

113學年度臺中市立文華高中  
科學展覽作品說明書

科 別：電腦與資訊學科

班級：219

指導老師簽名：

作品名稱：基於深度學習的羽毛球運動追蹤與姿態矯正研究

作者：21918張子祥

21925楊謹謙

## 摘要

隨著我國在羽毛球奧運賽事中取得勝利，這項快速且激烈的運動在全球的普及程度顯著提高，也吸引了越來越多的人參與其中。為了提升業餘選手的訓練效率，我們提出了一種創新的方法，開發一款能夠協助業餘玩家進行姿態矯正的軟體。我們成功選擇了最適合的人體關鍵點提取模型，分類器並已經能夠分類羽毛球中的四種基本動作，且有清晰的混淆矩陣。此外，我們訓練的 YOLO 模型在羽毛球軌跡追蹤上有低像素誤差的表現。該項目未來並計劃加入更多功能，以進一步完善羽毛球訓練的技術輔助。

## 壹、研究動機

羽毛球是一項遍及全世界知名度很高的運動。隨著羽毛球運動的普及，越來越多的人參與這項運動。然而，羽毛球是一項複雜且技術性很強的運動。要提高技術或進行練習，常見有四種主要的方法來學習：

實戰是許多羽毛球愛好者提升技巧的首選方式，通過與對手交鋒可以增加跑位的概念，理解打法策略。然而，這種方式的缺點在於缺乏動作矯正機制，錯誤的姿勢可能被反覆使用並固化，導致受傷風險增加或技術瓶頸難以突破。

聘請教練能夠糾正動作錯誤並給予專業建議，然而，聘請教練的成本較高，並且需要配合教練的時間安排，缺乏靈活性。此外，對於初學者來說，持續聘請教練可能在經濟上負擔較重。

YouTube等平台上的教學影片是許多羽球愛好者自學的主要來源，影片內容涵蓋從基本動作到高級技術的各種主題。然而，這些影片往往無法針對學習者的個人需求進行細緻的指導，尤其在細節動作或錯誤糾正方面有所不足。

目前，AI教練軟體尚未普及，而且大多數這類應用程式對羽球新手並不適用。例如，Clutch: AI for Racket Sports在iOS系統上可用於分析手持拍子的相關運動，該程式會幫忙定位羽球場地，並記錄一般羽球運動表現、剪輯精彩時刻(Clutch: AI for Racket Sports，取自：<https://www.clutchapp.io>)。動智AI羽球體驗中心是在桃園的一座羽球館，主要用於測試球速(動智AI羽球體驗中心，取自：<https://www.sportxai.com>)。成大和逢甲大學共同研發的智能球拍幫助球員們測量球速(智能球拍，取自：<https://pansci.asia/archives/332433>)，尚未能全方位輔助動作矯正。

綜合以上所述，目前尚未有AI教練能夠給予羽毛球新手球員專業的姿勢建議及球路軌跡分析或相關球場表現紀錄資訊。因此，我們希望利用不同類神經網路架構來打造一個姿勢矯正的軟體，並嘗試透過物件偵測與追蹤模型進行羽毛球軌跡分析，驗證羽球表現。隨著越來越多人學習這項迷人的運動，我們希望製作一個能夠在提供專業教練建議的同時，兼顧靈活性的AI教練。

## 貳、研究目的

本研究旨在發展一套利用不同類神經網路架構，結合多部視訊相機拍攝紀錄羽毛球新手在球場的動作姿勢表現，並協助調整姿勢的套件，最後再透過物件偵測與追蹤模型進行羽毛球軌跡分析驗證球員表現。研究目的可分為以下列點：

- 一、研究及實驗不同人體姿態辨識模型
- 二、研究及實驗2D人體姿態辨識姿勢並且嘗試分類
- 三、利用電腦視覺追蹤球的路徑

## 參、研究設備與器材

### 一、研究硬體

#### (一)、電腦

- CPU：Intel® Core™ 5 處理器 120U @5GHz
- GPU：Nvidia GeForce RTX 2050 4GB

#### (二)、羽毛球

#### (三)、球拍

#### (四)、手機

- 50MP
- 1080P@30fps/60fps，720p@30fps/60fps

### 二、研究軟體

(一)、Python 3.10.9 - 實驗主要程式語言。

(二)、opencv-python 4.10.0.84 - 視覺及影像處理函式庫，支援影片的讀取、物件追蹤。

(三)、pandas 2.2.2 - 資料分析與操作的函式庫，處理結構化資料

(四)、BlazePose 0.10.14 - 人體骨架關鍵點追蹤函式庫

(五)、ultralytics 8.2.56 - 提供 YOLOv8 物件偵測、人體骨架關鍵點追蹤模型的函式庫。

(六)、numpy 1.26.4 - 數學運算函式庫

(七)、matplotlib 3.9.1 - 資料視覺化的函式庫，主要用於繪製靜態圖表。

(八)、torch 2.3.0 - 開源的深度學習框架，適合用於深度學習模型訓練。

(九)、tensorflow 2.17.0 - 開源的深度學習框架，適合用於深度學習模型訓練。

## 肆、文獻探討

### 一、羽球常見動作姿勢

以下表格為研究者在進行相關文獻探討、以及諮詢專業教練後整理出的表格。

表(一)新手打羽球時常犯錯誤

| 姿勢              | 姿勢錯誤地方    | 詳細解說                                |
|-----------------|-----------|-------------------------------------|
| 放網<br>(net)     | 1.手臂蜷縮    | 球拍距離身體太近沒有伸展，手臂夾腳<120度              |
|                 | 2.出手點過低   | 球拍低於胸口，沒有在高點擊球                      |
|                 | 3.右腳膝蓋超伸  | 膝蓋超出腳尖，腿部夾腳<90度，身體壓力集中在膝蓋上，可能造成運動傷害 |
| 長球<br>(long)    | 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   | 手臂無伸展，手臂夾腳<70度，無法有效利用身體力量帶動擊球       |
|                 | 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                  |
| 預備姿勢<br>(ready) | 1.腳步站位錯誤  | 雙腳平行站立，沒有一前一後                       |
|                 | 2.雙手下垂    | 手臂下垂，無舉在胸前預備                        |
|                 | 3.身體直立    | 身體沒有稍微前傾壓低重心預備                      |
| 發球<br>(serve)   | 1.站位錯誤    | 發球時必須慣用腳在前，非慣用腳在後                   |
|                 | 2.發球位置過高  | 發球時擊球點不可高於腰部                        |

羽毛球運動中，急煞救球動作對膝蓋的傷害最為明顯，可能導致半月軟骨磨損等問題。為了保護膝蓋，在急煞時應該注意前腳避免讓膝蓋超過腳尖，避免重心過度向前，以減少膝蓋關節的磨損風險。正確的姿勢有助於利用臀部和另一隻腳的大腿力量，從而減輕膝關節的壓力。

上肢傷害則常見於肩膀、手肘和手腕，特別是肩部肌腱反覆承受壓力，可能導致肩夾擠症候群或旋轉肌腱炎。而肘部的網球肘多是由於擊球技巧不當引起的。(康健雜誌，取自<https://www.commonhealth.com.tw/article/80061>)。

## 二、羽球運動科技發展現況

Clutch: AI for Racket Sports 手機APP程式即時捕捉並分析運動員的擊球角度、揮拍速度及跑動路徑。攝影鏡頭對場地環境及拍攝高度需依規定佈置安裝，有良好的準確度及應用效果。

鷹眼系統(Hawk-Eye)可追蹤球路並進行精確判球。它提供選手移動熱點圖和擊球位置，適合比賽判決及技術訓練。在每次奧運中都會有此系統來協助裁判們決定是否球有出界，但是此系統對於業餘的人們成本太高。基於人體姿勢估計與機器學習的羽球球種分類研究(Shiela Mecha Cabahug, 2022)，使用人體姿態估計和機器學習去訓練個人和通用的羽球擊球分類模型，訓練模型的三種演算法分別是CNN、LSTM和CNN-LSTM，研究結果顯示多種羽球分類動作的3個模型皆有良好的辨識準確率。此外，也有蒐集專業比賽錄影作為訓練數據的研究，先進行人體關節點追蹤，再應用SMPL(Skinned Multi-Person Linear Model)模型來建構人體3D模型，最後可對七種羽球動作進行分類(Lin, Sheng-Jie, 2024)。

由於場地與經費的限制，要在保證不受傷的情況下打好球，對新手而言無疑是一項挑戰，我們希望可以鑒於此想法進行改良。

## 三、人體姿態檢測模型(Human Pose Estimation,HPE)

### (一)、BlazePose

BlazePose 是專為行動裝置設計的高效能即時人體姿態估測系統，採用了輕量化的卷積神經網路架構。其核心流程分為兩個階段：首先檢測人體的感興趣區域(ROI)，然後在該區域內預測 33 個關鍵點。BlazePose 結合了熱圖技術與座標回歸方法，確保了高精度和快速推理能力。(Bazarevsky et al., 2020)

### (二)、YOLO

YOLO-Pose 是一種基於 YOLO 框架的即時多人體姿態估測技術，結合了物體檢測與姿態估測，透過單一卷積神經網路(CNN)同時預測邊界框與關鍵點位置。它直接回歸關鍵點提升多人體邊界框檢測與姿態估測的效率與準確性。(Wang X et al., 2024)

### (三)、MoveNet

MoveNet 是一種先進的 Bottom-up 人體姿態估計模型，利用熱圖精確定位人體的關鍵點。其架構包括帶有特徵金字塔網絡(FPN)的 MobileNetV2 作為特徵提取器，以及預測頭。MoveNet 借鑒了 CenterNet 的方法，並進行增強以提高運算速度與準確性。特徵提取器生成高分辨率特徵圖，預測頭根據這些圖生成人物中心熱圖、關鍵點回歸場及偏移場。模型先識別人物中心並預測初步關鍵點，再進行精細調整，最終通過熱圖和偏移量選擇最準確的關鍵點位置。

MoveNet 的關鍵點格式與 YOLO 相同，都是使用 COCO17 關鍵點。BlazePose有33個，包含 COCO以及BlazePose的額外格式。

#### 四、羽球路徑追蹤(Shuttlecock Tracking)

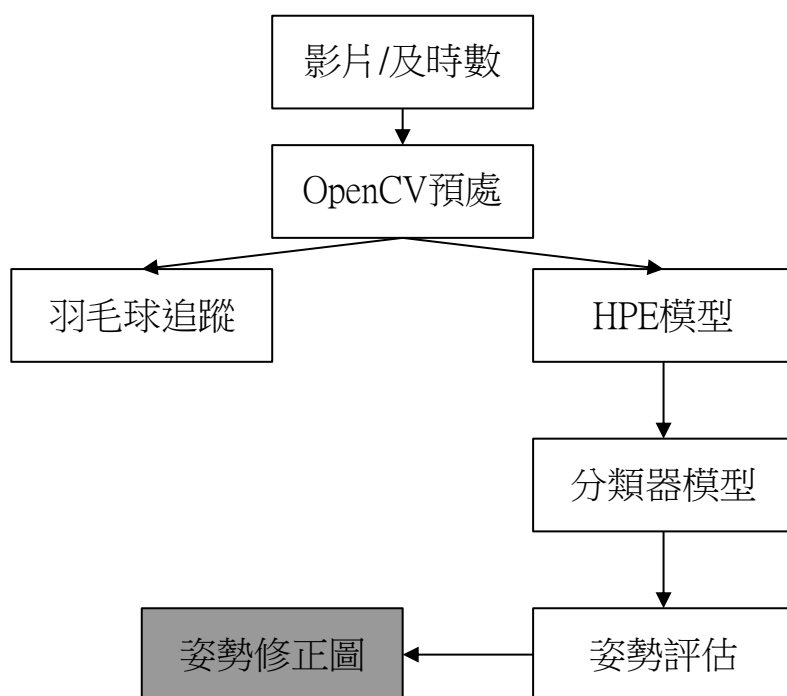
##### (一)、YOLO

YOLO(You Only Look Once)物體檢測算法使用單一卷積神經網絡(CNN)同時預測物體的邊界框和類別。其架構包含三個主要部分：骨幹網絡(Backbone，通常使用 CSP-Darknet53)、頸部(Neck，結合來自骨幹網絡的特徵圖)和頭部(Head，生成類別概率和邊界框)。YOLO 將輸入圖像分割成  $S \times S$  網格，每個網格負責檢測該區域內的物體，並預測邊界框及其置信度分數和類別概率。這種方法通過多部分的損失函數優化模型，使其能在單次前向傳播中完成高效、準確的實時物體檢測。(Redmon et al., 2016)

#### 伍、研究過程及方法

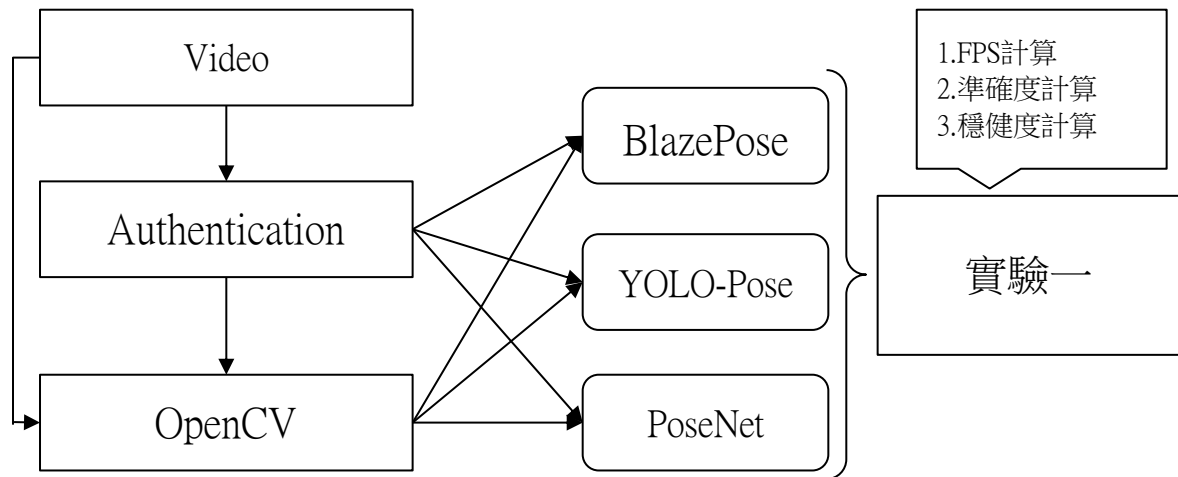
本研究比較了三種人體姿勢估測模型，並希望開發的軟體能成功捕捉人體的關鍵點以進行進一步分析。為此，我們正在測試四種常見的分類器模型(SVM、Dense、k-NN 和隨機森林)，以便理解玩家的實際動作。我們預期使用常見的方法進行分類，並計劃追蹤球的軌跡。數據集來自學校羽毛球隊及自錄影片，攝影機設置在羽毛球場邊界外1.5公尺處，捕捉球場的軌跡。該位置能有效提取人體關鍵點，並經測試發現，33公分的距離最適合用來練習動作，且適用於低成本相機。這一方法也適合業餘玩家進行動作捕捉。

下圖為研究架構圖，利用OpenCV進行資料前處理，HPE來判斷關鍵點(BlazePose、YOLO-Pose、MoveNet)，之後交由自己訓練的分類器模型，完成研究目的(一、二)。羽毛球的追蹤是YOLO、OpenCV負責，完成研究目的(三)。



## 一、實驗一、實驗不同人體姿態辨識模型

YOLO、Blaze Pose、MoveNet是三種最受歡迎的姿勢估計(HPE)模型。我們對這三種模型進行測試，檢驗它們的幀率(FPS)、準確性和穩健度。表(二)為研究者整理的表格。



圖(二)實驗一研究架構

表(二)YOLO、BlazePose、MoveNet歸納表

| 特徵    | BlazePose           | YOLO v8 Pose     | MoveNet            |
|-------|---------------------|------------------|--------------------|
| 拓撲    | 33個關鍵點              | 17個關鍵點           | 17個關鍵點             |
| 工作流程  | 檢測運行一次，隨後直到出現遮擋再跟蹤  | 每幀都進行檢測          | 每幀都進行檢測            |
| 分割    | 分割已整合到姿勢中           | 分割未直接整合到姿勢中      | 分割未直接整合到姿勢中        |
| 識別人數  | 單人                  | 多人               | 多人                 |
| 支持的語言 | Python & JavaScript | Python           | Python             |
| 兼容性   | TensorFlow          | PyTorch, Darknet | TensorFlow Lite    |
| 記憶體占用 | 低                   | 中                | 低                  |
| 種類    | 0, 1, 2             | n, s, m, l, x    | lightning, thunder |

### (一)、FPS(Frame Per Second) 計算

我們利用Python的time模組計時，關鍵點抓取程式碼的兩端放入time.time()，令此時間為 $t$ ，則 $FPS = 1/t$ ，雖然OpenCV也有自己的FPS計算：cv2.CAP\_PROP\_FPS，但是OpenCV的FPS代表的是讀取影片或攝像頭輸入影像的效率評估，並非代表的是模型一次處理完一幀所消耗的時間。

### (二)、準確性計算

在測試準確性時，我們使用了一種不太常見的方法，這與論文或研究中常見的測試方法有所不同。對於人體姿勢估計(HPE)模型，最常見的準確性評估方法是計算「平均關節精度」(Mean Joint Precision) $E_{MPJ}$ ：

$$E_{MPJ} = \frac{1}{N_s} \sum_{i=1}^{N_s} ||m_f S(i) - m_{gt} S(i)||$$

$N_s$ 是所有關節點個數， $m_f S(i)$ 是模型預測出的第 $i$ 個關節點， $m_{gt} S(i)$ 是自己標註的點 $(x, y)$ (Ground-Truth)。我們認為利用這種公式並不符合一般的需求，因為一個人體肩膀的定義並非單點，而是一個區域。由於上述原因和限制，我們選擇了不同的方法。我們編寫了一個程式，將影片分割成照片，最終獲得 4,523 張照片。我們會對這些照片進行人工審核，好的圖片時候標記為正確，在同時也會將正確的照片以及正確的關節點擷取除存在一個CSV檔案，不正確的將會刪除，準確率的計算就會是：正確圖片數量 / 4523。

### (三)、穩健計算(Robustness)

OpenCV除了可以讀取影片和攝像頭畫面外，還具備調整亮度、對比度和進行高斯模糊等常見照片編輯功能。我們旨在使此軟件在光強度或鏡頭限制下表現穩定，因此對經過人工審核的照片，我們利用OpenCV增強其亮度，保證調整後的圖像對人類依然清晰可辨。我們選取了1000張照片，通過OpenCV隨機更改亮度(0.5倍至1.5倍)、更改飽和度(0.5倍至1.5倍)，並隨機應用核大小為3、5或7的高斯模糊進行處理。這樣的處理流程確保圖像在不同條件下的可用性和穩定性，同時保持圖像的可識別性。

我們錯誤定義公式 $E_{AMPJ}$ 是：

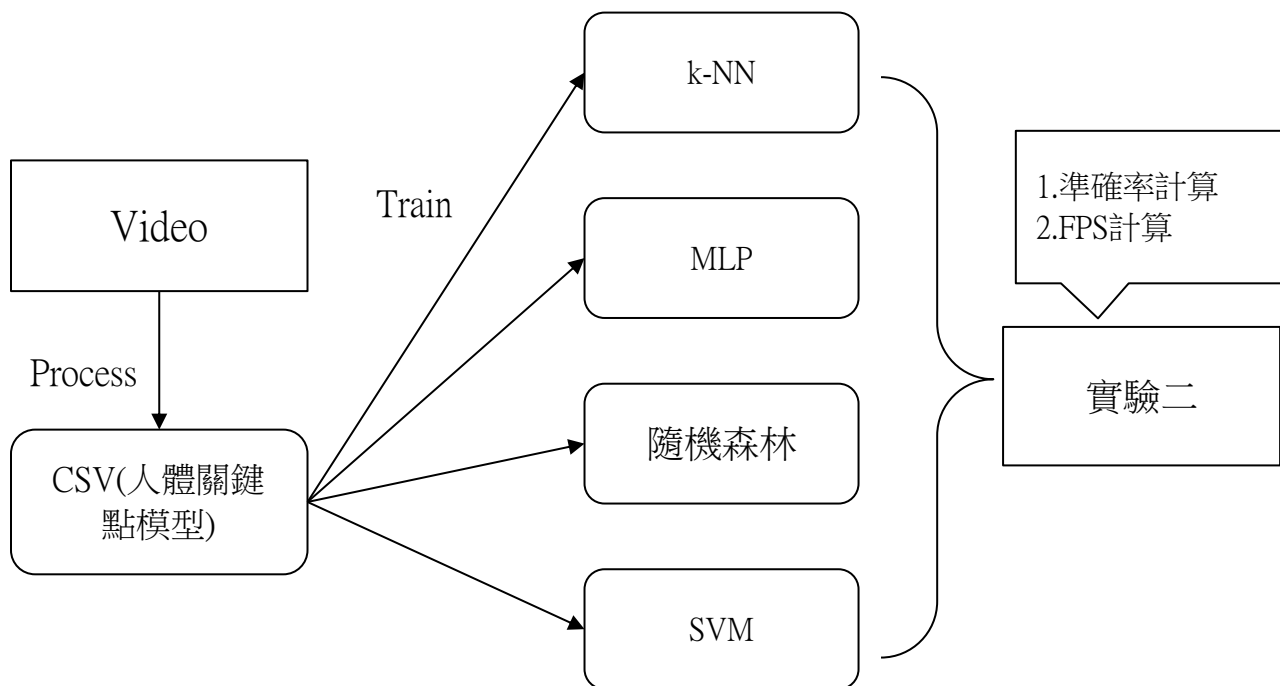
$$E_{AMPJ}(a) = \frac{1}{N_s} \sum_{i=1}^{N_s} ||m_a S(i) - m_o S(i)||$$

$N_s$ 是所有關節點個數， $m_a S(i)$ 是模型再以編輯過照片回傳的第 $i$ 個關節點， $m_o S(i)$ 是在位編輯照片原本回傳的第 $i$ 個關節點。



## 二、實驗二、實驗2D人體姿態辨識姿勢並且嘗試分類

圖(三)中展示的是實驗二的研究架構。在我們的實驗中，分類模型採用了兩種不同的輸入資料來源。其中，方塊中的模型(k-NN、MLP、Random Forest、SVM)使用的是關節座標點(x, y)作為輸入，因此需要先經由人體辨識模型擷取關節點後，才能進行分類。在完成實驗一後，我們將選用最佳的人體骨架辨識模型，用以提取關節點資料。至於訓練階段所使用的圖片數據，均經由實驗一前的人工審核過程整理而成，確保數據的準確性與可靠性。



圖(三)實驗二實驗架構

分類模型是機器學習中的重要工具，它們通過學習數據特徵來對新數據進行分類。常見的分類模型包括SVM、決策樹、隨機森林和神經網絡等，各自在不同的應用場景中表現出優勢。例如，SVM 在高維度數據中效果突出，而決策樹因其直觀易解釋的特性，特別適合處理缺失數據或異質數據集。隨機森林集成多個決策樹，能提升穩定性和準確性，而朴素貝葉斯則因計算效率高，對小樣本數據效果良好。

在測試階段，我們將針對不同資料類型進行評估，包括圖片以及連續的時間序列數據，以了解各模型在不同情境下的效能表現。我們希望可找出一個快速並且準確的模型，來提高軌跡預測的準確性。

|    |   |   |
|----|---|---|
| 名稱 | Dense   | k-NN  |
| 圖片 | <p>Ref: <a href="https://medium.com/@mlg788/deep-architecture-multi-layer-perceptron-764b0c938642">https://medium.com/@mlg788/deep-architecture-multi-layer-perceptron-764b0c938642</a></p> | <p>Ref: <a href="https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html">https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html</a></p> |
| 名稱 | SVM   | k-NN  |
| 圖片 | <p>Ref: <a href="https://vitalflux.com/classification-model-svm-classifier-python-example/">https://vitalflux.com/classification-model-svm-classifier-python-example/</a></p>               | <p>Ref: <a href="https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html">https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html</a></p> |

圖(四)分類模型架構示意圖

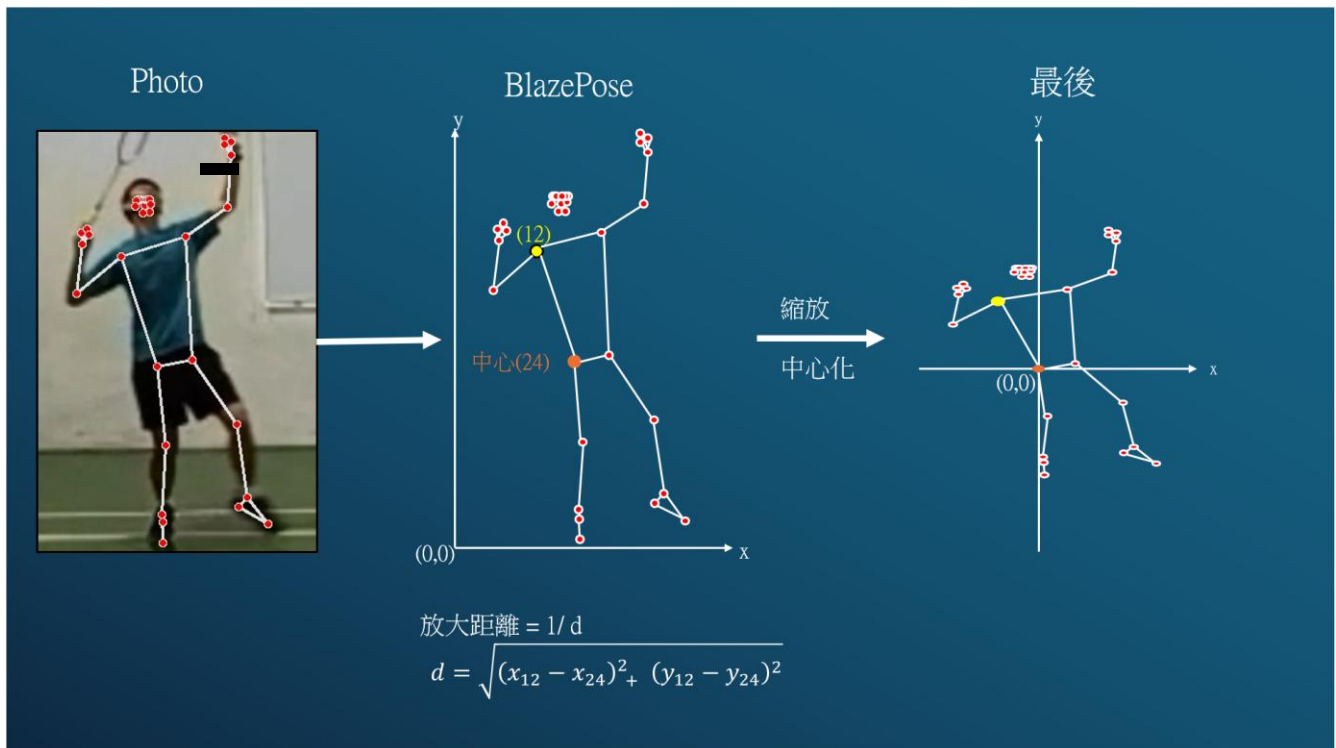
我們在實驗遇到了一個很大的問題：

|      |    |        |
|------|----|--------|
|      | 原圖 | 裁切過後的圖 |
| 無正規化 |    |        |
| 有正規化 |    |        |

圖(五)實驗二 正規化比對結果

圖(五)中，我們可以明顯看出，當圖片被裁剪至僅包含使用者的邊界框時，模型會做出錯誤的推論。兩個模型在原始圖片中都能成功推論(皆為 100% 正確)，然而在裁剪後的圖片中，無正規化模型自信地判斷該姿勢為發球，而有正規化模型則能正確分類為長球姿勢。這個問題乍看之下似乎很小，但其影響非常嚴重。BlazePose 的 x 座標格式表示到右邊界的像素距離，而 y 座標格式表示到下邊界的像素距離。這稱為正規化(Normalization)，即使資料範圍被壓縮到 0 至 1 之間，使用者的位移仍然會顯著影響分類結果。因此，如果我們直接使用 BlazePose 的格式進行訓練，場上不同位置的球員將會嚴重影響推論結果。

為了解決這個問題，我們採用了一種新的方法來降低誤差，首先，我們選擇了一個中心點(這指的是BlazePose中的關鍵點24，請參考圖(六))。選擇這個點的原因是因為右臀部在所有 Blaze Pose關鍵點中是最容易觀察到的點。接著，我們選擇了點 12 到 24 之間的距離作為主要的縮放距離，因此我們獲取的每個關鍵點都會根據這個距離進行定位和縮放，使得點 12 到 24 之間的距離正好為 1、使得點 24 的座標正好為 (0,0)。



圖(六)正規化示意圖

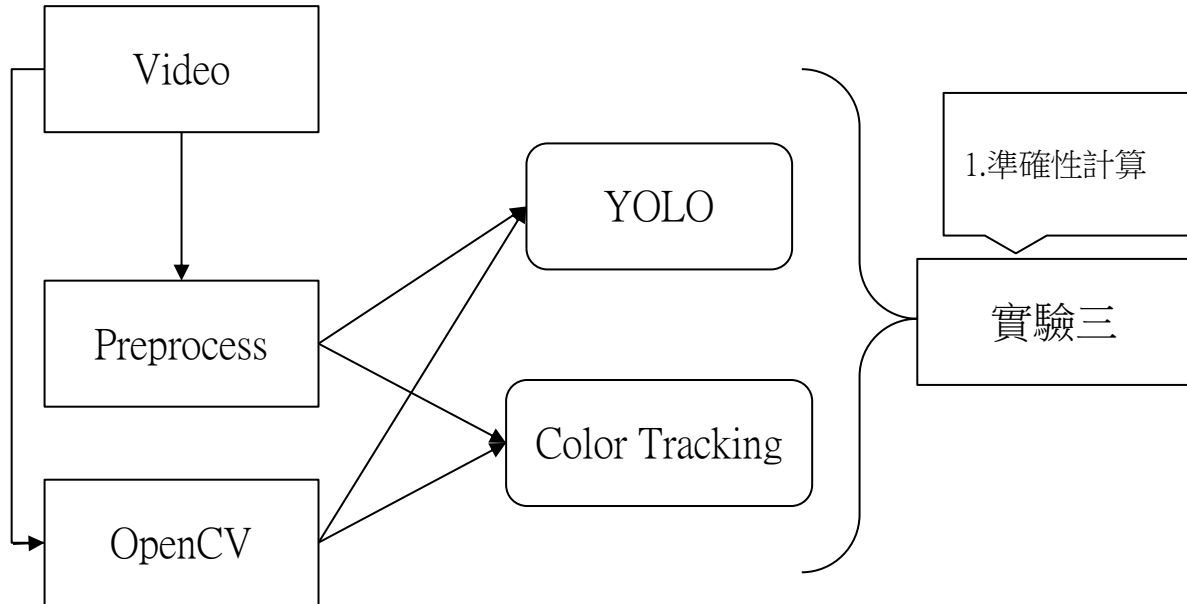
可知正規化的座標為：

$$\hat{x}_i = \frac{x_i - x_{24}}{\sqrt{(x_{12} - x_{24})^2 + (y_{12} - y_{24})^2}} ; \hat{y}_i = \frac{y_i - y_{24}}{\sqrt{(x_{12} - x_{24})^2 + (y_{12} - y_{24})^2}}$$

$(x_i, y_i)$ 是每一個BlazePose所載取的關鍵點座標， $(\hat{x}_i, \hat{y}_i)$ 是每一個在縮放和中心化過後的座標。

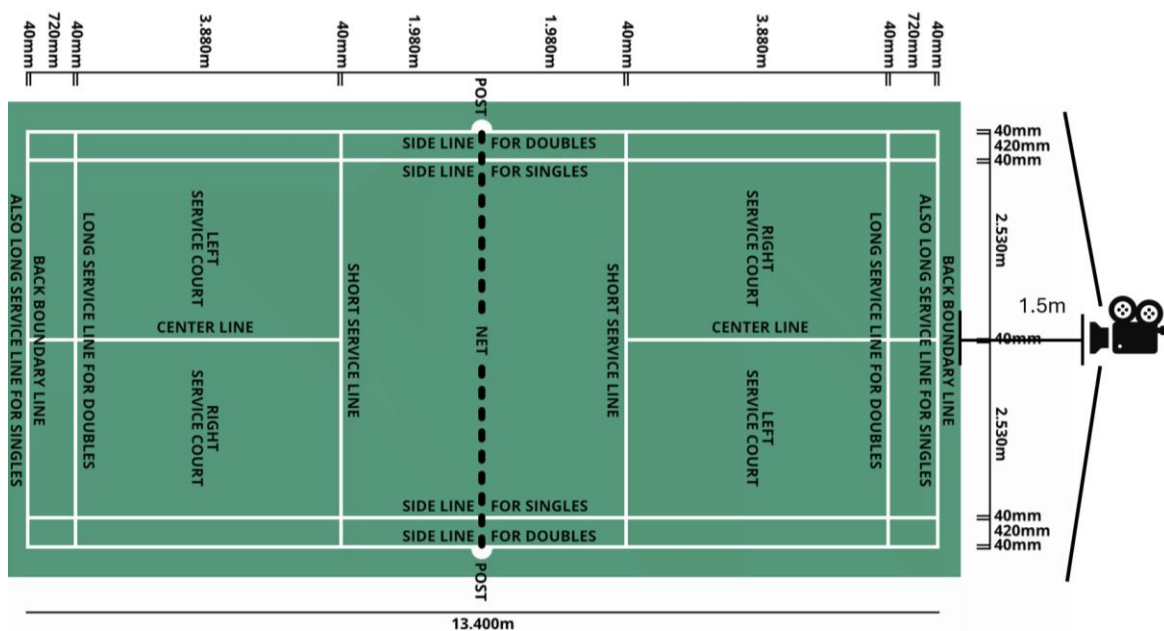
### 三、實驗三、利用電腦視覺追蹤球的路徑

圖(七)是實驗三的研究架構，我們的輸入資料來源會是錄不同羽毛球的飛行路徑，我們希望可以透過追蹤球的軌跡，來幫助使用者了解自己的姿勢改變後的成效。由於羽毛球非常地快速，全世界最快的，為了突破這個問題，我們選擇了SSD、YOLO、以及顏色追蹤，我們會利用CVAT 進行羽毛球的標記。



圖(七)實驗三實驗架構

我們選擇的位置是在對立於球員的球場的中間放置腳架，這個位置既可以拍到整個球場，也可以拍到上空的位置(黑色攝影機)，攝影機的選擇會採用(30FPS)。



圖(八)實驗三場地

目前最常見的追蹤方法是利用下列三種方法：

顏色追蹤基於球體與背景的顏色差異，可通過濾色或HSV空間分割精確定位，適合簡單場景。此方法易受光照變化和背景複雜度影響，可能導致準確性下降。雖然羽毛球的球場顏色也相當複雜，我們選擇了非常鮮豔的顏色，非常顯著，也不容易和球場的任何顏色會混合。但是也為了減少錯誤，我們會對輸入的影像進行預處理。例如高斯模糊減少雜訊、增加飽和度以凸顯顏色，也降低亮度以減少雜訊。對於顏色追蹤，我們也會測試不同的顏色。例如橘(255,165,0)、螢光粉紅(255,20,147)、水藍色(0,255,255)。

深度學習中的卷積神經網絡(CNN)在多物體場景中表現出色，應用於自駕車的行人和交通標誌識別、視頻監控中的異常行為檢測及醫療影像的精確分割和診斷。LeNet-5 奠定手寫數字識別的基礎；AlexNet 提升了圖像分類準確性；VGGNet 以深層結構和小卷積核取得成果；ResNet 用殘差塊解決深層網絡訓練難題。

我們的圖像資料會經過人工標註進行訓練，總共是\_張，誤差率公式 $E_{ST}$ ：

$$E_{ST} = \frac{1}{N} \sum_{i=1}^N ||K_f(i) - K_{gt}(i)||$$

$K_f(i)$ 是羽毛球追蹤模型回傳的座標點， $K_{gt}(i)$ 是人工標註的座標點。

## 伍、研究結果

### 一、實驗一、實驗不同人體姿態辨識模型

#### (一)、FPS測試(1/t)

表(三)實驗一FPS測驗結果

| 模型                | 平均    | 最大值    | 最小值   |
|-------------------|-------|--------|-------|
| YOLO_n            | 16.30 | 23.33  | 1.95  |
| BlazePose_0       | 15.09 | 22.67  | 1.54  |
| BlazePose_1       | 13.45 | 17.34  | 1.56  |
| BlazePose_2       | 6.66  | 1.33   | 7.81  |
| MoveNet_lightning | 83.6  | 127.33 | 10.61 |
| MoveNet_thunder   | 25.77 | 30.66  | 25.47 |

#### (二)、準確度測試(%)

表(四)實驗一 準確度測試結果

| 模型                | 準確率   |
|-------------------|-------|
| YOLO_n            | 62.1% |
| BlazePose_0       | 59.3% |
| BlazePose_1       | 65.2% |
| BlazePose_2       | 65.2% |
| MoveNet_lightning | 6.0%  |
| MoveNet_thunder   | 7.0%  |

#### (三)、穩健度測試(像素距離)

表(五)實驗一 穩健度測試結果

| 模型                | 原圖 | 亮度改變   | 高斯模糊過後 | 對比度增強  |
|-------------------|----|--------|--------|--------|
| YOLO_n            | 0  | 0.0000 | 0.0000 | 0.0000 |
| BlazePose_0       | 0  | 0.0163 | 0.0207 | 0.0150 |
| BlazePose_1       | 0  | 0.0110 | 0.0147 | 0.0106 |
| BlazePose_2       | 0  | 0.0096 | 0.0140 | 0.0088 |
| MoveNet_lightning | 0  | 0.0000 | 0.0000 | 0.0000 |
| MoveNet_thunder   | 0  | 0.0000 | 0.0000 | 0.0000 |

我們測試了相機的幀率(FPS)。當視頻輸入 OpenCV 時，OpenCV 通常會加速處理，但相機的幀率始終設定在 30 FPS。為了減少誤差，我們開啟網絡攝像頭來計算數據。在我們的數據集中，MoveNet 捕捉關鍵點的效果很差，而 YOLO 和 BlazePose 的表現都超過 50%。在像素級測試中，所有模型在照片編輯技能方面表現出色，但這不代表其精確度。然而，模型的穩健性依然明顯。

## 二、實驗二、研究及實驗2D人體姿態辨識姿勢並且嘗試分類

混淆矩陣是一種有效工具，用於評估分類模型的性能，通過比較實際和預測來展示結果。它包含四個部分：真正例(TP)表示正確預測的正類實例；負例(TN)表示正確預測的負類實例；假正例(FP)表示錯誤預測為正類的負類實例；假負例(FN)表示錯誤預測為負類的正類實例。混淆矩陣突顯出模型在哪些方面可能容易出錯。

表(六) 混淆矩陣示意圖

|       | 預測類別 1   | 預測類別 2   | 預測類別 3   | 預測類別 4   |
|-------|----------|----------|----------|----------|
| 實際類別1 | 真正例 (TP) | 假負例 (FN) | 假負例 (FN) | 假負例 (FN) |
| 實際類別2 | 假正例 (FP) | 真正例 (TP) | 假負例 (FN) | 假負例 (FN) |
| 實際類別3 | 假正例 (FP) | 假正例 (FP) | 真正例 (TP) | 假負例 (FN) |
| 實際類別4 | 假正例 (FP) | 假正例 (FP) | 假正例 (FP) | 真正例 (TP) |

- macro avg: 每個類別的Precision、Recall和 F1-score進行平均，類別的權重是相等的。
- weighted avg:每個類別的Precision、Recall和 F1-score進行平均，類別的權重是相等的
- Accuracy:  $\frac{TP+TN}{TOTAL}$
- Precision:  $\frac{TP}{TP+FP}$
- Recall:  $\frac{TP}{TP+FN}$
- F1-score:  $2 \times \frac{Precision \times Recall}{Precision + Recall}$

(一)-1、Dense(70 期、有正規化)

Accuracy≈98%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 146  | 2   | 8   | 1    |
|      | 放網   | 0    | 172 | 1   | 0    |
|      | 發球   | 2    | 1   | 128 | 1    |
|      | 預備姿勢 | 1    | 2   | 1   | 71   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 0.98      | 0.98   | 0.98     | 537     |
| weighted avg | 0.98      | 0.98   | 0.98     | 537     |

圖(九) Dense(有正規化) 模型訓練結果

(一) -2、Dense(70 期、無正規化)

Accuracy  $\approx$  93%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 149  | 0   | 6   | 2    |
|      | 放網   | 0    | 169 | 3   | 1    |
|      | 發球   | 2    | 1   | 128 | 1    |
|      | 預備姿勢 | 1    | 2   | 16  | 56   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 0.93      | 0.91   | 0.92     | 537     |
| weighted avg | 0.93      | 0.93   | 0.93     | 537     |

圖(十) Dense(無正規化) 模型訓練結果

(二)-1、隨機森林(有正規化)

Accuracy $\approx$ 98%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 151  | 2   | 3   | 1    |
|      | 放網   | 0    | 173 | 0   | 0    |
|      | 發球   | 2    | 1   | 128 | 1    |
|      | 預備姿勢 | 1    | 1   | 0   | 73   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 0.98      | 0.98   | 0.98     | 537     |
| weighted avg | 0.98      | 0.98   | 0.98     | 537     |

圖(十一) 隨機森林(有正規化) 模型訓練結果

(二)-2、隨機森林(無正規化)

Accuracy $\approx$ 100%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 156  | 0   | 1   | 0    |
|      | 放網   | 0    | 172 | 0   | 1    |
|      | 發球   | 0    | 1   | 131 | 0    |
|      | 預備姿勢 | 1    | 0   | 0   | 74   |



|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 1.00      | 1.00   | 1.00     | 537     |
| weighted avg | 1.00      | 1.00   | 1.00     | 537     |

圖(十二) 隨機森林(無正規化) 模型訓練結果

(三)-2、SVM(有正規化)

Accuracy≈96%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 147  | 3   | 7   | 0    |
|      | 放網   | 1    | 170 | 1   | 1    |
|      | 發球   | 0    | 2   | 128 | 2    |
|      | 預備姿勢 | 1    | 2   | 3   | 69   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 0.96      | 0.95   | 0.95     | 537     |
| weighted avg | 0.96      | 0.96   | 0.96     | 537     |

圖(十三) SVM(有正規化) 模型訓練結果

(三)-2、SVM (無正規化)

Accuracy≈99%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 157  | 0   | 0   | 0    |
|      | 放網   | 0    | 173 | 0   | 0    |
|      | 發球   | 1    | 1   | 129 | 2    |
|      | 預備姿勢 | 0    | 0   | 2   | 73   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 0.99      | 0.99   | 0.99     | 537     |
| weighted avg | 0.99      | 0.99   | 0.99     | 537     |

圖(十四) SVM (無正規化) 模型訓練結果

(四)-1、k-NN(有正規化)

Accuracy≈99%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 153  | 1   | 2   | 1    |
|      | 放網   | 1    | 172 | 0   | 0    |
|      | 發球   | 1    | 1   | 129 | 1    |
|      | 預備姿勢 | 0    | 0   | 0   | 75   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 0.98      | 0.99   | 0.98     | 537     |
| weighted avg | 0.99      | 0.99   | 0.99     | 537     |

圖(十五) k-NN(有正規化) 模型訓練結果

(四)-2、k-NN(無正規化)

Accuracy≈100%

|      |      | 預測類別 |     |     |      |
|------|------|------|-----|-----|------|
|      |      | 長球   | 放網  | 發球  | 預備姿勢 |
| 實際類別 | 長球   | 156  | 0   | 1   | 0    |
|      | 放網   | 0    | 173 | 0   | 0    |
|      | 發球   | 0    | 0   | 132 | 0    |
|      | 預備姿勢 | 0    | 0   | 0   | 75   |

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| macro avg    | 1.00      | 1.00   | 1.00     | 537     |
| weighted avg | 1.00      | 1.00   | 1.00     | 537     |

圖(十六) k-NN(無正規化) 模型訓練結果

### 三、實驗三、利用電腦視覺追蹤球的路徑

#### (一)、顏色追蹤

表(七)顏色追蹤(顏色差異)

| 顏色    | 誤差 |
|-------|----|
| 白色    | -  |
| 橘色    | -  |
| 螢光粉紅色 | -  |

#### (二)、YOLO

表(八)YOLO追蹤(顏色差異)

| 顏色    | 誤差     | 正確率   |
|-------|--------|-------|
| 白色    | 2.79px | 91.5% |
| 橘色    | 4.94px | 98.9% |
| 螢光粉紅色 | 7.49px | 89.5% |

當我們嘗試追蹤顏色時，我們已經預見到白色羽毛球的準確度會最差，誤差也會最大。然而，當我們嘗試追蹤橙色和粉紅色時，發現當羽毛球快速移動時，顏色會變暗，並轉變為更灰暗的顏色，因此顏色的遮罩會與其他顏色的HSV/RGB範圍重疊。

## 陸、討論

### 一、實驗一、實驗不同人體姿態辨識模型

根據實驗一的結果，可以看出 MoveNet 的每秒幀數 (FPS) 最高。我們認為這是因為 MoveNet 採用了 MobileNetV2 作為骨架，並結合了特徵金字塔網絡 (FPN)，從而提高了計算效率。我們發現 MoveNet 在近距離檢測時非常準確，但由於我們相機的安裝位置，MoveNet 被認為不適合於這種情況。此外，YOLO 和 BlazePose 在準確性測試中都表現良好，其中 YOLO 略勝一籌。最終，我們決定使用 BlazePose，因為它的實施相對簡單而且它的關鍵點數量更多。另外，當我們調整圖片(亮度、高斯模糊、對比度)時，YOLO 和 MoveNet 的性能沒有受到影響。然而，BlazePose 的性能受到了影響。我們懷疑這是因為 BlazePose 需要從 RGB 轉換為 BGR，然後再轉換為 HSV，我們猜測這就是它比其他兩個模型產生更多錯誤的原因。

## 二、實驗二、實驗2D人體姿態辨識姿勢並且嘗試分類

根據對八種不同模型類型的評估結果，我們觀察到：

- 混淆矩陣：所有模型的混淆矩陣都很清晰，表明在正確分類樣本方面表現良好。
- 性能指標：所有模型的精確度、召回率和F1分數均超過90%，這表明我們的預測具有高度的準確性和可靠性。

這些結果強烈表明我們的模型訓練是成功的。高精確度、召回率和F1分數表明我們的模型有效地捕捉到了數據中的模式並做出了準確的預測。清晰的混淆矩陣進一步證實了我們的模型穩健性。

針對這三種先進的機器學習算法，隨機森林(Random Forest)、k最近鄰居(k-NN)和支持向量機(SVM)可以進行微調。

- 隨機森林：最佳樹數為70，測試準確率為97.77%。
- 支持向量機(SVM)：在研究中，我們發現SVM有多種決策邊界函數(線性、多項式、RBF和Sigmoid)。結果顯示線性是最佳選擇，使用正則化參數100.0的線性SVM測試準確率為97.39%。
- k最近鄰居(k-NN)：當k估計值為1時，測試準確率最高，達到98.51%。這些結果顯示，我們對這些機器學習模型的微調是成功的，並且在準確率方面取得了優異的表現。

我們會比較有正規化以及沒有正規化模型的比較 $S_{p(i)}$ ：

$$S_{p(i)} = \frac{\text{有正規化} - \text{無正規化}}{\text{無正規化}}$$

$S_{p(i)}$ 是對於每一個姿勢的進步幅度， $S_{p(i)}$ 總和除以4就會是平均進步幅度。

### (一)、Dense正確率 (%)

表(九)Dense 實驗結果

|      | 有正規化   | 無正規化   |
|------|--------|--------|
| 長球   | 97.33% | 74.67% |
| 放網   | 98.67% | 28%    |
| 發球   | 33.34% | 65.33% |
| 預備動作 | 85.33% | 69.33% |

### (二)、隨機森林正確率 (%)

表(十)隨機森林實驗

|      | 有正規化   | 無正規化    |
|------|--------|---------|
| 長球   | 98.67% | 100.00% |
| 放網   | 86.67% | 72.00%  |
| 發球   | 30.37% | 26.67%  |
| 預備動作 | 86.67% | 32.00%  |

(三)、k-NN正確率 (%)

表(十一)k-NN 實驗結果

|      | 有正規化   | 無正規化   |
|------|--------|--------|
| 長球   | 96.00% | 98.67% |
| 放網   | 84.00% | 30.67% |
| 發球   | 52.00% | 26.67% |
| 預備動作 | 73.34% | 66.67% |

(四)、SVM正確率 (%)

表(十二)SVM 實驗結果

|      | 有正規化    | 無正規化   |
|------|---------|--------|
| 長球   | 98.67%  | 2.67%  |
| 放網   | 82.67%  | 0.00%  |
| 發球   | 0.00%   | 54.67% |
| 預備動作 | 100.00% | 66.67% |

(五)、每個模型平均推論時間 (t)

表(十三)分類模型所花時間

| 模型類型  | 平均時間   |
|-------|--------|
| Dense | 0.0407 |
| 隨機森林  | 0.0230 |
| k-NN  | 0.0362 |
| SVM   | 0.0010 |

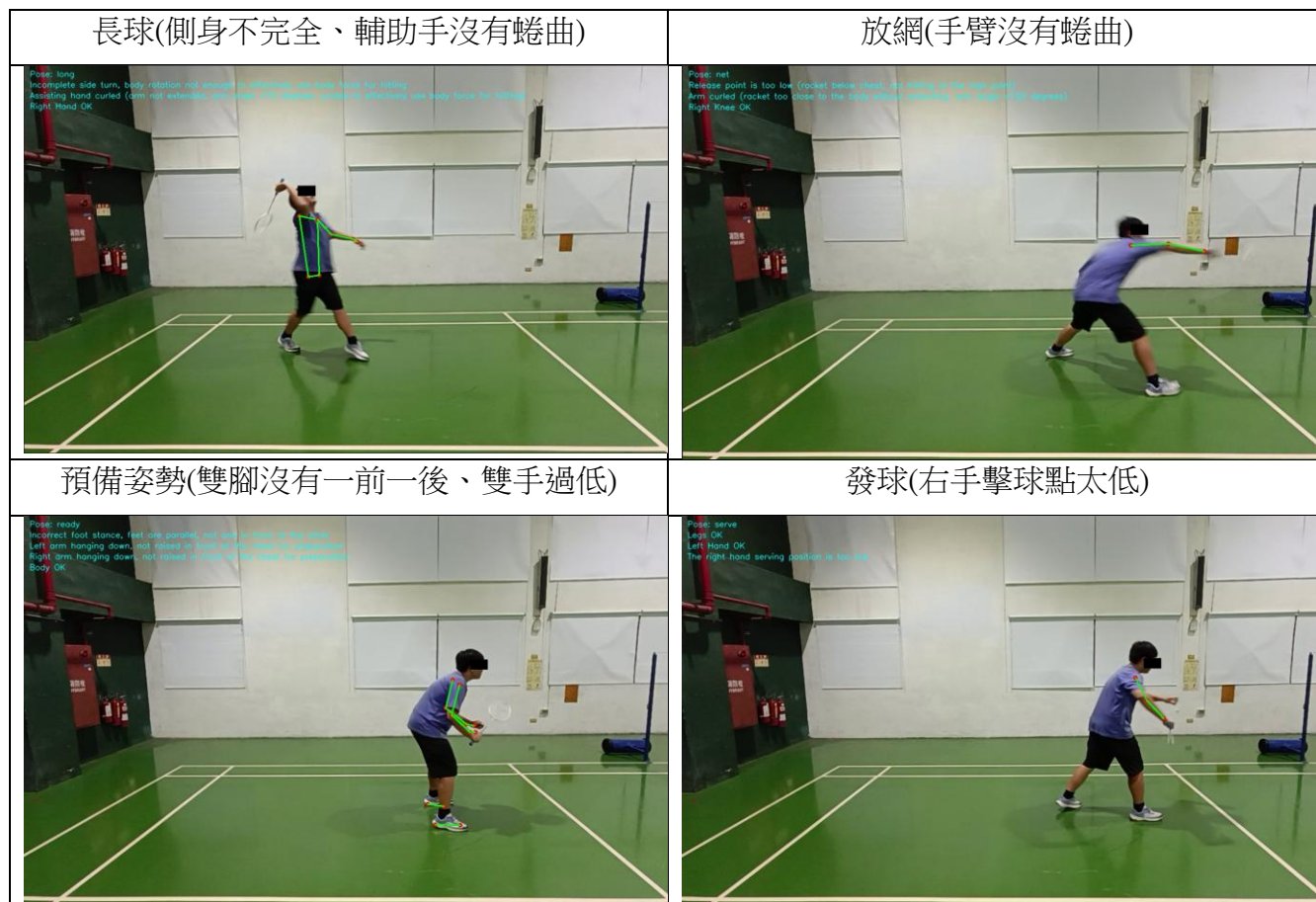
(六)、正規化提升幅度 (因為會出現除以零的情況，SVM 的增加無法量化。)

表(十四)分類模型提升幅度

| 模型類型  | 提升幅度        |
|-------|-------------|
| Dense | 64.212%     |
| 隨機森林  | 50.941%     |
| k-NN  | 69.039%     |
| SVM   | 增加157.3個百分點 |

從上述數據中，我們可以發現標準化坐標對推論有巨大的影響。經過標準化的 k-NN 模型比過去幾年未標準化的模型提高了69%。然而，當模型試圖區分發球和準備動作時，問題變得非常嚴重。我們猜測，數據不足以讓模型成功推斷。未來，我們會增加更多數據，或者專門錄製這些姿勢幾分鐘。此外，我們認為在未來我們可以省略某些點來減少錯誤，特別是頭部部分。

接下來，我們可以選擇最佳的模型。K-NN 能夠以 50% 的準確率分類預備姿勢、其他姿勢分類則高達八成。因此，我們終於可以開始進行姿勢清單檢查，動作請參見表格(一)。



圖(十七)姿勢分析功能

### 三、實驗三、利用電腦視覺追蹤球的路徑

從上表中，我們可以輕易看出YOLO模型在羽毛球捕捉上表現優異，且誤差距離不到10個像素。所有顏色中，橙色表現最好，我們對YOLO仍能捕捉到飛行中的白色羽毛球感到驚訝，羽毛球或許在未來不用塗色，透過增加訓練資料量，就可以使偵測更完美。然而，顏色追蹤的表現最差，原因在於球的顏色在高速中暗許多，甚至無法正確偵測。使用壓克力顏料可以覆蓋羽毛球，但經常會有一些小顆粒的乾漆脫落。我們仍然認為需要改進我們的羽毛球著色技術，或者未來可以考慮使用塑膠羽毛球。

## 柒、未來可改進項目

我們的項目是一個很好的主題，因為目前對新手或業餘選手的羽毛球姿勢矯正研究很少甚至沒有。現在的大多數產品要麼是報告你或你的對手喜歡的運動擊打方式，或是顯示基本的運動統計數據。Clutch: AI for Racket Sports 僅提供額外的統計數據，適合高成本、豪華的攝像機裝置。我們列出了一些可以改進的地方：

- 一、增加訓練數據。
- 二、使用生成對抗網絡(GAN)模型更好地視覺化使用者如何矯正姿勢。
- 三、分析正確姿勢和錯誤姿勢之間的軌跡作為評估依據。
- 四、定位球場關鍵點以及運用多相機來增強使用者的使用體驗。
- 五、增加更多類別(例如增加一個名為“其他”的類別以減少混淆)。
- 六、通過卷積神經網絡(CNN)或 CNN-LSTM 改進分類器。
- 七、實施追蹤模型(Deep SORT)。

## 捌、結論

- 一、BlazePose和 YOLO 都適用於更複雜的場景，更適合固定攝影機位置較遠的情況。
- 二、正規化對分類模型的影響是可觀察的，幾乎提升了1.6倍。
- 三、對於未正規化的模型，最佳分類模型是隨機森林(Random Forest)
- 四、對於數據正規化的模型，最佳分類模型是 k-NN。
- 五、顏色追蹤不適合用在羽毛球追蹤上。
- 六、使用顏色橘色的 YOLO 效果最佳。
- 七、這個主題有光明的前景，我們將致力於開發更多功能，為用戶帶來最佳體驗。

## 玖、參考文獻資料

- Bazarevsky, V., Grishchenko, I., Raveendran, K., Kartynnik, Y., Vakunov, A., Grundmann, M., & Slu, B. (2020). BlazePose: On-device real-time body pose tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (pp. 1004-1011). arXiv. <https://arxiv.org/abs/2006.10204>
- Huang, Y.-C., Liao, I.-N., Chen, C.-H., Ik, T.-U., & Peng, W.-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>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). arXiv. <https://arxiv.org/abs/1506.02640>
- 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). Retrieved from <https://etheses.lib.ntust.edu.tw/thesis/detail/63ccf6a6d4ad0399ee9a48cf8311984a>
- Toshniwal, D., Patil, A., & Vachhani, N. (2024). AI coach for badminton. arXiv preprint arXiv:2403.08956. Retrieved from <https://arxiv.org/abs/2403.08956>
- Wang, X., Zhang, R., He, D., Zhou, H., Liu, Y., & Tan, T. (2022). YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. arXiv preprint arXiv:2204.06806. <https://arxiv.org/abs/2204.06806>