

# 中華民國第 65 屆中小學科學展覽會

## 作品說明書

科 別：電腦與資訊學科

組 別：高級中等學校組

作品名稱：「羽」你同行－基於多模態深度學習與大語言模型的手機

端羽毛球姿態矯正系統

關 鍵 詞：羽球姿勢矯正、多模態分析、AI教練

編 號：

# 摘要

隨著台灣在奧運羽球雙打再奪金牌後，更多台灣人想嘗試這項運動。為提升新手學習效率，本研究旨在開發一套幫助新手透過手機進行姿勢矯正的軟體。首先，本研究選擇 BlazePose 模型當作人體姿態辨識模型，並訓練 MLP、KNN 及 YOLO 等姿勢分類器。其中，KNN 分類器在影片姿勢分類表現最佳，準確率達 **94.80%**。此外，本研究設計 Ball In Ball Out 流程可偵測羽球並自動剪輯單一打擊片段，準確率為 **95.89%**。經上述多模態資料分析後，再檢查常見錯誤姿勢及比對專家姿勢差異，最終 LLM 給予使用者姿勢錯誤矯正建議。羽球專家對姿勢矯正建議的平均認同百分比達 **82.40%**，羽球新手達 **81.16%**。

## 壹、前言

### 一、研究動機

羽球新手提升技能通常有三種方法，實戰對打訓練、聘請教練指導或觀看 YouTube 影片學習。然而，目前 AI 羽球教練軟體尚未普及，且相關研究尚無羽球新手姿勢指導功能，初學者無法即時調整動作，導致學習表現低落。

因此，本研究嘗試開發輔助羽球新手的姿勢矯正軟體，其中結合物件偵測與追蹤技術分析新手表現，檢查羽球專家認定的新手常見錯誤姿勢，再進行專家姿勢比對，並結合大型語言模型(LLM)，提供羽球新手有效的指導建議。此外，本研究也希望僅利用手機與腳架來運行此軟體，讓更多使用者都可方便操作與執行。

### 二、研究目的

本研究旨在開發一套羽球 AI 教練軟體，結合不同類神經網路模型，透過手機紀錄新手的動作影像進行電腦視覺分析，最後依據新手常見錯誤姿勢檢查錯誤，並結合 LLM 提供建議。

- (一) 探討人體姿態辨識(HPE)模型及機器學習技術應用在羽球姿勢分類。
- (二) 設計並實作 BIBO(Ball In Ball Out)方法應用於精確識別羽球動作分類的最佳時機。
- (三) 整合各 AI 模型的分析結果並給予新手完善的姿勢建議。

### 三、文獻回顧

#### (一) 羽球新手主要的學習姿勢

新手主要學習的姿勢有長球(clear)、平球(drive)、挑球(lift)、放網(net)與發球(serve)(VICTOR Taiwan, n.d.)[1]，以下為研究者示範這五類動作的連續影像。

1. 長球：常用於調動對手或擠壓對手空間。



圖 1-1：長球 (來源自行製作)

2. 平球：具攻擊性的技術，通常出現在接殺球的那一拍。



圖 1-2：平球 (來源自行製作)

3. 挑球：常出現於低於擊打腰部以下的球。



圖 1-3：挑球 (來源自行製作)

4. 放網：可以調動對手也可引導對手起高球，方便攻擊。

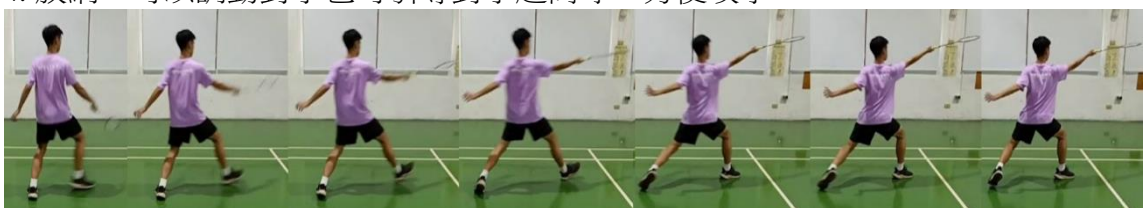


圖 1-4：放網 (來源自行製作)

5. 發球：在比賽開始時引導比賽的展開。



圖 1-5：發球 (來源自行製作)

## (二) 羽球新手常見的錯誤姿勢

錯誤姿勢可能影響擊球的準確性、穩定性以及增加受傷的風險。表 1-1 是研究者整理並示範羽球新手常見的錯誤姿勢(BadmintonInsight, n.d.)[2]。

表 1-1：羽球新手常見的錯誤姿勢 (來源自行製作)

姿勢	錯誤姿勢項目	圖示	詳細解說
長球 (clear)	1.側身不完全		身體轉動幅度不夠，無法有效利用身體力量帶動擊球。
	2.輔助手蜷縮		手臂未伸展，手臂與身體夾角 < 90 度、手臂夾角 < 70 度，無法有效利用身體帶動擊球。
	3.擊球手位置過低		大臂和身體夾角過小，無法在最高點擊球。
平球 (drive)	1.球拍未舉起		未能立即舉拍準備下一球
	2.身體直立		身體沒有稍微前傾壓低重心預備
挑球 (lift)	1.膝蓋超伸		膝蓋超出腳尖，腿部夾角 < 90 度，身體壓力集中在膝蓋上。
	2.無慣用腳在前		擊球手和前腳不同
	3.擊球手引拍過大		擊球前向後引拍過大
放網 (net)	1.手臂蜷縮		球拍距離身體太近沒有伸展，手臂夾角 < 120 度。
	2.出手點過低		球拍低於胸口，沒有在高點擊球。
	3.膝蓋超伸		膝蓋超出腳尖或腿部夾角 < 90 度
	4.無慣用腳在前		擊球手和前腳不同
發球 (serve)	1.無慣用腳在前		發球時必須慣用腳在前，非慣用腳在後。
	2.發球位置不當		發球時，擊球點應位於腰部附近。

### (三) AI 技術應用在羽球運動中的相關研究

目前 AI 技術應用在羽球專案與論文可分為兩類：第一類為羽球比賽數據分析，例如最長來回球數、球速及落點追蹤；第二類為羽球姿勢的分類，以探討比賽策略。參考的文獻中尚無針對羽球新手姿勢指導的研究。表 1-2 為研究者依據相關因素統整。

表 1-2：目前AI技術應用於羽球之相關研究主題

專案/論文	作者	年份	IOT裝置 及其種類	攝影機 架設角度	是否用於 賽事分析	姿勢分類 及方法
Clutch： AI for Racket Sports[3]	Kári Gunnarsson	2021	否	正後方	是	是 (未公開)
TrackNet： A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications [4]	Yu-Chuan Huang et al.	2019	否	正後方	是	否
Badminton Coach AI： 基於深度學習之羽球賽事 資訊分析平台 [5]	王威堯 et al.	2020	否	正後方	是 (球種分類)	是 (SVM、RF、XGBoost)
智能球拍[6]	成大與逢甲 共同研發	2020	是(陀螺儀)	無攝影機	否	否
基於人體姿勢估計與機器 學習的羽球球種分類 研究 [7]	Shiela Mecha Cabahug	2022	否	正前方	否	是 (CNN、LSTM、CNN-LSTM)
Deep Learning Based Technical Classification of Badminton Pose with Convolutional Neural Networks [8]	Tukino et al.	2024	否	正後方、 側邊	否	是 (CNN)
AI coach for badminton [9]	Toshniwal Dhruv et al.	2022	是 (IMU、Opal、XiaoYu 2.0感測器)	正後方	否	是 (揮拍軌跡 來做姿勢 分類)

## 貳、研究設備與器材

### 一、研究硬體

#### (一) 電腦

1.CPU：Intel® Core™ Ultra 9 285K

2.GPU：Nvidia Geforce RTX 4060 Ti 16GB

#### (二) 羽球

#### (三) 球拍

#### (四) 手機 (型號：OPPO RENO 12 F)

#### (五) 手機腳架

### 二、研究軟體

#### (一) 程式語言

1. Python：由於 Python 易於撰寫，本研究選用 Python 作為軟體開發程式語言。

#### (二) Python 程式庫

1. csv：Python 內建模組，用於儲存 CSV 檔案格式的姿勢數據。
2. flask (版本：3.1.0)：提供輕量級網頁框架功能。
3. joblib (版本：1.4.2)：用於序列化與儲存機器學習模型。
4. matplotlib (版本：3.10.0)：負責繪製實驗結果的圖表。
5. mediapipe (版本：0.10.20)：人體骨架關鍵點追蹤函式庫。
6. moviepy (版本：1.0.3)：負責影片的放慢。
7. numpy (版本：1.26.4)：數學運算函式庫，負責處理實驗數據的中心化。
8. opencv-python (版本：4.11.0.86)：負責影片的讀取、分割及預處理。
9. openai (版本：1.61.0)：負責連線得到 ChatGPT-4 的姿勢提供建議。
- 10.pandas (版本：2.2.3)：負責讀取 csv 檔案。
- 11.scikit-learn (版本：1.6.1)：開源機器學習庫，負責產生模型指標報告。
- 12.seaborn (版本：0.13.2)：負責混淆矩陣的生成。
- 13.tensorflow (版本：2.18.0)：開源的深度學習框架，負責深度學習模型訓練。
- 14.tensorrt (版本：10.8.0.43)：負責 YOLO 模型的量化。
- 15.time：Python 內建模組，負責計算程式推論時間。
- 16.ultralytics (版本：8.3.70)：YOLOv11 物件偵測模型的函式庫。

#### (三) 額外工具

1. CVAT：本研究用於標註邊界框，負責羽球影像的預處理。

## 參、研究過程及方法

### 一、研究架構

本研究架構分為三個部分：資料蒐集及預處理、模型訓練、與系統整合，使用者最後將會收到姿勢建議與姿勢比較影片。

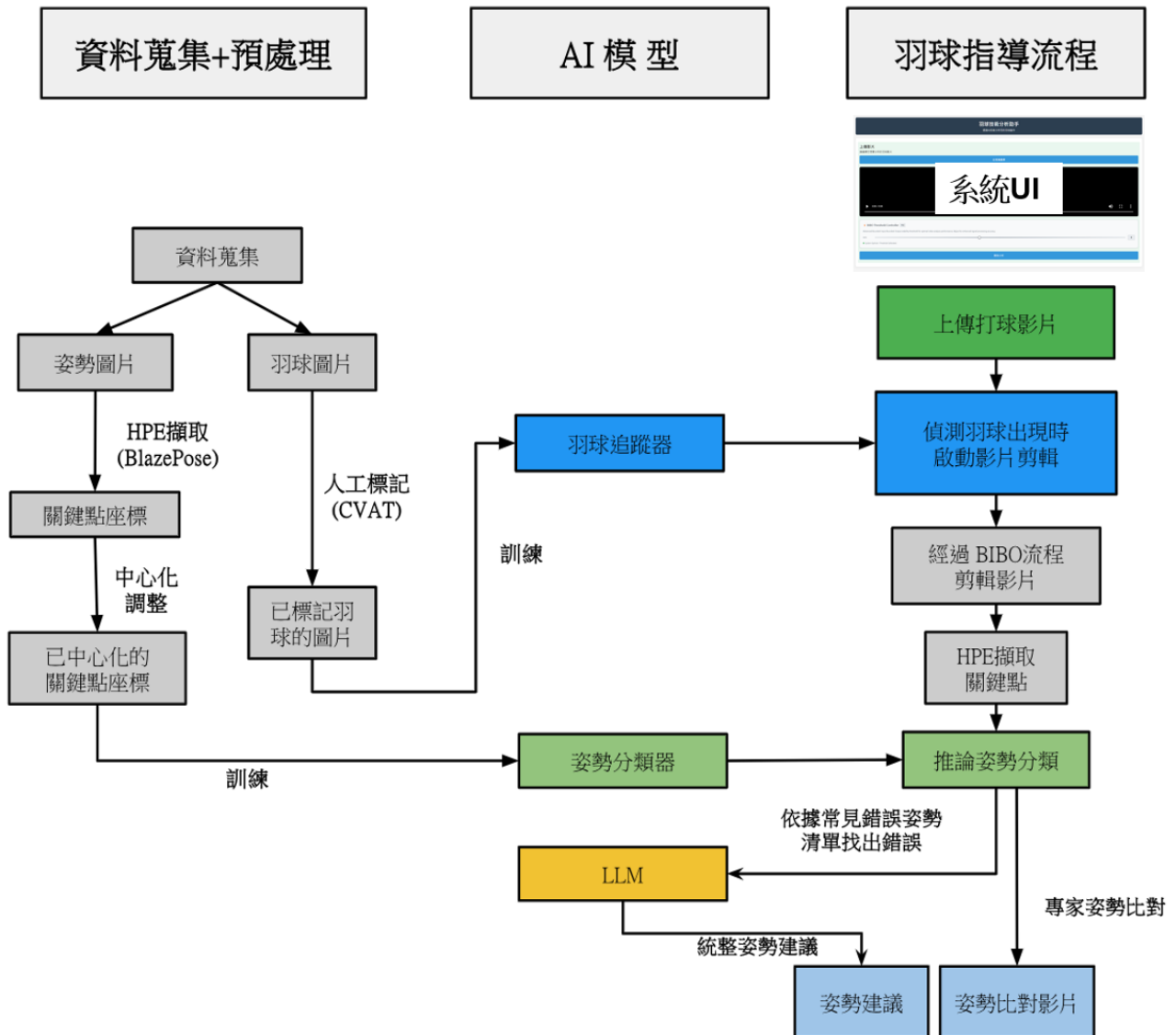


圖 3-1：研究架構 (來源自行製作)



## 二、相機類型與位置

目前 AI 羽球研究大多使用可覆蓋全場、高約兩公尺的攝影機腳架或壁掛式攝影機，但這類設備對新手而言，設置與操作較為不便(見表 1-2)，本研究希望新手可以以較低硬體限制獲得羽球姿勢改善建議。

### (一) 採用手機作為攝影工具

考量到手機普及性，使用者只需要一支能夠拍攝羽球半場的智慧型手機，配合高度 150 公分以上的腳架，即可進行錄影並獲得姿勢調整建議。

### (二) 腳架架設位置

如圖 3-2 所示，腳架應放置於距離中線 220 公分、右發球線 60 公分處。紅色為最佳攝影範圍，球場中粗黑線代表球網，紫色圓形標示新手站立位置。對側場地(藍色部分)可搭配發球機或與朋友練習，由於尚未進行大範圍的跑位訓練因此適合新手進行基礎羽球打擊練習。將攝影機架設在側邊相較於後方更能看到更多使用者的動作細節。

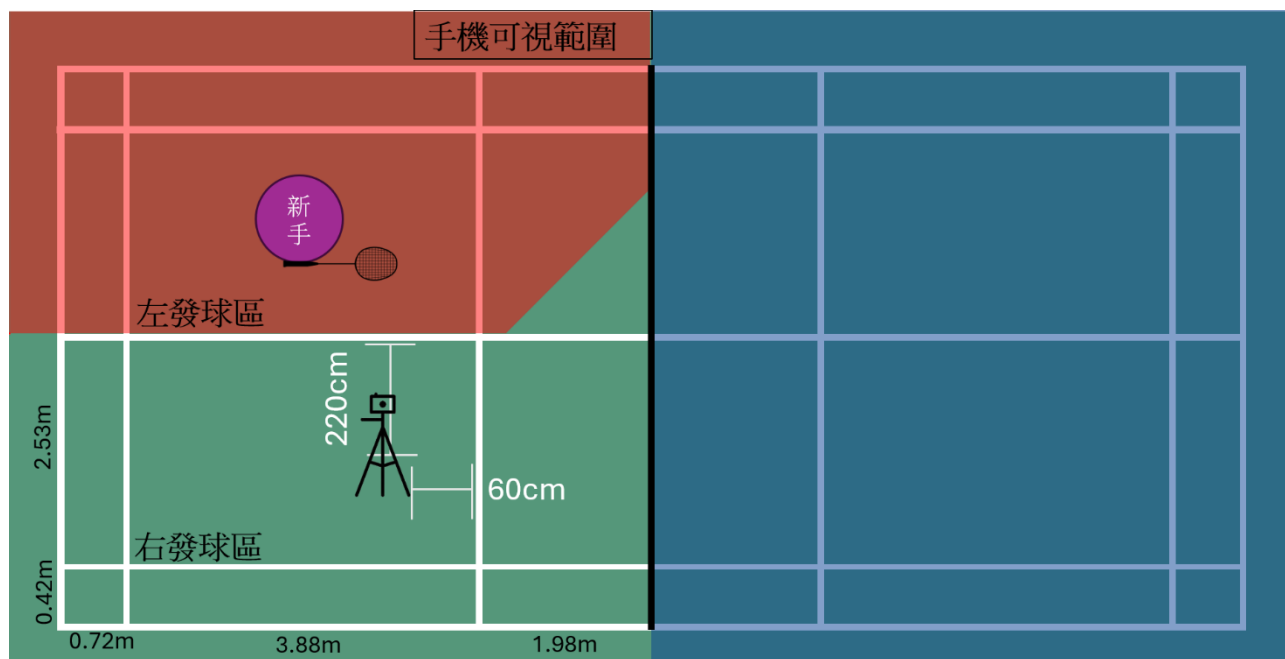


圖 3-2：攝影機架設示意圖 (來源自行製作)

## 三、研究限制

本研究要求使用者必須面向球網且與鏡頭平行。本研究對象僅限於右手持拍的羽球新手。



#### 四、資料蒐集

##### (一)資料來源

本研究所使用的羽球圖片與羽球姿勢的人體骨架關鍵點序列資料，皆於本校羽球場自行蒐集。此外，研究者錄製新手及羽球校隊球員的羽球打擊影片作為模型訓練學習之用。

##### (二)資料前處理

圖 3-3 為本研究資料收集的流程圖，羽球追蹤器及姿勢分類器的資料皆先使用 OpenCV 套件從影片中提取圖片幀，隨後進行預處理階段。

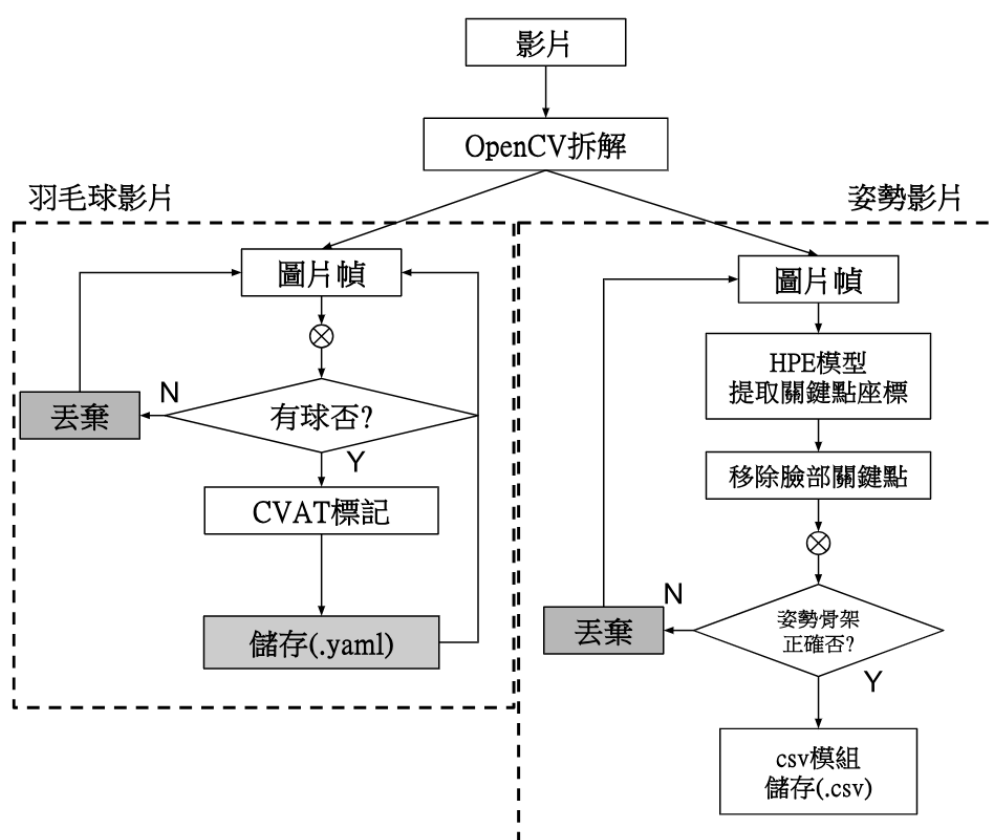


圖 3-3：資料收集及預處理流程圖 (來源自行製作)

如圖 3-3 所示，錄製完多段打羽球的影片後，先使用 HPE 模型(以 BlazePose 為例)對姿勢圖片提取關鍵點，接著移除臉部關鍵點，僅保留鼻子關鍵點作為頭部中心指標。接著透過程式自動將關鍵點連結為骨架資料，並人工檢視及移除不完整的姿勢骨架訓練圖片，以確保關鍵點序列的正確性。如圖 3-4，表格共 47 個欄位(46 個座標欄位及 1 個 Pose 標籤欄位)，Pose 標籤欄位含 clear, drive, lift, net, serve, other 等六類，最終將資料保存為 csv 檔案。

x0	y0	x11	y11	.....	x31	y31	x32	y32	Pose
0.287418	0.396891	0.309026	0.432695		0.332704	0.714211	0.253247	0.705173	clear
0.257729	0.350612	0.279224	0.402007		0.331536	0.70554	0.245155	0.776767	clear
0.412921	0.344964	0.427695	0.384592		0.462739	0.685123	0.411491	0.698327	clear

圖 3-4：關鍵點座標資料示意圖 (來源自行製作)

本研究蒐集羽球姿勢骨架關鍵點序列資料共計 19,885 筆。其中，「其他」類包含選手站立、轉身或蹲下等動作，旨在幫助模型更準確區分不同姿勢，「發球」類骨架關鍵點序列數量較少因為發球姿勢特徵明顯，加入過多資料會使姿勢分類模型對姿勢的決策有偏誤。

表3-1：羽球姿勢骨架關鍵點序列資料數量統計

	放網	長球	發球	平球	挑球	其他
數量(筆)	4,645	3,445	1,545	3,566	3,820	2,864

羽球圖片如透過 CVAT 標註羽球邊界框(class, x, y, w, h)，指邊界框的寬度、高度以及其中心點的位置。對於沒有球的照片，會將其刪除，不會用於訓練。圖 3-5 是使用 CVAT 平台標註羽球邊界框，用於訓練羽球追蹤器，其中「ball」指的是羽球，class 令其為 0。即使羽球看起來是模糊的，研究者仍然進行標註。本研究共蒐集了 5,962 張羽球圖片。

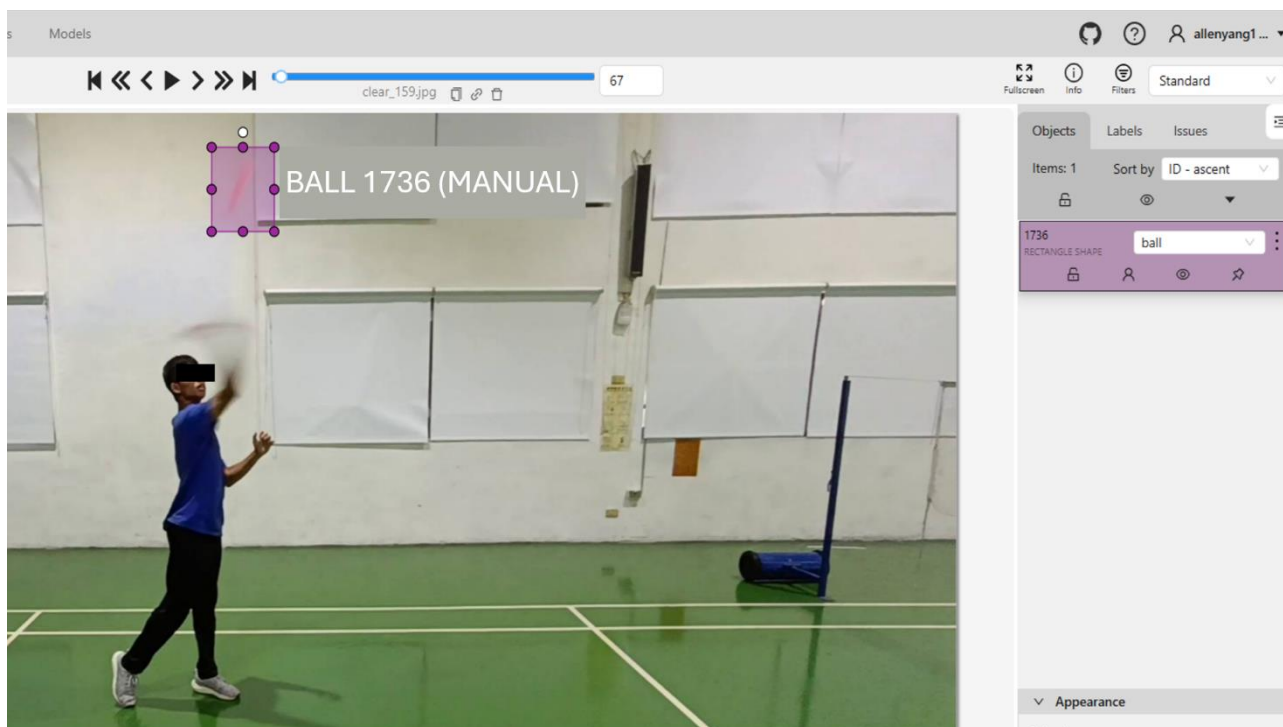


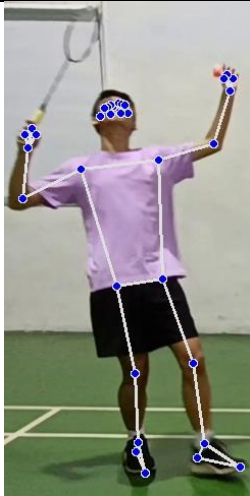

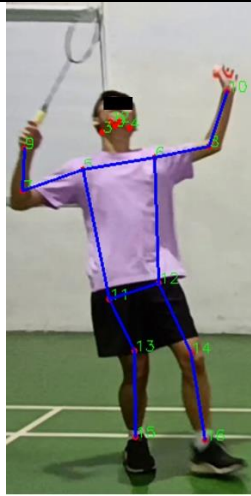
圖 3-5：利用 CVAT 標記羽球圖片 (來源自行製作)

#### 四、探討人體姿態辨識模型及機器學習技術應用在羽球姿勢分類

##### (一) HPE模型選擇

本研究比較 BlazePose(lite)、YOLOv11n-pose 和 MoveNet-lightning 三種常見姿勢估計模型。研究者使用了 300 張羽球姿勢圖片進行測試，本研究以人工審核 HPE 模型標註完成的圖片，標註正確的標準包括關鍵點是否對齊圖片及是否完整繪製，平均推論時間是使用 Python 的 time.time() 函式推算 HPE 模型標註 300 張圖片的平均時間。

表 3-2：BlazePose(lite)、YOLOv11n-Pose、MoveNet-lightning 比較分析表 (來源自行製作)

模型	BlazePose (lite)	YOLO v11n-Pose	MoveNet-lightning
關鍵點數量	33個	17個	17個
範例			
識別人數	單人	多人	多人
準確率(%)	218 張 / 300張 (72.67%)	208 張 / 300張 (69.33%)	12張 / 300張 (4%)
平均推論時間	0.01460s	0.01412s	0.00736s

BlazePose 額外提供腳趾與腳跟的位置資訊，並在手腕處多提供三個關鍵點。因此，本研究最終選擇 BlazePose 作為人體姿態辨識模型，以獲取更多可分析的關鍵點。

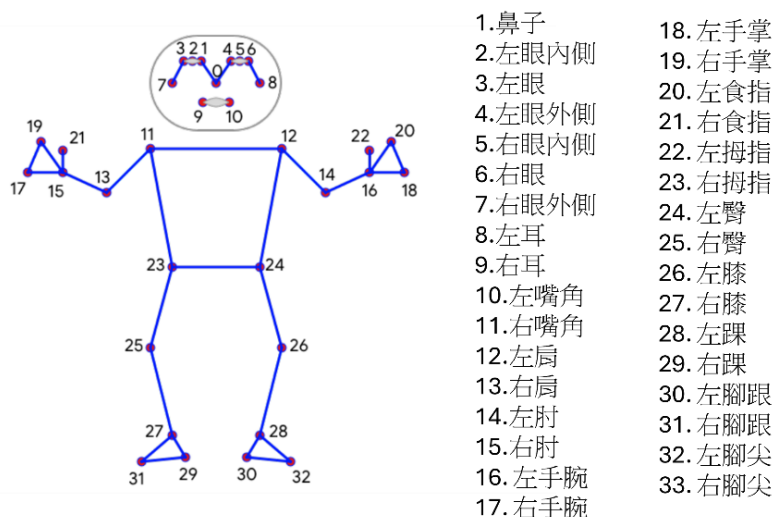


圖 3-6：BlazePose 關鍵點圖 (來源：Bazarevsky. V. et.al., 2017) [10]

## (二) 關鍵點序列預處理方法

球員在鏡頭前的位置對座標序列的絕對位置對姿勢判定影響巨大，因此將羽球姿勢骨架關鍵點序列進行「中心化」預處理。首先，定義身體的中心點座標為關鍵點 11、12、23、24 (見圖 3-6) 的  $x$  與  $y$  值的平均值，接著將每個關鍵點  $(x_i, y_i)$  的座標減去此中心點，表示將全身的骨架圖形中心點位移至座標(0, 0)位置。

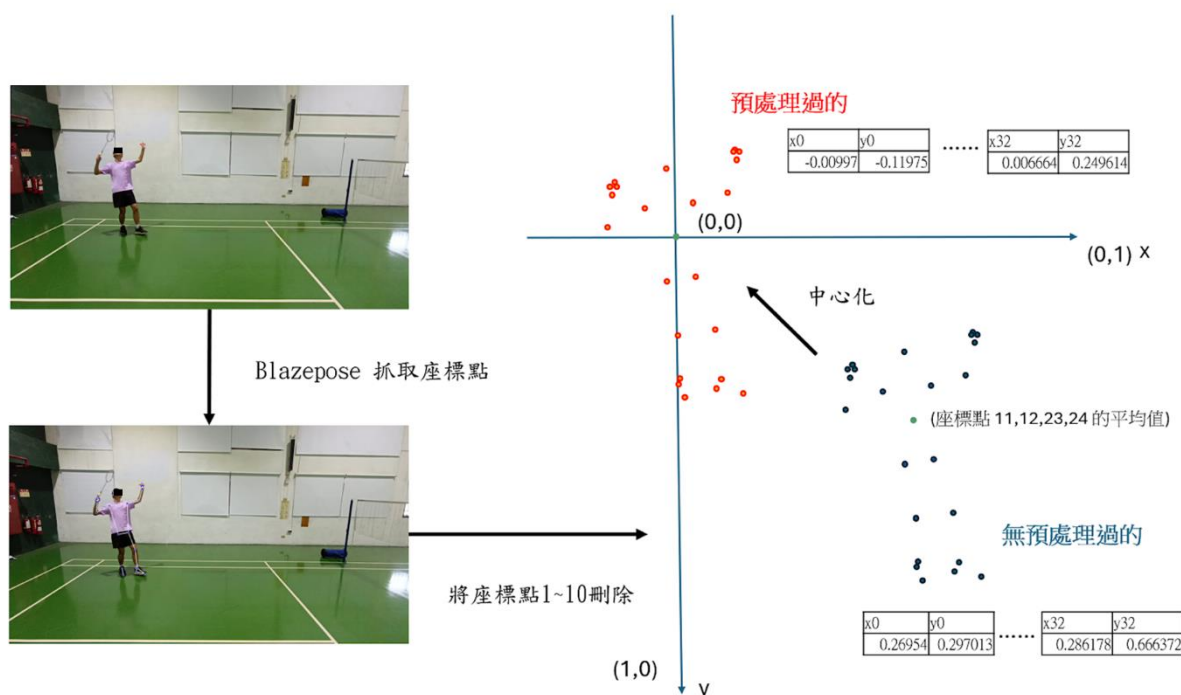


圖 3-7：人體骨架關鍵點序列的預處理示意圖 (來源自行製作)

## (三) 姿勢分類器

在訓練模型前將關鍵點序列與姿勢標籤分開，並使用 scikit-learn 套件的 `train_test_split()` 函式將數據集拆分為訓練集和測試集。其中，80% 的資料 (共 15,121 筆) 作為訓練集，剩餘的 20% (共 4,764 筆) 則作為測試集。本研究比較了多層感知器 (MLP)、K-近鄰演算法(KNN)以及 YOLO 等模型的建置方式，用於姿勢分類的評估，三種模型的說明如下。

### 1. 多層感知器(Multilayer Perceptron, MLP)

使用 Sequential 結構建置一個輸入層(Input Layer)、四個隱藏層(Hidden Layer)及一個輸出層(Output Layer)。輸入層接收 46 個特徵(features)，前三個隱藏層神經元數量分別為 128、128、64，激活函數設定為 ReLU，並應用 30% 的 Dropout 防止過擬合，並透過 Batch Normalization 穩定資料分佈加速收斂，第四個隱藏層，搭配 Batch Normalization、ReLU 激活函數。

輸出層的神經元數對應分類類別數，並採用 Softmax 激活函數，將輸出轉換為機率分佈。最後，採用 Adam 優化器和交叉熵損失函數(sparse categorical crossentropy)，並經過 100 個 epoch 的訓練。

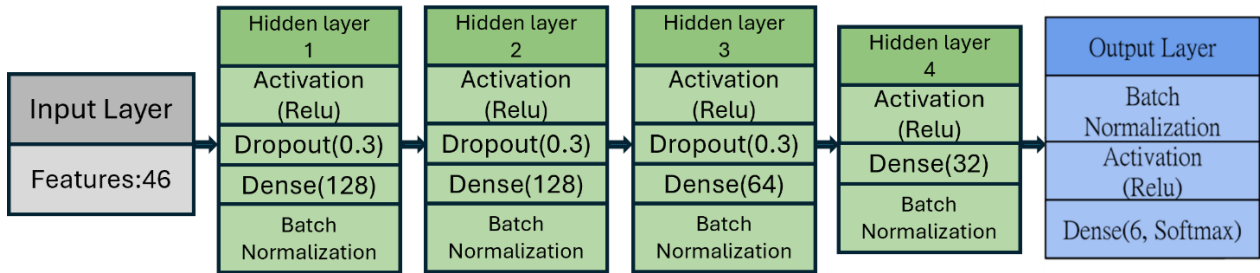


圖 3-8：MLP 模型架構 (來源自行製作)

## 2. K-近鄰演算法(K Nearest Neighbor, KNN)

本實驗使用的是 Scikit-learn 的 KNeighborsClassifier。k 值決定了在進行分類時，模型會參考 k 個最近的鄰居，本研究測試了不同的 k 值對模型表現的影響，研究者測試了從 1 到 50 不同的 k 值，並記錄每個 k 值下的訓練準確率和測試準確率。

## 3. You Only Look Once(YOLO)

YOLO 是一種物件分類及追蹤演算法，應用於多種領域，本研究想透過姿勢的靜態圖片進行分類，透過 BlazePose 與 OpenCV 套件繪製訓練資料的關節點及其連接，並為不同部位(如左右手、左右腿、身體)標記不同的顏色，接著再透過程式自動為 19,885 張圖片標記相應的 YOLO 格式、txt 文件，其中包含物件類別、物件邊界框的中心點  $x$  與  $y$  座標及物件邊界框的寬度與高度。物件邊界框的中心點座標計算方式是取出所有關鍵點序列中最大與最小的  $x$  與  $y$  座標，命其為  $x_{max}$ 、 $x_{min}$ 、 $y_{max}$ 、 $y_{min}$ ，再將其平均。如圖 3-9 所示為轉換公式。不同的姿勢類別定義如圖 3-10。

$$X = \frac{x_{max} + x_{min}}{2} \quad (1)$$

$$Y = \frac{y_{max} + y_{min}}{2} \quad (2)$$

$$W = x_{max} - x_{min} \quad (3)$$

$$H = y_{max} - y_{min} \quad (4)$$

圖 3-9：轉換公式 (來源自行製作)

1	names:
2	0: clear
3	1: drive
4	2: lift
5	3: net
6	4: other
7	5: serve

圖 3-10：姿勢類別編號 (來源自行製作)

如圖 3-11 展示關節點及其連接的 YOLO 格式標記資料轉換示意圖。

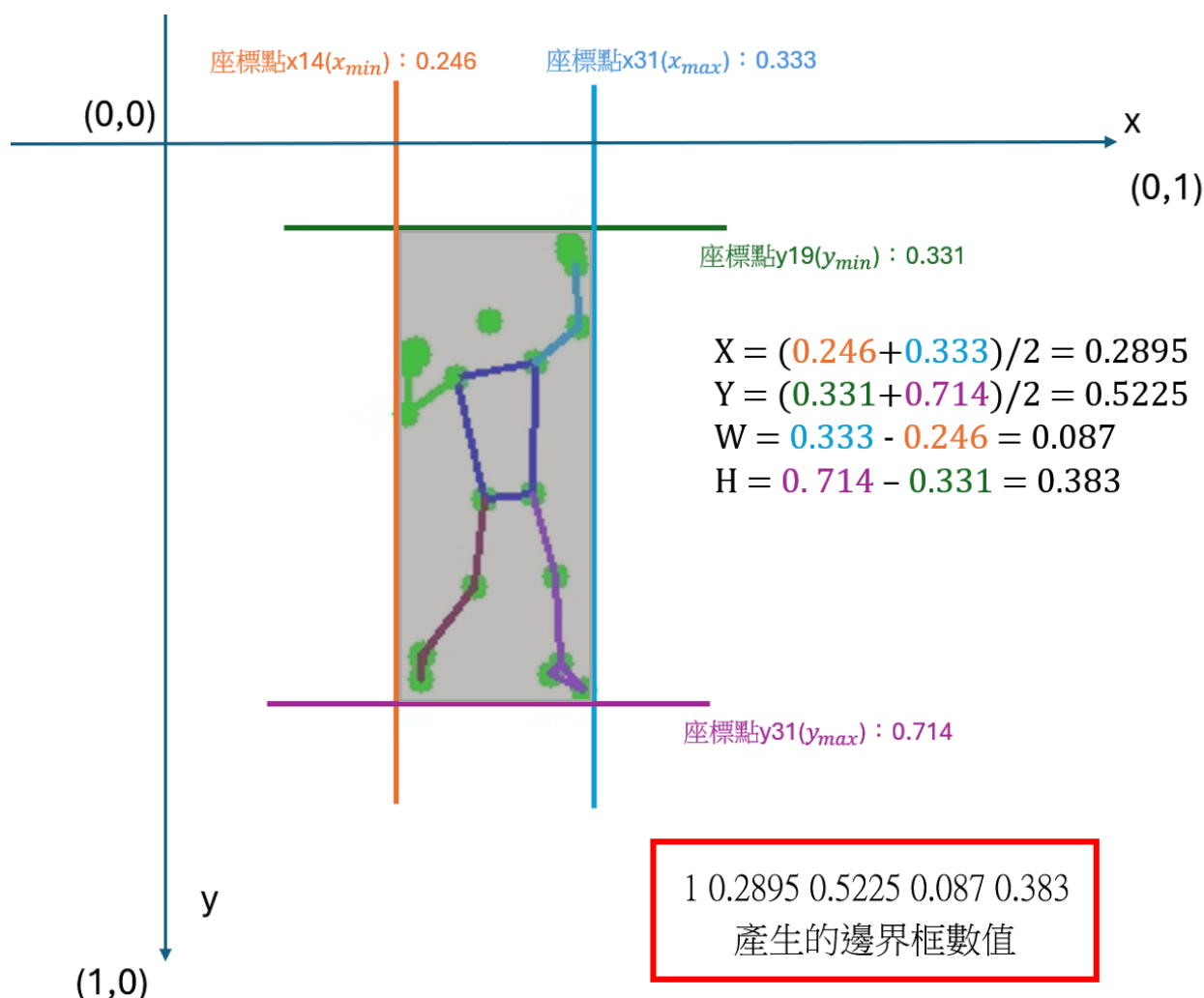


圖 3-11：YOLO標記資料自動轉換示意圖 (來源自行製作)

如圖 3-12 所示本研究使用的超參數(hyperparameters)為 100 個 epochs、img\_size 為 256，及 batch\_size 為 32。

```

5   # Resume training
6   model.train(
7       data='data.yaml', # Data Source
8       epochs=100,        # Epochs for Training
9       imgsz=256,         # Image Size for Training
10      batch=32,           # Batch Size for Training
11      device='0'          # Using GPU for Accelerating Training Process
12  )

```

圖 3-12：YOLO姿勢分類器訓練參數 (來源自行製作)



#### (四) 衡量指標

混淆矩陣用於評估分類模型性能的工具，通過比較實際值和預測值來展示結果，包含四個部分：

1. 真陽性(*True Positive*)：實際為陽性，且被正確預測為陽性的案例。
2. 偽陰性(*False Negative*)：實際為陽性，但被錯誤預測為陰性的案例。
3. 偽陽性(*False Positive*)：實際為陰性，卻被錯誤預測為陽性的案例。
4. 真陰性(*True Negative*)：實際為陰性，且被正確預測為非目標類別的案例。

表 3-3：混淆矩陣範例

	預測類別 1	預測類別 2	預測類別 3	預測類別 4	預測類別 5	預測類別 6
實際類別 1	<i>True Positive</i>	<i>False Negative</i>	<i>False Negative</i>	<i>False Negative</i>	<i>False Negative</i>	<i>False Negative</i>
實際類別 2	<i>False Positive</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>
實際類別 3	<i>False Positive</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>
實際類別 4	<i>False Positive</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>
實際類別 5	<i>False Positive</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>
實際類別 6	<i>False Positive</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>	<i>True Negative</i>

可以透過混淆矩陣的內容，來計算出衡量指標的前六項：

1. Accuracy：所有類別的 *True Positive* 除以總樣本數 (KNN、MLP、YOLO)
2. Precision：精確率衡量預測為陽性的樣本中實際陽性的比例(KNN、MLP、YOLO)

$$\text{Precision} : \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (5)$$

3. Recall：召回率衡量實際為陽性的樣本中被預測陽性的比例(KNN、MLP、YOLO)

$$\text{Recall} : \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (6)$$

4. F1-score：精確率和召回率的調和平均數 (KNN、MLP、YOLO)

$$\text{F1-score} : 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

5. mAP@0.5：評估YOLO模型的指標，表示  $\text{IoU} \geq 50\%$  時的平均精度 (YOLO)

6. 信心度：模型認為邊界框內含羽球的信心多寡，範圍為 0 至 1 (YOLO)

7. YOLO 損失函數：包含  $\text{cls\_loss}$ 、 $\text{dfl\_loss}$ 、 $\text{box\_loss}$  (YOLO)

8. FPS：模型平均完成單次推論完成所花時間的倒數 ( $s^{-1}$ ) (KNN、MLP、YOLO)



## 五、設計並實作 BIBO 方法應用於精確識別羽球動作分類的最佳時機

本研究提出 BIBO 方法可自動剪輯影片片段，以供羽球姿勢分類預測之用。如圖 3-13 所示，當羽球被偵測出現在鏡頭時，系統會開始收集影格，若鏡頭未偵測到羽球時，則開始計數若超過閾值 10 個影格，模型則視新手已回擊球作為結束。BIBO 方法最後還會在開始及結束前後額外增加 5 個影格確保捕捉足夠的影格數並據此剪輯影片。

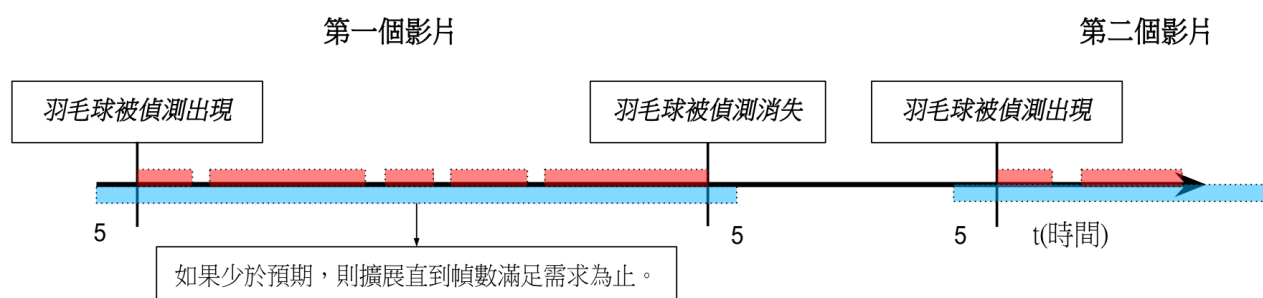


圖 3-13：BIBO 偵測剪輯示意圖 (來源自行製作)

本研究自行拍攝並標註羽球影像，用於訓練 YOLO 模型，最終運用訓練完的 YOLO 模型進行羽球的自動偵測與追蹤。

### (一) YOLO 羽球追蹤器訓練

追蹤羽球的 YOLO 模型將使用約 80% 的照片(4,789 張)來進行訓練，約 20% 的照片(1,173 張)來進行驗證。本研究使用兩種 YOLO 模型版本進行比較，YOLOv11n 和 YOLOv11s，並比較兩者性能表現。本研究使用的超參數為 100 個 epochs、img\_size 為 640，以及 batch\_size 為 32。

```
5 # Resume training
6 model.train(
7     data='data.yaml', # Data Source
8     epochs=100,        # Epochs for Training
9     imgsz=256,         # Image Size for Training
10    batch=32,           # Batch Size for Training
11    device='0'          # Using GPU for Accelerating Training Process
12 )
```

圖 3-14：訓練超參數 (來源自行製作)

### (二) TensorRT 優化

TensorRT 旨在透過量化(Quantization)等技術提升模型推論速度，例如將精度從 FP16 降至 INT8 以減少計算成本，一般用來優化和加速神經網路模型在 NVIDIA GPU 上的執行時間。本研究使用 TensorRT 將模型從 .pt 格式轉換為 .engine 格式，以提升推論速度。

### (三) 衡量指標

為驗證 BIBO 流程是否準確且可行，本研究將研究者自行錄製的羽球比賽影片提供給 BIBO 進行剪輯處理，該影片包含 73 個打擊片段(共 7,803 個影格)，最後人工逐一檢查所有 BIBO 剪輯的畫面，確認 BIBO 剪輯過的影片是否符合為一個完整羽球打擊動作，再計算其準確率評估是否可行，也利用以下衡量指標選擇最合適的模型。

1. Precision
2. Recall
3. F1-score
4. mAP@0.5
5. 信心度
6. YOLO損失函數
7. FPS

## 六、研究如何整合各 AI 模型的分析結果並給予新手完善的姿勢建議。

### (一) 影片姿勢判定、常見錯誤姿勢修正及專家姿勢比對

本研究實驗五種羽球姿勢類別，每類姿勢各測試 50 段 BIBO 剪輯後的影片，由於影片是由連續影格組成，在姿勢分類過程中，本研究測試三種姿勢分類模型對每一影格進行獨立分類，並透過多數決策確定影片最終的姿勢類別(如圖 3-15)。

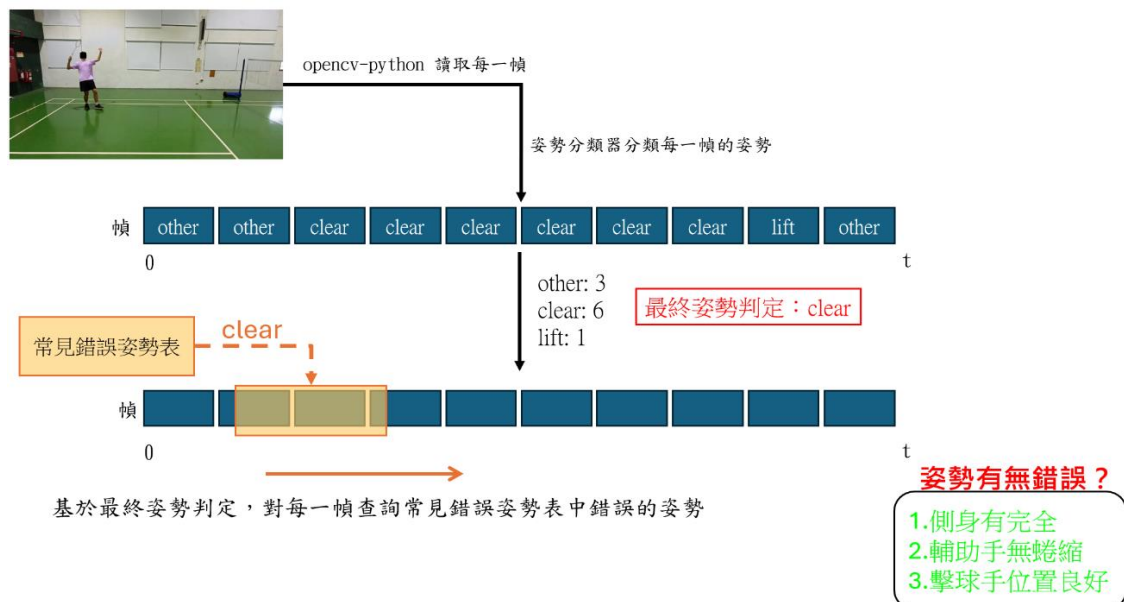


圖 3-15：影片姿勢判定 (來源自行製作)

在姿勢類別判定後，我們設計了一個檢查函式(如圖 3-16 所示)，依據表 1-1 檢查輸入影片中是否包含常見的錯誤姿勢，影片中將標示姿勢檢查結果，顯示是否存在錯誤。本研究透過關鍵點座標來檢查使用者是否做錯動作，例如，如果右膝過度伸展，

我們可以使用關鍵點 26 和 32 的座標來檢查使用者是否過度伸展膝蓋，請參考表 1-1 和圖 3-6。

```
291 if pose_name == "lift":
292     angle_24_26_28 = angle(lmk, 24, 26, 28)
293     if lmk[32].x < lmk[31].x:
294         P1_good=False
295     else:
296         P1_good = True
297
298     if lmk[16].x > lmk[24].x:
299         count += 1
300     else:
301         P2_good = True
302
303     if angle_24_26_28 > 90 and lmk[26].x < lmk[32].x:
304         count +=1
305     else:
306         P3_good = True
307
308
309     overlay_text(image, "P1: Correct stance" if P1_good else "P1: Adjust foot position", (50, 50))
310     overlay_text(image, "P2: Good posture" if not P2_good else "P2: Adjust right hand", (50, 100))
311     if P1_good:
312         overlay_text(image, "P3: Correct stance" , (50, 150))
313     else:
314         overlay_text(image, "P3: Please Change your feet first", (50, 150))
```

圖 3-16：偵測動作錯誤函式 (以挑球為例) (來源自行製作)

此外，本研究也對新手與專家的姿勢進行比對，首先記錄兩者的連續動作姿勢，接著繪製兩者的骨架圖，讓使用者直接觀察兩者的動作差異，進而識別偏差並改進。畫面中也會顯示使用者手部的揮拍軌跡，專家建議手部揮拍軌跡對新手的動作學習至關重要。

(二) LLM 整合

本研究受到論文：“Human-Centered LLM-Agent User Interface: A Position Paper” [11]的啟發，特別是其中的 FluteXGPT 示範，Chin 在該示範使使用者與大型語言模型互動，從而更快地提升學習效果。因此本實驗將利用大型語言模型進行姿勢建議的整合與重構，並透過 OpenAI 的 GPT-4 模型進行 API 呼叫與執行，最後提供新手個人化的姿勢矯正建議。表 3-4 是本研究 ChatGPT 主要 API 呼叫的設定，{errors} 是由常見錯誤姿勢偵測函式偵測出的名稱，{Pose\_Errors}是表 1-1 的項目內容，供 ChatGPT 讀取，{pose} 為姿勢分類器加上多數決機制所判定出的姿勢。

表 3-4：ChatGPT 提示詞設定

前置設定	你是一名羽球專業教練，您會根據使用者描述的姿勢和使用者的錯誤給出建議以及改善的方法，因此，您需要先研究羽球姿勢和常見錯誤，主要有五種姿勢：{Pose_errors} \n 請為使用者提供有用的繁體中文答案(2點條列式，第一行：此姿勢為何不好？第二行：此姿勢如何改善？)，姿勢改善建議中加入一些擊球練習，要有高清晰度、好的效果、高相關性、高可行性，不要有冗詞贅字。
指令(prompt)	{pose} --> {errors}。請提供具體的調整建議。

### (三) 衡量指標

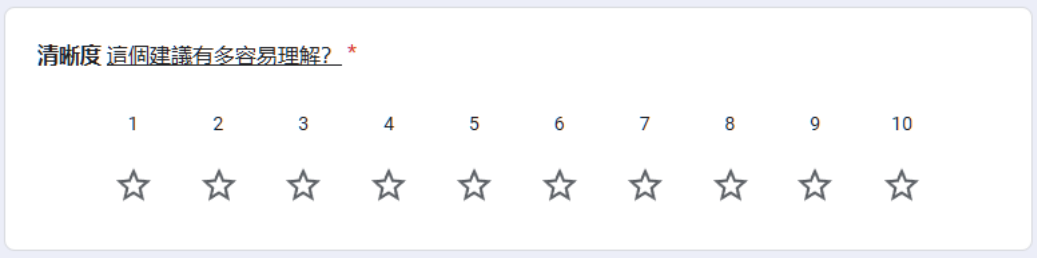
為評估 ChatGPT 提供的姿勢建議是否合適，本研究利用 Google Forms 設計了一套評分標準，包含清晰度、效果、相關性及可行性四個面向。測試內容包含 5 類新手學習的羽球姿勢，收集 10 位羽球專家(1 位羽球教練、2 位體育老師、7 位羽球校隊球員)對 ChatGPT 在姿勢矯正建議上的效果與相關性。針對初學者而言，研究者邀請 50 位羽球新手，請他們透過 Google Forms 評估 ChatGPT 姿勢建議之清晰度與可行性。

表 3-5：教練對 ChatGPT 提供的姿勢建議的認同百分比評分標準

衡量指標	配分	問題敘述
效果	10	此建議對錯誤姿勢問題解決有效？
相關性	10	此建議與當前錯誤姿勢問題直接相關？
合計	20	

表 3-6：新手對 ChatGPT 提供的姿勢建議的滿意程度評分標準

衡量指標	配分	問題敘述
清晰度	10	此建議對於新手多容易理解？
可行性	10	此建議是否可行，對新手是否窒礙難行？
合計	20	



清晰度 這個建議有多容易理解? \*

1 2 3 4 5 6 7 8 9 10

☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆

圖 3-17：表單評分範例 (來源自行製作)

## 七、系統流程及伺服器架設

本研究希望用戶能夠使用手機接收專業建議，並非將系統完全運行在手機上，本研究啟發於 Clutch: AI for Racket Sports 的方法，手機將作為發送影片和接受建議端。本研究利用 flask 來架設一個輕量化伺服器，使用者上傳影片後，在運行整個系統後(見圖 1-1)，系統將向用戶提供 BIBO 流程剪輯完的單一打擊片段、姿勢錯誤標示、姿勢糾正建議以及專家姿勢比對。此外也提供 BIBO 閾值調整的選項(見圖 3-13)，使系統剪輯出更準確的單一打擊片段。

## 肆、研究結果及討論

### 一、探討人體姿態辨識模型及機器學習技術應用在羽球姿勢分類

研究結果顯示 MLP 搭配 Dropout 可以大幅提升模型的準確率，避免 MLP 過度擬合，使用 KNN 則能夠透過增加大量數據來幫助模型辨識姿勢，增加模型的準確率，YOLO 能夠理解姿勢的空間特徵，使姿勢判斷更準確。本實驗利用 15,908 筆關鍵點序列用於訓練，3,977 筆用於測試，三種模型訓練結果圖表資料如下。

#### (一) MLP 分類報告 (Accuracy = 99.52%)

表 4-1：MLP模型訓練完結果 – 混淆矩陣 (來源自行製作)

	精確率	召回率	F1-Score	樣本數
長球	0.9940	0.9955	0.9947	663
平球	0.9972	0.9931	0.9952	728
挑球	0.9935	0.9923	0.9929	775
放網	0.9979	0.9990	0.9984	958
其他	0.9909	0.9927	0.9918	551
發球	0.9967	1.0000	0.9983	302
宏觀平均	0.9951	0.9954	0.9952	3,977
加權平均	0.9952	0.9952	0.9952	3,977

True Labels \ Predicted Labels	clear	drive	lift	net	other	serve
clear	660	1	1	0	1	0
drive	4	723	1	0	0	0
lift	0	1	769	0	4	1
net	0	0	1	957	0	0
other	0	0	2	2	547	0
serve	0	0	0	0	0	302

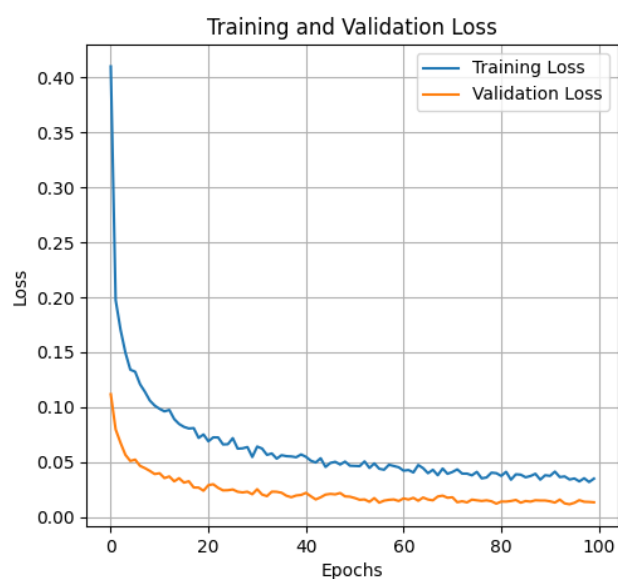
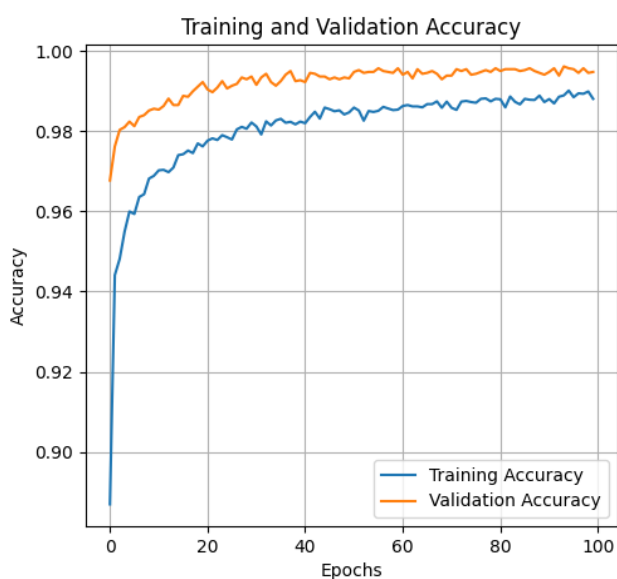


圖 4-1：MLP模型訓練完結果 – 訓練與驗證圖表 (來源自行製作)

(二) KNN 分類報告 (Accuracy = 99.67%)

表 4-2：KNN 模型訓練完結果 - 混淆矩陣 (來源自行製作)

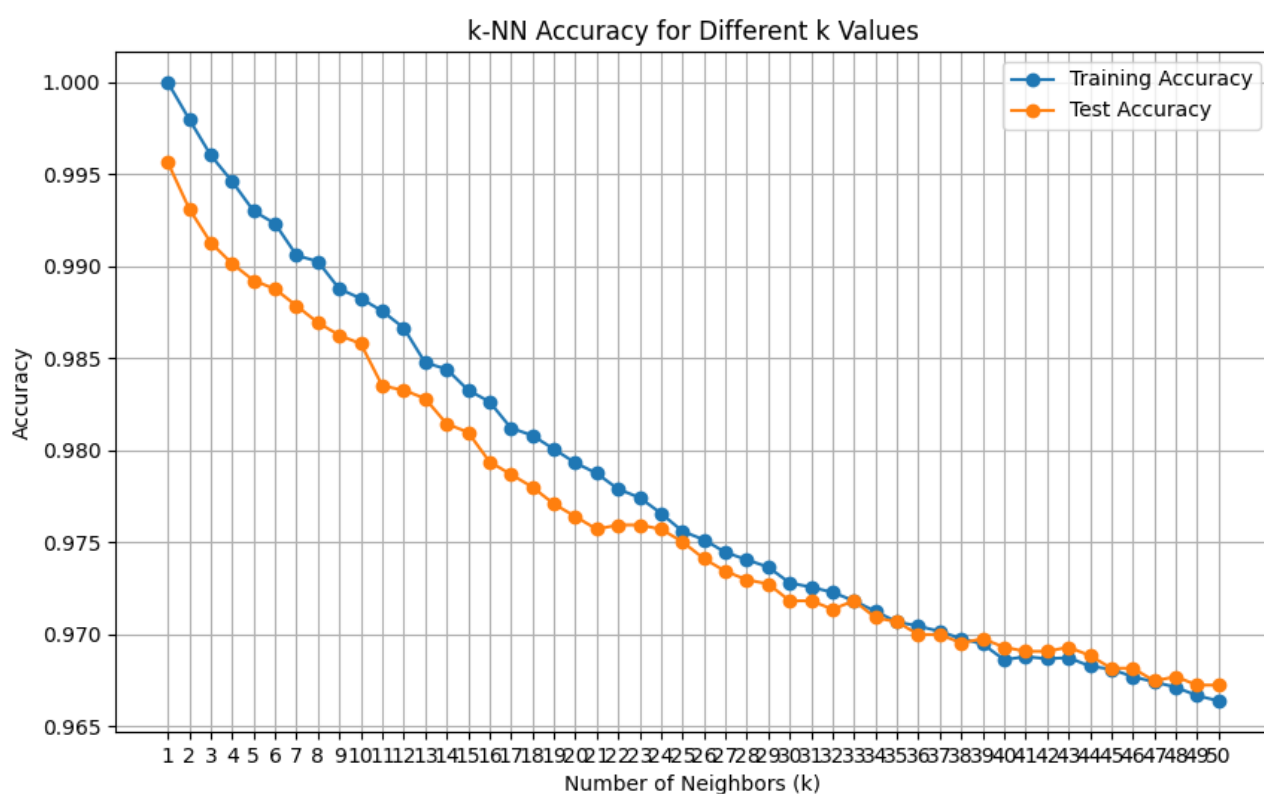
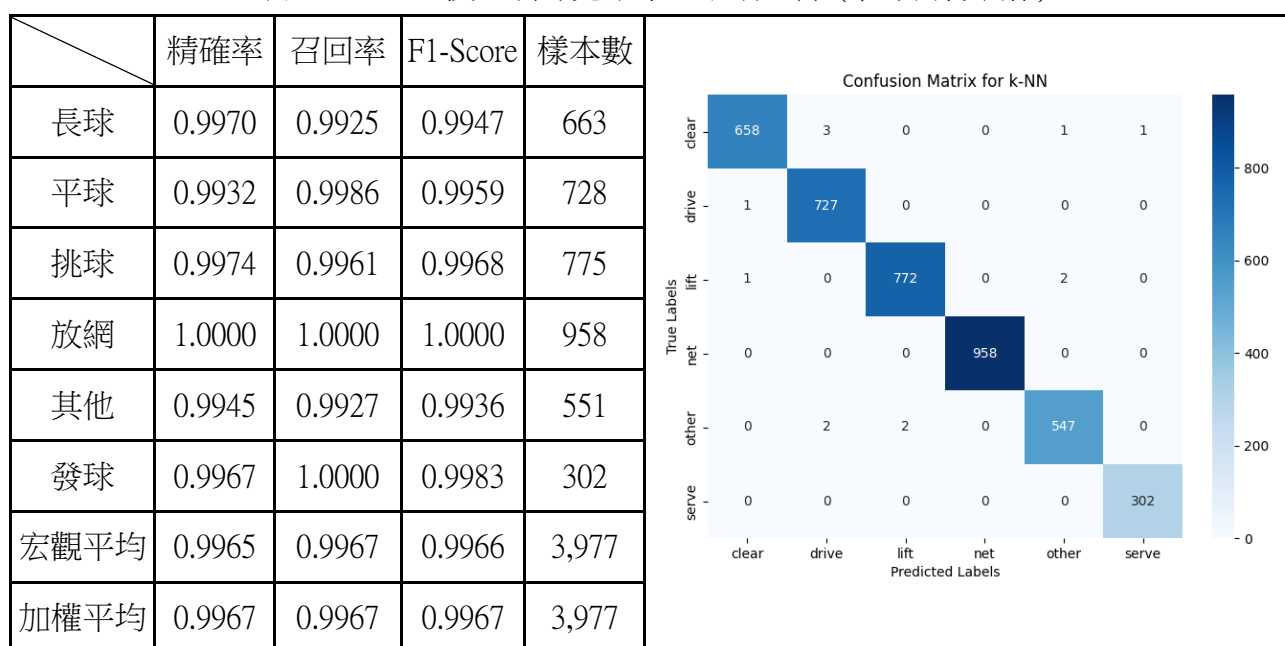
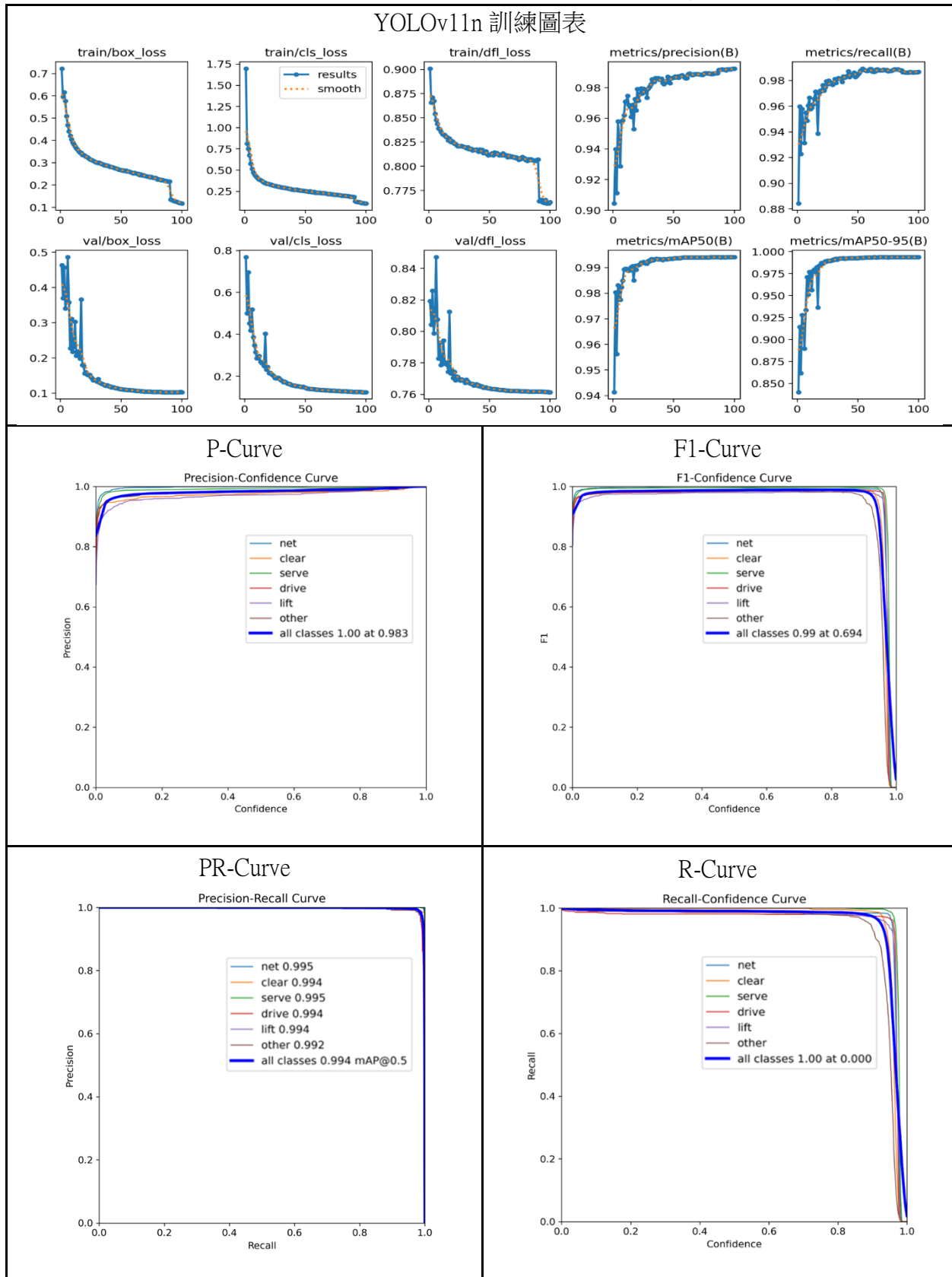


圖 4-2：KNN 模型訓練完結果 - 不同 k 值和其訓練結果變化圖 (來源自行製作)

(三) YOLO 分類報告 (Accuracy = 99.60%)

表 4-3：YOLOv11n 模型訓練完結果 - YOLO 損失函數 (來源自行製作)





Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	3977	3977	0.997	0.996	0.995	0.995
clear	704	704	0.995	0.996	0.995	0.995
drive	693	693	0.992	0.994	0.995	0.995
lift	765	765	0.999	0.994	0.995	0.995
net	952	952	1	0.999	0.995	0.995
other	545	545	0.998	0.994	0.995	0.995
serve	318	318	0.999	1	0.995	0.995

圖 4-3：YOLO 模型訓練結果 – Box Precision、Box Recall、mAP50、mAP50-95 (來源自行製作)

表 4-4：YOLO 模型訓練完結果 – 混淆矩陣 (來源自行製作)

	精確率	召回率	F1-Score	樣本數	True Class	Confusion Matrix for YOLOv11n						Predicted Class
						clear	drive	lift	net	other	serve	
長球	0.9943	0.9943	0.9943	704	clear	700	4	0	0	0	0	
平球	0.9899	0.9942	0.9920	693	drive	3	689	0	0	1	0	
挑球	0.9987	0.9948	0.9967	765	lift	1	1	761	0	2	0	
放網	1.0000	0.9989	0.9995	952	net	0	1	0	951	0	0	
其他	0.9945	0.9945	0.9954	545	other	0	1	1	0	542	1	
發球	0.9969	1.0000	0.9984	318	serve	0	0	0	0	0	318	
宏觀平均	0.9945	0.9945	0.9945	3,977								
加權平均	0.9960	0.9960	0.9960	3,977								

綜合上述資料，KNN 模型的準確率為 99.67%，這良好的表現可能與本研究提出的數據前處理方法(人體姿勢關鍵點序列中心化)密切相關。此外，本研究大量蒐集訓練資料，有助於 KNN 模型進行準確分類。從圖 4-2 中可以看出，訓練和驗證準確度差異不大，顯示模型並未發生過擬合現象。KNN 模型在  $k = 1$  時，模型的表現最佳，然而， $k = 1$  的 KNN 模型意味著模型自動找到最近的一筆關鍵點序列，這可能會失去其泛化能力。KNN 模型易混淆的類別為平球與長球，其中長球有 3 次被誤判為平球。

YOLO 模型準確率表現為 99.60%，位居第二，YOLO 在信心度 0.983 的情況下，所有類別的精確率均達到 100% (詳見表 4-3)。經過 1.58 小時的訓練後，YOLO 模型在多個姿勢類別上的表現均優異，mAP@0.5 超過 0.99。YOLO 模型易混淆的類別是發球與平球，其中發球 4 次被誤判為平球。

MLP 模型的分類表現排名第三，圖4-1 中驗證損失(val loss)並未隨著訓練時間增加，代表 MLP 模型未出現過擬合現象，在 MLP 模型中，挑球與「其他」及挑球與平球最易混淆，各發生 4 次錯誤分類。

在使用姿態分類進行單張影像推論後，研究者將此技術應用於影片分類上，並加入了多數決投票的處理過程。本研究實測五種羽球姿勢，每種類別各 50 段 BIBO 剪輯過的單一打擊片段，分類實測結果如下。

表 4-5：模型準確率評估 (使用 250 部影片進行測試)

	MLP	KNN (k=1)	YOLO
長球	50/50(100.00%)	50/50(100.00%)	49/50(98.00%)
平球	43/50(86.00%)	49/50(100.00%)	46/50(92.00%)
挑球	45/50(90.00%)	45/50(90.00%)	43/50(86.00%)
放網	41/50(82.00%)	43/50(86.00%)	39/50(78.00%)
發球	50/50(100.00%)	50/50(100.00%)	46/50(92.00%)
總共	229/250(91.60%)	<b>237/250(94.80%)</b>	223/250(89.20%)

表 4-6：模型平均推論速度 (FPS)

模型名稱	MLP	KNN	YOLO
運行速度	$11.26\text{ s}^{-1}$	<b><math>25.48\text{ s}^{-1}</math></b>	$7.55\text{ s}^{-1}$

根據表 4-5，KNN 具有最佳的準確率，而 YOLO 的準確率最低。考量FPS，從表 4-6 可看出 KNN 表現最佳，本研究不選擇 MLP 和 YOLO 的原因在於：YOLO 需先利用 OpenCV 套件額外繪製圖像會大幅增加計算時間，而 MLP 也需要進行大量矩陣運算，其運算時間也遠高於 KNN。三個姿勢分類器模型都有高準確率的原因可能是研究者最初訓練這些模型前，在資料處理過程中，每筆資料都經過仔細的審核和檢查錯誤地方。

針對已發生的這些錯誤，研究者整理了在日常使用中容易出錯的情況，其中有些動作姿勢很相似，從圖 4-4 可以看出，在測試時，平球姿勢與長球姿勢或發球姿勢相似。

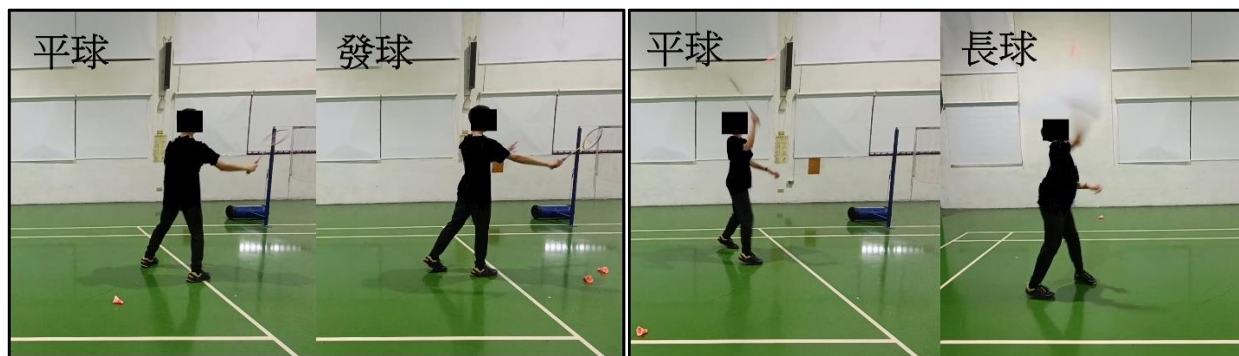


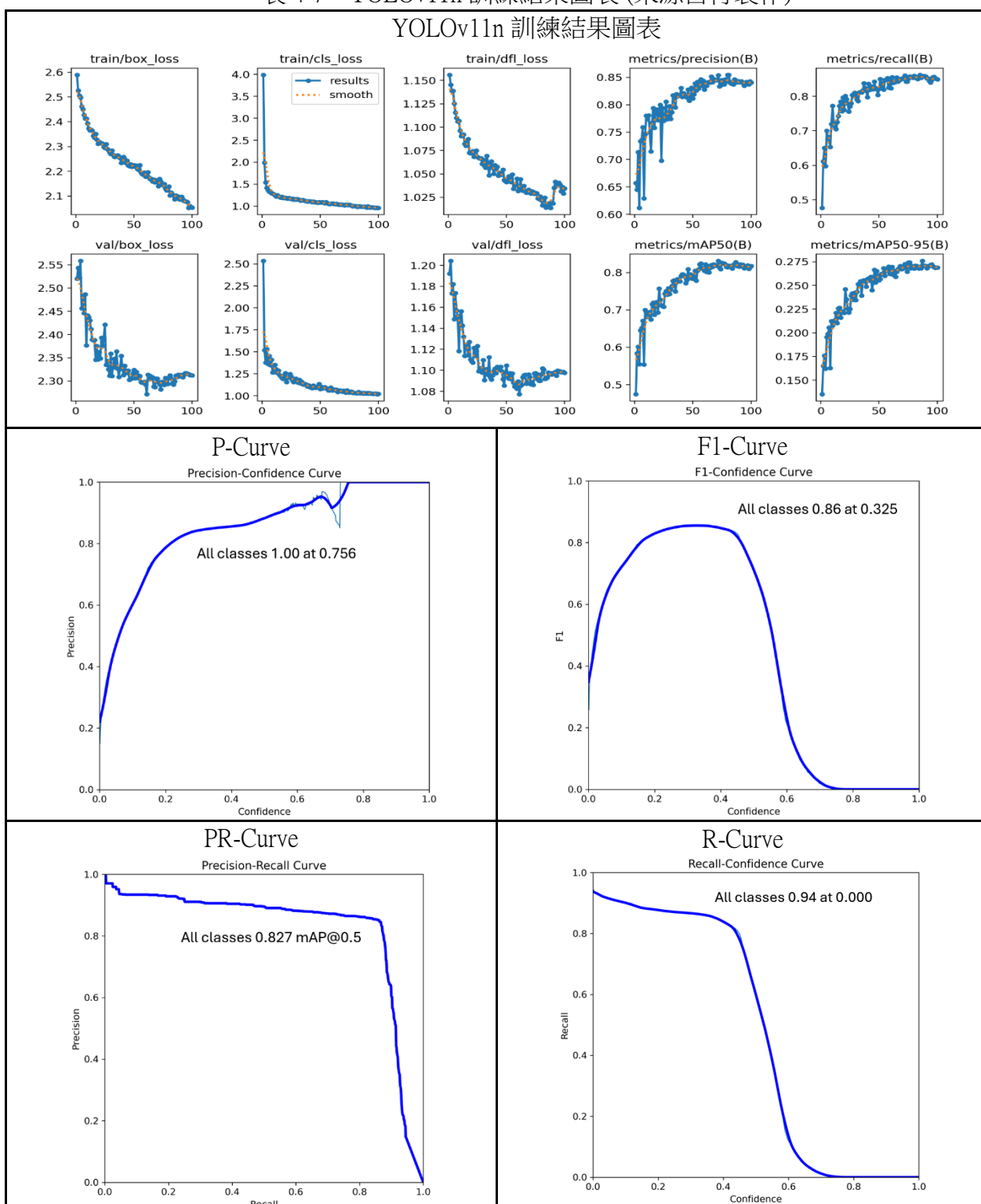
圖 4-4：姿勢相似地方 (以平球為例) (來源自行製作)

## 二、設計並實作BIBO方法應用於精確識別羽球動作分類的最佳時機

研究結果顯示，BIBO 方法能有效剪輯影片，且經過 TensorRT 量化後，效能有所提升。本研究使用約 80% 的照片(4,789 張)來進行訓練，約 20% 的照片(1,173 張)來進行驗證。兩種 YOLO 模型版本訓練結果圖表資料如下。

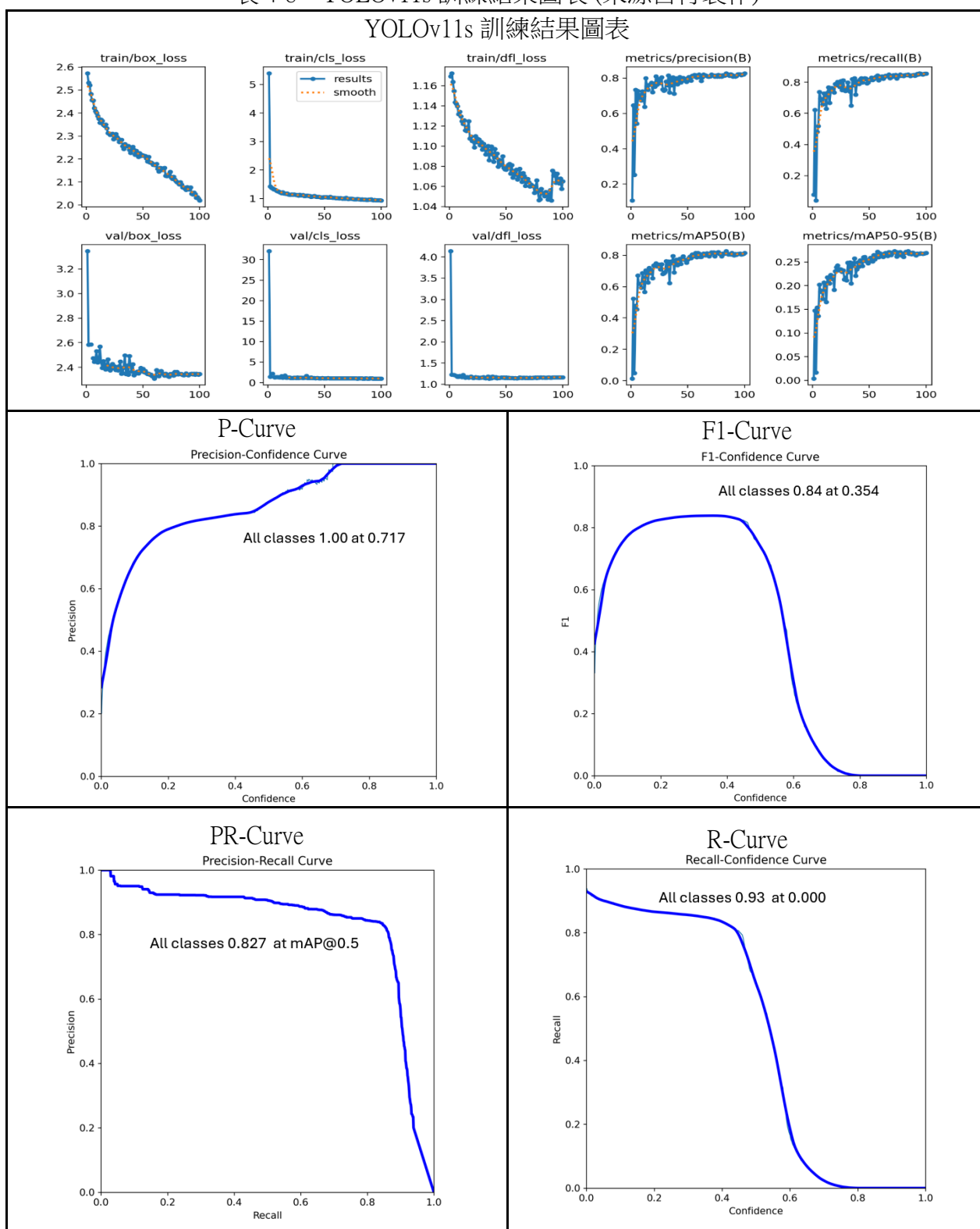
### (一) YOLOv11n (mAP@0.5 = 0.827)

表 4-7：YOLOv11n 訓練結果圖表 (來源自行製作)



(二) YOLOv11s ( $mAP@0.5 = 0.827$ )

表 4-8：YOLOv11s 訓練結果圖表 (來源自行製作)



在訓練完這兩個模型後，可以明顯看出兩個模型都沒有出現過度擬合的現象(見表 4-7、表 4-8)。兩個模型具有相同的  $mAP@0.5$  值，因此可以說這兩個模型差異並不大，僅在精確率曲線(P-Curve)、召回率曲線(R-Curve)以及 F1 曲線上存在少許差異。

為了選出合適的模型，本研究使用研究者自行錄製影片(73 個打擊片段，共 7,803 個影格)來測試 BIBO 流程，從表 4-7、表 4-8 的 F1 曲線可以看出，本研究對於兩類模型使用的最佳信心值分別為 0.325 和 0.354，以獲得精確度和召回率之間的最佳平衡。

表 4-9：YOLO 模型在 TensorRT 量化前後的準確率比較

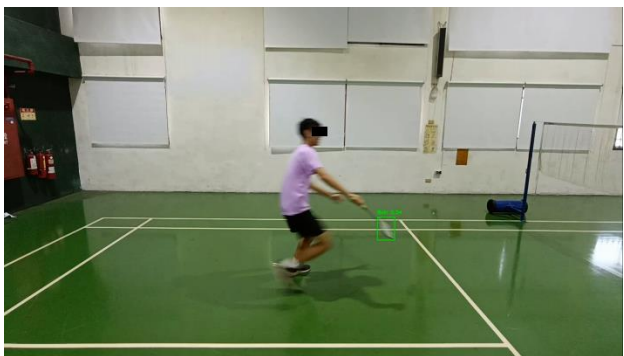
模型名稱	有 TensorRT 量化	無 TensorRT 量化
YOLO v11n	<b>70 out of 73 videos (95.89%)</b>	70 out of 74 videos (94.59%)
YOLO v11s	70 out of 74 videos (94.59%)	66 out of 73 videos (90.41%)

表 4-10：YOLO 模型在 TensorRT 量化前後的 FPS 比較

模型名稱	有 TensorRT 量化	無 TensorRT 量化
YOLO v11n	<b><math>69.32\text{ s}^{-1}</math></b>	$61.40\text{ s}^{-1}$
YOLO v11s	$65.57\text{ s}^{-1}$	$62.95\text{ s}^{-1}$

從表 4-9、表 4-10 可見，四種模型搭配皆能準確偵測羽球位置。其中，YOLOv11n 在 TensorRT 量化優化後，推論速度達 69.32 的 FPS，準確率達 95.89%，69.32 的 FPS 代表 BIBO 方法可以幾乎即時處理影片。儘管模型已具有高準確率，使用 BIBO 追蹤羽球時仍可能發生以下兩類錯誤(如表 4-11)。

表 4-11：BIBO 剪輯影片的問題 (來源自行製作)

問題	敘述	圖例
問題一	當對手挑球位置太近，球雖出現在影片中但持續上升，超出手機鏡頭的捕捉範圍，導致球沒有被偵測到的影格數已超過閾值(10 個影格)，使 BIBO 流程視為使用者已打擊完畢，導致剪輯錯誤。	
問題二	當球拍移動太快時，羽球拍的線材和羽球在高速運動的形狀非常相似，造成羽球偵測誤判。	



### 三、研究如何整合各 AI 模型的分析結果並給予新手完善的姿勢建議

下圖 4-5 是使用者與專家姿勢比對整合範例(以長球為例)，綠色骨架代表羽球專家的姿勢骨架，而白色骨架則是使用者的姿勢骨架，黃色線條為羽球專家的手部揮拍軌跡，而藍色線條則是使用者手部揮拍軌跡，畫面中可直接觀察兩者在姿勢與手部揮拍軌跡差異，有助於調整姿勢。此圖左上角也會顯示使用者目前姿勢的錯誤項目。紅色骨架線條也會輔助提醒使用者姿勢錯誤之處。

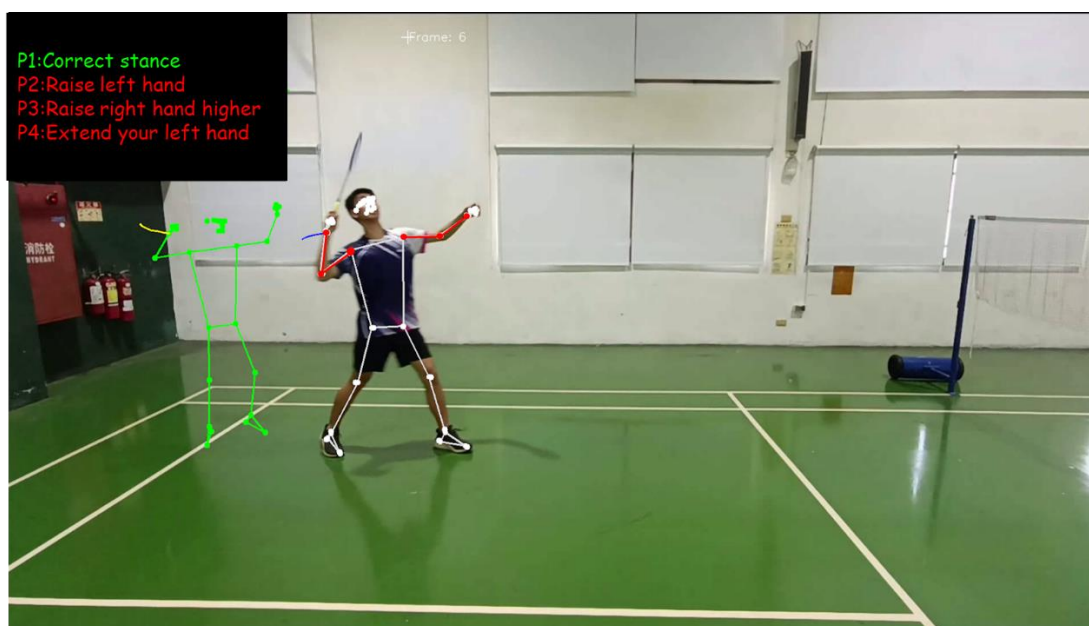


圖 4-5：修正整合範例 (以長球為例) (來源自行製作)

為了方便使用者可以使用我們的系統，研究者設計了一個使用者介面使新手可以透過手機來獲取姿勢建議，輸出的影片需要放慢，因此使用 MoviePy 將輸出的影片放慢 2 倍。



圖 4-6：使用者上傳影片的使用者介面 (手機畫面) (來源自行製作)



圖 4-7：系統產生結果 (手機畫面) (來源自行製作)

表 4-12 ChatGPT 提供的姿勢建議範例 (以長球為例)

長球姿勢改善建議	<p><b>此姿勢為何不好？</b></p> <ol style="list-style-type: none"> <li>1. 輔助手蜷縮會使你無法有效運用身體力量，導致擊球時可能力量不足。</li> <li>2. 擊球手位置過低會導致你無法在最高點擊球，影響長球的傳遞距離與落點控制。</li> </ol> <p><b>此姿勢如何改善？</b></p> <ol style="list-style-type: none"> <li>1. 要改善輔助手蜷縮的問題，你可以進行"身體旋轉練習"。緩慢地前後轉動身體，並注意身體始終朝向前方，藉此增加身體轉動幅度。此外，也可以嘗試"擴胸練習"，透過展開胸部和手臂來調整輔助手臂的姿勢。</li> <li>2. 如果擊球手位置過低，你可以進行"高點擊球練習"。請朋友或教練用球拍向高空擊球，您嘗試在高點擊球，並習慣這種擊球高度。同時，請理解擊球的力量來源並非只有手腕，而是源自全身力量的傳遞，需依序從腳底、腿部、腰部、背部、手臂、手腕傳遞出來。如此才能有效提升長球的距離和準確性。</li> </ol>
----------	---

為評估 ChatGPT 提供的姿勢建議是否有助於新手，本研究邀請 10 位羽球專家依據本研究設計的評分標準進行認同評分，表 4-13 為羽球專家認同百分比統計資料。

表 4-13：羽球專家對 ChatGPT 給新手姿勢建議的認同百分比

	平均認同分數	平均認同百分比	標準差	Cronbach's $\alpha$ 值
效果	8.120	81.20%	1.081	0.876
相關性	8.360	83.60%	0.933	0.760
總分	16.480	82.40%		



為評估 ChatGPT 提供的姿勢建議是否能讓新手了解且不會窒礙難行，本研究邀請 50 位羽球新手依據本研究設計的評分標準進行評分，表 4-14 為羽球新手認同百分比統計資料。

表 4-14：羽球新手對於 ChatGPT 給新手姿勢建議的平均認同百分比

	平均認同分數	平均認同百分比	標準差	Cronbach's $\alpha$ 值
清晰度	8.152	81.52%	1.700	0.921
可行性	8.080	80.80%	1.753	0.911
總分	16.232	81.16%		

由表 4-13、表 4-14 可見，羽球專家對 ChatGPT 整合的姿勢建議給予平均認同百分比達 82.40%，顯示其具備良好的可信度與專業性，在效果方面，羽球專家認為 ChatGPT 並無大多創新，但對於使用者解決問題是有效的。羽球新手認為 ChatGPT 給出的建議清晰度容易理解，且執行並不會窒礙難行，也給予平均 81.16% 的認同度。四項指標的 Cronbach's  $\alpha$  值均大於 0.75，顯示具有不錯的內部一致性。

## 伍、結論

- 一、本研究三種姿勢分類器中，KNN 的表現最優異，影片的姿勢分類準確率達 **94.80%**。
- 二、本研究對於 KNN 姿勢分類器模型的**最佳 k 值為 1 表現最佳**，此一結果應與資料預處理時已剔除姿勢標註錯誤及偵測異常的樣本有關，這也可能代表模型泛化能力較受限。
- 三、本研究設計的 BIBO 方法中，使用 TensorRT 量化後的 YOLOv11n 速度最快 (**69.32 FPS**)，亦具備最高的準確率 (**95.89%**)。
- 四、本研究 YOLO 羽球追蹤器經 TensorRT 量化之後可以**稍微提升推論速度及準確率**。
- 五、本研究 ChatGPT 整合之姿勢建議效果良好，專家平均認同百分比為 **82.40%**，羽球新手平均認同百分比為 **81.16%**，Cronbach's  $\alpha$  值均大於 **0.75**，具有不錯的內部一致性。

## 陸、未來可改進項目

- 一、未來可結合身體動作或重力加速度計等感測器，收集電腦視覺外重要的特徵參數，例如打球過程中的揮拍速度、出手時機及重心轉移等，整合更多模態的感測資料，將有助於描繪打羽球的動作特徵，進而提升動作分析與指導的正確性。
- 二、未來可結合大語言模型的微調技術(fine-tuning)、提示工程策略(Prompt Engineering)及檢索增強生成(RAG)機制，提升給予使用者建議的品質。
- 三、本研究所建置的羽球動作姿勢分類模型表現良好，顯示其具備有效的姿勢分類表現。未來可考慮羽球動作的連續性與時序關聯性，可結合具處理時間序列資料的遞迴神經網路(RNN)，如長短期記憶網路(LSTM)，增進模型對連續動作的理解與辨識能力。

四、未來可結合可立即接收回饋的裝置(如智慧手錶)，提升不良姿勢的察覺及修正效率，提高學習品質。

## 柒、參考文獻資料

- [1] VICTOR Taiwan. (n.d.). 周教練教室 [Playlist]. YouTube. <https://www.youtube.com/watch?v=dPPu uBabKAc&list=PL4F1FCAC06A7ACE92>
- [2] BadmintonInsight. (n.d.). Shot tutorials [Playlist]. YouTube. <https://www.youtube.com/playlist?list=P L6bpt8HeVN 95VIc- uRr iUPS-LG2a4n>
- [3] Dollentas, N. (2023, October 15). *Clutch AI Badminton App review: Great idea, but does it deliver? (No)* [Video]. YouTube. <https://www.youtube.com/watch?v=kfilVU56Jzw>
- [4] Huang, Y.-C., Huang, C.-H., Hsu, M.-H., & Li, K.-C. (2019). TrackNet: A deep learning network for tracking high-speed and tiny objects in sports applications. *arXiv preprint arXiv:1907.03698*. <https://arxiv.org/abs/1907.03698>
- [5] 王文義、張凱聲、陳廷發、王正昌、彭文忠、易子文 (2020)。Badminton Coach AI：基於深度學習的羽球比賽數據分析平台。《體育學報》，53(2)，201 – 213。 <https://www.airitilibrary.com/Article/Detail/10247297-202006-202007060015-202007060015-201-213>
- [6] 科技大觀園. (2021, October 18). AI 讓羽球訓練更聰明！智慧球拍上手，揮拍「爆發力」一測就知道. *PanSci 泛科學*. <https://pansci.asia/archives/332433>
- [7] Shiela, M. C. (2022). *Badminton strokes classification based on human pose estimation and machine learning* (Master's thesis, National Taiwan University of Science and Technology). <https://etheses.lib.ntust.edu.tw/thesis/detail/63ccf6a6d4ad0399ee9a48cf8311984a>
- [8] Tukino, T., Pratiwi, M., & Defit, S. (2024). Deep learning based technical classification of badminton pose with convolutional neural networks. *ILKOM Jurnal Ilmiah*, 16(1), 76 – 86. <https://doi.org/10.33096/ilkom.v16i1.1951.76-86>
- [9] Toshniwal, D., Patil, A., & Vachhani, N. (2024). AI coach for badminton. *arXiv preprint arXiv:2403.08956*. <https://arxiv.org/abs/2403.08956>
- [10] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhang, F., Zhu, T., Grundmann, M., & Flynn, J. (2020). BlazePose: On-device real-time body pose tracking. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 1004 – 1011. IEEE. <https://arxiv.org/abs/2006.10204>
- [11] Chin, D., Wang, Y., & Xia, G. (2024). Human-Centered LLM-Agent User Interface: A Position Paper. *arXiv:2405.13050v2 [cs.HC]*. <https://www.arxiv.org/pdf/2405.13050>