2025年臺灣國際科學展覽會
優勝作品專輯

作品編號　190006

參展科別　電腦科學與資訊工程

作品名稱　以深度學習進行籃球慣用動作分析

就讀學校　臺北市立第一女子高級中學

指導教師　潘則佑

　　陳怡芬

作者姓名　李信恩

　　陳敏瑄

關鍵詞　籃球、深度學習、動作辨識

作者簡介

我們是來自北一女中高二的李信恩(左)、陳敏瑄(右)。我們的興趣是打籃球，很高興這次有機會將興趣與研究結合，做與籃球有關的研究題目，希望我們開發的籃球慣用動作分析系統可以幫助更多喜歡打籃球的人，也謝謝我們的指導老師、教授及學長，未來我們會繼續努力！

2025 年臺灣國際科學展覽會

研究報告

區　　別：（編號由國立臺灣科學教育館統一填列）

科　　別：電腦科學與資訊工程

作品名稱：以深度學習進行籃球慣用動作分析

關鍵詞：籃球、深度學習、動作辨識

編　　號：（編號由國立臺灣科學教育館統一填列）

摘要

本研究聚焦於籃球員的慣用動作分析，透過深度學習技術開發了一套籃球動作分析系統，旨在準確分析籃球員在籃球運動中的個人動作特徵來進行動作辨識。我們透過自行蒐集籃球動作的影片，並使用 MMAction2 這個資源庫來進行動作辨識模型的訓練，將訓練好的動作辨識模型用開發慣用動作分析系統。系統流程首先使用滑動視窗（Sliding Window）的機制將即時拍攝的影像變成有序列的連續影像片段，再即時傳送至進攻動作辨識的深度學習模型中，來辨識出連續影像片段中的動作序列屬於何種特定動作，藉此將多個連續影像片段中的動作序列各自轉換為單一動作單元並依次輸出。最終，系統基於前述單一動作資料進行綜合分析，以統計使用者的籃球慣用動作。此分析系統能為籃球愛好者提供清晰的動作偏好資料，具有提升訓練成效的潛力，同時為籃球技術分析與訓練提供了一個精確的數據分析工具。

Abstract

This study focuses on analyzing habitual basketball moves by developing a basketball motion analysis system using deep learning techniques to accurately recognize and assess users' individual movement patterns in the sport. The system workflow begins with recognizing single basketball actions by utilizing MMaction2 technology to identify specific movements. Next, the system employs deep learning in combination with a sliding windows technique to segment and detect continuous basketball movements, transforming action sequences in videos into discrete motion units, which are then output sequentially. Finally, based on the compiled single-action data, the system performs a comprehensive analysis to determine users' habitual basketball moves. This analysis system is capable of providing personalized movement recommendations and clear data on users' movement preferences, enhancing training efficiency. It also serves as a precise data analysis tool to support technical analysis and training optimization in basketball.

壹、 研究動機

近幾年來，隨著人工智慧（Artificial Intelligence，簡稱 AI）技術的飛速發展，將人工智慧結合影像分析運用在籃球場上已屢見不鮮。舉例來說，透過 AI 影像分析技術自動化標註比賽時發生的重要事件，如灌籃、快攻得分，在賽後檢討快速找到影像並進行檢討；透過 AI 影像分析技術從球賽轉播影片分析對手常用的戰術，並藉由虛擬實境模擬對手站位，進行沉浸式訓練，這些技術的出現大幅降低教練團情蒐的時間，也帶給選手更好的訓練環境。身為籃球熱愛者的我們，在高中時都加入了籃球社，在比賽時常常會因為不了解隊友想要做什麼動作而出現失誤，例如：以為隊友要上籃，結果他傳球給我，卻因沒做好準備而失誤。或是面對一樣的進攻情境，A 隊友會採取直接切入進攻，而 B 隊友會採取繞步的方式進攻。然而，要了解一個隊友需要長時間的搭配合作，歷經各種情境模擬，才有機會了解其慣用動作。因此，我們想要透過深度學習來進行 AI 影像分析技術，建立一個籃球慣用動作的辨識模型，並且建構一個籃球慣用動作分析系統，讓我們可以在很短的時間內徹底了解隊友打球的習性。

貳、 研究目的

本研究旨在透過建構籃球慣用動作分析系統，使用者可上傳隊友持球進攻時的影片，或是即時拍攝隊友持球進攻時的連續影像，系統將運用 AI 影像分析的技術，精確辨識並分析出隊友在進攻時慣用的籃球動作，並以圖表和數據的方式呈現結果。透過這樣的系統，使用者可以在短時間內掌握隊友的慣用動作，像是是否偏好突破切入、運球假動作、或繞步等進攻傾向，進而在比賽中做出更有效的協作應對，逐漸增強團隊默契與戰術執行的精準度，提升整體場上表現。此外，若能進一步了解不同球員的進攻風格如何隨場上情境改變，以探討個人慣用動作和團隊戰術配合之間的關聯性，對於籃球戰技的學習勢必有個一大幫助。

為了實現上述系統，我們制定了以下研究內容及目的，來確保能夠完成系統開發：

(一)選定適合模型進行進行籃球動作辨識

(二)訓練籃球單一動作辨識模型

(三)訓練籃球連續動作辨識模型

(四)建立籃球慣用動作分析系統

參、 研究設備及器材

一、硬體

手機（Samsung A34 和 IPhone X）：用於影片拍攝及蒐集

電腦（NAVIDA GeForce RTX 4060 Ti）：用於模型訓練及系統建置

二、軟體

本研究使用的軟體如表 1。

表 1　研究使用軟體之介紹

軟體名稱	用途
Python	高階程式語言，語法簡潔，擁有豐富的函式庫，適合快速開發與原型製作
MMaction2	用於影片中的動作識別任務
Pytorch	用於構建和訓練人工神經網絡
MMCV	基礎函式庫，用於支援多媒體任務中的深度學習模型開發
OpenCV	開源的計算機視覺和影像處理函式庫，提供豐富的影像處理工具和功能
FFmpeg	用於處理、轉碼、編碼和解碼影片檔案
Scikit-learn	用於資料預處理、模型訓練、分類
CUDA	使程式能夠在 NVIDIA GPU 上進行計算
cuDNN	加速 GPU 上的深度學習運算

肆、 研究方法

本研究旨在運用深度學習技術開發了一套籃球動作分析系統，準確識別並分析使用者在籃球運動中的個人動作特徵。系統包含三個部分，單一動作辨識、連續動作辨識、以及籃球慣用動作分析系統，詳細內容將於後面介紹。其中，單一動作辨識系統為對一短影片（1～3 秒）進行動作辨識；連續動作辨識系統為對一長影片（30 秒左右）進行動作辨識，將長影片運用 Sliding Window 進行分割，再將分割後的短影片丟

入單一動作辨識系統中進行動作辨識；籃球慣用動作分析系統可輸入一長影片後，將其傳送至連續動作辨識系統，將得到的結果進行資料統計及分析，並輸出受測者的慣用動作。系統架構如圖 1、2 所示。

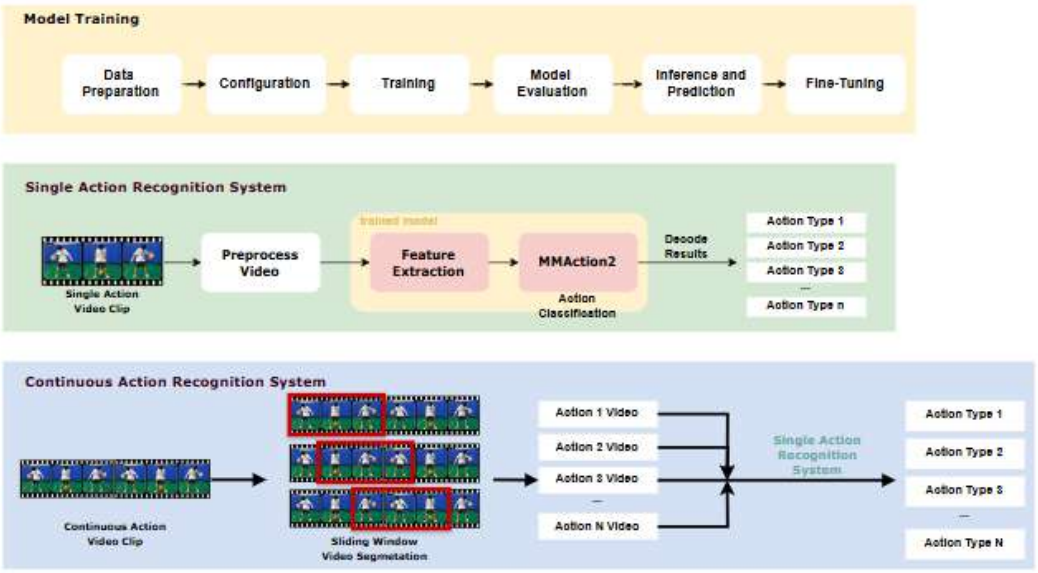


圖 1 籃球慣用動作系統架構圖(此圖為作者自製)

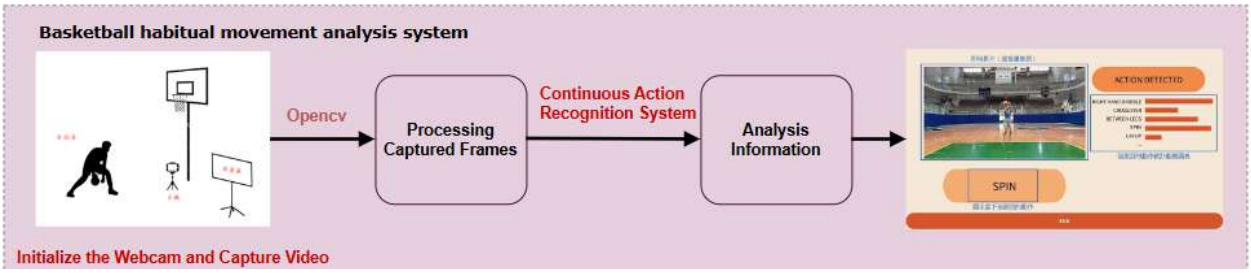


圖 2 單一動作及連續動作辨識系統架構圖(此圖為作者自製)

首先將針對研究所需深度學習技術進行說明。

一、研究背景

(一) CNN（Convolutional Neural Network，卷積神經網絡）

CNN 是一種專門用於處理圖像和影片數據的深度學習模型，能夠有效提取空間特徵和捕捉區域性模式，廣泛應用於圖像分類、物體檢測和語義分割等計算機視覺任務。CNN 主要由卷積層、激活函數、池化層和全連接層構成（如圖 3）。卷積層使用可學習的濾波器對圖像進行卷積，提取不同層次的特徵並生成特徵圖；激活函數（如 ReLU）提高模型的非線性表達能力；池化層通過縮小特徵圖尺寸來降低計算量並保留重要特徵；全連接層則用於分類，將提取到的特徵壓縮為固定大小向量，並使用 Softmax 函數輸出類別概率。

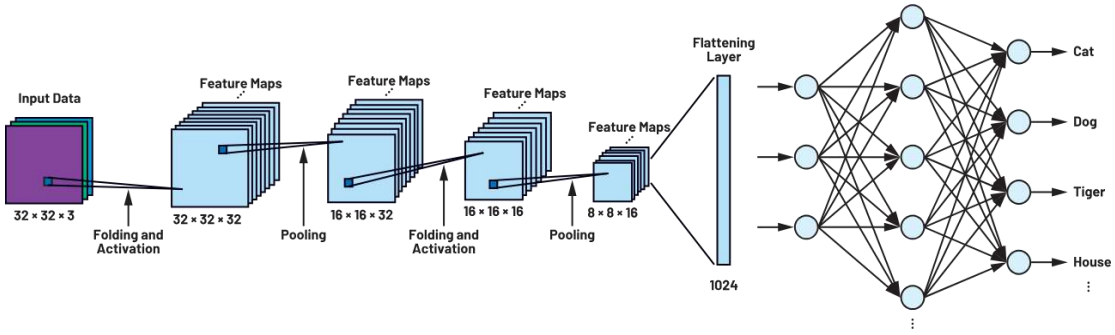


圖 3 CNN 架構圖取自[1]

(二) MMaction2 開源工具包

MMAction2 是一個基於 PyTorch 的開源工具包，支援大量的影片理解模型，包括動作辨識（action recognition）、基於骨架的行為識別（skeleton-based action recognition）、時空行為檢測（spatio-temporal action detection）（temporal action localization）等多個主要方向。它還支援大部分流行的學術資料集，並提供許多實用工具幫助使用者對資料集和模型進行多方面的探索與除錯。在本研究中會應用到其中的動作辨識模型（Action Recognition Models），而 MMaction2 中有支援多個動作辨識模型(如圖 4)，我們根據其方法優劣及新舊經過篩選後，將介紹其中幾個本研究中會使用的模型。

Action Recognition				
C3D (CVPR'2014)	TSN (ECCV'2016)	I3D (CVPR'2017)	C2D (CVPR'2018)	I3D Non-Local (CVPR'2018)
R(2+1)D (CVPR'2018)	TRN (ECCV'2018)	TSM (ICCV'2019)	TSM Non-Local (ICCV'2019)	SlowOnly (ICCV'2019)
SlowFast (ICCV'2019)	CSN (ICCV'2019)	TIN (AAAI'2020)	TPN (CVPR'2020)	X3D (CVPR'2020)
MultiModality: Audio (ArXiv'2020)	TANet (ArXiv'2020)	TimeSformer (ICML'2021)	ActionCLIP (ArXiv'2021)	VideoSwin (CVPR'2022)
VideoMAE (NeurIPS'2022)	MViT V2 (CVPR'2022)	UniFormer V1 (ICLR'2022)	UniFormer V2 (Arxiv'2022)	VideoMAE V2 (CVPR'2023)

圖 4 MMaction2 支援模型 擷取自[2]

1. TSN 模型

TSN（Temporal Segment Networks）是 MMAction2 中用於影片動作辨識的經典模型之一。旨在有效捕捉影片中的時序資訊，適用於需要理解全局時間資訊的場景，例如動作辨識。它的基本工作原理是透過分段抽樣處理長時間影片：將影片分割為若干相等長度的片段，從每個片段中隨機選擇一幀或多幀，再利用 CNN 模型提取片段特徵，最後將這些特徵進行融合，以得到整個影片的最終分類結果。TSN 的優點在於它能有效建模影片的全局時序資訊，同時降低了計算成本，尤其是相比直接使用 3D CNN 的模型，它具有更高的運算效率，非常適合需要捕捉較長時間動作變化的場景。在 MMAction2 中，TSN 透過高度模組化的方式實現，使用者可以靈活選擇 CNN 架構、設置片段數量等參數進行模型的訓練和推理，從而滿足不同的資料集和硬體需求。其架構如圖 5。

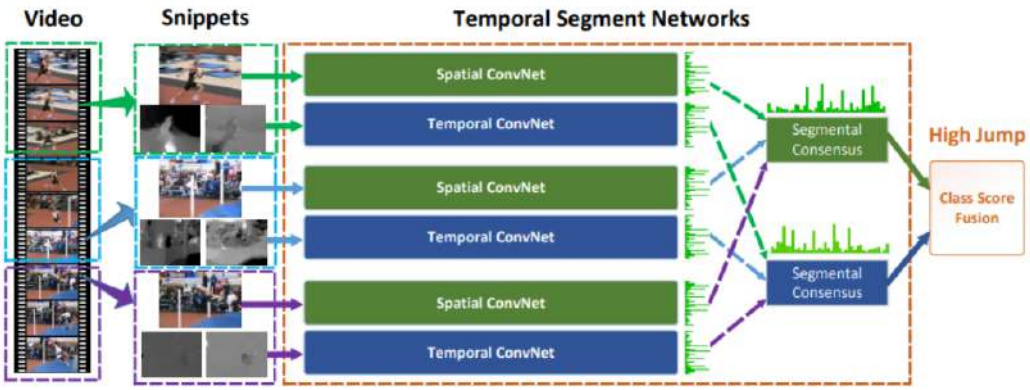


圖 5 TSN 架構圖取自[3]

2. **I3D 模型**

I3D（Inflated 3D ConvNet）是一種用於影片動作辨識的深度學習模型，旨在同時捕捉影片的空間和時間特徵。I3D 基於傳統的 2D 卷積網路擴展而來，將 2D 卷積膨脹成 3D 卷積，讓模型可以處理影片的時空維度資訊。其關鍵在於將經過預訓練的 2D 網路中的卷積核膨脹為 3D 卷積核，這樣可以在空間（寬和高）以及時間（深度）三個維度上進行卷積操作。這種擴展讓 I3D 能夠直接從原始影片中提取時空特徵，有效地理解和動作辨識。與傳統的 2D 卷積網路相比，I3D 在建模影片的時序資訊上更具優勢，並且能夠利用從大型圖像資料庫集中遷移學習而來的知識。其架構如圖 6。

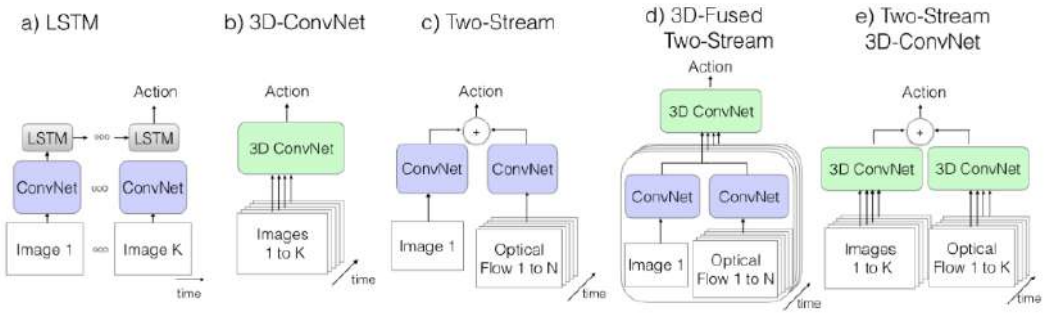


圖 6 I3D 架構圖取自[4]

3. **UniFormer2**

在前幾個方法的介紹中，會發現在影片分析中，核心問題是如何學習具區分性的時空表示。雖然視覺變壓器（ViTs）在捕捉影片中的長期依賴性上表現優異，但在處理局部冗餘上有其限制。UniFormer 結合了卷積和自注意力，成功解決這個問題，但仍需繁瑣的影像預訓練。UniFormerV2 進一步改進，結合預訓練的 ViTs 和 UniFormer 架構，實現局部與全局關係的聚合，並在多個影片基準上表現突出，成為首個在 Kinetics-400 上達到 90% top-1 準確率的模型。其架構如圖 7。

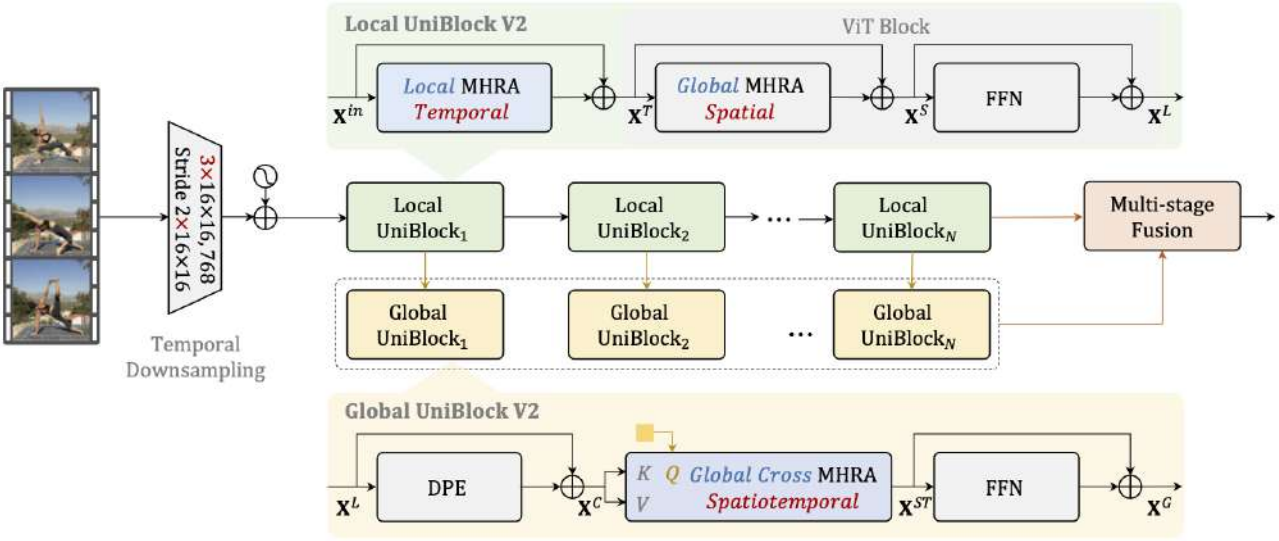


圖 7UniFormer 架構圖取自[5]

4. VideoSwin

VideoSwin 是一種純基於 Transformer 的影片建模演算法，並在主要的影片辨識基準測試中取得了最高的準確率。這個模型中引入了影片轉換器的局部性歸納偏差，與過去採用時空分解全局計算自注意力的方法相比，實現了更好的速度與準確率平衡。這種影片架構的局部性是透過使用專為影像領域設計的 Swin Transformer 來達成的，同時也繼續發揮了預訓練影像模型的強大功能。其架構如圖 8。

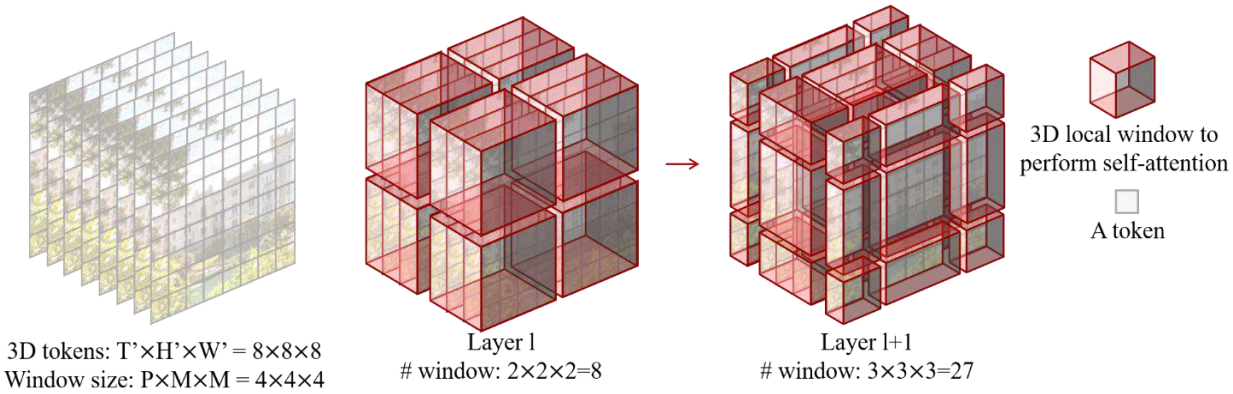


圖 8 Video Swim 架構圖取自[6]

5. MViT V2

MViT V2 在 MMAction2 中用於影片動作識別，透過多尺度結構和稀疏注意機制，同時處理空間與時間資訊。它能捕捉不同時間尺度的動作變化，並在多幀影片中建模時間維度，提升識別準確性。相比 3D 卷積神經網絡，MViT V2 計算效率更高，能在精度與速度間取得平衡，適用於不同長度的影片和動作模式。其架構基於多尺度視覺變壓器，結合稀疏注意，靈活應對大規模數據並提升效能。及架構如圖 9。

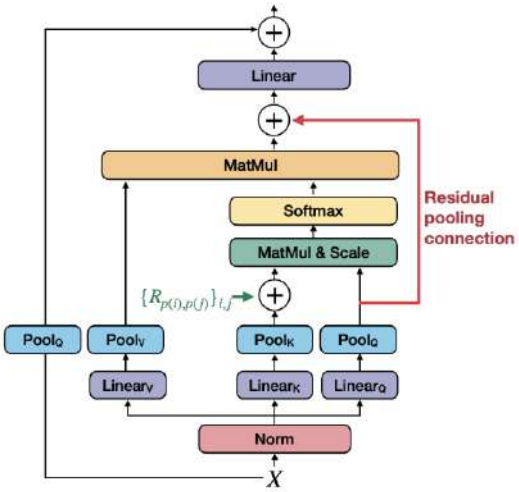


圖 9MViT V2 架構圖取自[7]

(三) Sliding Window

Sliding Window（滑動視窗）是一種常用於處理序列數據的技術，特別是在計算機視覺、影片分析和時間序列預測等領域。這個方法的主要目的是透過在序列中滑動一個固定大小的 window（視窗）來提取特徵或進行分類，從而捕捉局部資訊。其基本流程為，首先設定 window 及 step size（步長），window 是每個影片片段的長度，step size 是 window 每次滑動的距離；接著會依據設定的 window 及 step size 從影片的起始位置逐步滑動並將影片分割成多個片段；最後將多個片段分別進行處理或分析。示意圖如圖 10，其 window 為三，step size 為一。

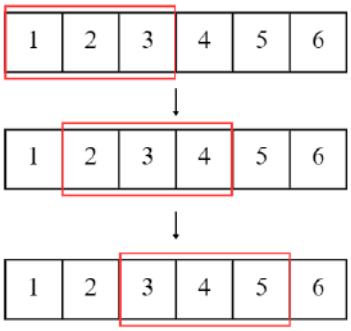


圖 10 Sliding Window 行進示意圖 (此圖為作者自製)

二、 籃球動作影片蒐集

使用手機作為攝影機。拍攝地點為學校活動中心。拍攝投籃、上籃相關影片時，攝影機架設位置為籃框下方，底線之外（如圖 11）。拍攝運球影片時則無特定地點。影像片段的收集，包含 11 名女性，其中 10 位慣用手為右手，1 位為左手。

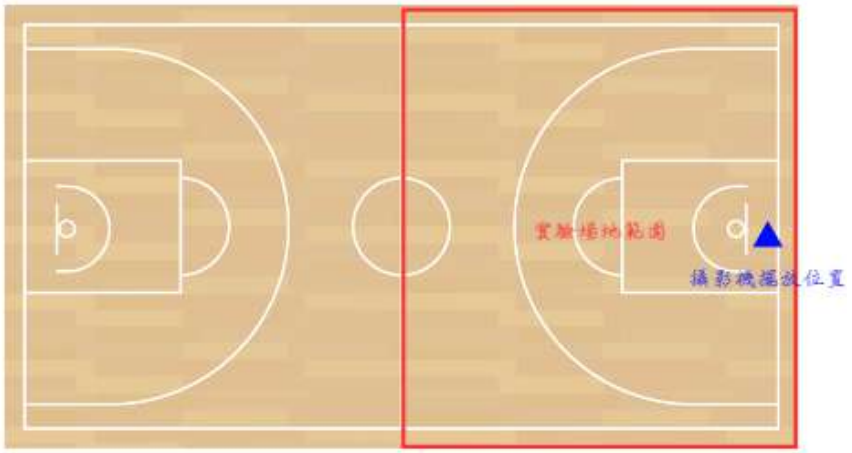


圖 11 影片蒐集場地示意圖 (此圖為作者自製)

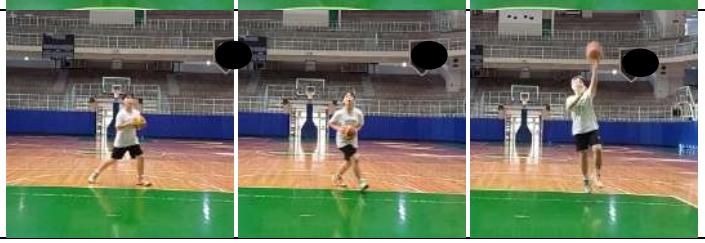
(一) 單一動作影片

為了辨識共 18 動作，我們共蒐集了 2600 個介於 1-3 秒間的影片，每個影片中有 1 個籃球動作。18 個動作類別如表 2 所示。

表 2 動作類別及示意圖（表中圖片皆為作者自行拍攝）

編號	動作	動作示意圖
0	背後運球 behind back	
1	胯下運球 Between legs	
2	換手運球 crossover	
3	拉桿 double clutch	
4	歐洲步 Europe step	
5	後仰跳投 fadeaway	
6	拋投 floater	

編號	動作	動作示意圖
7	左手運球 left hand dribble	
8	左手上籃 left hand layup	
9	急停跳投 pull up jump shot	
10	反手上籃 reverse layup	
11	右手運球 right hand dribble	
12	右手上籃 right hand layup	
13	投籃 shoot	

編號	動作	動作示意圖
14	轉身運球 spin	
15	轉身上籃 spin layup	
16	後撤步 step back	
17	踏跳 step hop	

(二) 連續動作影片

實驗人員在球場上隨機做運球、投籃等動作，時長約 30 秒，實驗人員出手後停止錄影（如表 3）。

表 3 連續動作影片示意圖（表中圖片皆為作者自行拍攝）

動作	示意圖
換手運球	
轉身運球	

動作	示意圖
反手上籃	

三、單一動作辨識系統

本研究訓練單一動作辨識系統更分為幾個部分，影片處理、模型選擇、模型預訓練及模型訓練。

(一) 籃球動作影片處理

我們將每個動作的影片分成訓練集及測試集，以利於後續模型的訓練。18 個動作的影片數目如表 4。

表 4 模型訓練資料集

編號	動作	訓練集(支)	測試集(支)
0	背後運球 behind back	96	41
1	胯下運球 Between legs	112	48
2	換手運球 crossover	66	42
3	拉桿 double clutch	94	40
4	歐洲步 Europe step	110	47
5	後仰跳投 fadeaway	86	37
6	拋投 floater	98	43
7	左手運球 left hand dribble	40	18
8	左手上籃 left hand layup	116	51
9	急停跳投 pull up jump shot	111	48
10	反手上籃 reverse layup	125	54

編號	動作	訓練集(支)	測試集(支)
11	右手運球 right hand dribble	43	18
12	右手上籃 right hand layup	120	50
13	投籃 shoot	139	60
14	轉身運球 spin	87	55
15	轉身上籃 spin layup	123	52
16	後撤步 step back	70	30
17	踏跳 step hop	113	48
總計		1749	782

(二) 選定適合模型

在 MMaction2 的模型中，我們選擇了 I3D、UniFormer V2、VideoSwin 及 MViTV2 共四個模型進行訓練，並比較個模型辨識籃球動作的準確率（結果詳見結果與討論）。

(三) 模型預訓練

我們使用 Kinetics-400 資料中已經經過預訓練（Pre-trained）的模型。Pre-trained 是指在大規模數據集上對模型進行的初步訓練，通過在大數據集上的訓練，模型可以學習到通用的特徵（如邊緣、形狀、紋理等），將其作為訓練的起點可大幅度降低訓練所需時間。Kinetics-400 是由 DeepMind 提出的超大型影片資料集，專門用來做動作辨識的研究。它包含了大約 30 萬段平均長度約 10 秒的影片，並涵蓋了 400 種不同的動作類別，常被模型用來作預訓練。

(四) 模型訓練

我們分別對模型進行了兩種微調（fine-tuning），分別為 Full Model fine-tuning 及 Head only fine-tuning 兩種微調方式。fine-tuning 是將預訓練的語言模型調整以適應特定任務的過程，透過使用新的數據集來更新模型的權重。與從零開始訓練不同，從零

開始訓練是隨機初始化模型，而微調則是在已有模型的基礎上進行調整，這些模型已經從大量的語料中學習過。

Full Model Fine-tuning 是指對模型的所有層進行參數更新，這意味著模型的每一層權重都能根據新資料進行調整。這種方法適用於需要高度準確性且與預訓練數據有顯著不同的特定任務。Head only fine-tuning 是一種特定的微調方法，專注於調整模型的輸出層，同時凍結其餘的層。這種方法因為只需訓練輸出層，所以訓練速度較快，適合時間緊迫的任務。且由於大部分參數保持不變，這種方法在數據量較少的情況下能有效減少過擬合的風險。它還能高效利用模型在大數據集上學到的知識，使其在特定任務上表現更佳，適用於訓練數據有限的情況下。我們也實驗兩種微調方式來進行模型訓練，來得知其表現。

四、 連續動作辨識系統

此系統目標為分析 20 秒左右的持球進攻長影片，並辨識出受試者作出的所有進攻動作。我們會將長影片用 Sliding window 分割為數個短影片，並將短影片丟入單一動作辨識模型進行動作辨識。我們設計了兩個實驗以求出最佳 window 大小及數目。

本研究的籃球動作主要分為兩大類別，運球及出手。每個運球的時間大約 1 秒，而出手的影片大約為 2~3 秒，因此若僅有一個 window，其大小皆相同，可能會導致影片分割效果不佳。我們設計了兩組實驗來探討最佳的 window 數目，第一組設置一個 window，第二個設置兩個 window，一個 window 大小較大，用來辨識出手動作，一個 window 大小較小，用來辨識運球動作(如圖 12)，並設計了不同 window 大小(如表 5)，找出最好的參數(結果詳見結果與討論)。

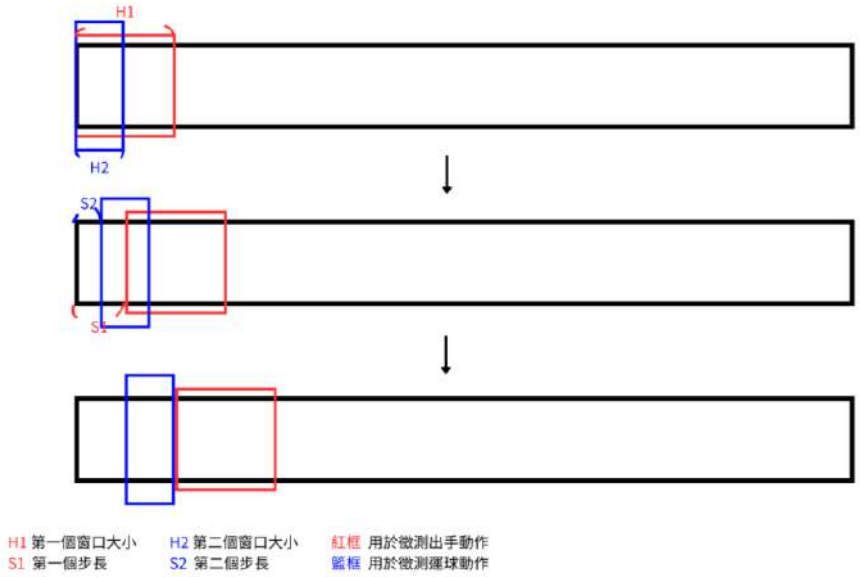


圖 12 兩個 window 的 Sliding window 示意圖(此圖為作者自製)

表 5 window 大小

	第一組 window size (frame)	第二組 window size (frame)	第三組 window size (frame)
一個 window	30	45	60
兩個窗口 window	20,50	30,60	40,70

五、 籃球慣用動作分析系統

此系統結合了連續動作辨識系統，使用者可藉由觀看實驗場地前投影幕中防守者的不同站位作出相應的籃球動作，並將手機架設前方同時進行錄影（如圖 13），此系統將即時列出使用者所作的籃球動作種類，並於錄影暫停時顯示此次的數據統計。

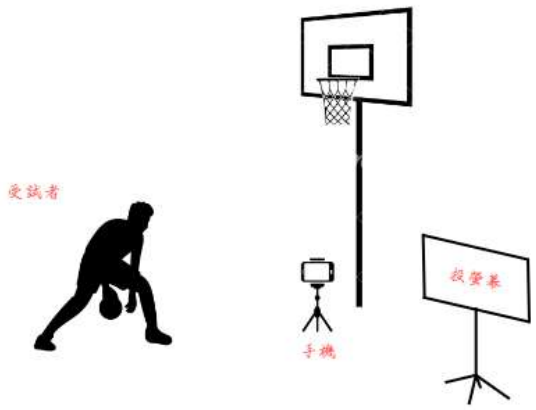


圖 13 使用情境示意圖 (此圖為作者自製)

(一) 使用者介面

本研究使用 PyQt 進行程式介面設計。PyQt 是基於跨平台圖形用戶界面（GUI）工具包 Qt 的強大 Python 框架，專門用來開發桌面應用程式。PyQt 為 Python 開發者提供完整的 Qt 功能，允許他們構建高效、美觀的桌面應用。表 6 將介紹我們使用到的功能。

表 6PyQt 功能介紹

PyQt Widget	功能描述	UI 中使用
QApplication	主應用程序類，所有 PyQt 程序都必須創建一個 QApplication 實例來運行應用程序。	
QMainWindow	應用程序的主窗口，所有的控件都放置在這個窗口內。	
QStackedWidget	允許將多個頁面堆疊在一起的控件，但在同一時間內只能顯示一個。可以透過程式控制來切換顯示不同的頁面	
QPushButton	可點擊的按鈕，執行特定動作	切換頁面、確認資料
QLabel	顯示靜態文字或圖片	展示文字、圖片
QLineEdit	用來讓使用者輸入單行文字的 PyQt 控件	用於輸入使用者名稱
QRadioButton	單選按鈕，在一組選項中只能選擇一個	選擇使用者性別（female、male、other）
QButtonGroup	用來將一組按鈕進行分組的控件。使按鈕之間會有互斥關係，確保在同一組中同一時間只能選擇一個按鈕	選擇使用者性別（female、male、other 只擇一）
QPixmap	用於處理圖片，QPixmap 加載圖片，QLabel.setPixmap() 顯示圖片。	處理並顯示拍攝到的使用者照片
QDateEdit	讓使用者選擇或編輯日期的控件，通常會顯示一個日期選擇框，並且允許使用者通過日曆選擇或者手動輸入日期	選擇使用者生日日期，讓用戶選擇日期並返回選擇結果
QFont	用於設置控件中文字的字體和大小。	所有文字
QIcon	用於在 QRadioButton 上設置圖像	選擇用者性別時的 male 和 female icon
QSize	用於設置圖標或圖片的大小	調整拍攝到的使用者照片在不同頁面中的大小

我們運用 MMaction2 所提供的 webcam demo 檔案，並串接於我們所設計的 PyQt 介面，來進行整個系統的操作，整體流程為（介面如圖 14）：

1. 獲取影像：使用 OpenCV 開啟攝影機並獲取影像幀。
2. 預處理：對於每個捕獲的幀，預處理影像以符合動作辨識模型的輸入要求。
3. 模型辨識：使用 MMAction2 所訓練完畢的模型來 Inference 以獲取動作辨識結果。

4. 顯示結果：更新 GUI 畫面，以顯示影像序列以及動作辨識的結果。

圖 14 使用者介面示意圖 (此圖為作者自製)

(二)防守影片錄製

我們錄製了三個不同狀況的防守影片，分別為：防守者壓迫防守，並留出右邊空檔、防守者壓迫防守，並留出左邊空檔以及防守者退一步防守（如表 7）。並記錄受試者在面對不同防守狀況時，常做籃球動作的分布圖，讓受試者能夠檢視自己在進攻時是否有根據防守者的佔位選擇進攻方式。

表 7 防守種類及示意圖（表中圖片皆為作者自行拍攝）

種類	示意圖
防守者壓迫防守，並留出右邊空檔	
防守者壓迫防守，並留出左邊空檔	

種類	示意圖
防守者退一步防守	

伍、 研究結果與討論

一、 選定適合的動作辨識模型

以下將分別分析 I3D 、UniFormer V2、VideoSwin 及 MViT V2 的訓練及辨識結果。其中 acc/top1 為模型在 top1 結果就辨識正確的比率；Acc/top5 為正確結果出現在 top5 以內的比率。

(一)I3D

I3D 辨識的正確率沒有達到預期的效果（如表 8），透過影像特徵可視化的 GradCAM 方法，發現 I3D 確實有辨識到持球者，然而當錄製背景不同時辨識結果便會出錯（如圖 15），也就是說 I3D 容易受背景的影響，導致訓練效果不佳。

表 8 I3d 辨識模型準確率

	Acc/top1	Acc/top5
Full model	64.79	89.64
Head only	58.22	87.07

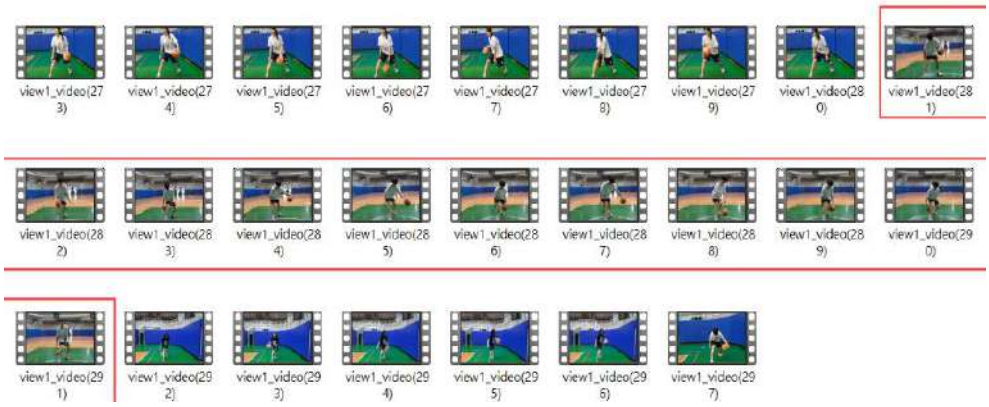


圖 15 受背景影響示意圖(此圖為作者自製)

I3D 的混淆矩陣（Confusion Matrix）如圖 16。其中有四個動作容易辨識錯誤，經

過排查，我們發現了動作辨識不佳的原因（如表 9）。

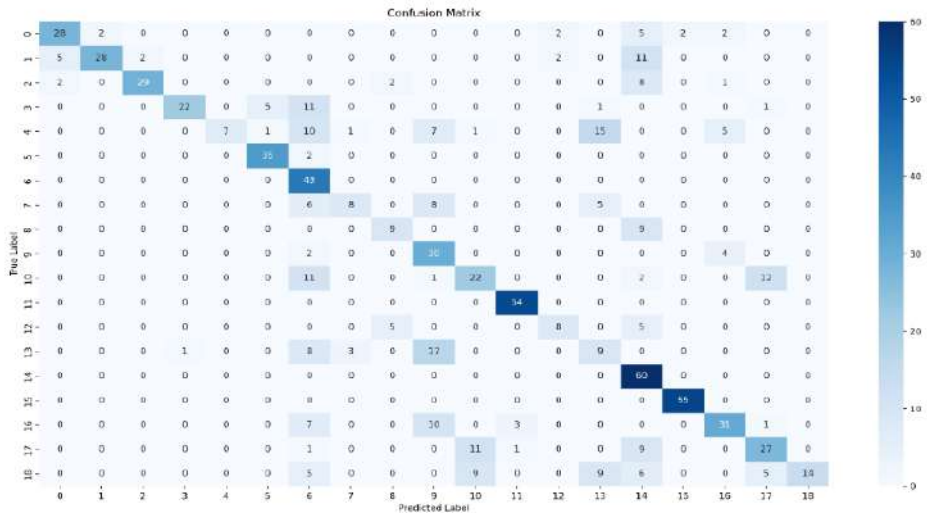


圖 16 I3D 的 Confusion Matrix (此圖為作者自製)

表 9 I3d 辨識結果分析

預設動作	Model 預測動作	預測原因
勾籃	左手上籃 右手上籃	實驗人員勾手動作不明顯，導致 model 認為是一般上籃。
拉桿	拋投	實驗人員手部動作像拋投，但應該以腰部為重點。
歐洲步	拋投 左手上籃 右手上籃	實驗人員作的歐洲步腳步不夠明顯，導致 model 判斷成一般上籃。
後撤步	急停挑投 投籃	後撤步的前進和後退做得不夠明顯，因此腳步看起來像急停跳投或單純的投籃。

(二)Uniformer V2

Uniformer V2 的訓練結果如表 10。Uniformer V2 因為模型太龐大，參數量太多，需要很多時間才能收斂，即使已經訓練至 100 Epochs，辨識結果仍然很不理想。

表 10Uniformer V2 模型辨識準確率

	Acc/top1	Acc/top5
Full model	0.4320	0.8365
Head only	0.3854	0.8002

(三)VideoSwin

我們使用 VideoSwin 共進行了兩次訓練，其結果如表 11。

表 11VideoSwin 模型辨識準確率'

	第一次		第二次	
	Acc/top1	Acc/top5	Acc/top1	Acc/top5
Full model	0.7235	0.9709	0.7253	0.9713
Head only	0.6512	0.9583	0.6529	0.9588

第一次實驗後，我們從該模型的 Confusion Matrix（圖 17）發現三項容易被誤判的動作，並分析誤判原因（如表 12）。

表 12VideoSwin 辨識結果分析

預設動作	Model 預測動作	推測原因	解決方案
後撤步	急停跳投	兩者動作從正面觀看過於相似、實驗人員動作不標準	重新錄製後撤步測資
急停跳投	後撤步		
左手上籃	右手上籃	模型訓練時會將影片翻轉以增加訓練資料集的多樣性，但這樣的操作反而導致無法分辨左右	將訓練時資料增量的翻轉（flip）操作從模型中移除

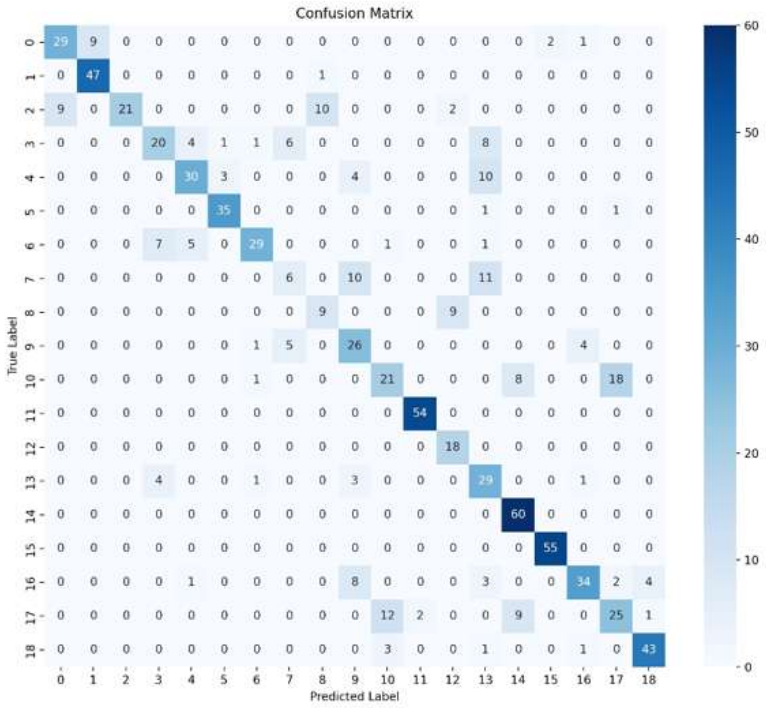


圖 17VideoSwin 的 confusion matrix (此圖為作者自製)

在進行上述更改後我們進行了第二次的實驗，其中準確率由 0.7235 上升到 0.7253，與第一次的實驗並沒有太大的區別。我們在 Confusion Matrix 中（如圖 18），發現如實驗一的情況並沒有減少，雖然左右手的上籃、後撤步和急停跳投已能正確分辨，但仍出現拉桿、歐洲步、轉身上籃被誤判成右手上籃的情況。

(四)MViT V2

MViTV2 的訓練結果如表 13，confusion matrix 如圖 19。

表 13 MViTV2 模型辨識準確率

	Acc/top1	Acc/top5
Full model	0.8913	0.9719
Head only	0.8042	0.9595

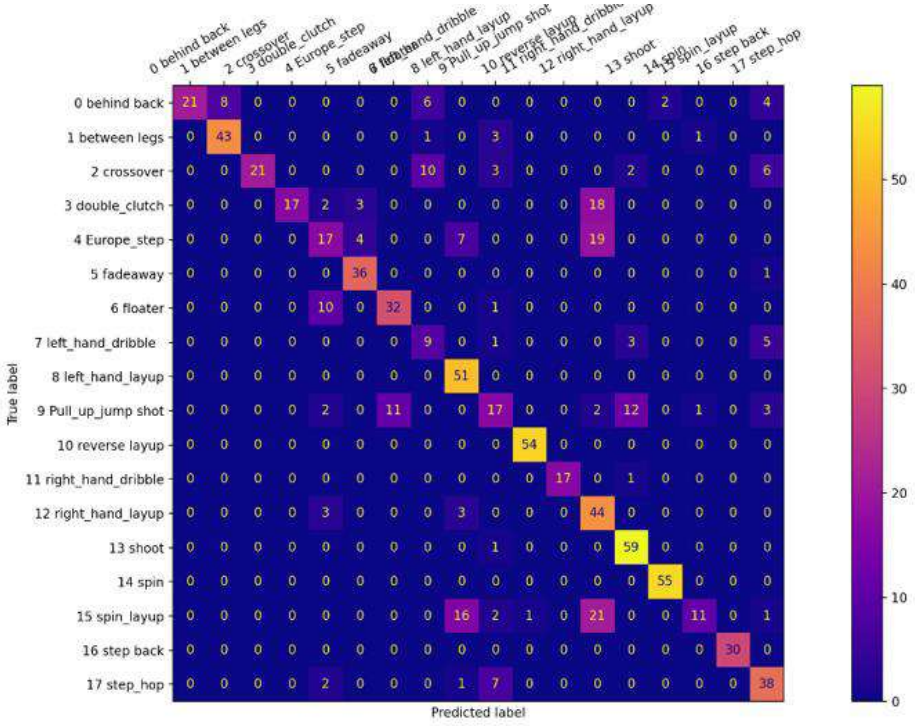


圖 18 VideoSwin 的 confusion matrix (此圖為作者自製)

經過上述的實驗及分析，四個模型的辨識結果如表 14。在四個模型中用 Full mode 進行 finne-tuning 效果皆比 Head only 理想。其中以 MViT V2 的辨識效果最好，因此我們將 MViT V2 選為單一動作辨識系統的辨識模型。

表 14 個模型辨識結果

Model name	I3D		Uniformer V2		Video swin		MViT V2	
	Acc/top1	Acc/top5	Acc/top1	Acc/top5	Acc/top1	Acc/top5	Acc/top1	Acc/top5
Full model	64.79	89.64	0.4320	0.8365	0.7253	0.9713	0.8913	0.9719

Head only	58.22	87.07	0.3854	0.8002	0.6529	0.9588	0.8042	0.9595

二、 連續動作辨識

(一) Window 正確率算法

我們將辨識過後的影片使用人工檢查，並依照表 15 規則進行計算

表 15 計算連續動作辨識準確率方法

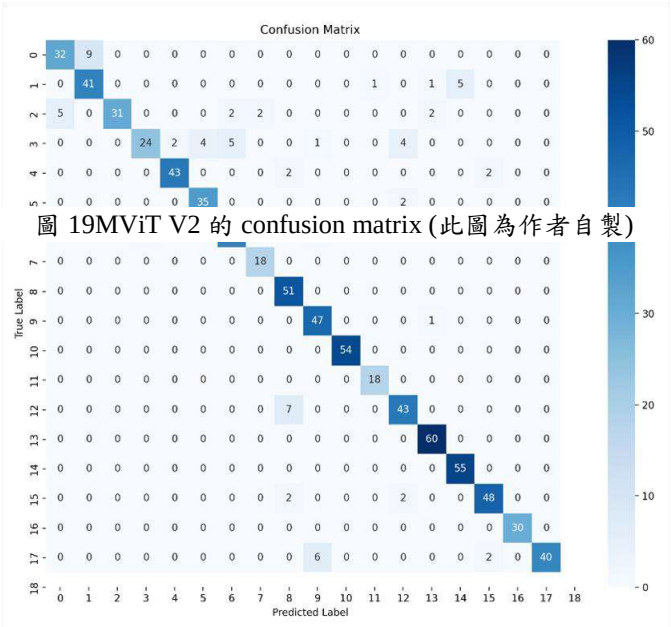


圖 19MViT V2 的 confusion matrix (此圖為作者自製)

辨識情況	計算方式
辨識結果與實際相同	正確
辨識結果與實際不同	錯誤
應辨識未辨識	錯誤
辨識結果正確但延遲過久	錯誤

表 16 為我們使用一個 window（45 frames）和兩個 window（[30,60]frames）對一影片進行辨識的結果。

表 16 辨識方式範例

一個 window (45 frames)		兩個 window ([30,60] frames)		辨識結果
Predicted label	計算方式	Predicted label	計算方式	True label
Right hand dribble	正確	Right hand dribble	正確	Right hand dribble

Double clutch	錯誤	Crossover	正確	Crossover
Left hand dribble	正確	Left hand dribble	正確	Left hand dribble
Spin	錯誤	Spin layup	錯誤	Crossover
Spin	正確	Spin	正確	Spin
Pull up jump shot	錯誤			
Behind back	錯誤	Behind back	錯誤	
Left hand dribble	錯誤	Left hand dribble	錯誤	Right hand dribble
Right hand dribble	正確	Right hand dribble	正確	
Shoot	錯誤	Crossover	正確	Crossover
Left hand dribble	正確	Left hand dribble	正確	Left hand dribble
Crossover	正確	Left hand dribble	錯誤	Crossover
Right hand dribble	正確	Right hand dribble	正確	Right hand dribble
Left hand layup	錯誤	Crossover	正確	Crossover
Pull up jump shot	錯誤	Pull up jump shot	錯誤	Right hand layup
正確率(%)	46.67		64.29	

(二) window 大小

我們用 10 段影片進行測試，表 17 為我們的實驗結果。其中我們發現使用 1 個 window、window 大小為 30 frame 時的辨識效果最佳。

表 17 窗口大小實驗結果

單一 window 大小(frame)	30	45	60
辨識準確率(%)	64.37	50.62	44.75
兩個 window 大小(frame)	[20,50]	[30,60]	[40,70]
辨識準確率(%)	61.44	57.43	51.86

雖然目前連續動作辨識結果準確率僅在 61%左右，原因是不管是單一還是兩個 Window 都有可能無法完整涵蓋到單一動作單元，而模型會傾向於硬是辨識該片段為某個動作，而容易有誤判的狀況發生。因此，我們採用兩個過濾策略來進行排除，第一個策略是將預測信心值（Confidence）較低的預測結果進行排除，維持前一個 Window 的預測結果，第二個策略為第一個和第三個 Window 及第四個 Window 間的辨識結果如果不同，則會有修正機制，將第二個 Window 修正為跟其他相同。透過該機制來進行處理後，將結果回傳至 GUI 能得到較正確之結果，來修正問題。

陸、 結論

在本研究中，我們成功建立了一個籃球慣用動作辨識系統，透過自行蒐集 18 種籃球員進攻動作的影片，並使用 MMAction2 資源庫裡面最新穎的動作辨識模型，來訓練動作辨識模型，完成系統中的慣用動作辨識模型。為了瞭解球員在進攻時的慣用動作，並以圖表和數據的方式呈現，我們使用 PyQt 來設計一個介面，將動作辨識模型的結果

得以進行統計及呈現。此外，為了完整分析球員在一次進攻中所會進行的慣用動作為何，我們也針對連續動作採用 Sliding Window 後再進行動作辨識的方式，來分析連續動作序列影片時的慣用動作。具體結論條列如下：

一、 單一動作辨識模型

(一) 在 MMaction2 的四個模型：I3D 、UniFormer V2、VideoSwin 及 MViT V2 中 MViT V2 來進行動作辨識，可以得到的效果最好。

(二) 在 Fine-tuning 的實驗中，用 full model 進行 fine-tuning 的效果比用 Head only 來的理想，能讓模型較快收斂，也較快達到比較好的結果。

二、 連續動作辨識模型

(一) 在使用 Sliding window 分割影片時，單一 window(30 frames)的效果最好，在採用過濾的策略後，能將辨識結果信心指數沒這麼高的結果排除，來避免誤判的結果。

三、 籃球慣用動作分析系統

(一) 使用 PyQt 設計出一個介面，讓使用者可以進行登錄，並在介面上展示拍攝影像、分析結果及相關圖表，方便使用者即時查看分析內容。

(二) 成功將動作辨識模型整合至 PyQt 介面，實現即時動作辨識，並統計球員的慣用動作，構建完整的智慧分析系統。

柒、 未來展望

我們的研究目前進行到一半，尚有地方可以精進，於 2024 年 11 月到 2025 年 1 月會繼續完善作品。以下為這段期間我們會持續探討的問題。

一、受試者可自由移動

在現有系統中，受試者（進攻者）只能在原地進行運球。因此我們希望模型能增加辨識距離的功能，讓受試者可以邊移動邊運球

二、模型準確率再提升

動作辨識模型的準確性是系統成功的關鍵因素之一，然而，目前所使用的 MViT V2 模型在長影片的辨識能力上仍有不足，無法達到我們預期的準確率。為了改善這個情況，我們希望透過調整和擴充訓練集，提高模型的準確率。我們計畫重新錄製一部份訓練集影片，特別針對那些在長影片中較難辨識的動作進行精細的錄製，使模型更有效的抓捕捉動作的細節。另外，我們希望能增加訓練集影片的數量以豐富數據集，透過增加更多的實驗人員、在不同的場地拍攝以及不同的拍攝角度，讓模型接觸到更多樣的動作變化和背景場景，從而提高其泛化能力。

三、結合 VR 技術

我們希望引入虛擬實境（Virtual Reality，簡稱 VR）技術，以便讓受試者在沉浸式的虛擬環境中進行訓練，模擬真實的籃球對抗情境。與傳統的螢幕觀看方式相比，VR 提供更高的真實感，使受試者能夠更有效地識別和發展其慣用的籃球動作。在 VR 環境中，受試者能夠進行即時互動，例如在虛擬場景中觀察防守者對進攻的反應並根據這些反應進行相應的進攻行為，這種互動性極大地提高了真實性和有效性。

為了實現 VR 與動作辨識的整合，我們計劃在 VR 設備上實施姿態估計和位置追蹤技術。這樣一來，進攻者的動作數據可以即時傳送至系統，確保防守角色能夠做出迅速且準確的反應。這一過程需要低延遲的技術支持，以確保動作的流暢性和即時互動效果。此外，我們還可以考慮在 VR 訓練中引入多樣化的場景，例如模擬不同風格的防守者，或者變換訓練場景和條件，以進一步觀察受試者的反應能力及慣用動作。

四、結合雲端運算處理以推廣至 App 或網頁

將系統結合雲端運算技術，以實現大規模數據處理與低成本的擴展應用，使更多用戶通過 App 或網頁即可享受系統的功能。目前我們的系統仍須依賴具有 GPU 電腦進行，但透過雲端處理，模型的計算與推理不用再依賴本地設備，而是在伺服器端完成，減輕了用戶端的硬體需求。不僅能支援更複雜的模型運行，還能確保系統在不同裝置

間的同步性與穩定性。雲端化的系統也可以輕鬆進行資料更新和模型升級，用戶無需手動更新應用程式即可自動享受最新的技術改進。此外，雲端架構可以支援即時的多用戶協作，或甚至可以構建出一個虛擬的訓練社群，讓使用者能夠分享訓練成果。使用者只需透過行動裝置或電腦便可輕鬆接入系統，不受地點和設備的限制。透過雲端運算，系統不僅能在使用者端更高效運行，也能實現即時性、擴展性與便利性的統一。

捌、 參考資料

[1] 卷積神經網路簡介：什麼是機器學習?
https://anstekadi.com/Article/Detail/3309

[2] https://github.com/open-mmlab/mmaction2

[3] Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.* 2018, *41*, 2740–2755.

[4] Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[5]Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, Yu QiaoUniFormerV2: Spatiotemporal Learning by Arming Image ViTs with Video UniFormer

[6]Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, Han Hu; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 3202-3211

[7]Yanghao Li; Chao-Yuan Wu; Haoqi Fan; Karttikeya Mangalam; Bo Xiong; Jitendra Malik. MViTv2: Improved Multiscale Vision Transformers for Classification and Detection

【評語】190006

本作品使用 MMaction2 開源工具包內的 I3D 、UniFormer V2、VideoSwin 及 MViT V2 等四種模型來做 fine-tuning，以辨識 18 項籃球動作。

研究中蒐集了 2,600 個介於 1-3 秒間的影片, 發現辨識率以 MViT V2 最好，可達 top1 0.89，但連續動作辨識率僅達 0,61~0.65。

研究過程，基本上實驗評估多個模型尚屬嚴謹、敘述清楚。

以下為建議供作者參考:

1. 連續動作若有多個動作應可包括召回率。
2. 可考慮使用 Contrastive Learning(對比式學習)等模型訓練機制。
3. 若可加強說明比較所提出方法與現有常見方法的差異、或本作品創新處則更佳。