

# 靜 宜 大 學

## 資 訊 工 程 學 系

### 畢 業 專 題 成 果 報 告 書

#### 專 題 名 稱

智 慧 排 球 系 統：A I 姿 勢 分 析 與 排 球 競 技 遊 戲

學 生：

資工四 B 411147429 劉姿均

資工四 B 411134395 劉濤澄

資工四 B 411134515 蔡承儒

資工四 B 411147160 吳沛芸

資工四 B 411147568 溫建翔

指導教授：葉介山 教授

西 元 二〇二五 年 十 二 月

靜宜大學資訊工程學系  
專題實作授權同意書

本人具有著作財產權之論文全文資料，授予靜宜大學資工系，  
為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種  
方法重製，不限地域與時間，惟每人以一份為限。授權內容均無須訂  
立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本  
授權所為之收錄、重製、發行及學術研發利用均為無償。

指導教授 葉介山

學生簽名：劉姿均	學號：411147429	日期：西元 2025 年 12 月 01 日
學生簽名：劉浩澄	學號：411134395	日期：西元 2025 年 12 月 01 日
學生簽名：蔡承儒	學號：411134515	日期：西元 2025 年 12 月 01 日
學生簽名：吳沛芸	學號：411147160	日期：西元 2025 年 12 月 01 日
學生簽名：溫建翔	學號：411147568	日期：西元 2025 年 12 月 01 日

指導教師簽章 葉介山

西 元 二 〇 二 五 年 十 二 月 一 日

靜宜大學資訊工程學系  
專題實作指導教師確認書

茲確認專題書面報告之格式及內容符合本系之規範

畢業專題實作名稱： 智慧排球系統：AI 姿勢分析與排球競技遊戲

畢業專題實作分組名單： 共計 5 人

組員姓名	學號
劉姿均	411147429
劉浩澄	411134395
蔡承儒	411134515
吳沛芸	411147160
溫建翔	411147568

指導教師簽章 葉介山

西 元 二 〇 二 五 年 十 二 月 一 日

## 誌 謝

本專題能順利完成，首先要衷心感謝指導教授葉介山教授，在專題過程中提供我們悉心的指導與建議，不僅在研究方向上給予我們寶貴的意見，也在技術與學術思維上給予啟發，讓我們在專題中學習到更多專業知識與實作能力。

同時，也感謝專題口試委員蔡奇偉教授、滕元翔教授、方百立教授，在口試過程中提供中肯的指導與建議，使我們能更全面地檢視研究成果與報告內容，並促使我們在未來的學習與研究中持續精進。

## 目錄

### 目錄

第一章、緒論 .....	1
第二章、專題內容與進行方法 .....	1
1.1 動機與目的 .....	1
1.2 專題相關現有系統回顧與優缺點分析 .....	1
1.3 專題進度規劃與進行方法說明 .....	1
第三章、專題流程與架構 .....	2
1.1 系統架構圖 .....	2
1.2 訓練流程 .....	2
1.3 排球競技遊戲: .....	5
第四章、專題成果介紹 .....	6
1 軟體硬體設備資訊 .....	6
2 硬體規格 .....	6
3 系統畫面(截圖) .....	8
第五章、專題學習歷程介紹 .....	8
1 專題相關軟體學習介紹 .....	9
2 專題製作過程遭遇的問題與解決方法 .....	10
2.1 姿勢辨識不穩定 .....	10
2.2 資料品質不足 .....	11
2.3 模型訓練不收斂 .....	11
2.4 資料格式錯誤 .....	11
2.5 模型與實際影片整合困難 .....	12
第六章、結論與未來展望 .....	12
1 結論 .....	12
2 未來展望 .....	12
1.1 資料擴充與模型優化 .....	13
1.2 跨運動應用 .....	13
1.3 醫療與復健輔助 .....	13
1.4 遊戲化互動與教育應用 .....	13
1.5 雲端與即時應用 .....	13
總結 .....	13

## 第一章、緒論

隨著科技的進步與健康意識的提升，運動訓練已逐漸結合人工智慧與智慧化技術發展，透過影像辨識與資料分析輔助動作判斷，能提升訓練效率並降低運動傷害風險。排球是一項高度仰賴動作協調性與正確技術的運動，例如發球、接球、扣球與攔網等動作，若姿勢不正確，不僅會影響比賽表現，也容易造成肩膀、手腕與腰部等部位的運動傷害。

近年來，人工智慧（Artificial Intelligence, AI）與電腦視覺（Computer Vision）技術快速發展，其中人體姿勢辨識技術已能準確擷取人體關鍵點資訊，並應用於運動分析、復健輔助與智慧訓練等領域。遊戲化學習（Gamification）的概念也逐漸被運用於運動與教育場域，透過互動式遊戲提升使用者參與度與學習動機。

## 第二章、專題內容與進行方法

### 1.1 動機與目的

排球是一項需要高度技巧與協調的運動，也是一項受歡迎及我們組員們熱愛的運動，正確的動作姿勢對表現與安全性非常重要，初學者常因為姿勢不正確而導致接球失誤、傳球不穩，甚至有手腕疼痛或其他運動傷害，因此，學習並掌握正確的排球姿勢，不僅能提升技術表現，也能減少受傷風險，讓運動變得更加輕鬆且有趣。

本專題的目標是以電腦視覺與 AI 為基礎，透過分析排球的基本姿勢與動作技巧，探討不同姿勢對球員表現的影響，並提供改進建議，希望藉由本研究，設計一套能即時辨識排球動作的系統，即時分析與指導學生動作，讓初學者了解正確的排球姿勢，及協助教練與選手提升訓練效率、排球技術，享受更安全、順暢的排球體驗。

### 1.2 專題相關現有系統回顧與優缺點分析

### 1.3 專題進度規劃與進行方法說明

- 1.3.1 **前期規劃階段：**蒐集相關文獻，了解姿勢辨識、動作分析與運動輔助系統之應用，並確立專題研究方向與系統架構。
- 1.3.2 **資料蒐集階段：**進行排球動作多角度拍攝，建立動作影像資料庫，並完成動作標註與分類。
- 1.3.3 **系統開發階段：**建置 MediaPipe Pose Landmark Detection 姿勢辨識模組，擷取人體 33 個關鍵點，並完成資料前處理與格式轉換。
- 1.3.4 **模型訓練階段：**使用 TCN（Temporal Convolutional Network）進行動作時間序列訓練，並透過交叉驗證與 Early

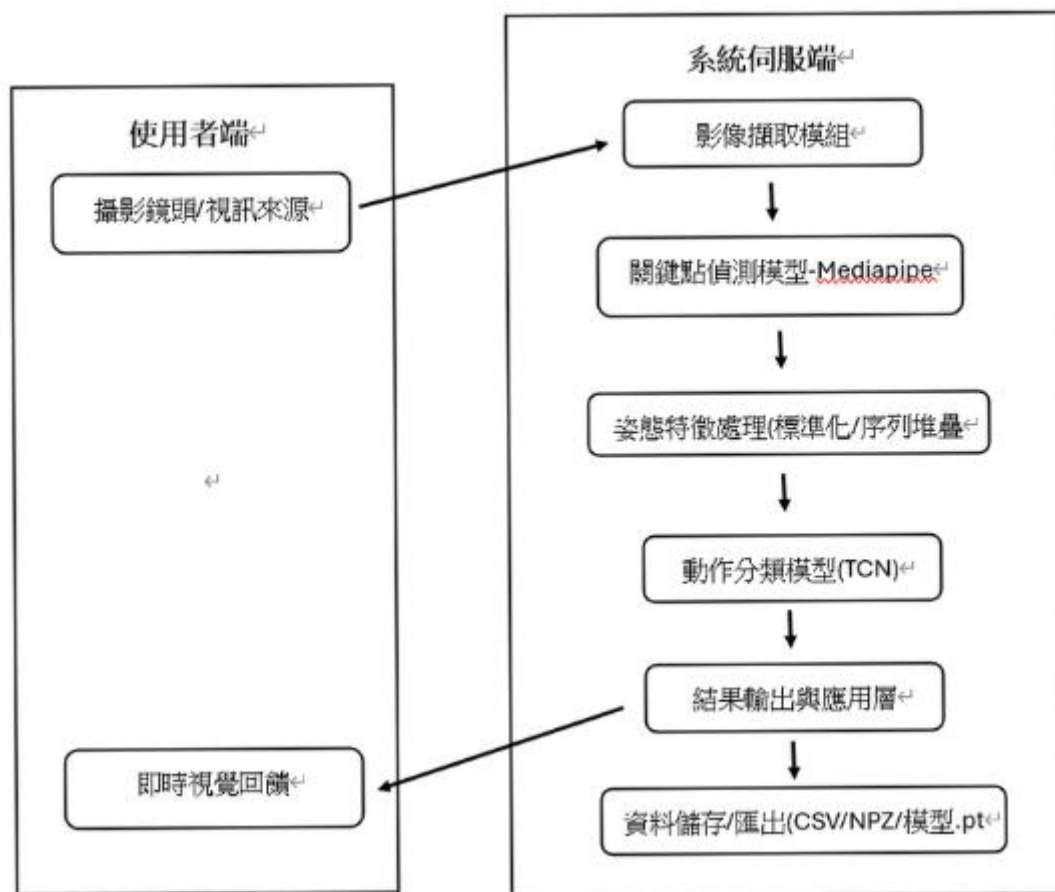
Stopping 提升模型穩定性。

### 1.3.5 成果整合階段：完成模型測試、效能評估與排球互動系統整合，進行成果展示與報告撰寫。

本系統首先以攝影機拍攝使用者排球動作，透過 MediaPipe Pose Landmark Detection 擷取人體關鍵點座標。接著將動作資料轉換為 CSV 與 NPZ 格式，並依每 30 影格切分為時間序列作為模型輸入。模型部分採用 TCN 進行動作分類訓練，以準確率、F1-score 及混淆矩陣評估模型效能，最後將訓練完成之模型應用於實際姿勢正確度辨識。

## 第三章、專題流程與架構

### 1.1 系統架構圖



### 1.2 訓練流程

步驟一：

每個動作採用固定的角度錄製影片，收集模型訓練資料集



## 步驟二：

批次讀取影片檔案，依照檔名規則提取動作類型、拍攝角度以及動作正確與錯誤資訊，整理為結構化的表格資料，會同時擷取影片的基本屬性（如影格總數、fps、起始與結束影格），最後將整理後的資訊輸出成 CSV 檔案。

filename	action	angle	quality	total_frame	start_frame	end_frame	frame_count	fps	n
underhandpass_90_correct_(10)	underhand	90	correct	76	30	60	76	30	
underhandpass_90_correct_(11)	underhand	90	correct	91	30	60	91	30	
underhandpass_90_correct_(12)	underhand	90	correct	141	30	60	141	30	
underhandpass_90_correct_(13)	underhand	90	correct	70	30	60	70	30	
underhandpass_90_correct_(14)	underhand	90	correct	73	30	60	73	30	
underhandpass_90_correct_(15)	underhand	90	correct	91	30	60	91	30	
underhandpass_90_correct_(16)	underhand	90	correct	85	30	60	85	30	
underhandpass_90_correct_(17)	underhand	90	correct	79	30	60	79	30	
underhandpass_90_correct_(18)	underhand	90	correct	79	30	60	79	30	
underhandpass_90_correct_(19)	underhand	90	correct	79	30	60	79	30	
underhandpass_90_correct_(1)	underhand	90	correct	68	30	60	68	30	
underhandpass_90_correct_(20)	underhand	90	correct	253	30	60	253	30	
underhandpass_90_correct_(21)	underhand	90	correct	70	30	60	70	30	
underhandpass_90_correct_(22)	underhand	90	correct	79	30	60	79	30	
underhandpass_90_correct_(23)	underhand	90	correct	71	30	60	71	30	
underhandpass_90_correct_(24)	underhand	90	correct	73	30	60	73	30	
underhandpass_90_correct_(25)	underhand	90	correct	72	30	60	72	30	
underhandpass_90_correct_(26)	underhand	90	correct	70	30	60	70	30	
underhandpass_90_correct_(27)	underhand	90	correct	89	30	60	89	30	
underhandpass_90_correct_(28)	underhand	90	correct	87	30	60	87	30	
underhandpass_90_correct_(29)	underhand	90	correct	53	0	30	53	30	
underhandpass_90_correct_(2)	underhand	90	correct	81	30	60	81	30	
underhandpass_90_correct_(30)	underhand	90	correct	84	30	60	84	30	

## 步驟三：

人工修改上一步驟 CSV 檔中的起始與結束影格，標註每段影片，依照起始與結束影格批次擷取影片，運用 MediaPipe Pose Landmark Detection 逐幀擷取人體關鍵點座標 (x, y, z) 及可見度(visibility)，將影片動作、角度、是否正確、fps 與關鍵點輸出為 CSV 檔案，並且輸出影片檔。



video	frame_index	frame_index	timestamp	fps	action	quality	is_correct	angle	x0	y0	z0	v0
underhandpass_90_correct_(43)	29	0	967	29.97146	underhandpass	correct	1	90	0.513306	0.358247	-0.19023	0.999968
underhandpass_90_correct_(43)	30	1	1000	29.97146	underhandpass	correct	1	90	0.512947	0.358219	-0.19183	0.999965
underhandpass_90_correct_(43)	31	2	1034	29.97146	underhandpass	correct	1	90	0.512691	0.358216	-0.1926	0.999963
underhandpass_90_correct_(43)	32	3	1067	29.97146	underhandpass	correct	1	90	0.512695	0.358143	-0.19092	0.999961
underhandpass_90_correct_(43)	33	4	1101	29.97146	underhandpass	correct	1	90	0.512782	0.357968	-0.19535	0.999959
underhandpass_90_correct_(43)	34	5	1134	29.97146	underhandpass	correct	1	90	0.513389	0.357846	-0.19801	0.999957
underhandpass_90_correct_(43)	35	6	1167	29.97146	underhandpass	correct	1	90	0.514547	0.357409	-0.20068	0.999957
underhandpass_90_correct_(43)	36	7	1201	29.97146	underhandpass	correct	1	90	0.515506	0.357178	-0.20546	0.999956
underhandpass_90_correct_(43)	37	8	1234	29.97146	underhandpass	correct	1	90	0.51655	0.356952	-0.1954	0.999954
underhandpass_90_correct_(43)	38	9	1267	29.97146	underhandpass	correct	1	90	0.518084	0.356514	-0.20167	0.999953
underhandpass_90_correct_(43)	39	10	1301	29.97146	underhandpass	correct	1	90	0.519984	0.356235	-0.20077	0.999956
underhandpass_90_correct_(43)	40	11	1334	29.97146	underhandpass	correct	1	90	0.522048	0.356235	-0.2014	0.999957
underhandpass_90_correct_(43)	41	12	1367	29.97146	underhandpass	correct	1	90	0.524313	0.356301	-0.206	0.99996
underhandpass_90_correct_(43)	42	13	1401	29.97146	underhandpass	correct	1	90	0.526841	0.356423	-0.21828	0.999962
underhandpass_90_correct_(43)	43	14	1434	29.97146	underhandpass	correct	1	90	0.529448	0.357013	-0.23587	0.999965
underhandpass_90_correct_(43)	44	15	1468	29.97146	underhandpass	correct	1	90	0.532211	0.357659	-0.23104	0.999967
underhandpass_90_correct_(43)	45	16	1501	29.97146	underhandpass	correct	1	90	0.534517	0.359261	-0.22515	0.999968
underhandpass_90_correct_(43)	46	17	1534	29.97146	underhandpass	correct	1	90	0.537157	0.360638	-0.23071	0.99997
underhandpass_90_correct_(43)	47	18	1568	29.97146	underhandpass	correct	1	90	0.539193	0.362903	-0.24389	0.999972
underhandpass_90_correct_(43)	48	19	1601	29.97146	underhandpass	correct	1	90	0.541281	0.36369	-0.25827	0.999974
underhandpass_90_correct_(43)	49	20	1634	29.97146	underhandpass	correct	1	90	0.543005	0.364448	-0.24902	0.999976
underhandpass_90_correct_(43)	50	21	1668	29.97146	underhandpass	correct	1	90	0.543352	0.364308	-0.24732	0.999977
underhandpass_90_correct_(43)	51	22	1701	29.97146	underhandpass	correct	1	90	0.543332	0.363352	-0.21524	0.999976
underhandpass_90_correct_(43)	52	23	1734	29.97146	underhandpass	correct	1	90	0.54326	0.359753	-0.21777	0.999976
underhandpass_90_correct_(43)	53	24	1768	29.97146	underhandpass	correct	1	90	0.543531	0.355646	-0.21567	0.999977
underhandpass_90_correct_(43)	54	25	1801	29.97146	underhandpass	correct	1	90	0.542668	0.352596	-0.21006	0.999976
underhandpass_90_correct_(43)	55	26	1835	29.97146	underhandpass	correct	1	90	0.541771	0.349588	-0.21507	0.999977
underhandpass_90_correct_(43)	56	27	1868	29.97146	underhandpass	correct	1	90	0.539797	0.349372	-0.22477	0.999979
underhandpass_90_correct_(43)	57	28	1901	29.97146	underhandpass	correct	1	90	0.537624	0.350219	-0.24076	0.99998
underhandpass_90_correct_(43)	58	29	1935	29.97146	underhandpass	correct	1	90	0.534283	0.353958	-0.23889	0.999982



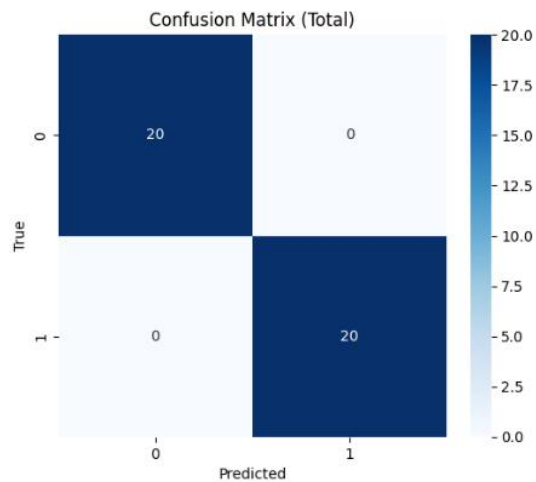
#### 步驟四：

依照每 30 影格為一個序列切片，將原始二維資料重組為三維結構 (n\_slices, 132 features, 30 fps)，以模擬時間序列的動作變化。quality 欄位進行二值化處理，使用 main() 函數，若該序列所有影格皆為正確，則標記為 1；若其中包含錯誤，則標記為 0，動作全程正確才算對。最後，將運算完的向量儲存為 .npy 及 .npz 格式。

#### 步驟五：

將每 30 影格切片成時間序列，並對特徵進行標準化後輸入 TCN 模型進行動作分類。模型利用多層卷積捕捉時間特徵，訓練時採 5 折交叉驗證與早停策略以避免過度擬合。最後以準確率、F1 分數及混淆矩陣評估模

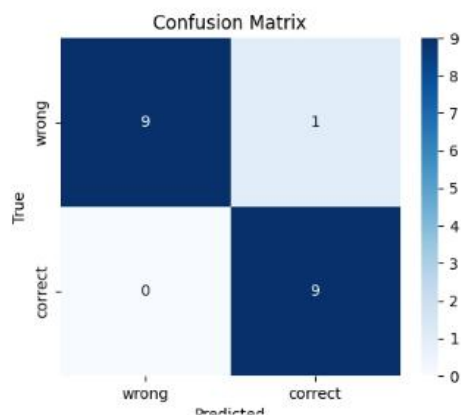
型表現，有效判斷每個動作是否正確。



```
best_model_fold1.pt
best_model_fold2.pt
best_model_fold3.pt
best_model_fold4.pt
best_model_fold5.pt
```

步驟六：

將測試資料標準化並轉換成 TCN 輸入格式後，載入訓練好的最佳模型進行預測。透過準確率、宏平均 F1 分數與混淆矩陣評估模型分類效果，並以圖形呈現，清楚檢視模型對「正確」與「錯誤」動作的判別能力。



```
pose.py"
Dataset shape: (19, 30, 132) (19,)
Accuracy: 0.947, Macro-F1: 0.947
Confusion Matrix:
[[9 1]
 [0 9]]
```

步驟七：

畫出 MediaPipe Pose Landmark Detection 辨識的 33 個關鍵點和連線的影片，利用訓練的模型辨識姿勢正確度，並進行即時影像辨識。

### 1.3 排球競技遊戲:

以 Unity 為開發平台，採用物理引擎 Physics2D 做出球體彈跳與碰撞反應。玩家可透過鍵盤控制角色進行移動、跳躍與擊球，以將球擊入對方場地得分。

主要元件：

Rigidbody2D：模擬重力與碰撞物理效果

BoxCollider2D：角色與球體的碰撞檢測

## Animation Controller：角色跳躍與擊球動作控制



### 第四章、 專題成果介紹

#### 1 軟體硬體設備資訊

##### 1.1 技術

深度學習框架

PyTorch

模型架構

TCN (Temporal Convolutional Network)

電腦視覺與姿勢估測

MediaPipe Pose Landmark Detection

OpenCV

#### 2 硬體規格

Python 套件版本

absl-py	2.3.1
astunparse	1.6.3
attrs	25.3.0
certifi	2025.10.5
cffi	2.0.0
charset-normalizer	3.4.4
colorama	0.4.6
contourpy	1.3.3
cycler	0.12.1
filelock	3.19.1
flatbuffers	25.2.10
fonttools	4.59.2
fsspec	2025.9.0
gast	0.6.0
google-pasta	0.2.0

grpcio	1.76.0
h5py	3.15.1
idna	3.11
jax	0.7.1
jaxlib	0.7.1
Jinja2	3.1.6
joblib	1.5.2
keras	3.12.0
kiwisolver	1.4.9
libclang	18.1.1
Markdown	3.9
markdown-it-py	4.0.0
MarkupSafe	3.0.2
matplotlib	3.10.6
mdurl	0.1.2
mediapipe	0.10.14
ml_dtypes	0.5.3
mpmath	1.3.0
namex	0.1.0
networkx	3.5
numpy	2.2.6
opencv-contrib-python	4.12.0.88
opencv-python	4.12.0.88
opt_einsum	3.4.0
optree	0.17.0
packaging	25.0
pandas	2.3.2
pillow	11.3.0
pip	25.3
protobuf	4.25.3
pyparser	2.23
Pygments	2.19.2
pyparsing	3.2.3
python-dateutil	2.9.0.post0
pytz	2025.2
requests	2.32.5
rich	14.2.0
scikit-learn	1.7.2
scipy	1.16.2
seaborn	0.13.2
setuptools	80.9.0
six	1.17.0
sounddevice	0.5.2

sympy	1.14.0
tensorboard-data-server	0.7.2
termcolor	3.2.0
threadpoolctl	3.6.0
torch	2.8.0
tqdm	4.67.1
typing_extensions	4.15.0
tzdata	2025.2
urllib3	2.5.0
Werkzeug	3.1.3
wheel	0.45.1
wrapt	2.0.0

### 3 系統畫面(截圖)

#### 第五章、 專題學習歷程介紹

本專題為「智慧排球系統：AI 姿勢分析與排球競技遊戲」，從最一開始的發想、技術學習、資料蒐集、模型訓練到系統整合，整個過程都是由團隊自行規劃並一步一步完成。

一開始在選題時，我們希望做一個**結合運動與人工智慧的應用系統**，不同於一般靜態影像辨識，而是能夠實際分析「動作過程」的專題，於是選擇以排球動作作為主題，設計一套能夠判斷動作正確性的智慧系統。

確定主題後，我們先從最基礎的姿勢辨識開始學習，包含什麼是人體關鍵點（Pose Landmark）、如何透過電腦視覺擷取人體動作、如何將影像轉換成可分析的數據格式。這一階段我們花了不少時間查資料、看論文與教學影片，才逐漸理解整個 AI 姿勢辨識的流程。

接著進入**資料蒐集階段**，我們實際拍攝排球動作影片，包括多個拍攝角度（正面、側面、斜前方），並區分正確動作與錯誤動作。這個階段最大的困難在於拍攝環境的不穩定，例如光線變化、人員走動等，都會影響辨識效果，因此我們後來調整拍攝場地與角度，提升影像品質。

拍攝完成後進入資料前處理階段，我們將影片透過程式逐一讀取，使用 MediaPipe Pose 將影片轉換成人體 33 個關鍵點的座標資料，並輸出成 CSV 檔，再整理成模型可使用的時間序列資料。這一部分讓我們實際體會到「原始影像」與「AI 模型輸入資料」之間的差異。

在模型訓練階段，我們選擇使用 TCN (Temporal Convolutional Network) 做為動作分類模型，主要用來處理時間序列資料。為了提升模型穩定性，我們加入 5-fold Cross Validation、Early Stopping、標準化等機制，讓訓練流程更完整。

到了後期，我們開始將模型與系統整合，讓使用者能夠透過影片輸入，即時觀看 Pose Landmark 辨識結果，並由模型輸出動作是否正確。這個階段也讓我們真正體會到「AI 系統從資料到應用」的完整流程。

整體而言，從完全不熟悉姿勢辨識，到能夠獨立完成模型訓練與系統整合，這段過程不但提升了我們的程式能力，也讓我們更了解 AI 在運動與醫療領域的實際應用價值。

## 1 專題相關軟體學習介紹

在本專題製作過程中，我們使用了多套軟體與程式工具，從影像處理、資料分析到深度學習模型訓練，皆有實際操作與應用。這些工具不僅幫助我們完成專題，也讓我們對整個 AI 系統開發流程有更深刻的理解，並體會到理論與實作之間的差異。

首先在姿勢辨識方面，我們使用的是 MediaPipe Pose。MediaPipe 是 Google 推出的電腦視覺工具套件，能夠即時偵測人體 33 個關鍵點位置，包括頭部、手臂、身體與腿部關節。我們透過 Python 介接 MediaPipe，讀取影片後即時取得每一幀的姿勢座標資料。在學習 MediaPipe 的過程中，我們不僅熟悉了關鍵點偵測的 API 使用方式，也理解了可見度 (visibility) 參數在資料過濾與錯誤點修正上的應用。此外，我們還嘗試整合平滑濾波與動作軌跡連線，使關鍵點在影片中呈現更穩定且連續的動作軌跡，提升辨識結果的直觀呈現效果。

接下來在資料整理與分析方面，我們使用 Python、Pandas、NumPy 進行數據處理。透過 Pandas 讀取 CSV 檔後，再由 NumPy 將資料切割為每 30 影格一組的時間序列資料，轉換成模型可訓練的格式。這個過程讓我們更加熟悉資料前處理的重要性，也學會如何進行標準化、缺失值處理、維度檢查以及異常資料過濾。為了提升模型訓練的效果，我們還實作了簡單的資料增強方法，例如座標鏡像翻轉、縮放與平移，以模擬不同拍攝角度與動作差異，提高模型對變化動作的辨識能力。這些資料處理經驗不僅強化了我們對時間序列資料的理解，也培養了面對大量資料時的分析能力。

在深度學習模型訓練方面，我們使用 PyTorch 框架，並成功實作 TCN



(Temporal Convolutional Network) 動作分類模型。透過自訂 Dataset 與 DataLoader，我們能夠靈活地控制資料輸入與批次大小，並設計合適的模型架構與訓練流程。我們學會如何調整卷積層的卷積核大小、步幅 (stride) 與層數，理解 Temporal Convolution 對時間序列資料的捕捉能力，並能夠針對不同動作特徵做模型設計優化。在訓練過程中，我們使用 Optimizer、Loss Function 以及 Learning Rate Scheduler 來控制模型收斂速度與穩定性，並觀察訓練曲線的變化，調整參數以避免過擬合或震盪。

為了讓模型訓練過程更完整與客觀，我們使用 Scikit-learn 進行 5 折交叉驗證 (Stratified K-Fold)，並計算 F1-score 以及輸出混淆矩陣。交叉驗證確保模型在不同資料切分下的穩定性，避免單一資料分割造成偏差；混淆矩陣則幫助我們觀察模型在各動作分類上的辨識情況，發現哪些動作容易混淆，並進行針對性資料增強與模型調整。

在結果視覺化方面，我們使用 Matplotlib、Seaborn 繪製混淆矩陣、損失曲線與準確率曲線，使模型訓練過程更加直觀易懂。我們也嘗試將這些視覺化結果整合到簡單的 GUI 介面中，方便非技術使用者理解模型效果，提升報告與展示的可讀性。

此外，在影片處理方面，我們使用 OpenCV 讀取影片畫面、標記姿勢關鍵點並輸出辨識後的影像。為了讓系統更易於觀察，我們加入了動作軌跡連線、顏色標示及即時姿勢顯示功能，使使用者可以直觀地觀察動作連續性與正確性。透過這個過程，我們更深入理解了影像座標轉換、畫面標記與即時影像處理的技術細節，也增強了系統整合能力。

透過這次專題，我們實際接觸到從「影像 → 資料 → AI 訓練 → 成果輸出」的完整流程，不再只是課堂上的理論，而是真正能夠獨立操作完整系統。在這個過程中，我們學會了如何選擇合適的軟體工具、分析問題、優化流程，以及將技術成果以視覺化方式呈現。這些經驗對未來進入研究或實務專案開發，都提供了非常扎實的基礎與實作能力。同時，我們也對 AI 系統在運動與教育領域的應用有更深入的理解，並意識到資料品質、演算法設計與系統整合對整個專案成功的重要性。

## 2 專題製作過程遭遇的問題與解決方法

在專題製作過程中，我們遇到了多種困難與挑戰，從資料收集、姿勢辨識到模型訓練與系統整合，每個環節都可能出現問題。透過不斷的討論、實驗與調整，我們逐步克服這些挑戰，也從中學習到寶貴的專題經驗。

### 2.1 姿勢辨識不穩定

初期使用 MediaPipe Pose 偵測人體關鍵點時，系統對光線、拍攝角度和背景非常敏感。特別是在光線不足或拍攝角度偏斜的情況下，關鍵點可能偵測錯誤或缺失。

解決方法：

改善拍攝環境，使用光線充足且均勻的場地。

固定攝影機位置與拍攝距離，減少人物移動造成的偵測誤差。

對關鍵點座標使用簡單平滑濾波，減少影格間抖動。

透過這些調整，姿勢辨識的穩定性明顯提升，也讓模型訓練的資料品質更可靠。

## 2.2 資料品質不足

原本收集的影片數量太少，且多數動作樣本過於單一，導致模型容易過擬合，準確率在測試集上起伏較大。

解決方法：

補拍更多影片，增加不同角度、距離和背景的樣本。

收集錯誤動作或不標準動作的影片，讓模型學習辨識「正確動作與錯誤動作」的差異。

對資料進行基本增強，例如簡單的鏡像翻轉或座標偏移，提高模型對不同動作變化的適應能力。

透過資料多樣化，模型的泛化能力明顯改善，測試結果也更穩定。

## 2.3 模型訓練不收斂

在初期訓練 TCN 模型時，準確率容易停滯或劇烈震盪，訓練效果不理想。

解決方法：

調整 Learning Rate 和 Batch Size，找到較合適的訓練參數。

加入 Early Stopping，避免模型因長時間震盪而過擬合。

對輸入資料進行標準化，確保不同關鍵點座標的數值範圍一致。

嘗試不同的卷積核大小與層數配置，改善 Temporal Convolution 對時間序列動作的捕捉能力。

經過多次實驗與參數調整後，模型訓練變得穩定，準確率曲線呈現平滑收斂。

## 2.4 資料格式錯誤

在將 CSV 轉換成 Numpy 時間序列資料的過程中，曾出現維度不一致或影格數不足的情況，導致模型無法順利訓練。



解決方法：

在資料處理過程中加入 shape 檢查與例外處理，確保每筆資料維度一致。  
針對影格不足的影片，進行補齊或刪除，避免異常資料影響訓練。  
編寫自動化資料檢查腳本，確保每次資料輸入模型前都經過驗證。  
這不僅解決了維度問題，也讓我們對資料前處理流程有更嚴謹的理解。

## 2.5 模型與實際影片整合困難

初期模型僅能對離線資料進行預測，無法即時應用在影片畫面上。這使得系統無法達到預期的互動效果。

解決方法：

重新設計資料流程，確保影片輸入格式與模型訓練時一致。  
將模型推論（inference）整合到即時影像處理管線中，搭配 OpenCV 即時繪製姿勢與分類結果。  
測試系統在不同影片解析度、影格率下的運行效率，確保即時辨識不卡頓。

## 第六章、 結論與未來展望

### 1 結論

本次專題「智慧排球系統：AI 姿勢分析與排球競技遊戲」順利完成了從影像擷取、資料前處理、模型訓練到成果呈現的完整流程。在過程中，我們使用 MediaPipe Pose 偵測人體 33 個關鍵點，並透過 Python、Pandas、NumPy 將影片資料整理成時間序列格式，最終以 TCN 模型進行動作分類。訓練完成後，我們能夠即時在影片中呈現姿勢辨識結果，並結合遊戲化互動介面，提供使用者即時動作回饋與趣味體驗。

透過專題實作，我們驗證了 AI 姿勢辨識技術在運動領域的應用可能性：不僅能協助運動員與學生檢視動作正確性，也能作為運動醫學與復健輔助工具。整個專題過程不只是技術操作，更讓我們深刻理解 AI 系統從資料收集、模型訓練到結果應用的完整開發流程，培養了程式設計、資料處理、問題解決以及系統整合的能力。

此外，透過資料整理與模型評估，我們學會了如何有效地檢測模型效能，例如使用 5 折交叉驗證、F1-score 與混淆矩陣，確保模型在各動作分類上具有穩定表現。整個過程也加深了我們對深度學習理論與實務應用的理解，從課堂理論走向實際操作，收穫相當豐富。

### 2 未來展望

### 1.1 資料擴充與模型優化

目前系統的資料量與動作種類仍有限，未來可透過收集更多不同角度、不同運動員的影片，增加資料多樣性。同時，可嘗試其他深度學習架構，如 LSTM、Transformer 或混合模型，進一步提升動作分類精準度與泛化能力。

### 1.2 跨運動應用

除了排球，我們希望將系統延伸至羽球、籃球、健身訓練等其他運動，提供更多動作分析與訓練輔助功能。跨運動應用不僅增加系統實用性，也能驗證模型對不同動作的適應性。

### 1.3 醫療與復健輔助

將系統應用於復健或物理治療領域，協助病患矯正姿勢與動作，並提供即時回饋。結合穿戴式感測器或生理訊號，可進一步提升復健效果的量化分析。

### 1.4 遊戲化互動與教育應用

系統可進一步加入排行榜、挑戰模式、多人互動或擴增實境（AR）功能，提升趣味性與使用者黏著度。同時，作為體育課程輔助工具，可讓學生在遊戲中學習正確動作，降低學習門檻。

### 1.5 雲端與即時應用

未來可將模型部署至雲端或移動裝置，實現更多即時辨識與跨平台應用，使系統更加方便使用者在不同場景下操作。

## 總結

總體而言，本次專題不僅完成技術目標，也讓我們實際操作完整的 AI 系統開發流程，從資料收集、前處理、模型訓練到成果展示都有深入體驗。專題過程中遇到的各種挑戰，不僅磨練了程式能力與問題解決技巧，也培養了跨領域協作與專案管理能力。展望未來，我們相信智慧排球系統具有廣泛的應用潛力，無論在運動訓練、教育、復健，甚至遊戲互動領域，都能發揮實際價值，也為後續研究與開發提供了穩固的基礎。