

基於電腦視覺的拳擊競賽即時計分輔助系統

A Boxing Competition Real-Time Scoring Assistance System Based on Computer Vision

王能中 胡凱瑋 林時瑋 蕭鈞永 洪銘佑

國立聯合大學資訊工程學系

苗栗市恭敬里聯大二號

ncwang@nuu.edu.tw

{U1024036, U1024055, U1024054, U1043038} @o365.nuu.edu.tw

摘要

全球技擊競賽中，誤判事件頻繁發生，對運動員的成績與職涯造成嚴重影響。特別是在臺灣，裁判系統缺乏更新機制，裁判水平參差不齊，進一步阻礙了選手的發展，導致觀眾對競賽結果的信任度降低，影響體育運動的吸引力與公信力。本研究旨在開發一套輔助裁判工作的系統，以提升競賽的公平性與透明度。系統結合 YOLOv8 模型進行邊緣偵測，並透過多鏡頭技術實現得分判定，最終利用 RTMP 與 HLS 串流技術將競賽畫面及分數即時傳送至網頁，記錄競賽過程，並提供直播及歷史競賽的觀看功能；本系統所使用之套件皆為開源程式碼，開發成本較低且有彈性。本系統能有效降低誤判率，進一步提升技擊類競賽的公信力以及公正性。

關鍵字：自動判分系統、拳擊競賽、公平競爭、誤判、即時、裁判系統。

Abstract

Misjudgments are common in global combat sports competitions, significantly impacted athletes' performance and careers. In Taiwan, the lack of a referee replacement mechanism and the inconsistent quality of referees further hinder athletes' development. These issues weaken public trust in competition outcomes and undermine the appeal and credibility of sports events. This study aims to develop a system to assist referees in enhancing fairness and transparency in competitions. The system integrates YOLOv8-based edge

detection and multi-camera techniques for scoring determination. It utilizes RTMP and HLS streaming technologies to deliver live and recorded match content to web platforms, enabling users to view both live broadcasts and historical matches. The system is built entirely with open-source software, offering cost-effective and flexible development. This system is expected to reduce the occurrence of misjudgments and elevate the overall quality and competitiveness of combat sports.

Keywords: automated scoring system, boxing competition, fair competition, misjudgment, real-time, referee system.

1. 簡介

在本系統中，我們設計了一套基於電腦視覺的拳擊競賽即時分析系統，為拳擊競賽提供更多輔助工具，減少裁判的主觀判斷，降低誤判和爭議的可能性。

1.1 研究背景

在全球範圍內，技擊類(例如拳擊、散打搏擊等)的競賽中，誤判事件層出不窮，這些誤判不僅對運動員的成績造成影響，更可能阻礙有潛力的年輕選手的職業生涯發展，也損害了觀眾對競賽結果的信任，從而削弱了體育運動的吸引力與信譽。

在場邊的裁判則會因為場地(例如拳擊擂台的邊繩、邊角等)以及心理狀態等因素，導致他的判分不夠公正，進

而導致真正獲勝的選手反而被判輸。這種情況在實力相近的競賽中特別常見，以拳擊為例，甚至會有三個裁判判出來的分數都可能不一樣，如表1所示。這種判決的不一致性進一步削弱了競賽的公正性和觀眾的信任度。因此，針對這些問題，我們需要一套能夠提供更公正判斷的系統來輔助裁判工作，確保競賽的公平性和透明度，並提升觀眾對競賽結果的信心。

表1：世界級的競賽仍有判分不同的情況。

WBC 重量級冠軍爭奪賽(2023 年10 月28 日)					
Tyson Fury vs Francis Ngannou					
裁判一		裁判二		裁判三	
96	93	95	94	94	95
Tyson Fury win		Tyson Fury win		Francis Ngannou win	

1.2 研究目的

為了有效解決上述問題，我們的研究目標是設計並開發一套能夠讓所有參與者，包括裁判、教練和選手都信賴的自動判分系統。這個系統應具備一定精確度和公正性，能夠取代或輔助現有的人為判決，從而提高競賽的公正性及透明度，並促進運動員的公平競爭環境。具體而言，我們希望這個系統能夠實時辨識競賽過程中的影像，並給出準確的得分判斷，從而減少人為誤判的機會。

1.3 相關研究

在技擊類競賽的分析與判分輔助領域，已有多項研究與系統如：拳擊分析系統[1]、The Automation of Computer Vision Applications for Real-Time Combat Sports Video Analysis (RTCSVA) [2]、DeepStrike 系統[3]及 Combat IQ [4]探索如何利用電腦視覺技術來提升競賽的公平性與效率。以下將比較三個具代表性的研究與系統，以及本專題系統的創新與差異：

a.拳擊分析系統

「拳擊分析系統」[1]是由國立清華大學運動科技中心所開發，此系統聚焦於競賽賽後的技術統計與亮點剪輯，為教練與選手提供戰術分析依據。系統依靠AI 自動篩選關鍵片段，但結果仍需人工參與解讀。

該系統雖對訓練與策略制定有幫助，但缺乏即時判分功能與自動化數據輸出。

b. RTCSVA

Quinn 與 Corcoran 提出了一個基於 YOLOv5 與姿態估計(HAR)的即時動作分析系統，專注於拳擊、MMA 和人對拳擊袋等多項格鬥運動的分類與追蹤。其系統成功應用於分析運動員的動作(如刺拳、踢擊、靜止)，並生成運動表現統計數據。該系統的強項在於多運動場景的適用性與即時動作分類的準確性(平均準確率達 95.5%)。然而，研究的側重點是動作分析與數據統計，未針對競賽裁判輔助功能進行優化，也未考慮實際應用中的多鏡頭協作問題。

c. DeepStrike 系統

DeepStrike [3]是 Jabbr.ai 開發的一套格鬥運動 AI 分析系統，據稱能進行即時動作分析與數據生成，包括拳擊命中率、力量統計、距離控制與選手表現指標(如平衡與攻擊節奏)。然而，該系統缺乏公開的詳細技術資料與實際測試結果，因此即時性與準確性尚待驗證；該系統有較多的數據統計及分析，數據統計和串流媒體類似，只要按下按鈕即可開始進行。

d. Combat IQ

Combat IQ [4]是一家專注於格鬥運動數據分析的科技公司，旨在提升拳擊、MMA 等技擊類競賽的公正性與觀賞價值。它運用先進的機器學習模型，結合視覺識別和動態追蹤技術，能即時捕捉並顯示關鍵數據，包括打擊速度、準確度、選手間距離、踢擊高度、纏鬥時間，以及選手身體部位的擊中記錄等。

本系統結合多項技術，特別是多鏡頭、多行程協作、YOLOv8 的 Instance Segmentation 模型以及基於 RTMP 串流的即時分數視覺化展示，確保使用者能即時獲得準確分數判定，同時提升使用者的觀賽體驗。相比其他系統，我們的重點在於網頁直播即時呈現、低成本應用及程式碼開源，同時找尋相關研究並列出優缺點，如表2所示。

表2：相關系統比較。

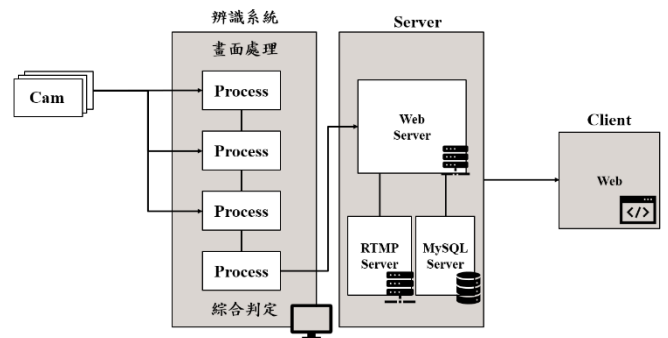
系統名稱	即時計分	程式碼 開源	目標用戶	應用場景
拳擊競賽 即時計分 輔助系統	有	是	- 裁判 - 觀眾 - 選手	- 競賽 - 訓練
拳擊分析系統 [1]	無	否	- 教練 - 選手	訓練
RTCSVA[2]	無	否	選手	訓練
DeepStrike 系統[3]	有	否	- 教練 - 裁判 - 觀眾	- 競賽 - 訓練
Combat IQ 系 統[4]	有	否	- 裁判 - 觀眾	- 競賽 - 訓練

2. 系統內容

本系統使用 YOLOv8 的 Instance Segmentation 模型來實作。使用者前端的部分利用了 XAMPP 架設網頁伺服器，後端則是使用 Nginx 來架設 MySQL 伺服器和 RTMP 伺服器。MySQL 伺服器用來存取資料，RTMP 伺服器則是提供串流服務；透過辨識系統即時判定擊中後將結果傳至網頁後端和資料庫來進行即時更新。

2.1 系統架構

使用者前端為網頁端；後端包含資料庫、RTMP 伺服器及「辨識系統」。後端電腦(PC 1)從三個鏡頭取得畫面後，透過三個行程進行邊緣分割處理，辨識完成後將三個不同的結果透過第四個行程進行綜合判定，此會以多數決的方式來決定此幀是否有擊中，其中的電腦傳輸方式會使用 Python 撰寫的 WebSocket 來收送資料。在 PC 1 上使用 FFmpeg 將直播畫面推流(Push Stream)到 RTMP Server 上，網頁端則會從 RTMP Server 上以 HLS 的協議將直播畫面拉流(Pull Stream)下來呈現，系統架構圖如圖一所示。



圖一：系統架構圖。

2.2 開發環境

本系統的開發環境包含 Python、Visual Studio Code、Spyder、TypeScript 及 JavaScript，說明如下。

2.2.1 Python

Python [5] 是多範式程式語言，它完全支援結構化程式和物件導向程式設計，同時支援多種套件。本系統辨識分析及資料傳輸都是使用此程式語言撰寫。

2.2.2 Spyder

Spyder [6] 是 Python 的整合開發環境，Spyder 包含了許多不同的開源套件。本研究 Python 開發皆透過 Spyder 來進行。

2.2.3 Visual Studio Code

Visual Studio Code [7] 是一個開源且免費的編輯器，可以支援幾乎所有程式語言。本系統使用 Visual Studio Code 來完成網頁的開發。

2.2.4 TypeScript

TypeScript [8] 一個基於 JavaScript 的超集合 (SuperSet)，提供型別系統 (Type System)，能夠在開發時期宣告型別。本系統使用 TypeScript 來完成網站後端的開發。

2.2.5 JavaScript

JavaScript [9] 是一個的基於原型 (Prototype-based programming) 的、多範型的、動態語言，支援物件導向及指令式，它以作為網頁的腳本語言而大為知名。本系統使用 JavaScript 來製作網站前端。

2.3 相關技術

2.3.1 yt_dlp

yt_dlp [10] 是一個強大的開源工具，用於從網站下載影片、音樂或其他多媒體內容。本系統利用 yt_dlp 來下載影片作為測試素材，以供後續影像處理與辨識。

2.3.2 subprocess

subprocess [11] 是 Python 的一個標準庫，允許程式在執行過程中啟動子進程並與其進行交互。在本系統中，subprocess 用來執行外部命令，像是觸發 FFmpeg 進行影片處理，實現自動化的影片格式轉換、解析度調整等操作。

2.3.3 FFmpeg

FFmpeg [11] 是一款開源的影片處理工具，提供了高效的影片解碼、編碼、轉換及編輯功能。本系統中使用 FFmpeg 處理影片素材，統一測試影片格式。

2.3.4 moviepy.editor

moviepy.editor [12] 是一個 Python 影片編輯套件，提供多種簡單的影片處理功能。在本系統中，moviepy.editor 用來對測試影片進行畫質調整，包括調整影片解析度、幀速率等，以確保輸入影像資料的格式。

2.3.5 OpenCV

OpenCV [13]是由英特爾(Intel)公開發行用來進行圖像處理、電腦視覺的套件，本系統透過 OpenCV 來進行影像的擷取、畫面的繪製。

2.3.6 labelme

labelme [14] 是一個圖像標註工具，專門用於手動標註影像資料，對於訓練深度學習模型具有重要意義。在本系統中，使用 labelme 來標註影像中的物件邊緣，為後續的物件辨識模型提供高品質的訓練資料。

2.3.7 YOLOv8

YOLOv8 [15]是即時物件辨識模型。YOLOv8 基於先前 YOLO 版本的改進，引入了新功能和優化；YOLOv8 模型是基於卷積神經網路(CNN)的方式進行，其中使用 CSPDarknet 實作 [16]；在本系統中使用了 Instance Segmentation 模型來取得選手的邊緣來進行辨識處理。

2.3.8 Nginx

Nginx [17]是非同步框架的網頁伺服器，也可以用作反向代理、負載平衡器和 HTTP 快取。它是一款面向效能設計的 HTTP 伺服器。本系統使用 Nginx 來架設 RTMP Sever 及 MySQL Server。

2.3.9 Real-Time Messaging Protocol (RTMP)

即時資訊傳輸協定(RTMP) [18]是一種基於傳輸控制協定(TCP) 技術的網路協定或系統，用於在網際網路上串流內容，也是一種被廣泛使用的串流媒體格式。本系統使用此網路協定將處理完的畫面推流到伺服器上供後續網頁直播畫面所用。

3. 系統功能

本章介紹本系統的主要功能，整體架構可分為「判分輔助系統」與「前端系統」兩大子系統。「判分輔助系統」旨在透過多鏡頭畫面同步處理技術與影像辨識模型，實現競賽自動化得分判定。該系統利用邊緣偵測與分類技術取得辨識所需資料，並進行綜合判定。「前端系統」則提供直觀的使用者介面，包含直播回放、即時串流及會員互動功能，讓觀眾能透過多視角呈現與即時比分顯示提升觀賽體驗。此外，會員系統強化了用戶間的交流與參與感，使競賽的娛樂性和社群互動性大幅提升。本章將詳細說明兩大子系統的設計理念、運作流程及核心功能。

3.1 判分輔助系統

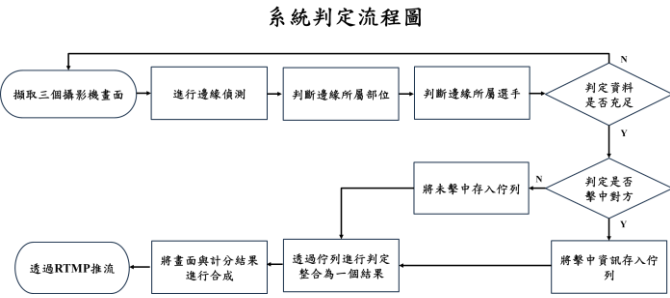
本系統啟動後，將截取三個鏡頭的畫面，接著進行邊緣偵測。偵測範圍涵蓋兩位選手的身體、拳套及頭部，並將偵測出的邊緣透過 `class_id` 進行分類。

分類完成後，系統以紅、藍兩色為主體，並透過 HSV 的 Hue(色相)作為判定，區分邊緣所屬的選手。

在成功區分選手後進行擊中的判定。判定邏輯為檢測雙方選手的拳套邊緣是否與對方的身體邊緣或頭部邊緣發生相交。若相交，系統會將擊中資訊加入佇列(Queue)，等待進一步的集合判定。

集合判定主要是將三個鏡頭的所提供的資訊來做判定，資訊包含「未擊中」、「資料不足」、「紅擊中藍身體」、「紅擊中藍頭部」、「藍擊中紅身體」、「藍擊中紅頭部」，在佇列中根據這些狀態進行多數決整合判定，判斷邏輯為：扣除「資料不足」的佇列，剩餘的佇列狀態如果有「紅擊中藍身體」、「紅擊中藍頭部」、「藍擊中紅身體」、「藍擊中紅頭部」以上三個狀態統計結果有過半，就存入資料庫，

並顯示在畫面上方，同時透過 FFmpeg 推流至 RTMP Server 上，本系統的判分輔助系統的流程圖如圖二所示。



圖二：判分輔助系統流程圖。

3.2 前端系統

前端系統包含了「直播回放系統」、「現場直播系統」和「會員系統」，網頁架構圖如圖三所示。

3.2.1 直播回放系統

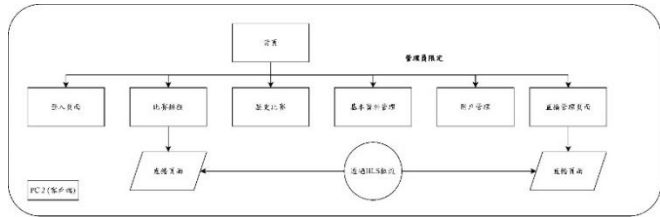
直播回放系統讓用戶能夠瀏覽所有過去競賽的完整影片。

3.2.2 現場直播系統

現場直播系統提供即時串流的競賽畫面，畫面包含三個不同視角，同時即時顯示競賽比分，為觀眾帶來更加身歷其境的觀賽體驗。

3.2.3 會員系統

會員系統提供會員註冊與登入功能。用戶登入後，可在聊天室中發表對競賽的評論，提升觀看競賽的參與感與交流樂趣。



圖三：網頁架構圖。

4. 系統實作成果

本系統的系統實作成果包含「前端」及「辨識系統」，說明如下。

4.1 前端

前端包含了「網站首頁」、「管理員介面」、「競賽管理」、「用戶管理」、「使用者介面」、「登入、註冊」、「會員資料修改」、「競賽排程」、「歷史競賽」及「聊天室」。

4.2 網站首頁

網站首頁包含系統介紹、拳擊規則以及開發人員，如圖四所示。



圖四：網站首頁。

4.3 管理員介面

管理員介面包含競賽管理、直播管理及用戶管理。

4.3.1 競賽管理

競賽管理介面可以查詢競賽標題、新增競賽、更改競賽標題、選手名稱、競賽時間、競賽狀態以及刪除競賽，如圖五所示。

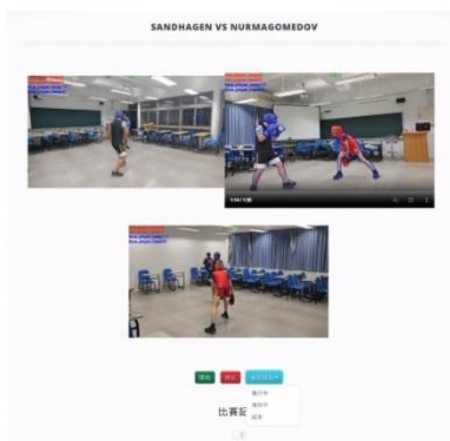
id	比賽標題	開始時間	狀態	操作
14	Vera vs Santhogen	2024-11-25 10:30:00	waiting	編輯 刪除
13	Santhogen vs Nurgamomedov	2024-11-22 23:20:00	proceeding	編輯 刪除
8	Jones vs Wendi	2024-10-28 10:40:00	waiting	編輯 刪除
7	Muhammad vs Pineda	2024-10-27 16:00:00	end	編輯 刪除
6	Toguri vs Makhachev	2024-10-25 14:30:00	end	編輯 刪除
5	Pantoja vs Dvalishvili	2024-10-23 10:30:00	waiting	編輯 刪除
10	Muhammad vs Khabibov	2024-10-10 07:00:00	waiting	編輯 刪除
9	Kari vs Figaev	2024-10-09 10:00:00	waiting	編輯 刪除
12	Jones vs Wendi	2024-10-09 00:00:00	end	編輯 刪除
11	Machado Garry vs Buckley	2024-10-01 08:00:00	end	編輯 刪除

圖五：競賽管理頁面。

競賽狀態共有三個選項，分別是 proceeding、waiting、end。proceeding 代表競賽在進行中，使用者可以進入這個直播間，waiting 代表競賽還在等待中，使用者無法進入直播間，end 代表競賽已結束，狀態為 end 的競賽會保存到歷史競賽頁面

4.3.2 直播管理頁面

直播管理頁可以進行開始直播、停止直播以及更改競賽狀態等操作，如圖六所示。點擊開始後，前端會使用 WebSocket 傳遞競賽開始的訊息到後端，後端再透過 WebSocket 傳遞訊息到判分程式上，開始判分以及串流。



圖六：直播管理頁面。

4.3.3 用戶管理

用戶管理可以查詢用戶名稱、編輯用戶以及刪除用戶。編輯用戶可以更改用戶名稱、帳號、密碼、啟動狀態以及權限。

4.4 使用者介面

使用者介面包含登入、註冊、修改會員資料、競賽排程、觀看直播以及歷史競賽。

4.4.1 競賽排程

競賽排程裡可以透過競賽名稱查詢競賽，同時它會顯示競賽標題、競賽開始時間以及競賽狀態，如圖七所示。競賽狀態分為兩種，分別是 proceeding 跟 waiting，當競賽狀態為 proceeding 代表一般使用者可以進入這個直播間。

比賽標題	開始時間	狀態
Yan vs Figueredo	2024-10-09 10:00:00	waiting
Muhammad vs Khabibov	2024-10-10 07:00:00	waiting
Pantoja vs Dzidzhevli	2024-10-23 10:00:00	waiting
Jones vs Miocic	2024-10-28 10:40:00	waiting
Sandhagen vs Nurmagomedov	2024-11-22 23:20:00	proceeding
Vera vs Sandhagen	2024-11-25 10:00:00	waiting

圖七：競賽排程頁面。

4.4.2 競賽記錄

競賽記錄是由 WebSocket 製作，當某一方得分時，判斷程式會透過 WebSocket 傳送擊中的畫面到後端，後端會把圖片保存在指定的資料夾，然後再透過 WebSocket 傳送圖片位置到前端以達到即時更新的效果，如圖九所示。



圖九：競賽記錄。

4.4.3 聊天室

聊天室是使用 WebSocket 製作，使用者可以從前端傳送訊息到後端 WebSocket 伺服器，後端 WebSocket 伺服器會把接收到的訊息再傳送給其他使用者，聊天室頁面如圖十所示。



圖十：聊天室頁面。

4.4.4 歷史競賽

歷史競賽裡保存著已經結束的串流直播，如圖十一所示。歷史競賽頁面能用名稱查詢競賽，會播放當期競賽的三個影片以及顯示雙方選手擊中各個部位的次數。

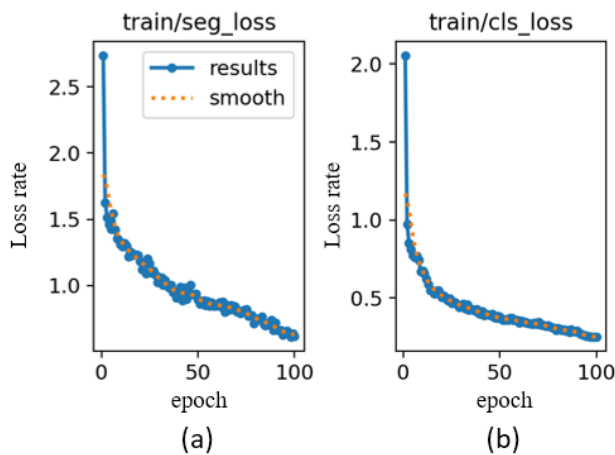


圖十一：歷史競賽頁面。

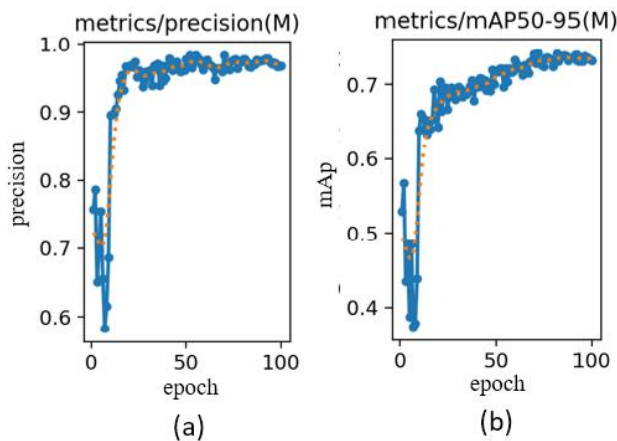
4.4.5 模型訓練過程分析

本系統使用 YOLOv8 的分割模型進行遷移式訓練，以符合本系統所需之邊緣分割模型。此模型的損失函數變化圖、準確率圖以及平均預測精度(mAP) 圖，如圖十二所示。從圖中可以發現損失函數隨著訓練過程逐漸遞減，

而準確率以及 mAP50-95隨著訓練過程逐漸遞增，如圖十三所示，表示模型的辨識準確度符合本系統之需求。



圖十二：訓練次數與損失函數關係圖：(a)分割損失曲線 (b) 類別分類損失曲線。



圖十三：訓練次數與準確率關係圖：(a)準確率曲線 (b) mAP 曲線

5. 結論

本系統實現了一套基於 YOLOv8 Instance Segmentation 模型的「即時拳擊判分輔助系統」。透過電腦視覺技術，本系統能即時辨識競賽場景並進行精準計分。使用者可在網頁上觀看競賽的即時直播畫面，並查看詳細的競賽記錄與計分數據。此外，系統整合了聊天室功能，讓使用者能分享對競賽或系統的看法，進一步提升觀賽體驗與互動樂趣。本研究旨在透過本系統的輔助，減少競賽中的誤判，進一步提升競賽的公平性。

參考文獻

[1] 本校運科所校友陳念琴- I'm Nien-Chin,CHEN 勇奪 2024 巴黎奧運女子拳擊 66 公斤量級銅牌。7 Aug. 2024

[2] Evan Quin, Niall Corcoran, “TheAutomation of Computer Vision Applications for Real-Time Combat Sports Video Analysis.”, 17 Nov. 2022

[3] Jabbr.ai, DeepStrike
<https://jabbr.ai/#beta>

[4] Jake Bickerton, “Combat IQ brings AI-based Real-Time Insights to Combat Sports.”, 2 May. 2023
<https://www.broadcastnow.co.uk/production/combat-iq-brings-ai-based-real-time-insights-to-combat-sports/5181557.article>

[5] AWS, 什麼是 Python？ Python 語言介紹。
<https://aws.amazon.com/tw/what-is/python/>

[6] Anaconda, Working with Conda (CLI), Anaconda Documentation.
<https://docs.anaconda.com/working-with-conda/ide-tutorials/spyder/>

[7] Visual Studio Code, Visual Studio Code - Code Editing.
<https://code.visualstudio.com/>

[8] Heidi-Liu, 學習筆記 TypeScript 基礎入門：從型別談起。
<https://hackmd.io/@Heidi-Liu/typescript>

[9] JavaScript , Mdn Web Docs.
<https://developer.mozilla.org/zh-TW/docs/Web/JavaScript>

[10] const 伐伐, yt-dl 使用教程。
<https://blog.csdn.net/u013905744/article/details/131142736>

[11] Arthur Wilton, Running FFmpeg Commands From a Python Script. 4 May. 2022
<https://artwilton.medium.com/running-ffmpeg-commands-from-a-python-script-676eaf2b2739>

[12] 夢想橡皮擦, Python Moviepy 的用法。15 Mar. 2022
<https://juejin.cn/post/7075130276380147725>

[13] Open Source Computer Vision, OpenCV-Python Tutorials.
https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

- [14] Ellie, Labelme 安裝與使用。22 Mar. 2024
<https://reurl.cc/Q59n72>
- [15] Ultralytics, Ultralytics YOLOv8.
<https://docs.ultralytics.com/models/yolov8/>
- [16] What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector.
<https://arxiv.org/abs/2408.15857>
- [17] 古古, Nginx 是什麼？認識反向代理、負載平衡
<https://kucw.io/blog/nginx/>
- [18] Kate Skavish, RTMP 完全指南：它是什麼，何時使用
https://wave-video.translate.google.com/zh/blog/rtmp-guide/?_x_tr_sl=zh-CN&_x_tr_tl=zh-TW&_x_tr_hl=zh-TW&_x_tr_pto=sc