

國立臺南大學資訊工程學系
113 級畢業專題報告

AI 人工智慧影片內容分析應用_以羽球競賽
對戰策略分析為例

*AI Video Content Analysis Application - Case Study:
Badminton Match Strategy Analysis*

專案編號：NUTN-CSIE-PRJ-113-016

組員：S10959024 吳定昌

S10959033 李崇楷

S10959036 陳柏舟

指導老師：林朝興 教授

中華民國 113 年 1 月

國立臺南大學資訊工程學系
畢業專題報告審定書

AI 人工智慧影片內容分析應用_以羽球競賽
對戰策略分析為例

*AI Video Content Analysis Application - Case Study:
Badminton Match Strategy Analysis*

學生 學號：_____ 姓名：_____

 學號：_____ 姓名：_____

 學號：_____ 姓名：_____

所提之報告內容符合本學系大學部畢業專題實作標準。

指導教授：_____

系主任：_____

中華民國 113 年 1 月

摘要

本研究旨在運用人工智慧 (AI) 技術於羽球比賽，以提升選手的技術水平和戰術意識。儘管已使用多種電腦輔助技術收集羽球數據，但手動標記仍需人力與時間，尤其在技術資料識別方面需要羽球專業人員。研究問題集中於如何結合運動分析方法，以開發準確辨識羽毛球飛行軌跡以及擊球姿態的模型。在方法與步驟中，本專題使用 TrackNetV2 追蹤球的座標，並對座標點進行預處理，匯集成訓練資料，並進行 CNN 訓練，以判斷羽毛球的飛行軌跡。同時，擊球姿態辨識模型是透過 YOLOv7 擷取選手，利用 OpenPose 偵測擊球者的擊球姿態，根據四種不同姿勢建立多個訓練資料，並進行 CNN 訓練，比較不同訓練資料的準確率，以找到最佳模型。最終，本專題將能透過辨識球路模型預測和策略建議，協助選手教練在比賽中做出更明智的決策。期望本專題成果不僅對羽球競技有所助益，也對運動科學領域帶來新的啟發。

關鍵詞: TrackNetV2、CNN 訓練、YOLOv7、OpenPose

***AI Video Content Analysis Application - Case Study: Badminton Match
Strategy Analysis***

ABSTRACT

This research aims to use artificial intelligence (AI) technology in badminton games to improve players' technical level and tactical awareness. Although a variety of computer-assisted technologies have been used to collect badminton data, manual labeling still requires manpower and time, especially badminton professionals for technical data identification. The research question focuses on how to combine motion analysis methods to develop a model that accurately identifies the flight trajectory and hitting posture of badminton. In the methods and steps, we use TrackNetV2 to track the coordinates of the ball, preprocess the coordinate points, collect them into training data, and perform CNN training to determine the flight trajectory of the badminton. At the same time, the batting posture recognition model uses YOLOv7 to capture players, uses OpenPose to detect the batsman's batting posture, creates multiple training data based on four different postures, and conducts CNN training to compare the accuracy of different training data. , to find the best model. Ultimately, we can help players and coaches make more informed decisions during the game by identifying ball path model predictions and strategic suggestions. It is expected that the results of this topic will not only help badminton competition, but also bring new inspiration to the field of sports science.

Keywords : TrackNetV2, CNN training, YOLOv7, OpenPose

致謝

時光飛逝，歲月如梭。為期一年的畢業專題也過去一半的時間了，首先感謝指導老師林朝興博士這半年年下來的細心指導，從文獻研讀開始慢慢帶領我們進入研究的世界中。並在過程中協助我們度過研究的瓶頸；在研究設備上，也會盡可能地讓我們擁有最大的運算資源，以達到最好的實驗成效。

感謝所有教導過我們的老師，有你們的教導才能讓我們擁有更多作專題的基礎。感謝同學在我們遇到問題時給予一些意見。感謝淑音姊在專題這條路上的行政作業給予我們幫助。

李崇楷 吳定昌 陳柏舟 謹致

國立臺南大學資訊工程學系

中華民國 113 年 1 月

目錄

摘要	i
ABSTRACT	ii
致謝	iii
目錄	iv
表目錄	v
圖目錄	vi
一、緒論	1
1.1 研究動機	1
1.2 研究目的	1
1.3 分工表	1
二、方法與流程	2
2.1 數據蒐集與處理	2
2.2 飛行軌跡的球種判斷模型設計與訓練	2
2.3 擊球姿態的辨識模型設計與訓練	9
三、實驗成果	17
3.1 實驗環境	17
3.2 飛行軌跡的球種判斷模型成果	18
3.3 擊球姿態的辨識模型成果	19
四、結論與未來展望	22
4.1 結論	22
4.1.1 飛行軌跡的球種判斷模型	22
4.1.2 擊球姿態的辨識模型	22
4.2 未來展望	23
參考文獻	25
附錄 1 專題簡報	26

表目錄

表 1 飛行軌跡的球種判斷模型實驗環境.....	17
表 2 擊球姿態的辨識模型實驗環境.....	17
表 3 PlayerA 打給 PlayerB 回球訓練成果.....	18
表 4 PlayerB 打給 PlayerA 回球訓練成果.....	18
表 5 PlayerA 姿態辨識的訓練成果.....	19
表 6 PlayerB 姿態辨識的訓練成果.....	19
表 7 PlayerA 姿態辨識訓練成果(訓練資料:二值圖).....	20
表 8 PlayerA 姿態辨識訓練成果(訓練資料:標準化+Z-score 的向量)	20
表 9 PlayerB 姿態辨識訓練成果(訓練資料:二值圖).....	20
表 10 PlayerB 姿態辨識訓練成果(訓練資料:標準化+Z-score 的向量).....	21

圖目錄

圖 1 飛行軌跡的球種判斷模型流程圖	3
圖 2 每一幀球的座標位置	3
圖 3 經過 denoise 後的座標位置	3
圖 4 擊球事件偵測原理示意圖	4
圖 5 發球判斷 PlayerA 跟 PlayerB 的 y 值大小差異	6
圖 6 平滑前與平滑後結果	6
圖 7 擊球姿態的辨識模型流程圖	9
圖 8 一幀擊球事件的照片	10
圖 9 經由 YOLO7 偵測擊球事件的選手	10
圖 10 裁剪後的擊球選手	11
圖 11 OpenPose 的 25 個關節點	11
圖 12 二值化前的擊球選手	11
圖 13 二值化後的擊球選手	11
圖 14 向量的位置示意圖	12
圖 15 訓練資料為原始圖片的 CNN 架構	15
圖 16 訓練資料為二值圖的 CNN 架構	15
圖 17 測試多種優化器的準確率和損失函數圖表	16
圖 18 訓練資料為向量的 CNN 架構	16

一、緒論

1.1 研究動機

人類對於體育分析的興趣已促使相關分析系統的發展。考慮到運動表現分析的重要性和其廣泛的應用潛力，本研究旨在開發一個精確的辨識羽毛球飛行軌跡的球種判斷模型和擊球姿態辨識模型，以提升競技水平並指導訓練實踐。

1.2 研究目的

本研究關注的問題集中於羽毛球運動中的以下方面：如何結合運動分析的方法，開發一個能準確預測羽毛球飛行軌跡的模型？該模型如何提供實時的球路預測和策略建議，以協助選手和教練在比賽中做出更為明智的決策？為了解答上述問題，本研究將著手於設計和研究一個羽毛球路徑分析系統，該系統能夠收集並處理大量的運動數據，並通過機器學習和運動生物力學方法的結合，提供個性化的訓練指導和戰術建議。此外，本研究還將探索將此類分析方法應用於其他運動領域的可能性，從而拓寬運動表現分析的應用範疇。最終的研究目的是提升選手的技術水平、戰術意識和競技成績，同時也推進運動科學的進一步發展和普及。

1.3 分工表

	負責項目	負責比例
吳定昌	還原模型、數據統計、數據研究、計畫進度、飛行軌跡的球種判斷模型	30%
李崇楷	還原模型、數據統計、數據研究、原理閱讀、資料搜查、擊球姿態的辨識模型	40%
陳柏舟	還原模型、數據統計、數據研究、資料撰寫、飛行軌跡的球種判斷模型	30%

表 1 人員分工表

二、方法與流程

本研究將採用實驗和理論分析相結合的方法，來實現羽毛球球路路徑預測。

以下是該研究的主要方法與流程概述：

2.1 數據蒐集與處理

本研究將收集羽毛球比賽中的相關數據，其中包括選手的動作軌跡、球的位置和速度等資訊。這些數據可透過輸入資料中獲得，之後將進行數據清理、去噪以及填補缺失值等數據處理步驟，以確保數據的質量與完整性，進而提升後續分析的準確性。

2.2 飛行軌跡的球種判斷模型設計與訓練

基於 TracknetV2 模型的框架，本研究獲得了羽球在影片中每一幀的座標位置。隨後，對這些數據進行了仔細的清理、去噪和缺失值填補。由於每一段球軌跡均呈現拋物線形狀，注意到存在 y 軸的最大值點，利用這種慣性特徵設計了一個 event detection (擊球事件檢測) 模型。

最終，本研究以各種球種的特性為基礎，建立了一個卷積神經網絡 (CNN) 模型，用於訓練和球種判斷。這一方法結合了對飛行軌跡的動態特性和每種球種的獨特特徵的深入理解，旨在實現高效且準確的球種分類。該方法不僅在數據處理方面進行了細致的優化，還採用了先進的事件檢測技術，為羽毛球運動中的球種識別提供了一個創新而可靠的解決方案。

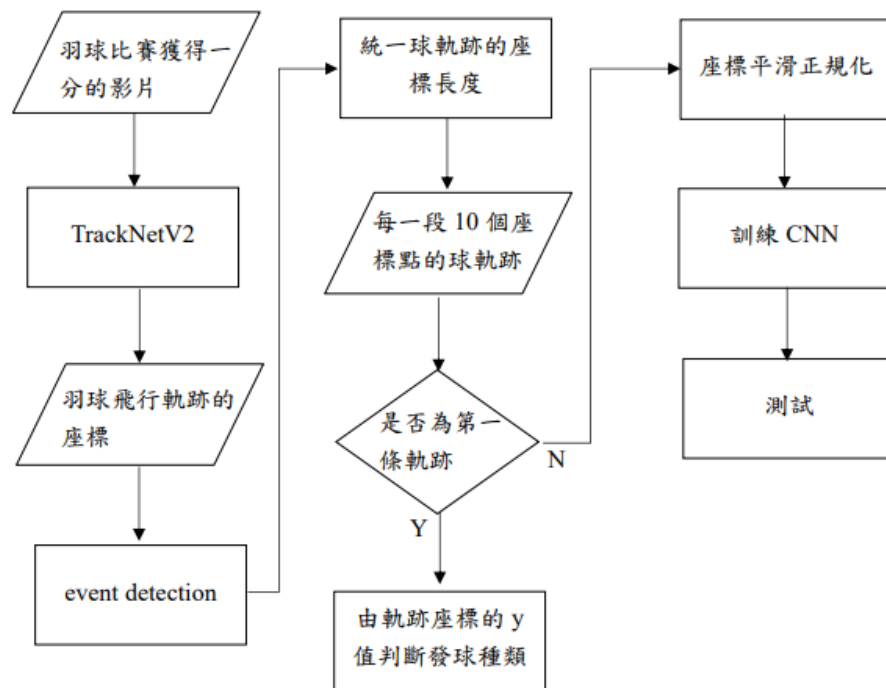


圖 1 飛行軌跡的球種判斷模型流程圖

表 1 TrackNetV2 的輸出結果

Frame	Visibility	X	Y	Time
2	0	0	0	00:00.1
3	0	0	0	00:00.1
4	0	0	0	00:00.1
5	1	623	468	00:00.2
6	0	0	0	00:00.2
7	1	620	476	00:00.2
8	1	620	475	00:00.3
9	1	618	472	00:00.3
10	0	0	0	00:00.3
11	0	0	0	00:00.4
12	0	0	0	00:00.4
13	1	601	411	00:00.4
14	1	602	408	00:00.5
15	1	598	390	00:00.5
16	1	596	372	00:00.5
17	0	0	0	00:00.6
18	0	0	0	00:00.6
19	0	0	0	00:00.6
20	0	0	0	00:00.7
21	1	580	323	00:00.7
22	1	582	317	00:00.7
23	0	0	0	00:00.8
24	0	0	0	00:00.8
25	0	0	0	00:00.8
26	1	575	307	00:00.9
27	0	0	0	00:00.9
28	0	0	0	00:00.9
29	1	571	311	00:01.0
30	0	0	0	00:01.0
31	0	0	0	00:01.0
32	1	568	317	00:01.1

圖2 每一幀球的座標位置

Frame	Visibility	X	Y	Time
2	0	0	0	00:00.1
3	0	0	0	00:00.1
4	0	0	0	00:00.1
5	1	623	468	00:00.2
6	0	621.5	472	00:00.2
7	1	620	476	00:00.2
8	1	620	475	00:00.3
9	1	618	472	00:00.3
10	0	0	0	00:00.3
11	0	0	0	00:00.4
12	0	0	0	00:00.4
13	1	0	0	00:00.4
14	1	0	0	00:00.5
15	1	0	0	00:00.5
16	1	0	0	00:00.5
17	0	0	0	00:00.6
18	0	0	0	00:00.6
19	0	0	0	00:00.6
20	0	0	0	00:00.7
21	1	580	323	00:00.7
22	1	582	317	00:00.7
23	0	0	0	00:00.8
24	0	0	0	00:00.8
25	0	0	0	00:00.8
26	1	575	307	00:00.9
27	0	0	0	00:00.9
28	0	0	0	00:00.9
29	1	571	311	00:01.0
30	0	0	0	00:01.0
31	0	0	0	00:01.0
32	1	568	317	00:01.1

圖3 經過denoise後的座標位置

denoise 原理：透過序列性地採用七個有效（非零）座標點，提出一種拋物線方程式的建立方法，用以推斷應填補的座標或需要移除的雜訊。此方法致力於有效處理拋物線軌跡資料，以提高數據的整體品質和精確性。本研究旨在透過數學建模和序列分析，為去噪過程提供更具科學基礎的框架，以增強對於座標點間相關性的理解，進而實現更可靠的雜訊去除和資料填補。

event detection (擊球事件檢測) 原理：本研究採用了一種基於 5 個連續區間的方法，以尋找 y 軸峰值來判斷羽球擊球時機點。這個方法旨在透過定義合適的區間，精確地捕捉拋物線運動中 y 軸的高峰值，進而識別擊球事件。這種事件檢測策略以一種數學上的局部極值分析為基礎，提供了一種有效的方式來確定羽球運動中的關鍵時刻，進而深化對擊球時機點的理解。該方法結合了區間設計和峰值檢測的技巧，為擊球事件檢測提供了一個更加科學和可靠的方法。

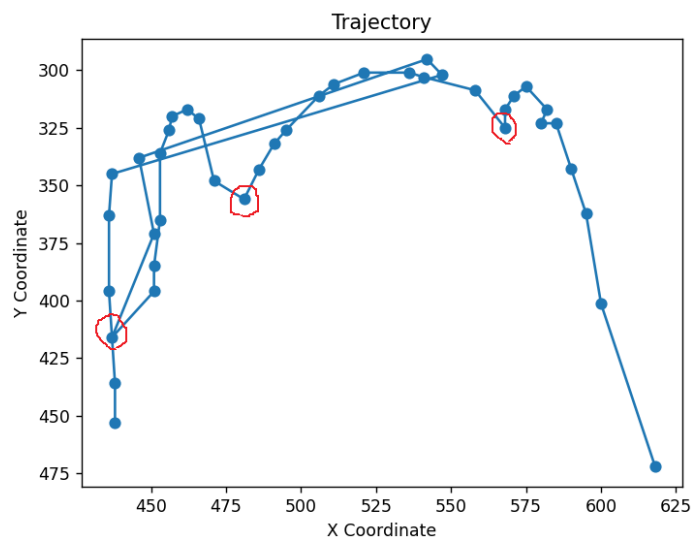


圖 4 擊球事件偵測原理示意圖

座標統一化方法：由於本研究欲將資料納入卷積神經網絡（CNN）進行判斷，必須確保球軌跡的座標數量一致。為實現此目的，採取了一種平均取樣的方法，將每一段軌跡均勻地轉換為包含 10 個座標點的統一形式。這個座標統一化的策略旨在確保每個軌跡都具有相同的數據結構，以有序且規律的方式輸入 CNN 進行後續的球種判斷，以確保 CNN 的有效運作。底下是飛行軌跡座標數量統一及落地判斷演算法：

考慮一段飛行軌跡，其座標數為 N 。本研究遵循以下步驟進行座標數量統一和球落地判斷：

移除內容為(0.0, 0.0)的座標，獲得 A 個有效座標。

若 A 大於等於 10：

- a. 選取頭尾各一座標，存入 `selected_coordinates` 陣列，剩下 $X = A - 2$ 個座標。
- b. 計算欲刪除的座標數，即 $(X \bmod 8)$ 。
- c. 計算新的座標數， $B = X / (X \bmod 8)$ 。
- d. 在原始座標陣列中將 B 的倍數的索引放入 `err` 陣列，表示需刪除的座標。
- e. 計算新的座標數， $C = X - (X \bmod 8)$ 。
- f. 計算新的座標數， $D = C / 8$ 。
- g. 在原始座標陣列中將 D 的倍數的索引放入 `selected_coordinates` 陣列。

若 A 不足 10：

輸出"座標不足 10 個"。

從 10 個座標中計算每一相鄰座標點之間的距離，並總和。若總和小於 10，表示球已落地。根據球是否已落地，決定是否保留該飛行軌跡。若球已落地，則移除相應的座標點。這一演算法結合了座標數量統一和球落地判斷，以提高飛行軌跡資料的一致性和實際應用性。

發球判斷：Player A(靠近鏡頭的選手)回擊給 Player B(離鏡頭較遠的選手)與 Player B(離鏡頭較遠的選手)回擊給 Player A(靠近鏡頭的選手)的差別在於：從 TracknetV2 偵測出的球座標中可以得知 y_1 (第一個座標點的 y 值)以及 y_{10} (第十個座標點的 y 值)，因此可以透過 y_1 、 y_{10} 的大小來判斷：如果 $y_1 < y_{10}$ 則為 Player B 回擊給 Player A； $y_1 > y_{10}$ 則為 Player A 回擊給 Player B。



圖 5 發球判斷 PlayerA 跟 PlayerB 的 y 值大小差異

平滑技術：本研究採取移動平均法，此方法對於每個座標的 $x(y)$ 值，取窗口中所有座標的 $x(y)$ 值之和，再除以窗口大小(本實驗使用 3)，窗口大小是一個由使用者定義的參數，根據數據特性進行調整以達到適當的平滑效果，若平滑程度太強烈，會造成球軌跡過於完美，會導致 CNN 訓練泛化不足。

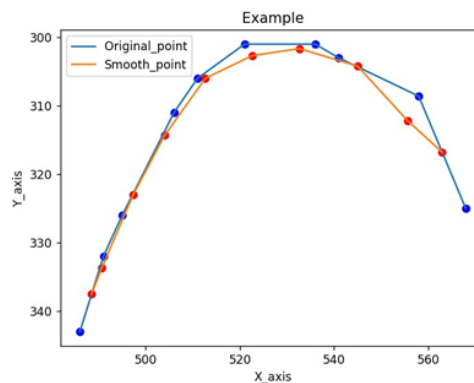


圖 6 平滑前與平滑後結果

標準化技術：本研究影片的解析度 1280*720，因此將 10 個球軌跡座標點的 x 除以 1280，再將 y 除以 720，可以將結果的範圍訂於[0,1]之間。主要辨識路徑分為：

PlayerA to PlayerB	Player B to Player A
Straight_short	
Straight_cut	
Straight_lift	
Straight_push	
Straight_long	
Cross_short	
Cross_cut	
Cross_lift	
Cross_push	
Cross_long	

表 2 主要回球辨識路徑

底下是主要回球的各特徵說明：

- straight 與 cross 分別為直線與斜線：差別在於 x1(第一個座標點的值)與 x10(第十個座標點的 y 值)有很大的落差。
- short 為短球：特性在於球軌跡弧度很小，max_y(y 最大值)與 min_y(y 最小值)落差不會很大，擊球與落點位置皆位於前場。
- cut 為切球：切球的球軌跡弧度也很小，但在 y1(第一個座標點的 y 值)與 y10(第十個座標點的 y 值)卻有著比較大的落差。擊球位置與落點位置分別為後場、前場。
- lift 為挑球：特性接近長球，與長球不同的是挑球為前場擊到後場，y1(第一個座標點的 y 值)與 y10(第十個座標點的 y 值)的落差不像長球這麼多。
- push 為推撲球：推撲球在這幾種球類中，弧度可以說是最小的，擊球與落地位

置分別為前場與後場。

- long 為長球：長球在所有球種中特性是最為突出的，其中 max_y 與 min_y 之間的落差最為巨大，時常會有球飛出鏡頭的狀況產生，所以球軌跡不為拋物線的時候，高機率為長球。

CNN 訓練流程&架構：將已標準化的向量送入 CNN 模型訓練，進而判斷出回球球種。用於十類標籤分類的全連接神經網路包括了輸入層、兩個密集層、兩個防止過擬合的丟棄層，最後是一個十單元的輸出層。

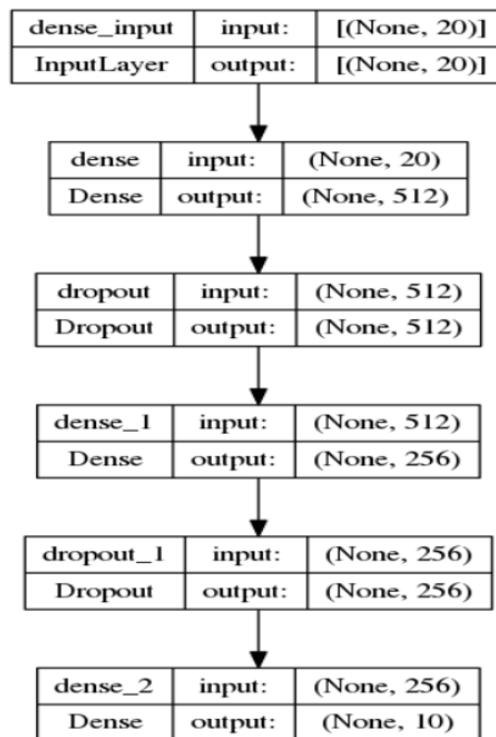


圖 7 CNN 訓練架構

2.3 擊球姿態的辨識模型設計與訓練

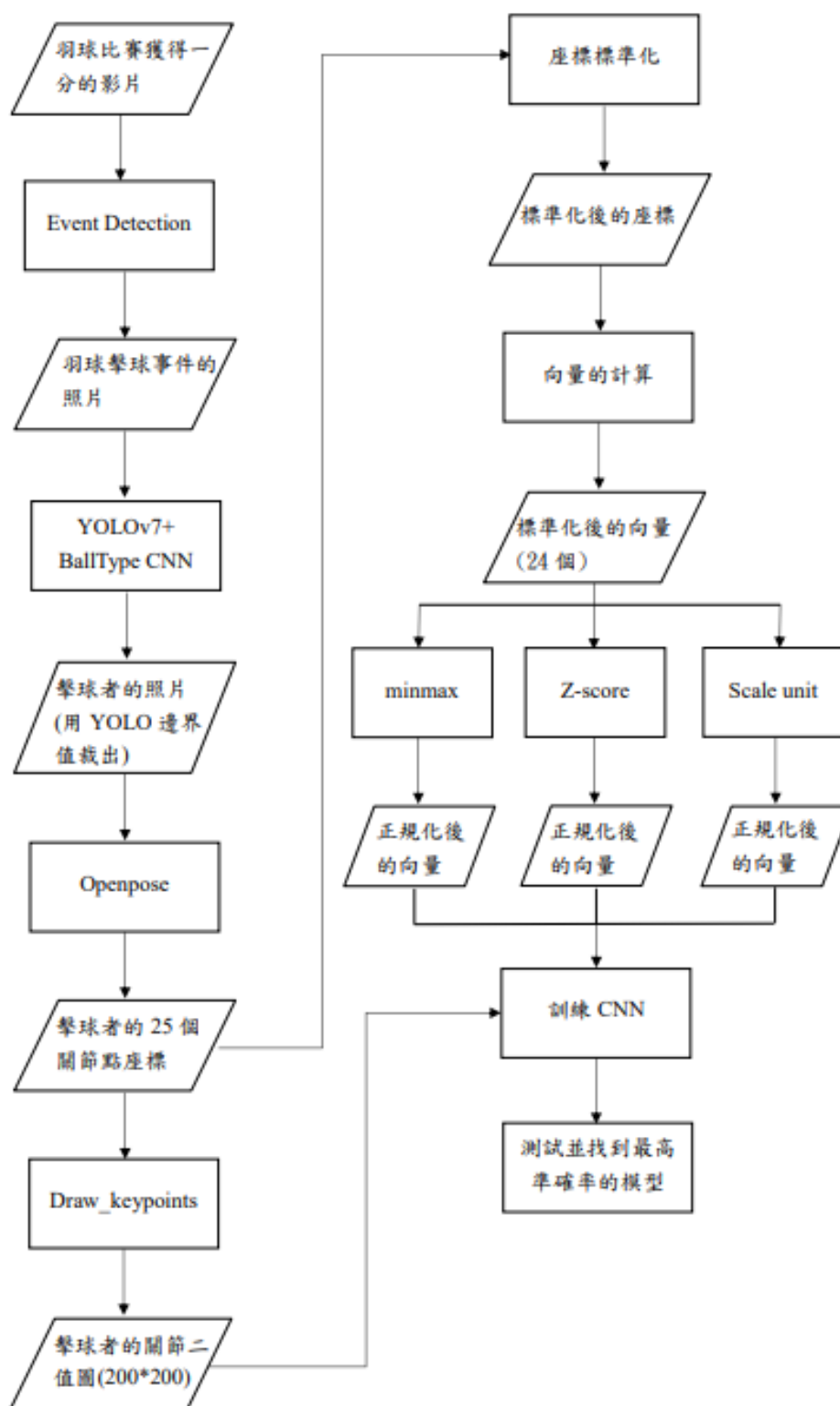


圖 7 擊球姿態的辨識模型流程圖

擊球姿態的辨識是羽球比賽分析的關鍵部分，主要涵蓋四種基本動作：反手擊球（Backhand）、吊球（Netdrop）、繞頭擊球（Roundhead）以及殺球（Smash）。在進行擊球姿態的辨識模型訓練過程中，首先透過事件檢測（Event Detection）技術來識別比賽影片中的關鍵擊球動作。

Event Detection 的原理和使用方式是根據輸出擊球事件的幀數（HitFrame）來裁剪 720x1280 解析度、30fps 的羽球比賽得分影片，製作成多幀的擊球事件照片，如圖 8。然後使用經過訓練的 YOLOv7 模型來辨識場上的選手，如圖 9，區分出是離鏡頭較近的選手（Player A）還是離鏡頭較遠的選手（Player B）。隨後，運用 Balltype CNN 來確定擊球事件照片中的擊球選手。在此過程中，YOLOv7 提供的邊界值（Bounding Box Values）能確定擊球選手在每一幀照片中的確切位置和範圍。具體來說，這些邊界值由四個部分構成：選手的中心點 X 座標、中心點 Y 座標、選手寬度、以及選手高度，所有這些都是相對於整個照片畫面的比例。這些精確的空間定位數據能夠只針對擊球選手進行裁剪，如圖 10，從而專注於分析擊球選手的擊球姿態，確保擊球姿態分析的準確性和有效性。



圖8 一幀擊球事件的照片



圖9 經由YOLO7偵測擊球事件的選手



圖10 裁剪後的擊球選手

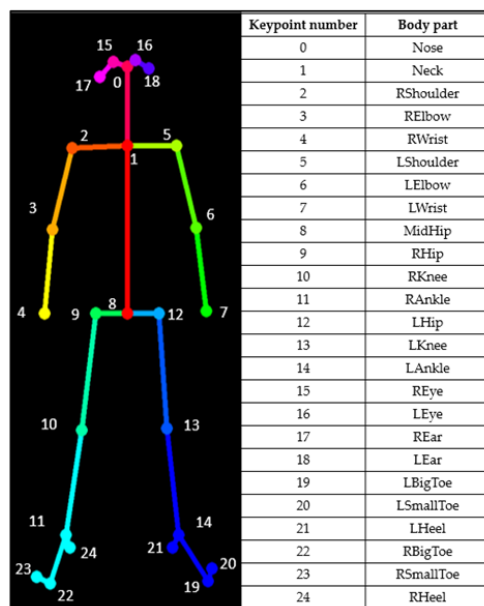


圖11 OpenPose的25個關節點

接著，運用 OpenPose 模型來捕捉裁剪後的擊球選手的關節點資訊。OpenPose 能夠識別出人體的 25 個關節點，如上圖 11。這些關節點的資訊被用於創建兩種不同的訓練資料：骨架二值圖和向量。

骨架二值圖：首先將 OpenPose 檢測到的第八個關節點 (Hip) 平移至一個 200x200 像素的圖片的中心 (100, 100 的位置上)，隨後將其餘關節點按相同的距離平移。這些平移後的點被用來繪製二值圖，即在每個關節點的位置上繪製一個點。如下圖 13。



圖12 二值化前的擊球選手

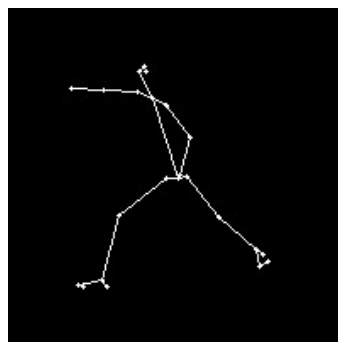


圖13 二值化後的擊球選手

向量：由於 OpenPose 的座標系統是基於裁剪後的照片尺寸，因此需要對座標進行標準化。通過將 x 座標除以圖片的長度，y 座標除以圖片的寬度來實現這一點。然後計算關節點之間的向量，利用一對關節點的座標來生成向量，如下圖 14，其中每個向量代表了兩個關節點間的相對位置和方向。具體而言，每個向量是通過減去一對關節點的座標值來計算的，使用 y 值較大的點座標(即是圖片中較下面的點)減去 y 值較小(即是圖片中較上面的點)的點座標。這些向量被定義如下：

$v1=(x17-x15, y17-y15); v2=(x18-x16, y18-y16); v3=(x0-x15, y0-y15);$
 $v4=(x0-x16, y0-y16);$
 等等以此類推...

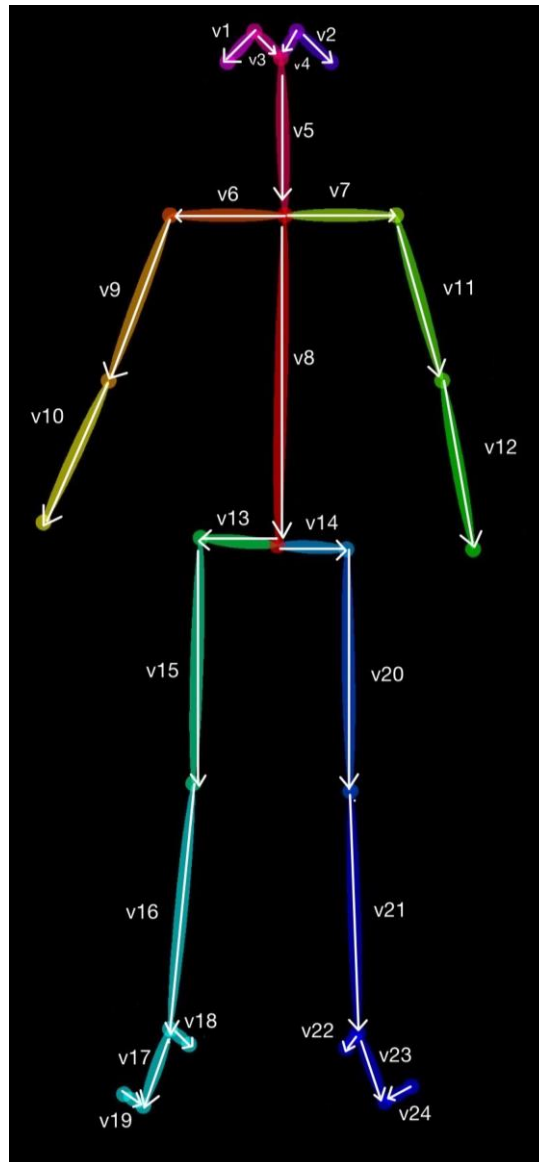


圖 14 向量的位置示意圖

在這裡， x 和 y 代表關節點在圖片中的座標位置，而數字標籤（如 15、16、17 等）則指的是 OpenPose 定義的特定關節點。通過這種方法，本實驗可以捕捉到每個關節點之間的相對運動和方向性。然而，在實際應用中，OpenPose 模型有時無法偵測出人體的完整 25 個關節點。在這種情況下，若計算向量時遇到關節中的其中一點為原點（即未被偵測到的點），則該向量將被視為 0 向量。因此，在進行向量正規化時，本實驗只針對非 0 向量進行處理。本實驗採用了三種主要的正規化技術來處理從 OpenPose 模型中提取的人體骨架向量數據，以增強擊球姿態辨識模型的性能和準確性。這些正規化方法如下：

1. Min-Max 正規化：此方法將每個特徵線性縮放到一個指定的範圍，通常是 $[0, 1]$ 或 $[-1, 1]$ ，本實驗是規範在 $[0, 1]$ 中。透過這種正規化，所有特徵的尺度被統一化，使得模型在訓練過程中不會因為特徵間尺度差異過大而產生偏差。對於人體骨架向量來說，Min-Max 正規化確保了不同身體部位和動作之間的比較是在相同的尺度下進行，從而提高了擊球動作辨識的精確度。演算法如下：

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

其中： X_{norm} 是正規化後的值、 X 是原始數據、

X_{min} 和 X_{max} 分別是數據集中該特徵的最小值和最大值。

2. Z-Score 標準化：經過這種標準化的數據會轉換為平均值為 0，標準差為 1 的分佈，即呈現標準正態分佈。對於人體骨架向量，這種方法有助於減少不同個體之間的尺寸差異，使得模型更能集中於學習動作模式的通用特徵，而不是被個體間的尺寸差異所干擾。演算法如下：

$$Z = \frac{X - \mu}{\sigma}$$

其中： Z 是標準化後的值、 X 是原始數據、 μ 是數據的平均值、

σ 是數據的標準差。

3. 單位長度標準化 (Scale to unit)：此方法將向量的長度標準化為 1，保持其方向不變。這在處理人體骨架向量時非常有用，因為它使得分析可以專注於向量的方向特徵，這對於理解特定的擊球姿態和動作特點尤為重要。通過這種標準化，模型可以更有效地識別和區分不同類型的擊球姿態。演算法如下：

$$V_{norm} = \frac{V}{||V||}$$

其中： V_{norm} 是標準化後的向量、 V 是原始向量、 $||V||$ 是原始向量的長度。

最後，本實驗使用了六種的訓練資料以及兩種專門設計的卷積神經網絡 (CNN) 架構來識別羽球運動員的擊球姿態。這些訓練資料分別是：裁剪後的擊球選手原始圖片 (200x200 解析度)、二值化圖片 (200x200 解析度)，原始的標準化向量座標以及三種正規化技術處理過的標準化向量座標。每種資料都獨立訓練並評估，以尋找最優的模型配置。以下是 CNN 的架構說明：

1. 圖片的 CNN 架構：對於圖片資料，本專題構建了一個先進的深度卷積神經網絡 (CNN)，它由多個卷積層 (Conv2D) 和最大池化層 (MaxPooling2D) 組成。這些層是以一種特定的順序排列的，旨在有效地從圖像中提取特徵。在我們的模型中，卷積層用於識別圖像中的各種細節和特徵，而池化層則用於減少參數的數量並防止過擬合，從而提高模型的泛化能力。此外，本實驗的輸入層設計是用於處理兩種不同類型的圖像：三通道的 RGB 彩色圖像和單通道的灰階圖像，分別具有 (200, 200, 3) 和 (200, 200, 1) 的輸入形狀。這種靈活的設計使得 CNN 可以有效地處理和分析不同類型的視覺資訊，並適用於多標籤分類任務。在優化器的選擇上，採用了 Adam 優化器，這是一種自適應學習速率的優化算法，它結合了 AdaGrad 和 RMSProp 優化算法的優點。這使得模型在學習過程中更加穩定和高效。具體而言，本實驗設定的學習速率為 0.001，這是一個比較謹慎的值，目的是在學習過程中避免過大的步伐，從而避免錯過最優解。塗在損失函數方面，採用了二元交叉熵損失函數 (binary_crossentropy)。這種損失函數適合於處理多標籤分類問題，這在案例中是識別不同的羽毛球

2. 向量的 CNN 架構：對於向量數據，採用了另一種 CNN 架構，如下圖 18，這是一種針對向量輸入優化的全連接網絡。它由一個輸入層開始，接著是兩個密集層，每個密集層後都有一個丟棄層以防止過擬合。該網絡最終通過一個四單元的輸出層來進行多標籤分類，該層能夠對標準化的向量數據進行精準的分類。在試過多種優化器之後，如下圖 17，最後選擇結果較優的 Adam 優化器。具體而言，本實驗設定的學習速率為 0.001，這是一個比較謹慎的值，目的是在學習過程中避免過大的步伐，從而避免錯過最優解。在損失函數方面，一樣採用了二元交叉熵損失函數（binary_crossentropy）。

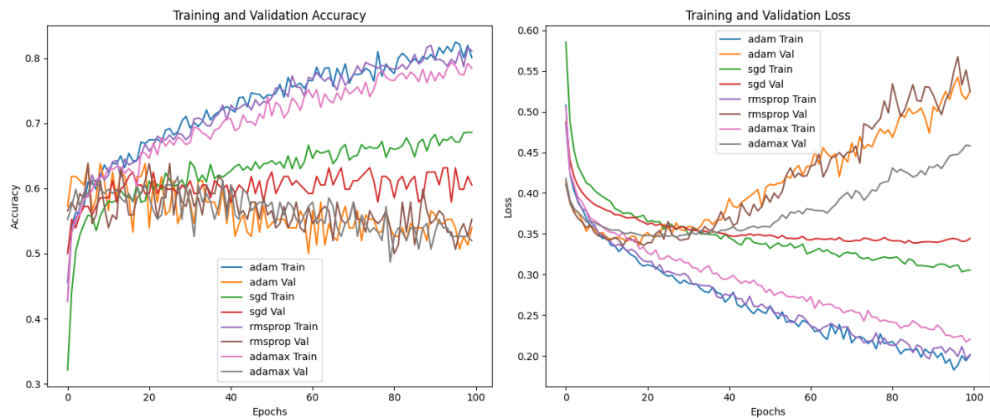


圖 17 測試多種優化器的準確率和損失函數圖表

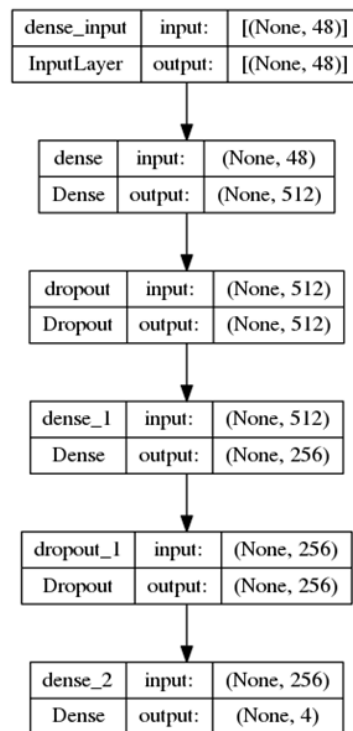


圖 18 訓練資料為向量的 CNN 架構

三、實驗成果

3.1 實驗環境

本研究後用 Python 語言做開發，並引入 TensorFlow 深度學習函式庫，由於深度學習網路所需計算量相當大，本研究使用 NVIDIA RTX3090 顯卡進行 GPU 加速，詳如實驗環境如下表 1、2。

表 1 飛行軌跡的球種判斷模型實驗環境

硬體	軟體
CPU:12th Gen Intel(R) Core(TM) i5-1235U	作業系統：Windows 11 家用版
GPU:Tesla T4 GPU	開發軟體：Python 3.9
RAM : 8GB	開發函式庫：tensorflow 2.15.0
	CUDA 版本：CUDA 12.2

表 2 擊球姿態的辨識模型實驗環境

硬體	軟體
CPU:Intel(R) Core(TM) i5-10500 CPU	作業系統：Ubuntu 16.04.7 LTS
RAM : 7.6GB	開發軟體：Python 3.9
	開發函式庫：tensorflow 2.13.0
	CUDA 版本：CUDA 11.2

3.2 飛行軌跡的球種判斷模型成果

表 3 PlayerA 打給 PlayerB 回球訓練成果

Precision	標準化+minmax	Precision	標準化+minmax
Straight_short	80.27%	Cross_short	84.95%
Straight_push	70.64%	Cross_push	88.12%
Straight_long	85.22%	Cross_long	59.45%
Straight_lift	72.31%	Cross_lift	85.34%
Straight_cut	61.25%	Cross_cut	62.93%
Total accuracy	73.93%	Total accuracy	76.15%

表 4 PlayerB 打給 PlayerA 回球訓練成果

Precision	標準化+minmax	Precision	標準化+minmax
Straight_short	88.49%	Cross_short	85.76%
Straight_push	80.31%	Cross_push	54.23%
Straight_long	84.21%	Cross_long	83.19%
Straight_lift	82.87%	Cross_lift	74.03%
Straight_cut	84.64%	Cross_cut	82.34%
Total accuracy	84.10%	Total accuracy	75.91%

3.3 擊球姿態的辨識模型成果

表 5 PlayerA 姿態辨識的訓練成果

Precision (%)	擊球選手 原始圖片	擊球選手 二值圖	標準化 向量	標準化+ Minmax 向量	標準化+ Z-score 向量	標準化+ ScaleUnit 向量
Backhand	41.5	65.22	73.86	60.44	78.84	81.85
Netdrop	52.84	42.62	39.73	48.42	50.88	34.39
Roundhead	71	62.86	76.31	76.31	72.72	81.57
Smash	48.64	59.12	47.02	40	63.85	49.18
Total accuracy	53.50	57.455	59.23	56.29	66.57	61.75

表 6 PlayerB 姿態辨識的訓練成果

Precision (%)	擊球選手 原始圖片	擊球選手 二值圖	標準化 向量	標準化+ Minmax 向量	標準化+ Z-score 向量	標準化+ ScaleUnit 向量
Backhand	50.13	71.86	73.30	80.59	78.84	70.06%
Netdrop	59.02	57.04	56.42	54.22	54.88	62.01%
Roundhead	66.66	21.43	86.36	68.18	72.72	63.63%
Smash	57.14	79.82	56.62	56.62	63.85	55.41%
Total accuracy	58.24	57.54	68.18	64.90	67.57	62.78%

表 7 PlayerA 姿態辨識訓練成果(訓練資料:二值圖)

Precision (%)	訓練次數 500	訓練次數 100	訓練次數 20	Best (取Val_loss最小的)
Backhand	77.94	65.31	16.52	65.22
Netdrop	25.93	31.51	50	42.62
Roundhead	40	62.86	31.43	62.86
Smash	79.01	65.75	85.58	59.12

表 8 PlayerA 姿態辨識訓練成果(訓練資料:標準化+Z-score 的向量)

Precision (%)	訓練次數 500	訓練次數 100	訓練次數 20	Best (取Val_loss最小的)
Backhand	65.23	65.4	78.92	78.84
Netdrop	62.56	62.78	54.55	54.88
Roundhead	36.36	40.90%	54.54	72.72
Smash	61.44	54.21	65.06	63.85

表 9 PlayerB 姿態辨識訓練成果(訓練資料:二值圖)

Precision (%)	訓練次數 500	訓練次數 100	訓練次數 20	Best (取Val_loss最小的)
Backhand	71.86	73.95	23.1	71.86
Netdrop	53.91	42.96	85.2	57.04
Roundhead	17.86	14.29	0	21.43
Smash	83.33	84.21	73.68	79.82

表 10 PlayerB 姿態辨識訓練成果(訓練資料:標準化+Z-score 的向量)

Precision (%)	訓練次數 500	訓練次數 100	訓練次數 20	Best (取Val_loss最小的)
Backhand	65.23	65.4	78.92	78.84
Netdrop	62.56	62.78	54.55	54.88
Roundhead	36.36	40.9	54.54	72.72
Smash	61.44	54.21	65.06	63.85

在本實驗中，模型的準確率是通過評估模型對於測試數據集中各個擊球動作的預測機率來計算的。具體而言，對於每一個動作類別，模型會輸出一個機率值。若這個機率值大於 0.5，則認為模型預測該動作發生。準確率的計算將這些正確預測的個數與測試數據集中總樣本數進行比較，即：

$$\text{準確率} = \frac{\text{正確預測的樣本數}}{\text{測試數據集的總樣本數}}$$

此方法確保了評估的客觀性和準確性，並使本實驗能夠量化模型在羽球擊球姿態識別任務上的表現效能。

四、結論與未來展望

4.1 結論

4.1.1 飛行軌跡的球種判斷模型

現階段的 TrackNet 技術，可能會因為場地、球網、選手的白衣、白色鞋子、白色球拍，會產生雜訊，所以在測試影片需要篩選過，準確率會比較高。另外，對於球種的判斷，event detection 判斷擊球的時間點現在是以「擊球位置會在每球軌跡的最低點」來判斷，導致像殺球以及平球這類型非拋物線軌跡的球種，就比較無法準確判斷。

目前 PlayerA 打給 PlayerB 的切球準確率相較於 PlayerB 打給 PlayerA 較低一些，可能是因為照相機是以俯拍的角度去紀錄，導致 PlayerA 的身體會去擋住球路，部分球路軌跡可能會因此沒有去擷取到，因此可能要將發力大小也考慮進去向量裡面，在測試選手手上穿戴感測器，去記錄發力大小。

4.1.2 擊球姿態的辨識模型

根據表 5 和表 6 的數據分析，觀察到在進行 OpenPose 動作特徵向量提取後，對於反手擊球（Backhand）與繞頭擊球（Roundhead）進行標準化及 Z-score 正規化處理，這些動作的準確率實現了顯著提升。此提升的可能原因是預處理步驟增強了模型對運動特徵的辨識能力。具體而言，在原始圖像中，關鍵的擊球動作特徵可能會受到球拍或其他障礙物的遮擋，而 OpenPose 所提取的特徵向量則能夠更穩定且準確地描繪出擊球姿勢的關鍵特點。

在吊球（Netdrop）的情況中，發現預處理後的準確率並未顯示出顯著提升。這可能是由於吊球動作的多樣性與複雜性，導致關鍵特徵的捕捉成為一項挑戰。此外，吊球動作的視覺微妙差異可能不足以為模型提供清晰的辨識路徑，且預處理階段可能未能充分針對這些變異進行調整。

從表 7、8 和表 9、10 的結果來看，在識別殺球（Smash）動作時，使用二值

化圖片的方法比 OpenPose 提取的特徵向量進行標準化和正規化後的準確率更高。這可能源於殺球動作中手臂揮動和身體姿態的明顯性，在二值化圖像中這些特徵更為突出，從而便於模型識別。相比之下，OpenPose 提取的特徵向量在這種情形下可能未能提供增加識別能力的額外資訊。

綜合所有擊球動作類型，應用標準化加 Z-score 正規化向量的方法所得到的整體準確率最高，達到 66.57%。這一結果表明，Z-score 正規化在整體上對於提升模型識別羽球擊球動作的準確性具有積極作用。推測其效果之所以顯著，可能是因為 Z-score 正規化通過將數據標準化到統一尺度，減少了特徵之間的變異，從而優化了模型學習與預測的過程。

4.2 未來展望

在未來的研究中，能夠期望通過以下途徑進一步增強飛行軌跡的球種判斷準確性和可靠性：

當前的研究主要集中於利用球員的動作和拍擊軌跡來辨識羽毛球的不同球種。然而，原本的飛行軌跡的球種判斷僅依賴於飛行軌跡的分析，並不能準確地區分這些球種。因此，本研究提出一種創新的方法，整合感應器技術和先進的深度學習模型。

此方法的核心在於高度精準和實時地分析球員的動作特徵，並結合球的飛行軌跡。通過這種整合，可以更有效地識別不同的球種，特別是在區分平球和殺球方面。未來的研究將專注於提高這一系統的準確性和實時性能，以便更廣泛地應用於羽毛球訓練和比賽場景。

在未來的研究中，能夠期望通過以下途徑進一步增強羽球運動員擊球姿態識別的準確性和可靠性：

1. 運用 3D 模型技術辨識選手姿勢：為了提升擊球姿態辨識的維度和精確度，採用創新的三維人體姿態模型技術。此技術將利用從多視角收集的數據，使得識別系統能從不同方向準確捕捉運動員的動態信息。藉由三維空間數據的整合，能夠提供更加

豐富的空間特徵，以優化模型對於擊球姿態的識別效能和整體的預測準確率。

2. 影片連續性動作的優化：現行的研究主要聚焦於單一幀圖像的分析。未來，將探索將連續的影片序列作為訓練資料的可能性。這將包括分析擊球前的準備姿勢與擊球後的完成動作，從而為模型提供一個時間連續性的上下文。這種連續性分析有望使模型更準確地理解和識別羽球動作的全過程，從而提升姿態識別的全面性和準確性。

參考文獻

- [1] Roboflow Universe Dataset. (2023). Retrieved from <https://universe.roboflow.com/>
- [2] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. GitHub Repository. Retrieved from <https://github.com/CMU-Perceptual-Computing-Lab/openpose>
- [3] Wong, K. Y. (2023). YOLOv7: YOLOv7: Real-Time Object Detection. GitHub Repository. Retrieved from <https://github.com/WongKinYiu/yolov7>
- [4] Aidea. (2023). Topic: Object Detection and Classification on Aidea Platform. Retrieved from <https://aidea-web.tw/topic/cbea66cc-a993-4be8-933d-1aa9779001f8>
- [5] H. Chen, H. Chen, M. Hsiao, W. Tsai, and S. Lee, "A trajectory-based ball tracking framework with visual enrichment for broadcast baseball videos," *Journal of information Science and Engineering*, vol. 24, no. 1, pp. 143-157, Jan. 2008, doi: 10.6688/JISE.2008.24.1.10
- [6] M. Cordeiro, L. Melo Junior, J. Fernandes de Macêdo, T. Coelho da Silva, and N. A. de Freitas, "Using Deep Learning for Trajectory Classification," *The International FLAIRS Conference Proceedings*, vol.34, Apr. 2021, doi: 10.32473/flairs.v34i1.128368.
- [7] N. Sun et al., "TrackNetV2: Efficient Shuttlecock Tracking Network," presented at the 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), pp. 86-91, Dec 2020, doi: 10.1109/ICPAI51961.2020.00023.
- [8] Y. Huang, I. Liao, C. Chen, T. İk, and W. Peng, "TrackNet: a deep learning network for tracking high-speed and tiny objects in sports applications," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1-8, Sep. 2019, doi: 10.1109/AVSS.2019.8909871.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137-1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [10] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390-391, Jun. 2020, doi: 10.1109/CVPRW50498.2020.00203.

附錄 1 專題簡報

113 TH
NUTN CSIE

AI 人工智慧影片內容分析應用
以羽球競賽對戰策略分析為例

AI Video Content Analysis Application
Case Study: Badminton Match Strategy Analysis

第16組 吳定昌、李崇楷、陳柏舟



Abstract

本研究旨在運用人工智慧 (AI) 技術於羽球比賽，以提升選手的技術水平和戰術意識。儘管已使用多種電腦輔助技術收集羽球數據，但手動標記仍需人力與時間，尤其在技術資料識別方面需要羽球專業人員。透過本專題，我們致力於開發高辨識率的羽毛球飛行軌跡模型和擊球姿態辨識模型，使得大量羽球數據的收集變得可行，推動羽球技戰術分析在科研和實際應用上更普及。

1

TABLE OF CONTENTS	
01 ————— 動機與問題 Motives and Problems	03 ————— 實驗結果 Experimental results
02 ————— 研究方法與步驟 Research methods and steps	04 ————— 結論與未來展望 Conclusion and future prospects

2

01

動機與問題

Motives and Problems

3



研究動機

人類對於體育分析的興趣已促使相關分析系統的發展。考慮到運動表現分析的重要性和其廣泛的應用潛力，本研究旨在開發一個精確的辨識羽毛球飛行球路模型和擊球姿態辨識模型，以提升競技水平並指導訓練實踐。

4

研究問題



如何結合運動分析的方法，開發一個能準確辨識羽毛球飛行球路的模型？



如何提供實時球路預測和策略建議，協助選手教練在比賽中做出更為明智的決策？

本研究將設計一個**辨識羽毛球飛行球路系統**，該系統能夠收集並處理運動數據，並通過機器學習和運動生物力學方法的結合，提供個性化的訓練指導和戰術建議。

最終的研究目的是提升選手的技術水平、戰術意識和競技成績，同時也推進運動科學的進一步發展和普及。

5

02

研究方法與步驟

Research methods and steps

6

建構模型

🔊 飛行軌跡的球種判斷



👁️ 擊球姿態的辨識



7

飛行軌跡的 球種判斷



8

飛行軌跡的球種判斷流程圖



9

TrackNet



輸入資料為選手獲得一分的影片 經過TrackNet產出由紅點追蹤球的影片

10

TrackNet



輸入資料為選手獲得一分的影片
經過TrackNet產出每一帧球的座標位置的CSV檔。

Frame	Visibility	X	Y	Time
2	0	0	0	00:00.1
3	0	0	0	00:00.1
4	0	0	0	00:00.1
5	1	623	466	00:00.2
6	0	0	0	00:00.2
7	1	620	475	00:00.2
8	1	620	475	00:00.3
9	1	618	472	00:00.3
10	0	0	0	00:00.3
11	0	0	0	00:00.4
12	0	0	0	00:00.4
13	1	621	471	00:00.4
14	1	622	468	00:00.5
15	1	598	390	00:00.5
16	1	596	372	00:00.5
17	0	0	0	00:00.6
18	0	0	0	00:00.6
19	0	0	0	00:00.6
20	0	0	0	00:00.7
21	1	590	323	00:00.7
22	1	592	317	00:00.7
23	0	0	0	00:00.8
24	0	0	0	00:00.8
25	0	0	0	00:00.8
26	1	575	307	00:00.9
27	0	0	0	00:00.9
28	0	0	0	00:00.9
29	1	571	311	00:01.0
30	0	0	0	00:01.0
31	0	0	0	00:01.0
32	1	568	317	00:01.1

CSV檔經過denoise
將模糊的球做填補座標
的動作

Frame	Visibility	X	Y	Time
2	0	0	0	00:00.1
3	0	0	0	00:00.1
4	0	0	0	00:00.1
5	1	623	466	00:00.2
6	0	0	0	00:00.2
7	1	620	475	00:00.2
8	1	620	475	00:00.3
9	1	618	472	00:00.3
10	0	0	0	00:00.3
11	0	0	0	00:00.4
12	0	0	0	00:00.4
13	1	621	471	00:00.4
14	1	622	468	00:00.5
15	1	598	390	00:00.5
16	1	596	372	00:00.5
17	0	0	0	00:00.6
18	0	0	0	00:00.6
19	0	0	0	00:00.6
20	0	0	0	00:00.7
21	1	590	323	00:00.7
22	1	592	317	00:00.7
23	0	0	0	00:00.8
24	0	0	0	00:00.8
25	0	0	0	00:00.8
26	1	575	307	00:00.9
27	0	0	0	00:00.9
28	0	0	0	00:00.9
29	1	571	311	00:01.0
30	0	0	0	00:01.0
31	0	0	0	00:01.0
32	1	568	317	00:01.1

1

step	Variable X	Y	Time	error
2	0	0	0.0001	0
3	0	0	0.0001	0
4	0	0	0.0001	0
5	1	623	450	0.0002
6	0	52-5	472	0.0002
7	1	620	472	0.0002
8	1	620	475	0.0003
9	1	618	472	0.0003
10	0	0	0.0003	0
11	0	0	0.0004	0
12	0	0	0.0004	0
13	0	0	0.0004	0
14	0	0	0.0005	0
15	1	0	0.0005	0
16	0	0	0.0005	0
17	0	0	0.0006	0
18	0	0	0.0006	0
19	0	0	0.0006	0
20	0	0	0.0006	0
21	1	380	325	0.0007
22	2	562	317	0.0007
23	0	0	0.0008	0
24	0	0	0.0008	0
25	0	0	0.0008	0
26	1	525	307	0.0009
27	0	0	0.0009	0
28	0	0	0.0009	0
29	571	311	0.0010	0
30	0	0	0.0010	0
31	0	0	0.0010	0
32	1	568	317	0.0010

將資料整理成每拍的軌跡，方便在測試時觀察

618.0, 472.0), (609.0, 401.42), (595.0, 362.21), (590.0, 342.61), (585.0, 323.0), (580.0, 323.0), (582.0, 317.0), (575.0, 307.0), (571.0, 311.0), (568.0, 317.0), (568.0, 325.0), (558.0, 308.67), (541.0, 303.0), (536.0, 301.0), (521.0, 301.0), (511.0, 306.0), (506.0, 311.0), (495.0, 326.0), (491.0, 332.0), (486.0, 343.0), (481.0, 356.0), (471.0, 348.0), (466.0, 321.0), (462.0, 317.0), (457.0, 320.0), (456.0, 326.0), (453.0, 336.0), (453.0, 365.0), (451.0, 385.0), (451.0, 396.0), (437.0, 416.0), (451.0, 371.0), (446.0, 338.0), (542.0, 295.18), (547.0, 302.0), (437.0, 345.0), (436.0, 363.0), (436.0, 366.0), (438.0, 462.0), (438.0, 451.0)



統一球軌跡的座標數量的演算法

```
# 確定要刪除的座標數
to_remove = X % 8
# 計算 B 並找到要刪除的索引
err.extend(range(0, X, to_remove))
# 計算 D 並找到要保留的索引
selected_coordinates.extend(remaining_coordinates[i
* 8] for i in range(X // 8))
else:
print("座標不足10個")
return []
return selected_coordinates
```

0 1 2 3 4	21 個座標
5 6 7 8 9	21-2=19 個座標
10 11 12 13 14	19/8=2.3 需要刪除 3 個座標
15 16 17 18 19	Floor(19/8)=6 刪除 6 個座標
20	19-3=16 個座標
	16/8=2 刪除 2 個座標

(範例)

額外，從10個座標中計算出每一段點到點的距離，
加起來小於10表示球已落地，以此來刪除不必要的飛行軌跡

15

發球球種判斷

我們知道在選手獲得一分的過程中，第一條球軌跡必為發球。

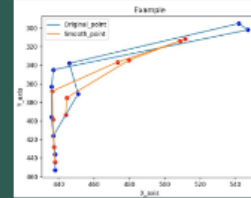
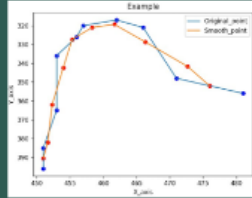
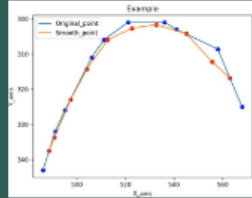
`(118.0, 477.0), (109.0, 471.0), (105.0, 462.0), (102.0, 451.0), (100.0, 439.0), (100.0, 427.0), (101.0, 415.0), (102.0, 403.0), (104.0, 391.0), (107.0, 379.0), (112.0, 367.0), (118.0, 355.0), (126.0, 343.0), (136.0, 331.0), (147.0, 319.0), (159.0, 307.0), (172.0, 295.0), (186.0, 283.0), (201.0, 271.0), (217.0, 259.0), (234.0, 247.0), (252.0, 235.0), (271.0, 223.0), (291.0, 211.0), (312.0, 199.0), (334.0, 187.0), (357.0, 175.0), (381.0, 163.0), (406.0, 151.0), (432.0, 139.0), (459.0, 127.0), (487.0, 115.0), (516.0, 103.0), (546.0, 91.0), (577.0, 79.0), (609.0, 67.0), (642.0, 55.0), (676.0, 43.0), (711.0, 31.0), (747.0, 19.0), (784.0, 7.0), (822.0, 0.0), (861.0, 0.0), (901.0, 0.0), (942.0, 0.0), (984.0, 0.0), (1027.0, 0.0), (1071.0, 0.0), (1116.0, 0.0), (1162.0, 0.0), (1209.0, 0.0), (1257.0, 0.0), (1306.0, 0.0), (1356.0, 0.0), (1407.0, 0.0), (1459.0, 0.0), (1512.0, 0.0), (1566.0, 0.0), (1621.0, 0.0), (1677.0, 0.0), (1734.0, 0.0), (1792.0, 0.0), (1851.0, 0.0), (1911.0, 0.0), (1972.0, 0.0), (2034.0, 0.0), (2097.0, 0.0), (2161.0, 0.0), (2226.0, 0.0), (2292.0, 0.0), (2359.0, 0.0), (2427.0, 0.0), (2496.0, 0.0), (2566.0, 0.0), (2637.0, 0.0), (2709.0, 0.0), (2782.0, 0.0), (2856.0, 0.0), (2931.0, 0.0), (3007.0, 0.0), (3084.0, 0.0), (3162.0, 0.0), (3241.0, 0.0), (3321.0, 0.0), (3402.0, 0.0), (3484.0, 0.0), (3567.0, 0.0), (3651.0, 0.0), (3736.0, 0.0), (3822.0, 0.0), (3909.0, 0.0), (3997.0, 0.0), (4086.0, 0.0), (4176.0, 0.0), (4267.0, 0.0), (4359.0, 0.0), (4452.0, 0.0), (4546.0, 0.0), (4641.0, 0.0), (4737.0, 0.0), (4834.0, 0.0), (4932.0, 0.0), (5031.0, 0.0), (5131.0, 0.0), (5232.0, 0.0), (5334.0, 0.0), (5437.0, 0.0), (5541.0, 0.0), (5646.0, 0.0), (5752.0, 0.0), (5859.0, 0.0), (5967.0, 0.0), (6076.0, 0.0), (6186.0, 0.0), (6297.0, 0.0), (6409.0, 0.0), (6522.0, 0.0), (6636.0, 0.0), (6751.0, 0.0), (6867.0, 0.0), (6984.0, 0.0), (7102.0, 0.0), (7221.0, 0.0), (7341.0, 0.0), (7462.0, 0.0), (7584.0, 0.0), (7707.0, 0.0), (7831.0, 0.0), (7956.0, 0.0), (8082.0, 0.0), (8209.0, 0.0), (8337.0, 0.0), (8466.0, 0.0), (8596.0, 0.0), (8727.0, 0.0), (8859.0, 0.0), (8992.0, 0.0), (9126.0, 0.0), (9261.0, 0.0), (9397.0, 0.0), (9534.0, 0.0), (9672.0, 0.0), (9811.0, 0.0), (9951.0, 0.0), (10092.0, 0.0), (10234.0, 0.0), (10377.0, 0.0), (10521.0, 0.0), (10666.0, 0.0), (10812.0, 0.0), (10959.0, 0.0), (11107.0, 0.0), (11256.0, 0.0), (11406.0, 0.0), (11557.0, 0.0), (11709.0, 0.0), (11862.0, 0.0), (12016.0, 0.0), (12171.0, 0.0), (12327.0, 0.0), (12484.0, 0.0), (12642.0, 0.0), (12801.0, 0.0), (12961.0, 0.0), (13122.0, 0.0), (13284.0, 0.0), (13447.0, 0.0), (13611.0, 0.0), (13776.0, 0.0), (13942.0, 0.0), (14109.0, 0.0), (14277.0, 0.0), (14446.0, 0.0), (14616.0, 0.0), (14787.0, 0.0), (14959.0, 0.0), (15132.0, 0.0), (15306.0, 0.0), (15481.0, 0.0), (15657.0, 0.0), (15834.0, 0.0), (16012.0, 0.0), (16191.0, 0.0), (16371.0, 0.0), (16552.0, 0.0), (16734.0, 0.0), (16917.0, 0.0), (17101.0, 0.0), (17286.0, 0.0), (17472.0, 0.0), (17659.0, 0.0), (17847.0, 0.0), (18036.0, 0.0), (18226.0, 0.0), (18417.0, 0.0), (18609.0, 0.0), (18802.0, 0.0), (18996.0, 0.0), (19191.0, 0.0), (19387.0, 0.0), (19584.0, 0.0), (19782.0, 0.0), (19981.0, 0.0), (20181.0, 0.0), (20382.0, 0.0), (20584.0, 0.0), (20787.0, 0.0), (20991.0, 0.0), (21196.0, 0.0), (21402.0, 0.0), (21609.0, 0.0), (21817.0, 0.0), (22026.0, 0.0), (22236.0, 0.0), (22447.0, 0.0), (22659.0, 0.0), (22872.0, 0.0), (23086.0, 0.0), (23301.0, 0.0), (23517.0, 0.0), (23734.0, 0.0), (23952.0, 0.0), (24171.0, 0.0), (24391.0, 0.0), (24612.0, 0.0), (24834.0, 0.0), (25057.0, 0.0), (25281.0, 0.0), (25506.0, 0.0), (25732.0, 0.0), (25959.0, 0.0), (26187.0, 0.0), (26416.0, 0.0), (26646.0, 0.0), (26877.0, 0.0), (27109.0, 0.0), (27342.0, 0.0), (27576.0, 0.0), (27811.0, 0.0), (28047.0, 0.0), (28284.0, 0.0), (28522.0, 0.0), (28761.0, 0.0), (29001.0, 0.0), (29242.0, 0.0), (29484.0, 0.0), (29727.0, 0.0), (29971.0, 0.0), (30216.0, 0.0), (30462.0, 0.0), (30709.0, 0.0), (30957.0, 0.0), (31206.0, 0.0), (31456.0, 0.0), (31707.0, 0.0), (31959.0, 0.0), (32212.0, 0.0), (32466.0, 0.0), (32721.0, 0.0), (32977.0, 0.0), (33234.0, 0.0), (33492.0, 0.0), (33751.0, 0.0), (34011.0, 0.0), (34272.0, 0.0), (34534.0, 0.0), (34797.0, 0.0), (35061.0, 0.0), (35326.0, 0.0), (35592.0, 0.0), (35859.0, 0.0), (36127.0, 0.0), (36396.0, 0.0), (36666.0, 0.0), (36937.0, 0.0), (37209.0, 0.0), (37482.0, 0.0), (37756.0, 0.0), (38031.0, 0.0), (38307.0, 0.0), (38584.0, 0.0), (38862.0, 0.0), (39141.0, 0.0), (39421.0, 0.0), (39702.0, 0.0), (39984.0, 0.0), (40267.0, 0.0), (40551.0, 0.0), (40836.0, 0.0), (41122.0, 0.0), (41409.0, 0.0), (41697.0, 0.0), (41986.0, 0.0), (42276.0, 0.0), (42567.0, 0.0), (42859.0, 0.0), (43152.0, 0.0), (43446.0, 0.0), (43741.0, 0.0), (44037.0, 0.0), (44334.0, 0.0), (44632.0, 0.0), (44931.0, 0.0), (45231.0, 0.0), (45532.0, 0.0), (45834.0, 0.0), (46137.0, 0.0), (46441.0, 0.0), (46746.0, 0.0), (47052.0, 0.0), (47359.0, 0.0), (47667.0, 0.0), (47976.0, 0.0), (48286.0, 0.0), (48597.0, 0.0), (48909.0, 0.0), (49222.0, 0.0), (49536.0, 0.0), (49851.0, 0.0), (50167.0, 0.0), (50484.0, 0.0), (50802.0, 0.0), (51121.0, 0.0), (51441.0, 0.0), (51762.0, 0.0), (52084.0, 0.0), (52407.0, 0.0), (52731.0, 0.0), (53056.0, 0.0), (53382.0, 0.0), (53709.0, 0.0), (54037.0, 0.0), (54366.0, 0.0), (54696.0, 0.0), (55027.0, 0.0), (55359.0, 0.0), (55692.0, 0.0), (56026.0, 0.0), (56361.0, 0.0), (56697.0, 0.0), (57034.0, 0.0), (57372.0, 0.0), (57711.0, 0.0), (58051.0, 0.0), (58392.0, 0.0), (58734.0, 0.0), (59077.0, 0.0), (59421.0, 0.0), (59766.0, 0.0), (60112.0, 0.0), (60459.0, 0.0), (60807.0, 0.0), (61156.0, 0.0), (61506.0, 0.0), (61857.0, 0.0), (62209.0, 0.0), (62562.0, 0.0), (62916.0, 0.0), (63271.0, 0.0), (63627.0, 0.0), (63984.0, 0.0), (64342.0, 0.0), (64701.0, 0.0), (65061.0, 0.0), (65422.0, 0.0), (65784.0, 0.0), (66147.0, 0.0), (66511.0, 0.0), (66876.0, 0.0), (67242.0, 0.0), (67609.0, 0.0), (67977.0, 0.0), (68346.0, 0.0), (68716.0, 0.0), (69087.0, 0.0), (69459.0, 0.0), (69832.0, 0.0), (70206.0, 0.0), (70581.0, 0.0), (70957.0, 0.0), (71334.0, 0.0), (71712.0, 0.0), (72091.0, 0.0), (72471.0, 0.0), (72852.0, 0.0), (73234.0, 0.0), (73617.0, 0.0), (74001.0, 0.0), (74386.0, 0.0), (74772.0, 0.0), (75159.0, 0.0), (75547.0, 0.0), (75936.0, 0.0), (76326.0, 0.0), (76717.0, 0.0), (77109.0, 0.0), (77502.0, 0.0), (77896.0, 0.0), (78291.0, 0.0), (78687.0, 0.0), (79084.0, 0.0), (79482.0, 0.0), (79881.0, 0.0), (80281.0, 0.0), (80682.0, 0.0), (81084.0, 0.0), (81487.0, 0.0), (81891.0, 0.0), (82296.0, 0.0), (82702.0, 0.0), (83109.0, 0.0), (83517.0, 0.0), (83926.0, 0.0), (84336.0, 0.0), (84747.0, 0.0), (85159.0, 0.0), (85572.0, 0.0), (85986.0, 0.0), (86401.0, 0.0), (86817.0, 0.0), (87234.0, 0.0), (87652.0, 0.0), (88071.0, 0.0), (88491.0, 0.0), (88912.0, 0.0), (89334.0, 0.0), (89757.0, 0.0), (90181.0, 0.0), (90606.0, 0.0), (91032.0, 0.0), (91459.0, 0.0), (91887.0, 0.0), (92316.0, 0.0), (92746.0, 0.0), (93177.0, 0.0), (93609.0, 0.0), (94042.0, 0.0), (94476.0, 0.0), (94911.0, 0.0), (95347.0, 0.0), (95784.0, 0.0), (96222.0, 0.0), (96661.0, 0.0), (97101.0, 0.0), (97542.0, 0.0), (97984.0, 0.0), (98427.0, 0.0), (98871.0, 0.0), (99316.0, 0.0), (99762.0, 0.0), (100209.0, 0.0), (100657.0, 0.0), (101106.0, 0.0), (101556.0, 0.0), (102007.0, 0.0), (102459.0, 0.0), (102912.0, 0.0), (103366.0, 0.0), (103821.0, 0.0), (104277.0, 0.0), (104734.0, 0.0), (105192.0, 0.0), (105651.0, 0.0), (106111.0, 0.0), (106572.0, 0.0), (107034.0, 0.0), (107497.0, 0.0), (107961.0, 0.0), (108426.0, 0.0), (108892.0, 0.0), (109359.0, 0.0), (109827.0, 0.0), (110296.0, 0.0), (110766.0, 0.0), (111237.0, 0.0), (111709.0, 0.0), (112182.0, 0.0), (112656.0, 0.0), (113131.0, 0.0), (113607.0, 0.0), (114084.0, 0.0), (114562.0, 0.0), (115041.0, 0.0), (115521.0, 0.0), (116002.0, 0.0), (116484.0, 0.0), (116967.0, 0.0), (117451.0, 0.0), (117936.0, 0.0), (118422.0, 0.0), (118909.0, 0.0), (119397.0, 0.0), (119886.0, 0.0), (120376.0, 0.0), (120867.0, 0.0), (121359.0, 0.0), (121852.0, 0.0), (122346.0, 0.0), (122841.0, 0.0), (123337.0, 0.0), (123834.0, 0.0), (124332.0, 0.0), (124831.0, 0.0), (125331.0, 0.0), (125832.0, 0.0), (126334.0, 0.0), (126837.0, 0.0), (127341.0, 0.0), (127846.0, 0.0), (128352.0, 0.0), (128859.0, 0.0), (129367.0, 0.0), (129876.0, 0.0), (130386.0, 0.0), (130897.0, 0.0), (131409.0, 0.0), (131922.0, 0.0), (132436.0, 0.0), (132951.0, 0.0), (133467.0, 0.0), (133984.0, 0.0), (134502.0, 0.0), (135021.0, 0.0), (135541.0, 0.0), (136062.0, 0.0), (136584.0, 0.0), (137107.0, 0.0), (137631.0, 0.0), (138156.0, 0.0), (138682.0, 0.0), (139209.0, 0.0), (139737.0, 0.0), (140266.0, 0.0), (140796.0, 0.0), (141327.0, 0.0), (141859.0, 0.0), (142392.0, 0.0), (142926.0, 0.0), (143461.0, 0.0), (143997.0, 0.0), (144534.0, 0.0), (145072.0, 0.0), (145611.0, 0.0), (146151.0, 0.0), (146692.0, 0.0), (147234.0, 0.0), (147777.0, 0.0), (148321.0, 0.0), (148866.0, 0.0), (149412.0, 0.0), (149959.0, 0.0), (150507.0, 0.0), (151056.0, 0.0), (151606.0, 0.0), (152157.0, 0.0), (152709.0, 0.0), (153262.0, 0.0), (153816.0, 0.0), (154371.0, 0.0), (154927.0, 0.0), (155484.0, 0.0), (156042.0, 0.0), (156601.0, 0.0), (157161.0, 0.0), (157722.0, 0.0), (158284.0, 0.0), (158847.0, 0.0), (159411.0, 0.0), (160000.0, 0.0), (160589.0, 0.0), (161179.0, 0.0), (161770.0, 0.0), (162362.0, 0.0), (162955.0, 0.0), (163549.0, 0.0), (164144.0, 0.0), (164740.0, 0.0), (165337.0, 0.0), (165935.0, 0.0), (166534.0, 0.0), (167134.0, 0.0), (167735.0, 0.0), (168337.0, 0.0), (168940.0, 0.0), (169544.0, 0.0), (170149.0, 0.0), (170755.0, 0.0), (171362.0, 0.0), (171970.0, 0.0), (172579.0, 0.0), (173189.0, 0.0), (173799.0, 0.0), (174410.0, 0.0), (175022.0, 0.0), (175635.0, 0.0), (176249.0, 0.0), (176864.0, 0.0), (177480.0, 0.0), (178097.0, 0.0), (178715.0, 0.0), (179334.0, 0.0), (179954.0, 0.0), (180575.0, 0.0), (181197.0, 0.0), (181820.0, 0.0), (182443.0, 0.0), (183067.0, 0.0), (183692.0, 0.0), (184318.0, 0.0), (184945.0, 0.0), (185573.0, 0.0), (186202.0, 0.0), (186832.0, 0.0), (187463.0, 0.0), (188094.0, 0.0), (188726.0, 0.0), (189359.0, 0.0), (189993.0, 0.0), (190628.0, 0.0), (191263.0, 0.0), (191899.0, 0.0), (192536.0, 0.0), (193174.0, 0.0), (193812.0, 0.0), (194451.0, 0.0), (195091.0, 0.0), (195732.0, 0.0), (196374.0, 0.0), (197016.0, 0.0), (197659.0, 0.0), (198303.0, 0.0), (198948.0, 0.0), (199593.0, 0.0), (200239.0, 0.0), (200885.0, 0.0), (201532.0, 0.0), (202180.0, 0.0), (202828.0, 0.0), (203477.0, 0.0), (204127.0, 0.0), (204778.0, 0.0), (205429.0, 0.0), (206081.0, 0.0), (206733.0, 0.0), (207386.0, 0.0), (208040.0, 0.0), (208694.0, 0.0), (209349.0, 0.0), (210004.0, 0.0), (210660.0, 0.0), (211316.0, 0.0), (211973.0, 0.0), (212630.0, 0.0), (213288.0, 0.0), (213946.0, 0.0), (214605.0, 0.0), (215264.0, 0.0), (215924.0, 0.0), (216584.0, 0.0), (217245.0, 0.0), (217906.0, 0.0), (218568.0, 0.0), (219230.0, 0.0), (219892.0, 0.0), (220555.0, 0.0), (221218.0, 0.0), (221881.0, 0.0), (222545.0, 0.0), (223209.0, 0.0), (223874.0, 0.0), (224539.0, 0.0), (225204.0, 0.0), (225869.0, 0.0), (226534.0, 0.0), (227200.0, 0.0), (227865.0, 0.0), (228531.0, 0.0), (229197.0, 0.0), (229863.0, 0.0), (230530.0, 0.0), (231196.0, 0.0), (231863.0, 0.0), (232530.0, 0.0), (233197.0, 0.0), (233864.0, 0.0), (234531.0, 0.0), (235198.0, 0.0), (235865.0, 0.0), (236532.0, 0.0), (237200.0, 0.0), (237867.0, 0.0), (238534.0, 0.0), (239201.0, 0.0), (239868.0, 0.0), (240535.0, 0.0), (241202.0, 0.0), (241869.0, 0.0), (242536.0, 0.0), (243203.0, 0.0), (243870.0, 0.0), (244537.0, 0.0), (245204.0, 0.0), (245871.0, 0.0), (246538.0, 0.0), (247205.0, 0.0), (247872.0, 0.0), (248539.0, 0.0), (249206.0, 0.0), (249873.0, 0.0), (250540.0, 0.0), (251207.0, 0.0), (251874.0, 0.0), (252541.0, 0.0), (253208.0, 0.0), (253875.0, 0.0), (254542.0, 0.0), (255209.0, 0.0), (255876.0, 0.0), (256543.0, 0.0), (257209.0, 0.0), (257876.0, 0.0), (258543.0, 0.0), (259209.0, 0.0), (259876.0, 0.0), (260543.0, 0.0), (261209.0, 0.0), (261876.0, 0.0), (262543.0, 0.0), (263209.0, 0.0), (263876.0, 0.0), (264543.0, 0.0), (265209.0, 0.0), (265876.0, 0.0), (266543.0, 0.0), (267209.0, 0.0), (267876.0, 0.0), (268543.0, 0.0), (269209.0, 0.0), (269876.0, 0.0), (270543.0, 0.0), (271209.0, 0.0), (271876.0, 0.0), (272543.0, 0.0), (273209.0, 0.0), (273876.0, 0.0), (274543.0, 0.0), (275209.0, 0.0), (275876.0, 0.0), (276543.0, 0.0), (277209.0, 0.0), (277876.0, 0.0), (278543.0, 0.0), (279209.0, 0.0), (279876.0, 0.0), (280543.0, 0.0), (281209.0, 0.0), (281876.0, 0.0), (282543.0, 0.0), (283209.0, 0.0), (283876.0, 0.0), (284543.0, 0.0), (285209.0, 0.0), (285876.0, 0.0), (286543.0, 0.0), (287209.0, 0.0), (287876.0, 0.0), (288543.0, 0.0), (289209.0, 0.0), (289876.0, 0.0), (290543.0, 0.0), (291209.0, 0.0), (291876.0, 0.0), (292543.0, 0.0), (293209.0, 0.0), (293876.0, 0.0), (294543.0, 0.0), (295209.0, 0.0), (295876.0, 0.0), (296543.0, 0.0), (297209.0, 0.0), (297876.0, 0.0), (298543.0, 0.0), (299209.0, 0.0), (299876.0, 0.0), (300543.0, 0.0), (301209.0, 0.0), (301876.0, 0.0), (302543.0, 0.0), (303209.0, 0.0), (303876.0, 0.0), (304543.0, 0.0), (305209.0, 0.0), (305876.0, 0.0), (306543.0, 0.0), (307209.0, 0.0), (307876.0, 0.0), (308543.0, 0.0), (309209.0, 0.0), (309876.0, 0.0), (310543.0, 0.0), (311209.0, 0.0), (311876.0, 0.0), (312543.0, 0.0), (313209.0, 0.0), (313876.0, 0.0), (314543.0, 0.0), (315209.0, 0.0), (315876.0, 0.0), (316543.0, 0.0), (317209.0, 0.0), (317876.0, 0.0), (318543.0, 0.0), (319209.0, 0.0), (319876.0, 0.0), (320543.0, 0.0), (32`

平滑飛行軌跡座標點

(563.0,316.8),(555.7,312.2),(545.0,304.2),(532.7,301.7),(522.7,302.7),(512.7,306.0),(504.0,314.3),(497.3,323.0),(490.7,333.7),(488.5,337.5)
(476.0,352.0),(472.7,341.7),(466.3,328.7),(461.7,319.3),(458.3,321.0),(455.3,327.3),(454.0,342.3),(452.3,362.0),(451.7,382.0),(451.0,390.5)
(444.0,393.5),(444.7,375.0),(479.7,334.7),(511.7,311.7),(508.7,314.0),(473.3,336.7),(436.3,368.0),(436.7,398.3),(437.3,428.3),(438.0,444.5)

我們採取移動平均法，公式如下：

對於每個座標的 x (y) 值，取窗口中所有座標的 x (y) 值之和，再除以窗口大小(本實驗使用3)。
窗口大小是一個由使用者定義的參數，根據數據特性進行調整以達到適當的平滑效果。
若平滑程度太強烈，會造成球軌跡過於完美，會導致CNN訓練泛化不足。



18

正規化

(563.0,316.8),(555.7,312.2),(545.0,304.2),(532.7,301.7),(522.7,302.7),(512.7,306.0),(504.0,314.3),(497.3,323.0),(490.7,333.7),(488.5,337.5) 第二條
(476.0,352.0),(472.7,341.7),(466.3,328.7),(461.7,319.3),(458.3,321.0),(455.3,327.3),(454.0,342.3),(452.3,362.0),(451.7,382.0),(451.0,390.5) 第三條
(444.0,393.5),(444.7,375.0),(479.7,334.7),(511.7,311.7),(508.7,314.0),(473.3,336.7),(436.3,368.0),(436.7,398.3),(437.3,428.3),(438.0,444.5) 第四條

我們影片的解析度1280*720，因此我們將10個球軌跡座標點的x除以1280，再將y除以720。
可以將結果的範圍訂於[0,1]之間。

第二條標準化後的結果

0.35686275,0.45138889
0.54705882,0.42870833
0.53049216,0.47083333
0.52549020,0.11805556
0.51078431,0.41805556
0.50098039,0.42500000
0.49607843,0.43194444
0.48579417,0.45777778
0.48137225,0.46111111
0.47647059,0.47688889

第三條標準化後的結果

0.47156863,0.49444444
0.46176471,0.48333333
0.45686275,0.44583333
0.45294118,0.44077778
0.44803922,0.44444444
0.44705882,0.45277778
0.44411765,0.46666667
0.44111765,0.50694444
0.44215686,0.53422222
0.44215686,0.55000000

第四條標準化後的結果

0.42845137,0.57777778
0.44215686,0.51527778
0.43725190,0.46944444
0.53137225,0.40992222
0.53827451,0.41944444
0.42845137,0.43916667
0.42745098,0.50416667
0.42745098,0.55000000
0.42941176,0.60555556
0.42941176,0.62916667

將標準化後的結果

丟入CNN模型中去訓練

19

主要辨識的路徑

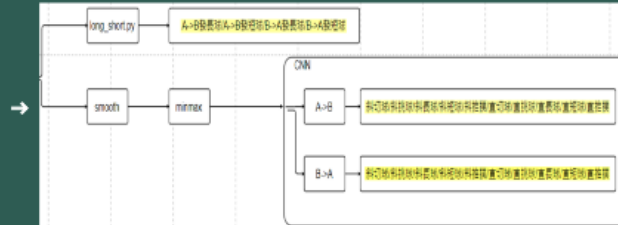
PlayerA to PlayerB	PlayerB to PlayerA
Straight_short	
Straight_cut	
Straight_lift	
Straight_push	
Straight_long	
Cross_short	
Cross_cut	
Cross_lift	
Cross_push	
Cross_long	

20

球種路徑的辨識流程圖

0.47156863,0.49444444
0.46176471,0.48333333
0.45686775,0.44583333
0.45294118,0.41027778
0.44803922,0.44444444
0.44705882,0.45277778
0.44411765,0.46666667
0.44411765,0.50694444
0.44215686,0.53472222
0.44315686,0.55000000

已正規化後
的向量結果

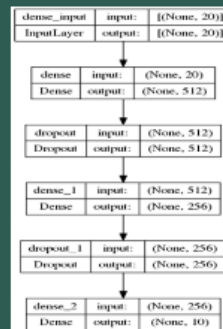


21

訓練CNN(向量)&CNN架構

0.55686275,0.45138889
0.54705882,0.42870833
0.53039216,0.42083333
0.52549020,0.41805556
0.51078431,0.41805556
0.50098039,0.42500000
0.49607843,0.43194444
0.48529412,0.45277778
0.48137255,0.46111111
0.47647059,0.47638889

輸入資料：
正規化後的向量結果



用於 1 類多標籤分類的全連接神經網路
包括了輸入層、兩個密集層、
兩個防止過擬合的丟棄層，
最後是一個十單元的輸出層。

22

擊球姿態 的辨識



23

主要辨識的動作



Backhand



Netdrop



Roundhead



Smash

24

擊球姿態的辨識流程圖



25

Event Detection



Badminton Scoring One Point Video

B	C
HitFrame	Hitler
5	B
14	A
27	B
54	A
80	B
104	A
155	B
168	A
189	B
201	A
213	B
234	A
255	B
282	A



Badminton IHitler Frame

輸入資料為選手獲得一分的影片 經過Event Detection產出由哪一位選手打到球目事件發生在哪一幀

26

Yolov7



輸入資料為選手打擊到球的那一個照片
經過Yolov7可以判斷出圖片中的選手



經由yolov7提供的邊界值可以對
有選手的畫面作裁剪

27

openpose



輸入資料為攻擊方選手
打擊到球的那一個照片

```
49.137889 92.491026 0.04077777
80.72321 49.02392 0.43469917
49.172875 44.709905 0.49416913
47.432602 41.744943 0.30442029
6.444884 46.434950 0.00486719
84.98358 92.98878 0.80655181
70.32354 92.95021 0.6411910
73.44242 52.22100 0.40566564
71.11202 95.35022 0.5750134
76.42943 84.18860 0.47019127
84.11044 124.10788 0.8183688
117.73808 237.4330 0.49918079
74.53117 58.425605 0.500000
16.126483 -17.72252 0.000004
26.42542 156.4711 0.4684251
0.0 0.0 0.0
91.42278 28.342592 0.118600138
0.0 0.0 0.0
22.37428 22.476255 0.47279227
13.514443 156.14638 0.42460743
18.892028 139.99892 0.8936862
19.833819 140.00083 0.84467358
119.129 147.85774 0.17002573
124.94238 144.41909 0.20222297
132.04918 146.11474 0.3846711
```

經由openpose可以提供
選手的骨架資訊以及
conf

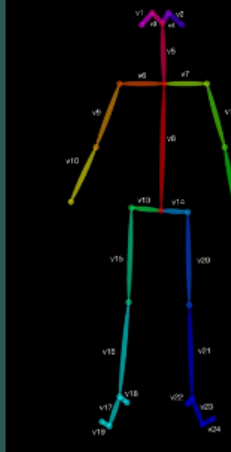
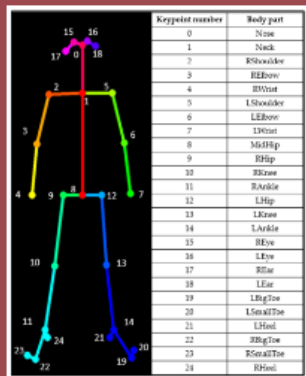
```
1.3904891212332346 1.289811810893822 0.30427579
0.45700004570004 0.37461628888762 0.40488961
0.33487787752042 0.234270747123212 0.4043021
0.232120559425244 0.242712424948224 0.30482008
0.037138448912351 0.137658812475121 0.3187737
0.47470124117947 0.134175 0.304331
0.378881124117947 0.402032357777888 0.401444
0.377408848483244 0.402032357777888 0.401444
0.12289602224944 0.12403179712408 0.470008
0.42138474747474 0.42138474747474 0.470008
0.708884708884708 0.468421279720207 0.423886
0.301121212121212 0.7788881848484 0.8942614
0.47000847000847 0.402032357777888 0.401444
0.20403423212121 0.40217040402121 0.800499
0.137881234117947 0.474804848484848 0.404823
0.0 0.0 0.0
0.374238484848484 0.118484848484848 0.118484848
0.0 0.0 0.0
0.100739240822024 0.100739240822024 0.47578217
0.404212121212121 0.404212121212121 0.404212121
0.121212121212121 0.808888888888889 0.404212121
0.121212121212121 0.808888888888889 0.404212121
0.870484848484848 0.404212121212121 0.17002573
0.810578 0.810578 0.810578 0.810578 0.810578
0.810578 0.810578 0.810578 0.810578 0.810578
```

標準化向量 = (畫面的高, 畫面的寬)

因openpose的座標系統是根據裁切
過的圖片大小, 所以須對座標作標
準化。

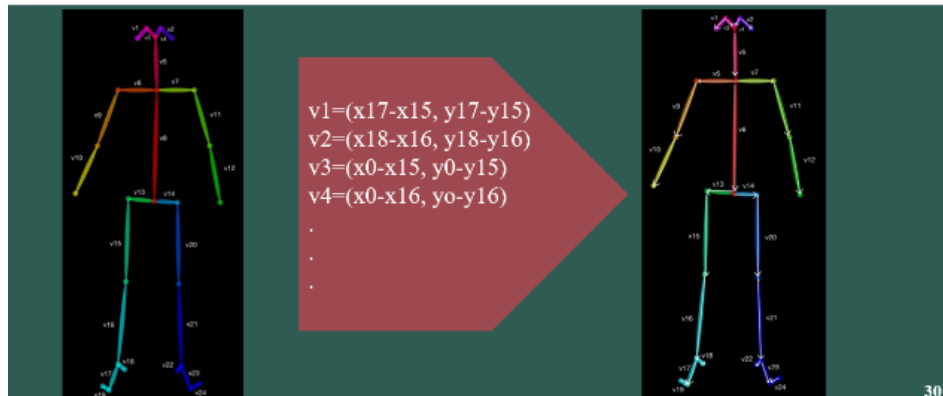
28

向量的計算



29

向量的計算



整理訓練資料(向量)



正規化方法說明



Min-Max 正規化是種資料正規化的方法，用於將每個特徵以線性方式縮放到一個指定的範圍內，通常是 [0, 1] 或 [-1, 1]，讓所有特徵具有相似的尺度，有助於更好地訓練模型。

$$z = \frac{x - \min(x)}{[\max(x) - \min(x)]}$$



資料經過 Z-Score 標準化後，將資料轉換為平均值為 0，標準差為 1，會呈現標準正態分佈 (高斯分布, Gaussian distribution)。

$$z = \frac{x - \text{mean}(x)}{\text{stdev}(x)}$$



Scale to unit

這個方法的目的是將向量的長度標準化為 1，從而保持其方向但改變其大小。這種方法在處理向量數據時非常有用，特別是在需要比較或分析不同向量方向而不是其實際大小時。

$$\text{normalized vector} = \frac{\text{vector}}{\|\text{vector}\|}$$

32

整理訓練資料(二值圖)

```
49.027415 32.457043 0.06421177
56.71811 49.027415 0.3246047
49.027415 44.702015 0.49516921
27.40666 43.734943 0.20442008
8.888483 42.294493 0.30438715
64.35204 52.39475 0.5943331
78.32234 71.59201 0.8431311
74.04843 68.21164 0.40565746
71.11231 95.15022 0.3750324
76.40961 94.1866 0.4701617
96.11104 119.19795 0.6193569
117.73806 137.41159 0.69919473
84.20117 95.62505 0.800045
38.538293 117.72551 0.808849
26.62562 135.6771 0.4868251
0.0 0.0 0.0
51.42275 29.741381 0.115608188
0.0 0.0 0.0
52.37432 32.476552 0.47570537
32.514843 106.55118 0.42480799
35.361032 158.94943 0.4515452
29.202875 140.90183 0.44487186
119.189 147.50374 0.17303513
124.44230 144.61969 0.2022297
122.04025 147.11424 0.35497117
```

Origin keypoints

將 OpenPose 檢測的第八點(Hip)平移到 200x200圖片的(100, 100)位置，並將其餘點按相同距離平移，然後在這些點上繪製二值圖。



origin



Monochromatic Skeletal

33

為什麼要先使用Yolov7而不直接使用Openpose?



34

為什麼要先使用Yolov7而不直接使用Openpose?



Openpose的模型無法辨識人物，所以需要先用yolov7

去辨識上下半場的選手，才能使用openpose

35

訓練資料

原始圖片
(調成200*200)



標準化加minmax
向量



二值化的圖片



標準化加Z-score
向量



標準化向量



標準化加scale to unit
向量

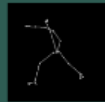


36

訓練CNN(圖片)



輸入資料為攻擊方選手打擊到球的那一幀照片



輸入資料為攻擊方選手打擊到球的那一幀照片在作二值化



03

實驗結果

Experimental results

39

PlayerA to PlayerB 直(斜)線回球

Threshold>0.5 precision	標準化+minmax	Threshold>0.5 precision	標準化+minmax
Straight_short	80.27%	Cross_short	84.95%
Straight_push	70.64%	Cross_push	88.12%
Straight_long	85.22%	Cross_long	59.45%
Straight_lift	72.31%	Cross_lift	85.34%
Straight_cut	61.25%	Cross_cut	62.93%
Total accuracy	73.93%	Total accuracy	76.15%

40

PlayerB to PlayerA 直(斜)線回球

Threshold>0.5 precision	標準化+minmax	Threshold>0.5 precision	標準化+minmax
Straight_short	88.49%	Cross_short	85.76%
Straight_push	80.31%	Cross_push	54.23%
Straight_long	84.21%	Cross_long	83.19%
Straight_lift	82.87%	Cross_lift	74.03%
Straight_cut	84.64%	Cross_cut	82.34%
Total accuracy	84.10%	Total accuracy	75.91%

41

PlayerA Posture Accuracy

Threshold >0.5 precision	badminton- pose- analysis	Origin picture	二值圖	標準化	標準化 +minmax	標準化 +Z-score	標準化 +Scale to unit
Backhand	5.08%	41.5%	65.22%	73.86%	60.44%	78.84%	81.85%
Netdrop	18.07%	52.84%	42.62%	39.73%	48.42%	50.88%	34.39%
Roundhead	--	71%	62.86%	76.31%	76.31%	72.72%	81.57%
Smash	78.59%	48.64%	59.12%	47.02%	40%	63.85%	49.18%
Total accuracy	33.91%	53.50%	57.455%	59.23%	56.29%	66.57%	61.75%

42

PlayerB Posture Accuracy

Threshold >0.5 precision	badminton- pose- analysis	Origin picture	二值圖	標準化	標準化 +minmax	標準化 +Z-score	標準化 +Scale to unit
Backhand	4.44%	50.13%	71.86%	73.3%	80.59%	78.84%	70.06%
Netdrop	2.1%	59.02%	57.04%	56.42%	54.22%	54.88%	62.01%
Roundhead	--	66.66%	21.43%	86.36%	68.18%	72.72%	63.63%
Smash	70.84%	57.14%	79.82%	56.62%	56.62%	63.85%	55.41%
Total accuracy	25.79%	58.24%	57.5375%	68.18%	64.90%	67.57%	62.78%

43

PlayerA Posture Accuracy

Precision (%)	Epoch 500	Epoch 100	Epoch 20	Best	Precision (%)	Epoch 500	Epoch 100	Epoch 20	Best
Backhand	77.94	65.31	16.52	65.22	Backhand	65.23	65.4	78.92	78.84
Netdrop	25.93	31.51	50	42.62	Netdrop	62.56	62.78	54.55	54.88
Roundhead	40	62.86	31.43	62.86	Roundhead	36.36	40.9%	54.54	72.72
Smash	79.01	65.75	85.58	59.12	Smash	61.44	54.21	65.06	63.85

Training data: 二值圖

Training data: 標準化+Z-score

44

PlayerB Posture Accuracy

Precision (%)	Epoch 500	Epoch 100	Epoch 20	Best	Precision (%)	Epoch 500	Epoch 100	Epoch 20	Best
Backhand	71.86	73.95	23.10	71.86	Backhand	65.23	65.4	78.92	78.84
Netdrop	53.91	42.96	85.20	57.04	Netdrop	62.56	62.78	54.55	54.88
Roundhead	17.86	14.29	0	21.43	Roundhead	36.36	40.9	54.54	72.72
Smash	83.33	84.21	73.68	79.82	Smash	61.44	54.21	65.06	63.85

Training data: 二值圖

Training data: 標準化+Z-score

45

04

結論與未來展望

Conclusion and future prospects

113 TH
NUTN CSIE

46

結論

現階段的TrackNet技術，可能會因為場地、球網、選手的白衣、白色鞋子、白色球拍，會產生雜訊，所以在測試影片需要篩選過，準確率會比較高。另外，對於球種的判斷，event detection判斷擊球的時間點現在是以「擊球位置會在每球軌跡的最低點」來判斷，導致像殺球以及平球這類型非拋物線軌跡的球種，就比較無法準確判斷。

47

結論



48

結論

目前PlayerA打給PlayerB的切球準確率相較於PlayerB打給PlayerA較低一些，可能是因為照相機是以俯拍的角度去紀錄，導致PlayerA的身體會去擋住球路，部分球路軌跡可能會因此沒有去擷取到，因此可能要將發力大小也考慮進去向量裡面，在測試選手手上穿戴感測器，去記錄發力大小。

49

結論



Roundhead: 高速揮拍下的照片無法正確的提取特徵
Backhand: 在反手擊球時可能會被身體或物體擋住，照片無法正確的提取特徵。

50

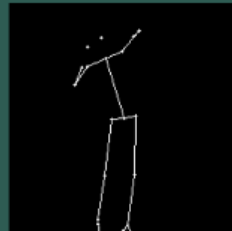
結論



吊球動作有多種變化形式，預處理過程難以有效捕捉到吊球動作的關鍵特性。

51

結論



殺球動作有更為明顯的手臂揮動和身體姿勢，這些特徵在二值化圖片中可能更加突出，使得模型更容易識別出殺球動作。

52

結論



綜合所有擊球動作，應用Z-score標準化後的總體準確率最高，達到66.57%。這顯示了Z-score標準化在整體上對於提高模型識別羽毛球擊球動作的準確性有正面影響，可能因為它能夠有效地將數據歸一化到同一尺度，減少了不同特徵間的變異性，從而幫助模型更好地學習和預測。

53

未來展望



運用3D模型技術辨識選手姿勢

應用先進的人體姿態3D模型技術，可以更全面且精確地辨識羽球選手的擊球姿勢。這包括運用多視角的資訊，使模型能夠在不同方向上更準確地捕捉選手的動作。透過整合3D技術，將能夠提供更豐富的空間信息，進一步改進擊球姿態辨識的效能你集準確率。



影片連續性動作的優化

將影片序列作為訓練資料，影片包括擊球前的起始姿勢和擊球後的收拍動作。這種方法可以更全面地考慮擊球前後的幀數變化，從而提供更精確的姿態辨識。

54