案出現次數較多導致訓練偏誤。可惜的是，對於損失函數的部分，label smoothing 與權重功能無法同時開啟，否則會跳出錯誤，因此我無法開啟 label smoothing 以更進一步的增加訓練效果

1. 輸入資料
   繼承 PyTroch 的 Dataset class，自行撰寫屬於自己的 Dataset 並使用 PyTorch 的 DataLoader 處理資料 I/O 部分。
   DataLoader 有開啟 shuffle 功能、pin_memory 以及 drop_last 選項。
   每次進行訓練都會隨機取 160 支 train 影片 (20%) 做為該次訓練的驗證資料集，其餘 640 支影片為訓練資料集。
   訓練資料與驗證資料各自從其資料集萃取，故訓練資料與驗證資料將不會有同一部影片的不同片段。
   Batch size 設定通常為 80，有時為 64。

2. 雜項 (Misc)
   Optimizer 選用 Adam，代入整個模型的參數，並未將模型切分成不同區域分別計算反向傳播並優化。
   Learning Rate Scheduler 採用 Exponential。

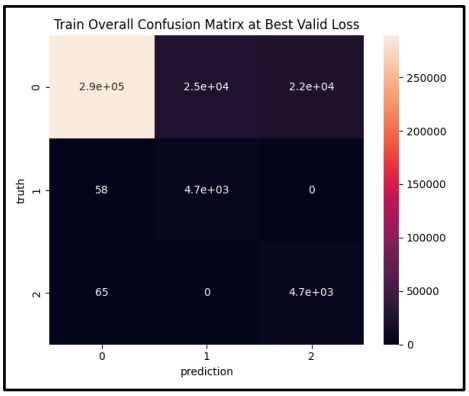
3. 記錄
   透過 TensorBoard 進行每次訓練的紀錄，並且對於類別偵測型的模型，也透過 confusion matrix 觀察模型效果。
   下圖為打擊事件偵測之最佳模型的 confusion matrix 示意圖：

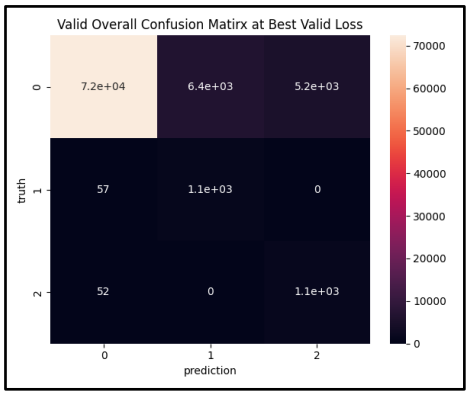


陸、分析與結論

一、 模型成效

1. 打擊事件偵測

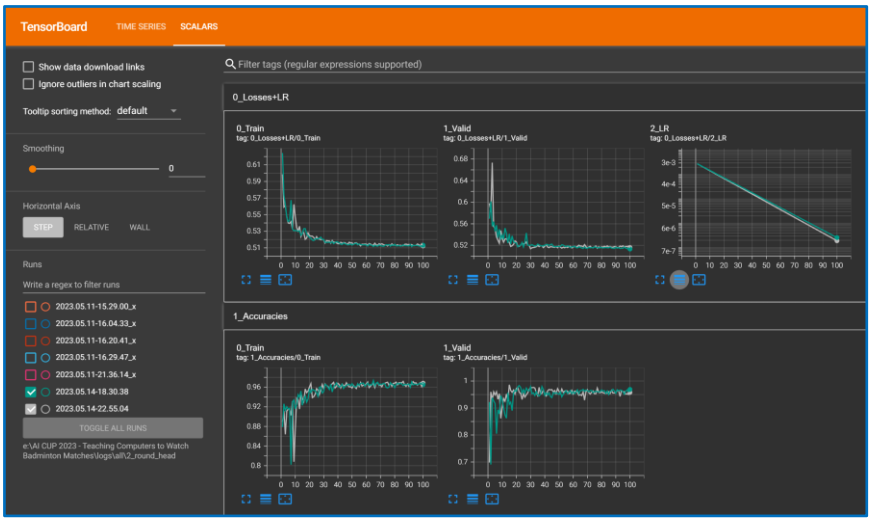




由於若要計算打擊事件偵測的準確度，還需對其做高斯模糊、尋找波峰、過濾波峰、幀數配對、幀數相差不超過 2 等等驗證，較為複雜而沒有實作，因此僅能透過 loss 判斷模型的好壞。

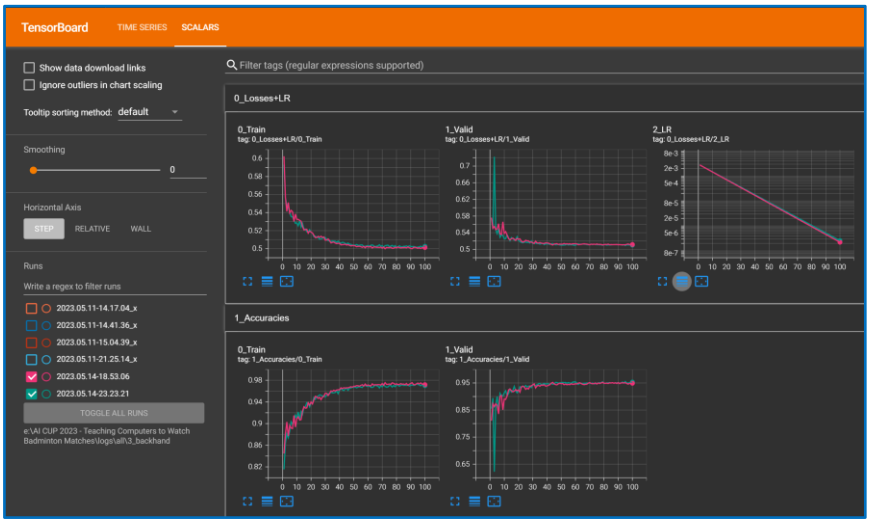
2. RoundHead

由下圖可以看出模型的驗證準確率高達 95%以上，效果非常好。



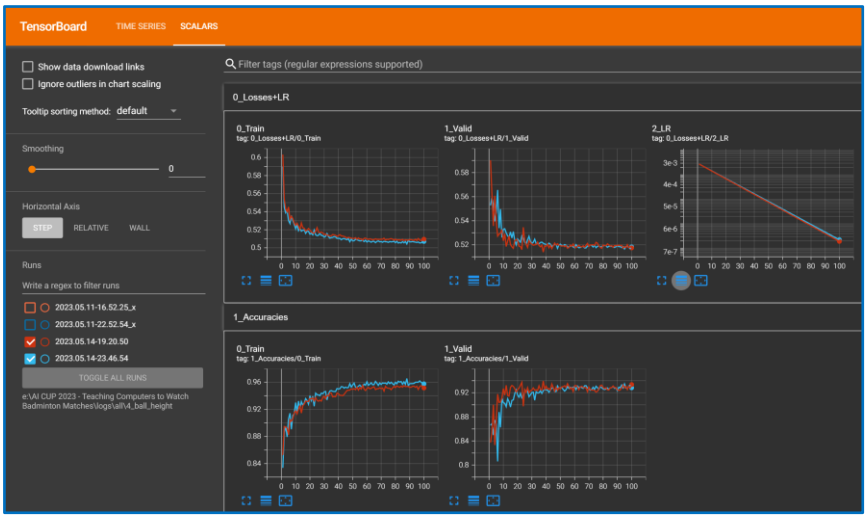
3. Backhand

由下圖可以看出模型的驗證準確率大略為 95%，效果非常好。



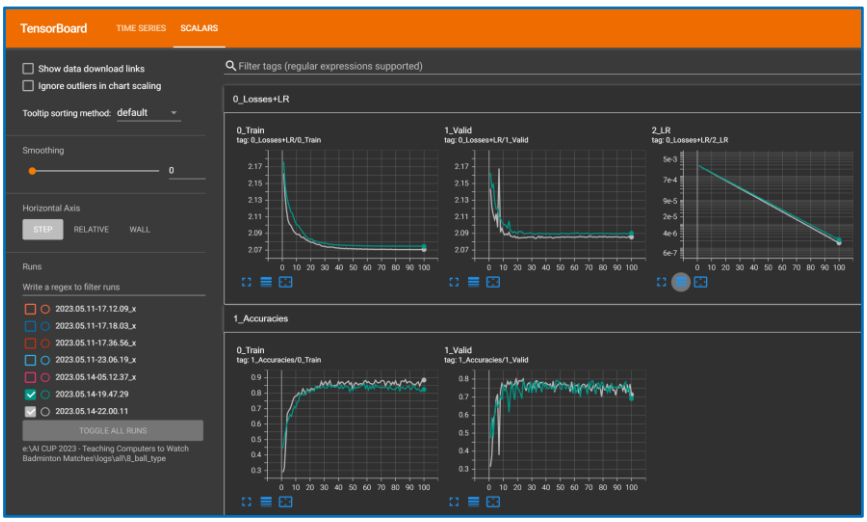
4. BallHeight

由下圖可以看出模型的驗證準確率大略為 93%，效果非常好。



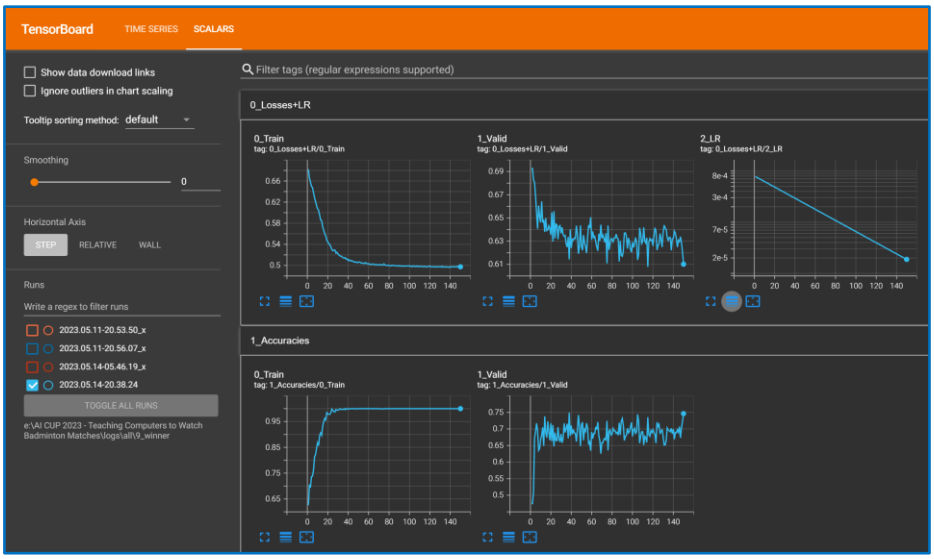
5. BallType

由下圖可以看出模型的驗證準確率大略為 75%，效果普通。



6. Winner

由下圖可以看出模型的驗證準確率大略為 70%，效果普通。
值得注意的是，透過訓練準確度早已到達 100% 能明顯看出模型的泛化能力不足，問題可能出在模型本身、輸入資料，抑或是最後一拍的片段數量太少。(訓練資料僅有 640 支影片，故僅有 640 個最後一拍的片段)



二、 未來展望

1. TrackNetV2

縱歷本次競賽，個人認為 TrackNetV2 有著最大的改進空間。將影片分割成數個各 1 幀或各 3 幀之片段顯然不像是處理連貫型資料的好方法；若能同時將選手姿態以更清楚的資料形式以及前數幀的羽球位置作為輸入，也許能使這個模型變得更有連貫性，也更能精準地預測羽球的下一個軌跡位置。

在開發出能準確預測羽球位置的模型後，就能將羽球比賽從原有的影片資料形式，萃取成僅包含「場地-選手-羽球」的精緻資料，不僅可以消除所有雜訊，也得以大幅減少訓練所需的 RAM。屆時，由於資料已萃取至最精緻，也許不需要輸入任何輔助資訊，也能有很好的辨識結果。

2. Pose Estimation

我發現在 Pose Estimation 的步驟中，可能是由於在影片中的人像較遠較小，又時常被網子擋住，非正式比賽中的 Player A 時常未被 Faster-RCNN 偵測到，或是低於用以過濾誤判的信心門檻。
因此，我認為下次可以採取更新更準確的辨識模型如 YOLOv7 來取代 Faster-RCNN 的部分。

3. Ball Type 與 Winner 辨識

我認為對於專門用以辨識 Ball Type 與 Winner 的模型，可以多給予一個「上一球的 Ball Type」作為輸入資料，希望可以達到類似於馬可夫鏈的效果。另外，我也希望可以進一步加強模型的泛化能力。

柒、程式碼

雖已附於信件附件之壓縮檔中，仍另外再附上 Google Drive 連結:
https://drive.google.com/file/d/1CUUWnb13W9LFIAeySchJnyvkOOuIlp9V/view?usp=share_link

捌、使用的外部資源與參考文獻

Ren, S., He, K., Girshick, R., & Sun, J. (2016, January 6). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* ArXiv. https://arxiv.org/abs/1506.01497

Xiao, B., Wu, H., & Wei, Y. (2018, August 21). *Simple Baselines for Human Pose Estimation and Tracking.* ArXiv. https://arxiv.org/abs/1804.06208v2

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019, February 25). *Deep High-Resolution Representation Learning for Human Pose Estimation.* ArXiv. https://arxiv.org/abs/1902.09212

Duan, H., Zhao, Y., Chen, K., Lin, D., & Dai, B. (2022, April 2). *Revisiting Skeleton-based Action Recognition.* ArXiv. https://arxiv.org/abs/2104.13586

Jiang, S., Campbell, D., Lu, Y., Li, H., & Hartley, R. (2021, July 29). *Learning to Estimate Hidden Motions with Global Motion Aggregation.* ArXiv. https://arxiv.org/abs/2104.02409

報告作者聯絡資料表

隊伍名稱	TEAM_2956	Private Leaderboard 成績	0.4449	Private Leaderboard 名次	Rank 5
身分（隊長/隊員）	姓名	學校系所中文全稱	學校系所英文全稱	電話	E-mail
隊長	洪偉倫 Wei-Lun, Hung			0975-848-033	aisu.programming@gmail.com
隊員 1					
隊員 2					
隊員 3					
隊員 4					
指導教授資料					
每隊伍至多填寫兩名	指導教授中文姓名	指導教授英文姓名	任職學校系所中文全稱	任職學校系所英文全稱	E-mail
教授 1					
教授 2					

★註 1：請確認上述資料與 AI CUP 報名系統中填寫之內容相同。自 2023 年起，獎狀製作將依據報名系統中填寫內容為準，有特殊狀況需修正者，請主動於報告繳交期限內來信 moe.ai.ncu@gmail.com。，報告繳交截止時間後將不予修改。

★註 2：繳交程式碼檔案與報告，請 Email 至：evawang.cs11@nycu.edu.tw，並同時副本至：moe.ai.ncu@gmail.com。缺一不可。報告「檔名」與「信件主旨」請寫「AI CUP 競賽報告與程式碼 / TEAM_？？？？/「教電腦看羽球」競賽」