University of Winchester

Department of Digital Technologies

Computer Science Bsc(Hons)

BS3203 Computing Project Portfolio:

AI-Powered 3D Basketball Shot Analysis System

Integrating Pose Estimation and Rapid Feedback for Performance Enhancement

Supervisor: Aimee Robinson

Lecturer: Software Engineering

Student: Ashya Gray

Student Number: 2111692

Academic Year 2024-2025

Contents

Glossary

Term	Definition
Pose Estimation	The process of determining the position and orientation of human joints from visual data, used here to assess player biomechanics.
MMPose	A modular, open-source pose estimation framework used to extract 2D and 3D keypoints from basketball shooting footage.
YOLOv11	An object detection model (You Only Look Once version 11) used here to detect the basketball within video frames.
Dynamic Time Warping (DTW)	A sequence alignment algorithm that compares motion data between student and teacher keypoint sequences despite timing variations.
Savitzky-Golay Filter	A digital smoothing filter applied to wrist position data to calculate release points more robustly.
Shooter Locking	The initial phase in which the system identifies which player in the video is performing the shot using wrist proximity to ball detections.
Key Phase Extraction	The process of identifying distinct biomechanical stages in the shooting motion: Preparation, Load, and Release frames.
3D Lifting	The transformation of 2D keypoint data into estimated 3D joint positions using pretrained pose models.
Fallback Logic	Backup detection methods triggered when ball or wrist detection fails, ensuring continuity of analysis.
Gradio GUI	The user interface framework used to allow easy user interaction with the system inside Google Colab.

1 Introduction

1.1 Background and Problem Statement

Basketball coaching has long relied on manual observation and video analysis to evaluate shooting mechanics. Coaches and players extensively study footage to identify flaws in form, release angle or follow-through, however this process is inherently subjective, time-consuming, and inaccessible to athletes without professional training resources. While recent advancements in computer vision have introduced tools for sports analytics, most systems and studies, such as (Chen and Lu 2022) 3D Pose Estimation analysis, focus on only static 3D pose comparisons derived from images neglecting efficient video processing and integration with ball trajectory data. This limits practical application during dynamic training sessions where immediate feedback is critical. Furthermore, existing systems like VideoPose3D have limited accessibility and documentation for public usage making general utilization and deployment challenging.

This project addresses these gaps by developing an AI-powered basketball shot analysis system that combines MMPose's hybrid 2D/3D inferencing with Ultralytic's YOLOv11 ball tracking to provide rapid, actionable feedback. By leveraging MMPose's modular architecture, predicting 2D keypoints from video frames before lifting them to 3D, the system balances accuracy and speed, enabling seamless integration with dynamic, real-world training environments.

1.2 Research Aims and Objectives

Building upon the identified gaps in real-time basketball shot analysis, this project aims to achieve the following interconnected objectives:

First, the system will acquire and preprocess both professional and amateur basketball shooting videos, ensuring a diverse dataset that reflects real-world variability in player techniques and recording conditions. This includes sourcing NBA footage under fair use guidelines and capturing videos from university team members, with particular attention to resolution and frame rate.

Second, the methodology will implement MMPose's hybrid 2D-to-3D pose estimation pipeline to address the limitations of prior work. Unlike Chen & Lu's static 3D analysis, this approach leverages MMPose's HRNet-W48 model for robust 2D keypoint detection across varying video qualities, followed by 3D pose lifting. This two-stage process not only improves computational efficiency but also enables temporal analysis of shooting mechanics; capturing critical phases such as shot preparation, release, and follow-through that are essential for biomechanical evaluation.

Third, the project will integrate YOLOv11 for basketball detection and trajectory analysis, extending the system's functionality beyond skeletal tracking alone. By fine tuning the model on a basketball specific dataset, the system will address challenges such as ball occlusion during shooting and provide a holistic view of shot execution. The combined pose and ball data will then be analysed using Dynamic Time Warping (DTW) to align amateur and professional shooting sequences, identifying deviation in joint movement and release timing.

Finally, the systems practical utility will be realizes through an intuitive feedback interface designed for coaches and players. While not explicitly real-time, the pipeline's short processing time represents a significant improvement over manual analysis methods. The interface will highlight biomechanical insights (e.g., "Elbow joint angle is 18° behind professional benchmarks before release") alongside trajectory metrics, enabling data driven adjustments during training sessions.

Collectively, these objectives advance the field of sports analytics by bridging the gap between theoretical pose estimation research and applied coaching tools, while addressing computational and ethical constraints.

1.3 Significance of Study

This Project makes significant contributions to both the academic field of computer vision and the practical domain of sports analytics. By integrating MMPose's hybrid 2D/3D pose estimation with YOLOv11-based ball tracking, the system advances the state of automated basketball shot analysis in three key ways.

1.3.1 Inclusive access through the cloud

This system will be deployed on Google Colab, eliminating hardware barriers, and enabling coaches, athletes, and researchers without high-end GPU's to access professional-grade biomechanical analysis. This cloud based approach democratises advanced training tools for grassroots sports communities, aligning with the growing demand for inclusive, equitable sports science. Prior frameworks for computer vision require specialised NVIDIA hardware, this pipeline prioritises usability on consumer-grade devices, broadening its societal impact.

1.3.2 Computational Efficiency and Precision

The hybrid 2D pose estimation and selective 3D lifting methodology represents a significant methodological innovation. By focusing on 3D analysis only on critical shot phases (e.g., release, load point), the system reduces computational overhead while maintaining biomechanical analysis across the most important action phases. This approach addresses a key limitations in general frameworks, such as VideoPose3D, which process entire videos in 3D from video input, leading to impractical resource demands. Integrating the most recent YOLO model (v11) for ball tracking further enhances efficiency, enabling swift detection times even on limited hardware.

1.3.3 Actionable Biomechanical Insights

The system’s ability to reliable isolate key shooting phases (e.g., elbow angle at release, knee flexion during loading) and correlate them with ball trajectory data provides coaches with targeted interpretable feedback. This advances beyond generic pose estimation tools by linking specific joint kinematics to performance outcomes- a contribution with implications for injury prevention and skill refinement in sports science.

1.3.4 Scalability and Adaptability

The modular design and concepts the system incorporates gives the potential for scalability beyond just the context of basketball free throws. For instance:

- The ball tracking logic can be adapted to different YOLO models to function with other ball sports such as football, or rugby.
- The key phase extraction modules and logic are also adaptable. Key phase extraction operates based on general rules followed during specific actions, therefore sports with identifiable key phases (e.g., tennis serves, volleyball spikes.) could follow similar pipelines, modified to sport specifics, to make use of the general structure that this system follows.
- The Google Colab deployment framework lowers barriers to adoption in resource-constrained environment (e.g., schools, developing regions.)

2 Literature Review

The integration of computer vision and artificial intelligence into sports analytics has revolutionised how athletes and coaches analyse performance, yet significant gaps persist in delivering accessible, biomechanically granular feedback. Prior research has explored pose estimation, object tracking, and AI-driven coaching tools in isolation, often overlooking the computational and practical constraints of real-word deployment. This review critically evaluates these strands of literature, focusing on, the evolution of pose estimation techniques in sports, advancements in ball tracking methodologies, and the limitations of existing AI coaching systems. By evaluating these areas, this discussion highlights the need for integrated, efficient frameworks such as the system proposed in this project; a hybrid 2D/3D pipeline optimised for usability on consumer grade hardware.

2.1 Pose Estimation in Sports Analytics

Pose estimation has emerged as a critical component in modern sports analytics, greatly enhancing the biomechanical assessment of athletes by employing computer vision and machine learning technologies. Traditional motion capture methods, which are typically marker-based, offer a high degree of accuracy through optical markers that track body segments (Figure 1). This position is supported by authors who note the effectiveness of marker-based systems in controlled laboratory environments but also highlight their significant limitations, such as high costs, complex setup requirements, and the effects cause by occlusions during real-world applications

(Vasileiadis et al. 2019)(Hwang et al. 2023). Due to these drawbacks, the transition towards marker-less pose estimation has been increasingly favoured in sports analytics, where the need for real-time and practical solutions outside of lab settings is critical.

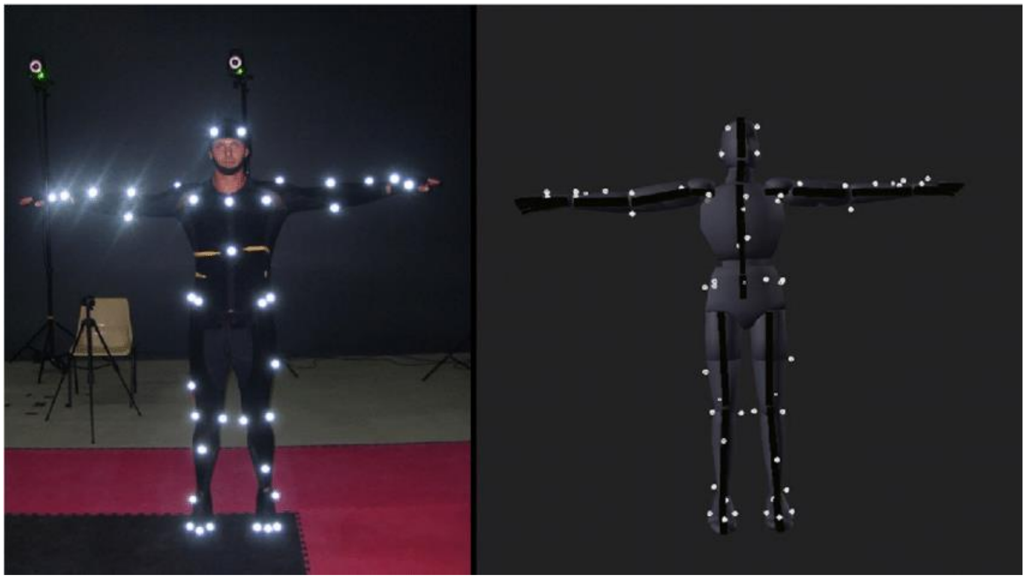

Figure 1: Marker-based motion capture in controlled environment example (AnimostStudio, 2024)

Recent innovations in marker-less pose estimation highlight significant progress in addressing the limitations of traditional methods. For instance, deep learning models such as VideoPose3D utilise temporal convolutional networks and semi-supervised learning methodologies to derive 3D human poses from standard video inputs, thus removing the dependency on physical markers altogether. This transformative approach has been implemented in many sports scenarios (Baumgartner et al. 2023), including basketball, where studies have demonstrated effective extraction of skeletal data from shooting videos and the comparative analysis between amateur and professional athletes using methodologies such as Dynamic Time Warping (DTW) (Chen and Lu 2022). Such initiatives highlight the importance of implementing strong metrics for performance assessment, including the mean distance to key frames, which quantitatively evaluates shooting form accuracy.

Nonetheless, several challenges persist in the real-world application of pose estimation technologies. factors such as variable outdoor conditions, fluctuations in lighting, and the rapid dynamics of athlete movements can severely compromise the accuracy of pose estimation systems as identified in numerous studies (Šajina and Ivašić-Kos 2022) (Stenum et al. 2021).

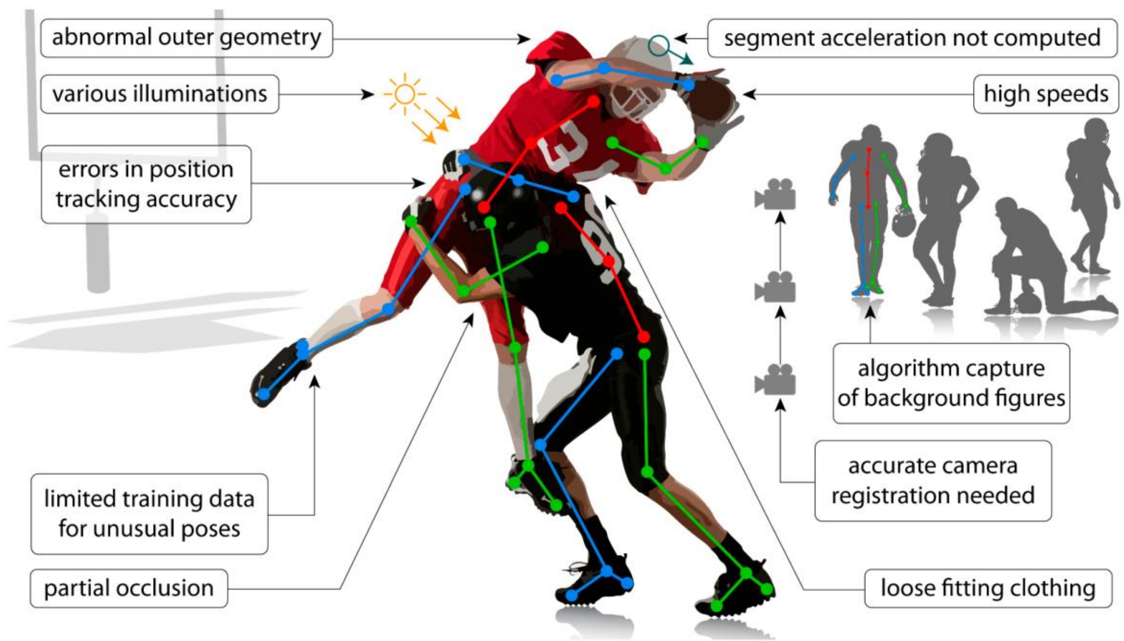

Figure 2: Examples of pose estimation limitations (Stenum et al. 2021)

Furthermore, many existing frameworks remain limited to post-even analysis, offering delayed insights that fall short of the timely feedback athletes increasingly demand (Nie et al. 2019). The evolution from 2D to 3D pose estimation technologies has enhanced depth perception, yet many systems still struggle with accurate lifting from custom videos or 3D lifting directly from videos or images with multiple people (Pavllo et al. 2019). Such gaps reveal the potential for future advancements in creating robust systems that synthesize, pose estimation, ball tracking, and multi-person handling, to deliver comprehensive analyses of athletic performance (Baumgartner et al. 2023)

In sum, the field of pose estimation in sports analytics is witnessing rapid advancement, particularly with marker-less methods that offer practical applications outside controlled environments. Although considerable progress has been made, ongoing challenges related to environmental factors and the necessity for fast, actionable feedback reveal important areas for future research and development. Integrating these systems with comprehensive analytics will be essential for transforming athletic training methodologies and performance outcomes.

2.2 Ball Tracking in Sports Analytics

Ball tracking has evolved from rudimentary annotation to sophisticated AI-driven systems, playing a pivotal role in sports performance analysis. In early approaches, colour-based segmentation, particularly utilising the Hue-Saturation-Value (HSV) colour model, was prevalent as it effectively distinguishes between different visual elements based on their colour characteristics (Ji et al. 2018). However, most early models faced challenges with occlusions, rapid motion, and varying light conditions (Naik and Hashmi 2022) (Robertson et al. 2023) For instance, the need for high shutter speeds (e.g., 1/1000s) to properly capture fast-moving objects like a volleyball (reaching up to 90Km/h on serve), exemplifies the constraints of traditional video capture technologies when accompanied by simplistic tracking algorithms.

Recent advancements incorporate occlusion-aware strategies to enhance tracking accuracy, addressing previous shortcomings. For example, occlusion detection techniques have been developed to cope with visibility issues encountered during high-speed actions, ensuring continuous tracking even when objects become partially or fully obscured (Kong et al. 2020) (Rathnayake et al. 2020) As a result, the integration of intelligent algorithms has transformed object/ball tracking into a more reliable tool for analytics in sports, allowing for improved performance evaluation and strategic decision making (Naik and Hashmi 2022).

Moreover, the adoption of deep learning frameworks has shown promise in managing these challenging conditions by utilising historical frame data to inform current tracking decisions. On top of this, these frameworks, such as YOLO and SSD (Single shot MultiBox Detector) (Liu et al. 2016) bypass manual feature engineering by learning hierarchical representations directly from data (Erabati et al. 2020).

Figure 3: Workflow of (a) traditional methods and (b) deep learning methods (Erabati, Gonçalves, and Araújo 2020)

YOLO (You Only Look Once) architectures, particularly YOLOv11 (Khanam and Hussain 2024) have become benchmarks for real-time performance, achieving sub-millisecond inference times while maintaining high accuracy. Similar to Kong and Rathnayake's work, YOLO's design enable robust detection even under partial occlusions through its anchor free design and multi-scale predictions, further reinstating it at the forefront of object detection. However, it is also important to note that general object detection implementation focuses on 2D planar tracking, giving some inaccuracy when attempting to comprehensively predict or display ball trajectory data without multiple viewpoints (Wu et al. 2020).

2.3 AI in Sports Coaching and Basketball

Artificial intelligence (AI) has increasingly become an integral component of modern sports coaching, where it plays a significant role in performance enhancement and injury risk management. The application of AI-driven coaching technologies enables athletes to gain insights into their performance, health and techniques, thereby facilitating transformative learning experiences (Ghezelseflou and Choori 2023) These technologies utilise various data analytics methods, such as pose estimation, ball tracking, and player tracking, to asses player and team performance. Studies suggest that advanced algorithms can accurately detect and analyse critical game aspects, providing coaches with essential data to adjust training regimens for optimal outcomes (Claudino et al. 2019).

In basketball, AI technologies have shown promise in predictive performance analytics and injury prevention. Strategies employing AI enable targeted feedback that athletes can use to improve specific skills, such as shooting accuracy and ball handling (Bin and Xu 2021). Feedback in sports is critical as athletes rely on intrinsic feedback (self-evaluation), alongside extrinsic feedback (provided by coaches) to evaluate their performance effectively (Hammes et al. 2022). The integration of AI allows for a deeper understanding of the mechanics of athletic movements, allowing of real-time correction and adaptations in player technique. This methodological approach emphasises AI's capability to process large data volumes, which can identify trends and patterns that might be overlooked in a traditional coaching context(Molavian et al. 2023).

However, while the benefits of AI in sports coaching are notable, challenges remain concerning ethical considerations and the importance of maintaining the human element in coaching (Demenius and Kreivytė 2017). As sports increasingly integrate AI technologies, balancing computational support with personalised human interactions will be vital for creating a sustainable and effective coaching environment. The future of AI in sports coaching, particularly basketball, holds promising potential, but ongoing research should ensure that it complements the invaluable role of coaches in athlete development, rather than undermining them.

In conclusion, AI technologies represent a transformative force in basketball coaching, offering innovative solutions for skill enhancement, performance analysis, and injury prevention. By harnessing advance analytics and machine learning, coaches can provide tailored training programs that respond to player needs. Nonetheless, the successful integration of these technologies necessitates a mindful approach that retains the essential aspects of human coaching.

2.4 Edge/Cloud Computing in Computer Vision

In recent years, the introduction of edge and cloud computing has become a driving force behind innovation in computer vision applications. This architectural shift is designed to strengthen the performance of numerous systems, particularly those involving artificial intelligence, and Internet of Things (IoT) technologies, such as sports analytics. By leveraging the computational power distributed across edge and cloud environments, these systems enable efficient model optimisation and deployment, making applications like AI-driven basketball shot analysis more widely accessible across various platforms.

The shift towards cloud solutions in the context of computer vision has led to significant improvements. Cloud platforms provide on-demand access to expansive computing power and storage capabilities, which are essential for processing large volumes of video data generated from activities such as basketball free throw analysis. By utilising cloud-based services, developers can deploy complex machine learning models without the burden of local resource limitations, enabling advancements like real-time analytics that are critical for sports coaching and player performance assessment (Sithipolvanichgul et al. 2021) . These resources can be efficiently managed through techniques such as load balancing and resource allocation to ensure optimal performance during peak usage times (Li 2022).

Despite the benefits, the challenge of security within cloud computing environments cannot be overlooked. With the transition to utilising cloud services, there are increased concerns regarding data privacy and integrity. Cloud infrastructures must implement robust security protocols to protect sensitive data such as player analytics and personal performance records (Zhang et al. 2018).Moreover, ensuring that cloud services meet the necessary compliance and regulatory standards is crucial for fostering trust among users who may be hesitant to adopt these technologies (Alqahtani 2019).

The potential of AI systems, optimised through cloud computing for real-time applications is considerable. Enhancements in machine learning methodologies, combined with the structure provided by cloud environments, allow for innovative solutions that combine accessibility with high performance. Academic research indicates that cloud computing's inherent flexibility and scalability are pivotal for meeting the varied demands of modern computing applications, especially in fields like computer vision(Rawashdeh et al. 2023).

Integrating edge and cloud computing solutions presents a multitude of benefits for advancing computer vision applications, particularly for AI-based evaluation systems aimed at sports analytics. By optimising model performance and ensuring accessibility, this structure supports not only technical advancement but also the democratisation of sophisticated tools for a broader audience of users.

3 Research Methodology and Design

3.1 Research Design Overview

This project follows an applied, experimental research design aimed at developing a functional AI-based system for biomechanical evaluation in basketball shooting. The goal is not only to explore the feasibility of 3D pose estimation and ball tracking in a constrained cloud environment, but also build a prototype capable of delivering biomechanically meaningful feedback based on real-world performance data. The research process was divided into interconnected technical stages, each aligning with a core component of the system: video acquisition, 2D and 3D pose estimation, ball tracking, motion comparison, and feedback generation.

To guide development, this project followed a hybrid Agile (Figure 4), Incremental methodology. This approach supported iterative refinement and modular construction, allowing key component, such as, pose estimation, ball tracking, and pose alignment, to be designed, tested, and improved in isolation before full integration. The Agile framework enabled adaptability when unexpected implementation issues arose, such as the pivot from VideoPose3D to MMPose due to compatibility limitations. This flexibility was essential for maintaining momentum within the project's time-frame and aligning development decisions with the overarching research objectives.

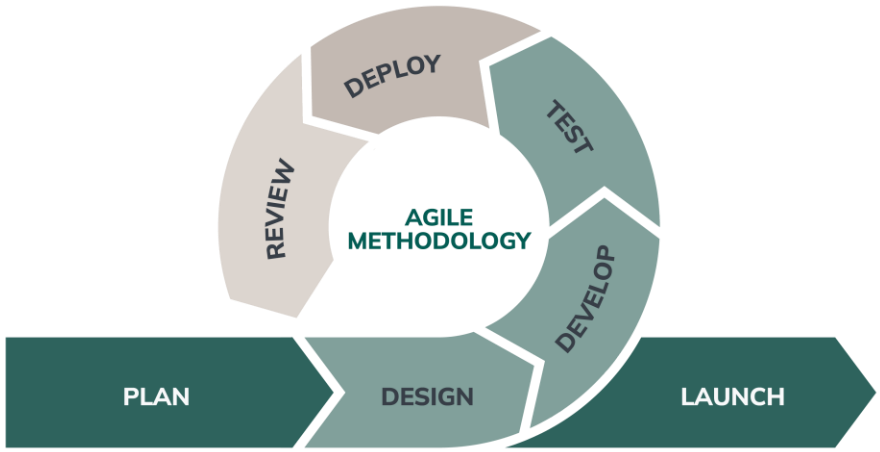

Figure 4: Agile Software Methodology Diagram (Michaud 2024)

The architecture was designed to process video inputs from both professional (NBA) and amateur (University player) sources. Videos are processed frame-by-frame for 2D keypoint extractions, with MMPose then lifting selected frames to 3D. Simultaneously, the basketball's position is tracked using the YOLOv11 object detection framework, fine-tuned for basketballs in motion. The two data streams (skeletal keypoints and ball trajectories) are synchronised using frame indexing and later analysed using Dynamic Time Warping (DTW) to align student an professional shooting phases for comparative analysis.

While not strictly real-time, the system prioritises rapid feedback with minimal processing delay, supporting its practical application in live coaching scenarios. As a secondary focus, The entire system is deployable through Google Colab, ensuring users without specialised hardware (such as NVIDIA GPUs) can still execute the pipeline and receive actionable feedback.

In summary, the research design combines modular software engineering with cloud deployment, balancing the practical constraints of resource-limited execution environments with the demand for accurate motion analysis. This design enables both technical exploration and applied evaluation.

3.2 Data Collection and Ethical Considerations

3.2.1 Participant Recruitment and Data Collection

To evaluate the system under real-world conditions, custom data was collected via voluntary participation from members of the University of Winchester basketball team. Participants were recruited directly through the first team's communication channel with clear communication that participation was entirely voluntary and unrelated to team membership or any other obligations. Participants were asked to complete several free-throw shooting repetitions, which were recorded using a smartphone camera. The recording session lasted approximately 10-15 minutes and was conducted in the participants' normal training environment to ensure no irregular interference with their shooting motion. The videos served as primary data inputs for assessing the system's shooter detection, pose estimation, and feedback generation components.

In addition to the collected university footage, publicly available NBA free-throw footage sourced from NBAPlayDB was also used. This dataset served both as reference material for system development and as training data for the YOLOv11 ball detection model. These NBA samples do not contain identifiable personal information and were used under fair use provisions for academic purposes.

3.2.2 Ethical approval and Informed Consent

Prior to beginning data collection, full ethical approval was obtained through the University of Winchester's ethics process (see approved Form B). As the project involved the collection of non-anonymous video data, additional safeguards were implemented throughout the course of development and report writing to ensure participant confidentiality and voluntary informed consent.

Participants received a comprehensive Participant Information Sheet and Consent Form

outlining the nature, purpose, and risks of the study. All participants consented in writing prior to participation. Participants were also informed of their right to decline or withdraw at any point prior to data anonymisation. All faces or identifying features within any visualisations or screenshots included in this report have been anonymised (e.g., blurred), and any personal identifiers have been removed in accordance with GDPR (2018) regulations. All data was securely stored using university-provided cloud storage with password protection. Upon project completion, all collected data will be securely deleted as per university research data management policies.

3.2.3 Ethical Considerations of AI integration

The integration of artificial intelligence (AI) in sports analytics must be approached with careful consideration of ethical implications, particularly concerning algorithmic bias and participant data privacy. As with any machine learning model, including the custom YOLO basketball detection system, there exists the potential for algorithmic bias primarily stemming from the data used in training. Research indicates that if models are primarily trained on a limited dataset, such as curated NBA footage, they may not adequately represent the diversity found in amateur, youth, or non-professional athlete data, thereby raising fairness concerns when applied across various populations (Lotfi and Rebbouj 2021)(Terven et al. 2023) . This remains a critical issue as technical advancements happen without thoughtful consideration of their broader societal implications (Padala et al. 2019)

Additionally, the reliance on pretrained models for tasks such as 2D pose estimation raises significant concerns about the accuracy and fairness of AI-driven analytics across diverse demographics. Studies have demonstrated that many of these models inherit biases present in the original datasets, often gathered under controlled conditions with limited demographic variation (Lotfi and Rebbouj 2021) Consequently, this limitation can influence the accuracy of biomechanical assessments for athletes of varying body types, ages, or genders. As highlighted by the literature, it is essential to employ diverse datasets when training these models to enhance fairness and robustness in performance assessments (Terven et al. 2023)

As the AI system aims to enhance coaching efficiency by providing objective biomechanical feedback, it is crucial to underscore that such technology should not replace the judgement and intuition of qualified coaches. Over-reliance on AI assessments can unintentionally reduce athlete confidence by creating a view of performance, that is robotic and mechanistic. Experts have pointed out that while technology can serve as a valuable tool in sports decision-making, the human element remains irreplaceable, illustrating that a collaborative effort between human judgement and machine-driven analytics is vital for effective coaching(Padala et al. 2019).

As a final concern, privacy constitutes an ongoing challenge within the realm of sports AI, especially regarding the potential use of live-streamed or competitive footage. Although the current research ensures that participant data is stored securely and anonymized, scaling the system would require further attentiveness in regards to the handling of data. It would be imperative to ensure that the correct guidelines are followed regarding ownership, access, and retention, particularly in compliance with regulations such as GDPR (Mane 2024). Existing literature emphasizes that proper governance frameworks are essential to safeguard participant privacy and maintain trust in the deployment of AI in sports analytics (Joshi 2023).

3.3 Model Training and Preparation

A critical component of the proposed system is the detection and tracking of the basketball during shooting sequences. To achieve accurate and robust ball detection across diverse video conditions, a custom object detection model was trained using the YOLOv11 architecture provided by the Ultralytics framework.

307 frames were manually annotated using Roboflow from five free throw clips sourced from NBAPlayDB. These frames were selected to capture diverse ball positions, lighting conditions, occlusion scenarios, and camera angles. The Roboflow web interface allowed for bounding box annotations specific to the "basketball" class, ensuring consistent labelling standards across the dataset. Once annotated, the dataset was downloaded directly into a Colab environment using the Roboflow API key, which streamlined integration with the training pipeline.

Figure 5: Example Frame From Dataset With Annotation (Purple Box Highlights Basketball)

Model training was conducted in Google Colab, utilising GPU acceleration to train the YOLOv11 model for 100 epochs. Training the model over 100 epochs means the entire dataset was passed through the neural network 100 times during training. Increasing the number of epoch improves the model's ability to generalise, but also increases the risk of overfitting if not monitored. In this project 100 epochs provided a balance between training duration and model accuracy, producing a stable object detector suitable for use in basketball specific conditions. Training performance was monitored over the course of the 100 epochs using the real-time visual metrics for training loss and object confidence, alongside the post training metric graphs that are produces by the train function.

```
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key="[API KEY REDACTED FOR PRIVACY]")
project = rf.workspace("basketballcustom").project("basketball-
player-detection-pcmsl")
version = project.version(4)
dataset = version.download("yolov11")
!yolo detect train model=yolo11s.pt data=
/content/Basketball/player-detection-4/data.yaml/epochs=100
imgsz=640
```

Listing 1: YOLOv11 model training code snippet

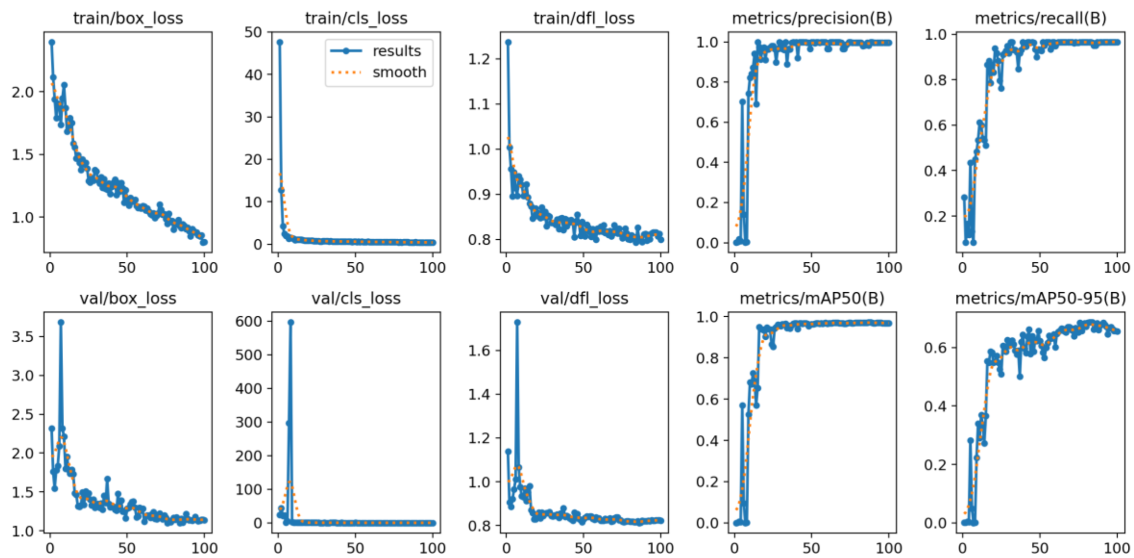

Figure 6: YOLOv11 training metrics across 100 epochs.

he mAP50-95(B) graph (bottom right) represents the model's mean average precision (mAP) across multiple intersection-over-union thresholds, giving an indicator of both detection quality and localisation accuracy. The upward trend and eventual plateau near 0.65 demonstrate that the model successfully generalised across diverse shot scenarios, making it well-suited for integration into the real-world analysis pipeline.

The Recall graph (top right) illustrates the model's ability to detect all instances of the basketball across test frames. The curve rising toward 1.0 indicates minimal missed detections, confirming the model's reliability for continuous ball tracking throughout a shot sequence.

After completion and review, the best performing model weights were saved and exported in .pt format for seamless deployment.

This basketball-specific YOLOv11 model was chosen over generic pre-trained models to improve accuracy in detecting basketballs under varying conditions, such as occlusion, blur, or motion. It serves as the backbone of the full analysis pipeline, being an essential marker in player detection and tracking, and giving the positional and trajectory data of the ball for analysis.

3.4 System Architecture and Technical Workflow

3.4.1 Shooter Identification and Initialisation

The pipeline begins by identifying the shooter in the early portion of the video, which is essential for isolating pose data relevant to shot execution. Since MMPose returns keypoints for every detected individual, the system must reliably distinguish the shooter from other people in frame

This is handled through a two-stage approach. First, a custom function lock_shooter() scans the first 100 (adjustable) frames to locate a candidate frame that includes both a basketball detection and at least one person. Once found, the system evaluates all detected persons in the frame and compares the distance of each individuals wrist keypoint locations, and the centre of the basketballs bounding box. Wrist keypoints are retrieved using the COCO 17-joint format (Figure 7), which designates the left and right wrists as keypoints 9 and 10 respectively. The person whose wrist position is closest to the ball is selected as the shooter.

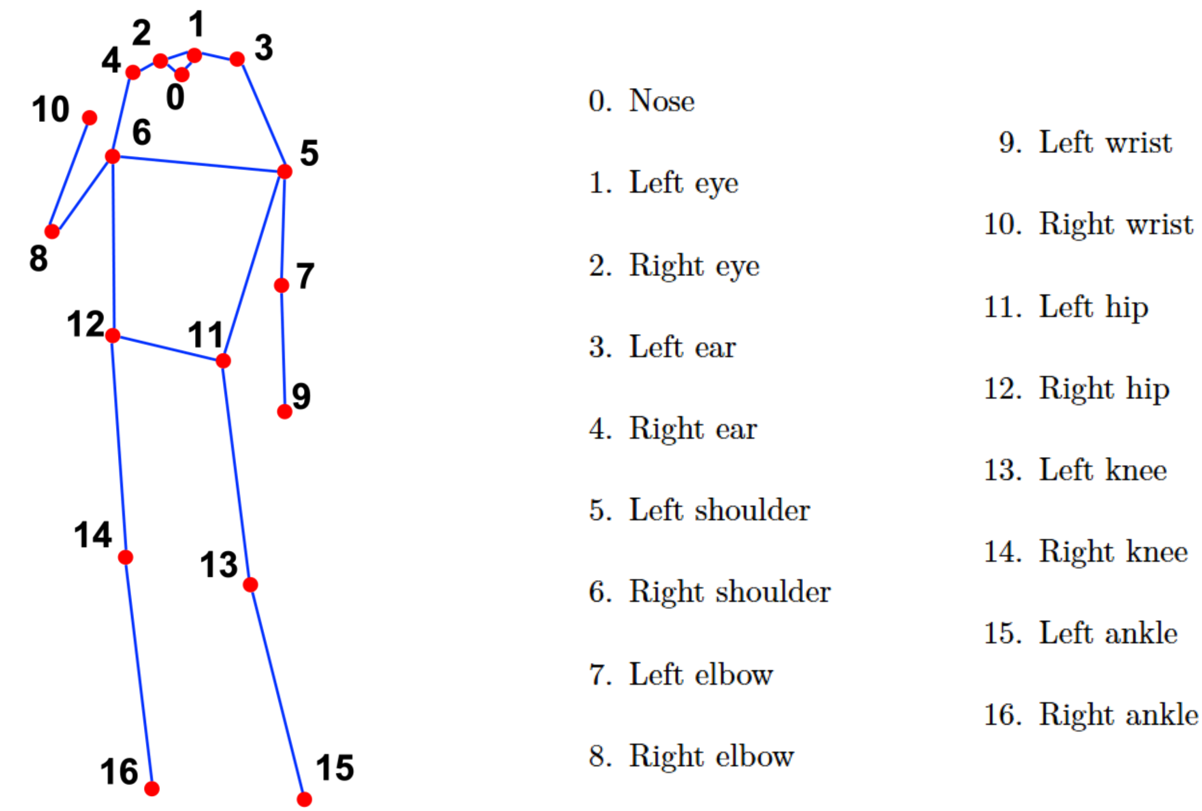

Figure 7: COCO keypoints format (Lin et al. 2014)

3.4.2 Keypoint Extraction and Ball Tracking

After the shooter is locked in the initial frame, the system proceeds to track them across the entire video using the track_shooter_full_video() function. This function uses an Intersection over Union (IoU) method to compare each new frame's person detections with the bounding box of the initially identified shooter. A match is confirmed once all new detections have been compared and the highest IoU value has been found, after which, the shooter bbox is updated for comparison next frame. This gives consistency in shooter tracking even as the subject moves, jumps, or rotates.

```
def calculate_iou(box1, box2):
    # Calculates overlappng bounding box for tracking
    x1_inter = max(box1[0], box2[0])
    y1_inter = max(box1[1], box2[1])
    x2_inter = min(box1[2], box2[2])
    y2_inter = min(box1[3], box2[3])

    width_inter = max(0, x2_inter - x1_inter)
    height_inter = max(0, y2_inter - y1_inter)
    area_inter = width_inter * height_inter

    area_box1 = (box1[2] - box1[0]) * (box1[3] - box1[1])
    area_box2 = (box2[2] - box2[0]) * (box2[3] - box2[1])

    area_union = area_box1 + area_box2 - area_inter

    return area_inter / area_union if area_union > 0 else 0
```

Listing 2: IoU function code snippet.

Alongside this tracking function, when the shooter is identified each frame, their keypoint positions for this frame are also extracted and saved to a dictionary that will contain their position for every frame. This dict is returned at the end of the function when the video has finished being processed and is later used to extract biomechanically relevant frames and analyse joint angles at critical shot phases. Also running in parallel, the YOLOv11 model is used to detect the basketball on each frame. As the use case for this systems assumes a single ball in frame, the YOLOv11 model is configured to only return the highest confidence prediction per frame, ensuring consistency and avoiding false positives. Once again, all ball positions for each frame are appended to the dictionary to aid with key frame analysis and identification explained in Section 3.4.3.

```
"""
    Returns:
        shooter_keypoints_data: Dictionary with:
        {
            frame_num: {
                'bbox': [x1, y1, x2, y2],
                'keypoints': np.array of shape (N, 3)  # (x, y,
score)
                'keypoint_scores': np.array of shape (N,)
                'ball_bbox': [x1, y1, x2, y2] or None
            }
        }
"""
```

Listing 3: : Returned dictionary structure

This combination of spatial proximity for initial shooter selection and IoU-based frame-to-frame tracking provides a reliable foundation for extracting clean, subject-specific pose and ball data in dynamic game environments.

3.4.3 Keyphase Identification and Preparation

Once a full dictionary of shooter keypoints and ball positions is returned, the system uses this data to identify and isolate critical frames corresponding to the **preparation**, **load**, and **release** phases of a basketball shot. These phases represent key biomechanical markers: the preparation frame reflects the athlete's pre-shot stance, and marks the beginning of the shooting sequence. The load frame marks the initiation of upward force generation, and the release frame captures the moment the ball leaves the shooter's hand. Isolating these moments allows for targeted 2D and 3D pose analysis while optimising computational efficiency.

Release Frame Detection

The release frame is the first frame to be detected and is used as a basis for detecting later frames. The find_release_frame() function prioritises frames where the shooters dominant wrist is at its highest vertical position and the basketball is within close proximity. Candidate frames are sorted based on wrist height and evaluated in batches to ensure that the ball is sufficiently near the wrist (pixel-based proximity, set to 80px by default). The selected release frame corresponds to the frame where the ball is most likely released Figure 8(c), after which, the elbow angle is computed using the calculate_elbow_angle() function in preparation for the analysis phase.

```python
def calculate_elbow_angle(keypoints, shoulder_idx=12,
elbow_idx=14, wrist_idx=16):
"""Calculates elbow angle (shoulder-elbow-wrist) in degrees"""
shoulder = keypoints[shoulder_idx][:2]
elbow = keypoints[elbow_idx][:2]
wrist = keypoints[wrist_idx][:2]
# Vectors from elbow to shoulder and wrist
vec_se = np.array(shoulder) - np.array(elbow)
vec_ew = np.array(wrist) - np.array(elbow)
# Calculate angle
cosine = np.dot(vec_se, vec_ew) / (np.linalg.norm(vec_se) *
np.linalg.norm(vec_ew))
angle = np.degrees(np.arccos(np.clip(cosine, -1, 1)))
return angle
```

Listing 4: Elbow joint angle calculation function snippet

Load Phase Detection

Following the identification of the release frame, the find_load_frame function performs a backwards search to locate the load phase. This is defined as the last frame before the release, in which the elbow is at or below the shoulder level, indicating the conclusion of the preparatory motion Figure 8(b). A tolerance of 10 pixels is allowed to account for variability or potentially skipped frames. This function returns the frame index number alongside both the elbow angle and the knee angle at this stage, the latter of which is computed using calculate_knee_angle(), being stored for later to provide insight into the lower-body involvement and power generation during the shot.

Preparation Phase Detection

The find_preparation_frame() function is used to estimate the beginning of the shooting sequence by analysing the wrist's vertical motion. Working backwards this time from the load frame, the function monitors upwards movement of the dominant wrist (keypoint 9 for left-handed shooters, 10 for right-handed). If the wrist shows no upward displacement for a set number of consecutive frames (default threshold: 3), that frame is marked as the beginning of the preparation phase Figure 8(a). This technique assumes that shooters remain relatively still before initiating a free throw, and as such, it provides a reliable checkpoint for the initiation of the shot phase. Whilst the preparation phase is not explicitly used in analysis, It is essential for marking the beginning of the shot sequence, and gives reference for the generation of analytical metrics later in the pipeline.

Frame Preparation: Cropping for 3D Pose Estimation

After detecting the preparation, load, and release frames, the system isolates the shooter visually using the crop_shooter() function for the load and release frames. This operation extracts a padded, square region centred on the shooter's bounding box and resizes it to 256x256 pixels. This is done to ensure compatibility with the 3D pose lifting model as the compute time increases extensively when multiple people are detected in frame. For debugging purposes, this function can be configured with a paint_joints argument that overlays the specified joints with visual markers

Figure 8: Crop_shooter() function example outputs with paint_joints (a), (b), (c)

Left-Handed Shooter Adaptation

It should be noted that all key phase identification functions include logic to dynamically adjust joint indices for left-handed shooters (an is_lefty argument, default set to False, can be passed). This ensures that angle measurements and motion tracking are anatomically consistent regardless of handedness, increasing the robustness of the system.

```
# Switching joints for left handed shooters
if lefty:
shoulder_idx, elbow_idx, wrist_idx = 5, 7, 9
else:
shoulder_idx, elbow_idx, wrist_idx = 6, 8, 10
```

Listing 5: left-handed handling logic (included in most functions)

3.4.4 Visualisation and 3D Pose Lifting

With the shooter's keypoints and ball trajectory data extracted and stored throughout the shot sequence (as detailed Section 3.4.2) the system proceeds to generate meaningful representation of this information and process selected frames through 3D pose estimation.

2D Visualisation of Motion and Ball Trajectory

To validate pose tracking and enhance interpretability, the system uses a custom visualisation function visualise_shooter_with_ball_trajectory() to overlay the shooter's 2D skeleton and ball movement path directly onto the original video frames. Rather than focusing on solely static poses, this dynamic visualisation illustrates:

- the continuous pose evolution across frames, using MMPose's skeletal keypoint renderer
- A plotted trajectory of the basketball, derived from bounding box centre positions, beginning at the identified release frame.

Figure 9: Example frame taken from visualise_shooter_with_ball_trajectory() output

Selective 3D Pose Lifting for Biomechanical analysis

As stated before, rather than applying 3D inference to every frame, the system lifts only the biomechanically critical frames (load and release frames identified in Section 3.3.3. The cropped frames received from the previous steps are fed into a pre-trained 2D-to-3D lifting model implemented through MMPose.

The model predicts a 17-joint skeleton in a three-dimensional space, with joint coordinates expressed in (x, y, z) relative to the camera. The lifted outputs are rendered using 3D skeletal plotting, enabling a clear comparison of join positions between load and release phases.

Figure 10: 3D Visualisation Example Outputs

3.4.5 DTW Alignment and Feedback Generation

To evaluate shot mechanics and provide constructive performance insights, the system employs a Dynamic Time Warping (DTW)–based motion comparison framework, alongside 2D joint comparison. This approach enables the comparison of temporal pose sequences and key frames between the user (student) and a chosen reference (teacher), accounting for differences in execution speed or frame alignment.

Joint Sequence Comparison Using DTW

Upon completion of shooter tracking and key phase identification, the extracted 2D keypoints are used to compute time-series joint angle sequences, specifically focusing on the elbow and the knee. These joint sequences represent biomechanical trends over the course of the shot, from the preparation phase, to the moment of release.

The generate_dtw_scores_and_angles() function performs DTW alignment between the student and teacher joint sequences. The teacher reference data is loaded from a pre-processed .pkt file containing structured keypoints, selected based on user input. The DTW algorithm dynamically aligns the sequences and outputs a normalised score, representing the average angular deviation between the two sequences.

This score allows for quantitative comparison of movement fluidity and coordination, regardless of the sequence length or speed.

```
    # Compares 2 joint sequences using DTW
    distance, _, _, path = dtw(seq1, seq2, dist=lambda x, y:
np.abs(x - y))
    normalised_score = (distance / len(path[0]))
    return normalised_score
```

Listing 6: compare_angle_sequences_dtw() snippet

Flexible Teacher Keypoint Integration

The comparison framework is designed to support flexible teacher reference selection. All reference data are stored in .pkt files containing structured keypoint dictionaries, which can be dynamically loaded at runtime. As long as the file naming convention aligns with system expectations, any saved teacher sequence can be used for comparison.

This extensibility not only allows comparisons across multiple expert shooters, but also provides a basis for support of user-uploaded teacher examples, enabling custom evaluations, and peer-to-peer comparison.

Interpreting alignment scores and Static joint angle feedback

The DTW scores are categorised using a tiered interpretation with lower scores indicating greater similarity between the student and reference movement:

DTW Score Range	Interpretation
< 5	Excellent alignment with reference motion
5 ≤ score < 10	Good, minor deviations in joint motion
10 ≤ score < 20	Moderate, noticeable biomechanical variation
≥ 20	Needs improvement, significant divergence

Table 1: Interpretation bands for DTW-based motion alignment scores.

In addition to dynamic DTW scores, the system evaluates static joint angles at critical frames. The generate_joint_recommendations() function compares the student's elbow and knee angles at both the load and release frames with those of the teacher, generating descriptive feedback.

```
  def generate_joint_recommendations(student_angle,
teacher_angle, joint_name):
    diff = student_angle - teacher_angle
    if abs(diff) < 5:
        return f"Your {joint_name} angle is well aligned with the
reference."
    elif diff > 0:
        return f"Try reducing your {joint_name} angle slightly
during the shot for a more compact form."
    else:
        return f"Consider increasing your {joint_name} angle to
better mirror the reference motion."
```

Listing 7: generate_joint_recommendations() function

After this, the vertical motion of the basketball is used to compute a release angle, offering trajectory-specific feedback using generate_trajectory_recommendation(). These trajectory metrics complement pose feedback by addressing timing, follow-through, and arc.

Constructing Final Summary and Feedback Output

At this stage, DTW alignment, joint angle comparison, and trajectory assessment, are combined into two final paragraphs that document the feedback and summary of calculations. These paragraphs are in a structured markdown format and formatted for rendering within the systems user interface (detailed more in Section 3.4.6)

By combining temporal alignment and pose accuracy evaluation, the system provides a comprehensive biomechanical feedback loop designed to improve shot technique over time.

3.4.6 GUI and System Output

To facilitate accessibility and streamline system interaction, the full pipeline was deployed through a browser accessible graphical user interface (GUI). The interface was developed using the Gradio framework, which supports rapid deployment of machine learning applications with minimal frontend overhead

(Figure 11) displays the initial wireframe of the interface, used during the planning stage to conceptualise a user friendly interface that will seamlessly integrate with system behaviour. The layout was kept intentionally minimalistic to prioritise usability and clarity.

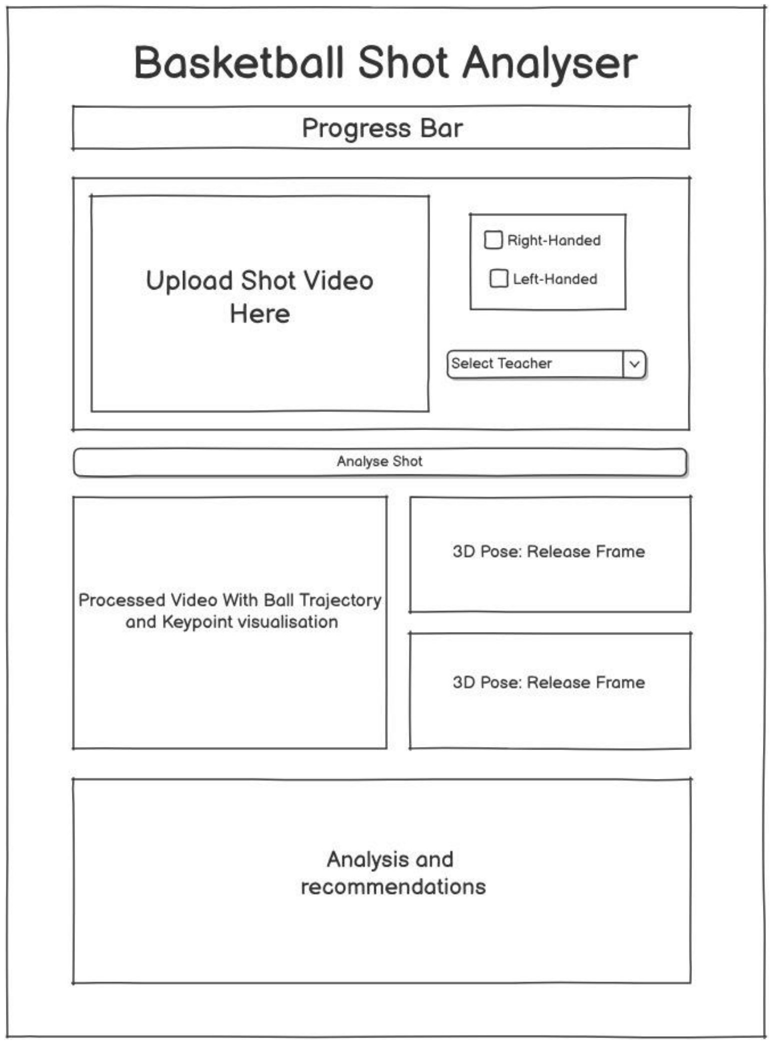

Figure 11: Initial GUI design (created using Balsamiq)

Interface Components and Interaction Flow

The final interface (See Figure 12) follows a similar structure to the initial wireframe, and consists of the following key elements**:**

- **Input Section:** Allows users to upload a video, select their dominant hand, choose a teacher from a predefined list (e.g., “SGA” or “Vanfleet”), and begin the analysis process through an "Analyse Shot" submit button
- **Output Section:** Displays the processed video overlaid with skeleton keypoints and ball trajectory. Static images of the 3D pose at load and release frames shown side by side for comparative analysis.
- **Analysis Section:** Provides a quantitative summary of pose and trajectory metrics, including DTW alignment scores and joint angles
- **Recommendations Section:** Outputs natural-language suggestions based on the identified discrepancies between student and reference motions.

Internally, clicking the "Analyse Shot" button triggers the analyse_video() function, which begins the pipeline outlined in the above sections (Tracking shooter and ball -> identifying and cropping key frames, etc.)

behind the frontend, each visual or textual output corresponds to a yield output within the

analyse_video() pipeline function. The Gradio layout leverages Blocks, Rows, and Columns to maintain a coherent structure while allowing for responsive layout changes depending on screen dimensions.

```
  # upload video and other input args
    with gr.Row():
        with gr.Column(scale=2):
            video_input = gr.Video(label="Upload Shot Video",
height=320)
        with gr.Column(scale=1):
            hand_radio = gr.Radio(
                ["Right-handed", "Left-handed"],
                label="Shooter's Dominant Hand",
                value="Right-handed"
            )
            teacher_dropdown = gr.Dropdown(
                ["SGA", "Vanfleet"],
                label="Select Reference Player (Teacher)",
                value="SGA"
            )
            submit_btn = gr.Button("Analyse Shot",
variant="primary")
```

Listing 8: Gradio GUI build Snippet (input args section)

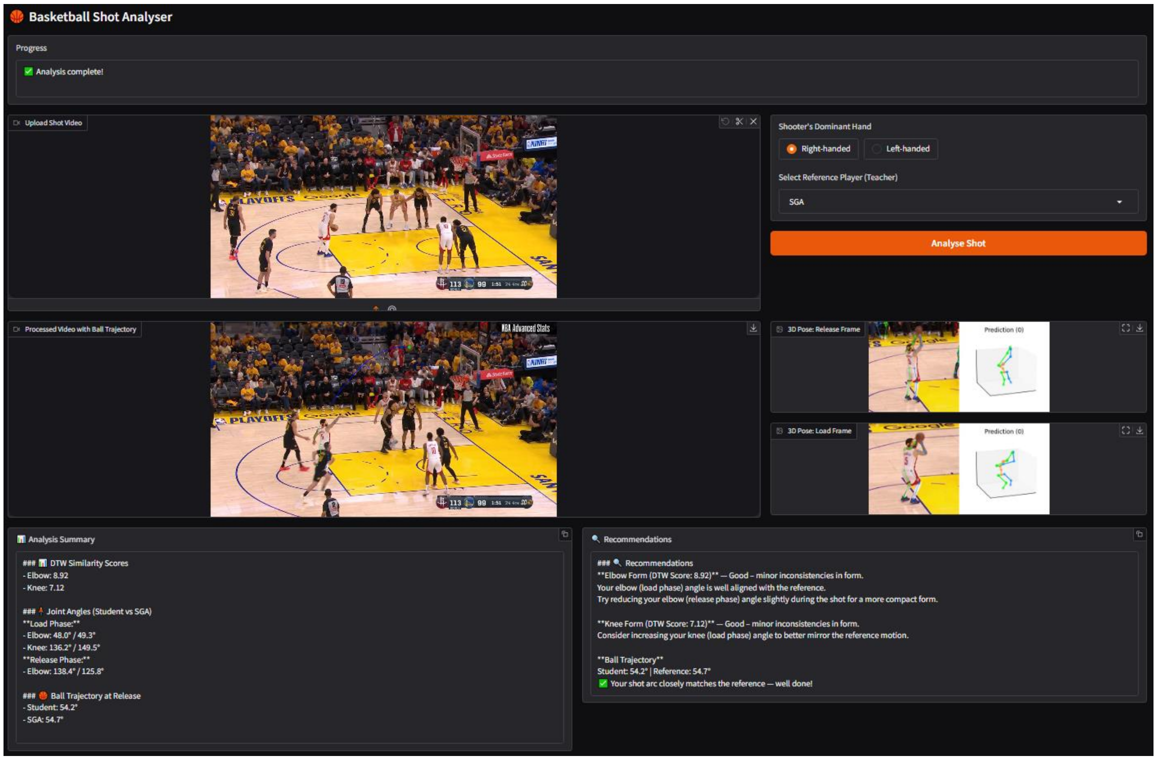

Figure 12: Final GUI implementation after analysis completion

Gradio was selected as the deployment interface due to its natural compatibility with Python-based machine learning workflows and its abstraction of front-end complexities. Compared to alternatives such as Flask or Streamlit, Gradio offered superior support for the media types typically associated with computer vision, including video frames and PIL-based pose images. This allowed for rapid prototyping and iterative testing throughout development, in line with the agile development approach outlined earlier in Section 3.1.

Moreover, Gradio's seamless integration with Google Colab and its support for yield-driven outputs made it ideal for long running processes like pose tracking and 3D lifting. These characteristics ensured that system feedback could be delivered as the process progressed.

3.5 Testing and Initial Deployment

Initial system testing was conducted using a combination of NBAPlayDB footage and the custom-collected real-world free throw videos recorded from university basketball participants. This phase of testing primarily focused on assessing the robustness of the full processing pipeline, including shooter identification, ball tracking, key phase detection, 3D lifting, and feedback generation.

Testing revealed that while the system operated consistently with NBA footage, real-world variability introduced notable challenges. Factors such as differing camera angles, lighting conditions, frame quality, and partial occlusions affected object detection accuracy. Most significantly, the YOLOv11-based ball detector struggled in scenarios where the ball was sized differently or more partially occluded, which resulted in missed detections that directly impacted shooter locking and phase detection, both of which initially depended on accurate ball tracking.

3.5.1 Adaptations to Shooter Locking Logic

To address the first limitation, modifications were made to the lock_shooter() function Originally, the system exclusively relied on identifying the shooter via proximity between detected wrist keypoints and the basketball's location. In the adapted version, a fallback mechanism was implemented: if no ball was detected but only one person was present in frame, that individual was automatically designated as the shooter.

This modification greatly increased the system's resilience in single-person video scenarios which is a common condition when capturing isolated practice footage. The fallback reduced the dependency on ball detection in cases where video clarity or camera angle made YOLO predictions unreliable.

```
# FALLBACK: Only one person -> assume shooter
        elif len(pred_instances) == 1:
            shooter_data = pred_instances[0]
            shooter_bbox = shooter_data['bbox'][0]
            print(f"✅ Shooter locked at frame {frame_id}
(fallback: only one person)")
            cap.release()
            return frame_id, shooter_bbox
```

Listing 9: Simple single person fallback logic snippet

3.5.2 Adaptations to Release Frame Identification

Ball detection difficulties also significantly impacted key phase detection. The original Release frame identification method relied solely on ball proximity to the wrist to determine the moment of release and as such, struggled when ball detection was not consistent. To improve robustness, a secondary fallback was incorporated into the find_release_frame() function. If no ball data was available, the system instead monitored vertical wrist velocity, identifying release candidates based on rapid upward wrist acceleration followed by peak wrist height.

This fallback utilised Savitzky–Golay filtering to smooth the wrist's vertical position and applied a gradient-based threshold to detect rapid upward motion, giving a biomechanically meaningful indicator of release.

```
# Smooth wrist positions
    y_array = np.array(all_wrist_y)
    smoothed_y = savgol_filter(y_array, 5, 2)
    """Inbetween code ..."""
    # Fallback velocity check if no ball detection
    elif ball_bbox is None:
        idx_in_full = frame_nums.index(candidate_frame)
        if idx_in_full < 3:
            continue

        prior_y_vals = smoothed_y[idx_in_full - 3: idx_in_full +
1]
        velocity = np.gradient(prior_y_vals)
        avg_upward = -np.mean(velocity[:-1])

        if avg_upward >= 3.0:  # threshold
            release_angle =
calculate_elbow_angle(candidate_data['keypoints'], shoulder_idx,
elbow_idx, wrist_idx)
            return candidate_frame, release_angle
```

Listing 10: Velocity-based fallback snippet

The implementation of these adaptive fallback mechanisms significantly improved the system's ability to process diverse real-world videos, making it better suited for use in real coaching contexts

Figure 13: Successful university student key phase isolation (faces blurred for identity protection).

3.5.3 System limitations and Remaining Challenges After Fixes

Despite these improvements, several challenges remain. The system's accuracy is still partially dependent on initial detection confidence, and multi-person scenarios with complex occlusions remain difficult to resolve without more sophisticated shooter identification.

Additionally, lighting variance, video resolution, and camera positioning continue to affect both object detection and pose inference quality. These factors highlight potential directions for future work, including multi-frame temporal smoothing, combination detection models, and model fine-tuning based on domain-specific training data.

3.6 Algorithmic Design Justifications

While developing the system pipeline, multiple design decisions were made regarding model selection, implementation logic, and computational trade-offs. This section provides justification for the key algorithmic and technical choices that shaped system implementation.

3.6.1 Selection of Dynamic Time Warping for Motion Alignment

Dynamic Time Warping (DTW) has proven to be an effective technique for comparing joint motion sequences in biomechanical assessments due to its ability to normalise and compare sequences with differing time-frames. Unlike traditional methods such as frame-by-frame comparisons or Euclidean distance metrics, DTW offers a greater degree of flexibility by allowing variable motion speeds and frame lengths to be aligned effectively (Wang and Piccardi 2016). This adaptability is particularly important in physical activities like shooting, where even skilled performers will exhibit variations in timing across performances. DTW addresses this challenge by identifying optimal temporal alignments, thus providing a more comprehensive measure of biomechanical similarity that does not punish slight differences in pacing.

In addition, research shows that DTW has lower computational overhead than other time-series comparison techniques, like hidden Markov models, which may increase its suitability for dynamic feedback settings in sports (Li and Wang 2019). The lightweight computational cost that DTW provides aids in the system's goals of delivering timely and effective feedback. Experimental results have demonstrated that DTW not only yields more accurate alignments in motion data but also scales efficiently, therefore supporting real-time application needs in the context of sports biomechanics (Wang and Zheng 2024)}. By utilising this analytical approach, the system can offer immediate insights that help users refine their techniques, ultimately contributing to improved performance and skill development.

3.6.2 Use of Savitzky-Golay Filtering for Release Detection

In scenarios where ball detection fails, using wrist velocity as a metric for estimating the release frame is an effective strategy. To accurately compute this velocity, the wrist Y-coordinate time series must be smoothed using a Savitzky-Golay filter. This choice of smoothing technique is supported by its capability to effectively suppress noise while retaining essential features of motion, particularly peaks in wrist elevation that correlate with the release of the ball (Syahrial et al. 2024). This is crucial, as accurate timing of these peaks is necessary for reliable release detection based on this value.

The Savitzky-Golay filter differs from more straightforward techniques like moving average smoothing, which may unintentionally mask inflection points that are essential for precisely identifying releases. Instead of reducing the importance of these crucial elements through over-smoothing, its design enables it to preserve them (Syahrial et al. 2024). In sports biomechanics, where fast and accurate identification of critical motion events can have a direct influence on performance evaluation and feedback, this trait is very advantageous.

Selecting Savitzky-Golay filtering aligns seamlessly with the overarching goal of providing immediate and accurate feedback based on the movement of joints.

3.6.3 Model Selection for Pose Estimation and Ball Detection

Because of its modular open-source structure, state-of-the-art accuracy on the COCO keypoint dataset, and ease of integration with pretrained weights, MMPose was chosen for 2D pose estimation. Following shooter identification, person-centric keypoint extraction was made possible by the top-down inference approach, and qualitative assessment of model performance throughout development was made even easier by MMPose's internal visualisation modules. Experiments with 3D inference were also made possible by its support for lifting pipelines, which eliminated the need for significant external code changes.

The selection of YOLOv11 for basketball-specific object identification is primarily due to its robust capabilities in accurately detecting the objects in motion across video frames. The YOLO (You Only Look Once) family of models is widely regarded for its exceptional balance between detection accuracy and processing speed, making it well-suited for real-time or near-real-time applications in sports analytics (Redmon et al. 2016). YOLO's single-pass architecture significantly reduces inference latency while maintaining precision, especially when compared to two-stage models like Faster R-CNN, which can experience increased computation time and latency. This efficiency is crucial in dynamic environments such as basketball, where timely feedback is essential for performance assessment.

Furthermore, the capability to fine-tune the YOLOv11 architecture on situationally distinct datasets allows for improved object detection tailored to specific contexts, such as free-throw footage (Li 2024). Previous studies support the assertion that fine-tuning enhances accuracy and robustness in identifying sports-related objects, as it optimizes the model's parameters based on relevant training data (Tripathi et al. 2016) The bounding box outputs generated by YOLO play a vital role not only in analysing wrist-ball proximity during shooting but also in visualizing the trajectory of the basketball, ensuring that all critical aspects of the shooting process are captured and visualised effectively.

4 Results and Evaluation

4.1 Quantitative performance

The system was evaluated across multiple video sources to assess both its biomechanical analysis capability and its robustness under real-world testing conditions. Two primary test scenarios were used: curated footage sourced from NBAPlayDB (representing ideal video conditions), and custom footage recorded from university basketball participants (representing uncontrolled, real-world settings).

For each test case, the system produced joint angle comparisons, DTW similarity scores, and natural language feedback based on the extracted biomechanics. Table 2 summarises representative quantitative outputs for both an NBA reference shot and a university student shot:

Test Case	DTW Elbow	DTW Knee	Trajectory Angle (°)
NBA Player (Fred Vanfleet)	9.46	7.22	56.4
University Student 4	44.27	23.81	N/A

Table 2: DTW scores and ball trajectory comparisons across test cases.

For the NBA test case, the system generated low DTW scores in both elbow and knee joints (9.46 and 7.22, respectively), consistent with expected high biomechanical similarity when another NBA player is used as reference (SGA -Shai Gilgeous-Alexander). The ball trajectory angle also closely matched the expert benchmark (56.4° vs 54.7°).

For the university student footage, the DTW scores were significantly higher (elbow: 44.27, knee: 23.81), indicating greater biomechanical deviation from the reference form, also consistent with an expected greater difference in form between amateur and professional. Due to YOLOv11's failure to detect the basketball upon release in this particular custom video, no ball trajectory output could be generated, though joint angles and dynamic comparison remained functional.

4.2 Visual Output Evaluation

In addition to numerical outputs, qualitative evaluation of visual outputs was conducted to verify the accuracy of 2D pose tracking and ball trajectory rendering. Figure 14 displays example visualisations produced by the system.

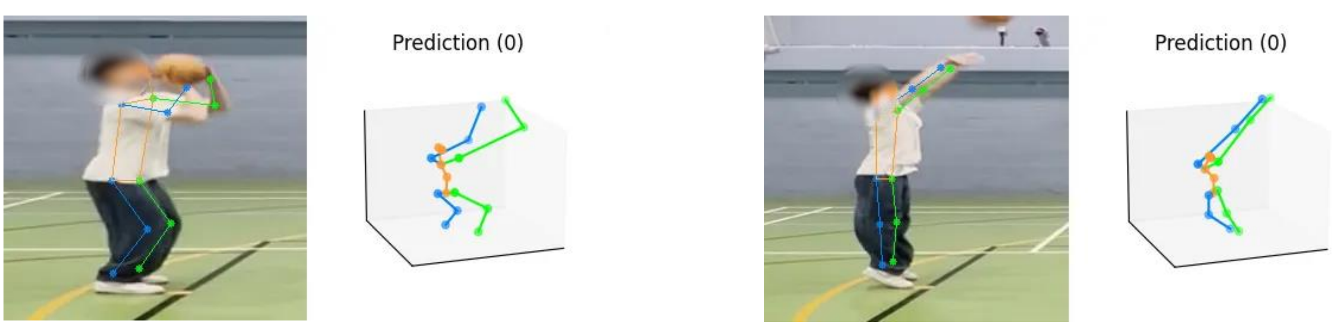

Figure 14: Student 3D visualisation example outputs (blurred)

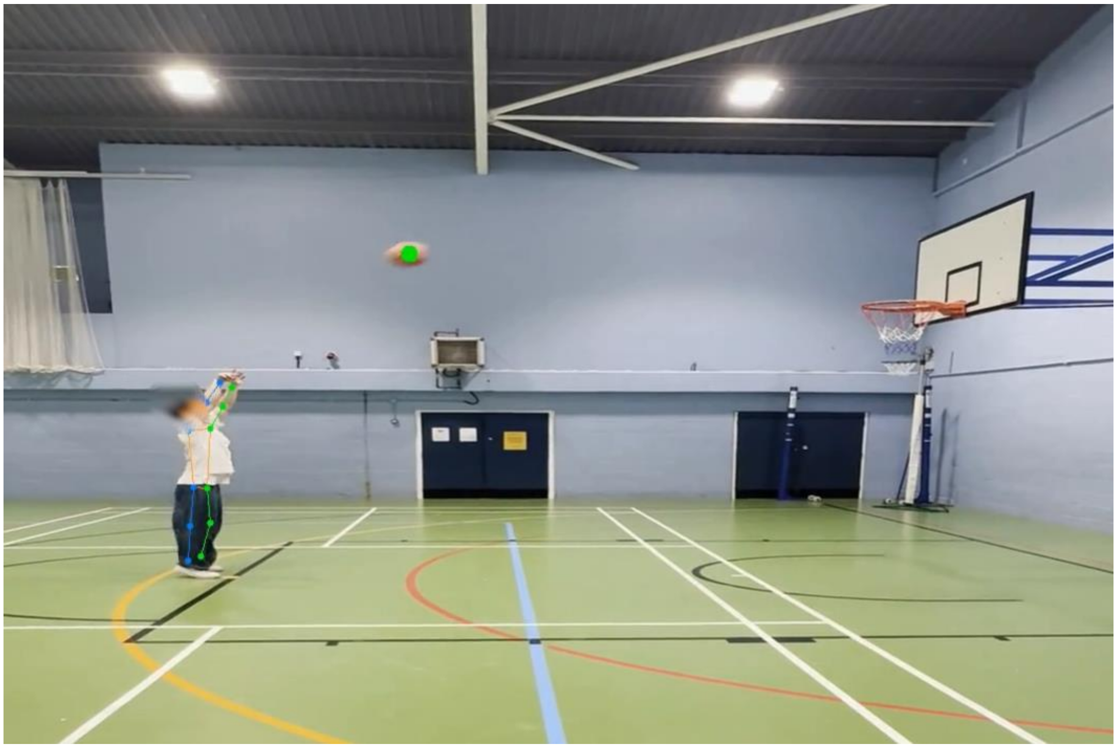

Figure 15: Student visualisation frame (blurred)

Despite variability in lighting and camera angles, 2D skeleton overlays remained stable across most frames in both NBA and university footage. The continuous tracking of the shooter across frames performed consistently, aided by the implemented fallback shooter-locking logic.

However, 3D pose lifting produced unstable joint estimations in both NBA and university test cases. The lifting model, trained primarily on controlled laboratory datasets, exhibited sensitivity to when lifting from singular frame inputs. As a result, the 3D reconstructions

4.3 Limitations and Failure cases

While the system successfully executes its core analysis pipeline across both controlled (NBAPlayDB) and real-world (university) footage, several limitations remain which affect certain modules under less constrained conditions.

As previously mentioned, the most significant limitation stems from the 3D pose lifting component. As the lifting model operates on single cropped frames, it lacks temporal context that could otherwise stabilise predictions. This makes it highly sensitive to small variations in scale, cropping, and input resolution across both NBA and university test cases. Consequently, the generated 3D reconstructions frequently appeared visually warped and biomechanically implausible, limiting their practical utility for detailed kinematic interpretation. Despite this, the extracted 2D keypoints remained stable, allowing joint angle computation and DTW-based temporal alignment to operate as intended.

A second major challenge lies in ball detection robustness. The custom-trained YOLOv11 model performed well on the NBAPlayDB training domain but struggled with university footage captured under differing lighting, resolution, and camera angles. In certain cases, complete failure to detect the ball prevented ball trajectory visualisation and eliminated trajectory-based feedback generation. This dependency remains a key source of vulnerability when generalising to broader deployment scenarios.

Additionally, while the adaptive fallback logic for shooter locking and release frame identification substantially improved performance, these heuristics remain best suited to stationary free-throw scenarios where shot phases are biomechanically predictable. Although it falls outside of the scope of this project, more dynamic shooting conditions, such as contested jump shots or partial occlusions, may reduce the reliability of these fallback mechanisms.

Finally, system runtime can become a practical consideration for full-length or high frame-rate videos as all clips passed through during development and testing did not exceed 30 fps, dramatically decreasing total compute time.

Despite these constraints, the system remains functional within its intended scope, providing actionable feedback for isolated shooting sessions and demonstrating a strong proof of concept for integrating AI-based pose analysis into applied sports training contexts.

5 Discussion and Implications

5.1 Reflection on System Performance

The system developed within this project successfully integrates multiple AI-driven components to automate the biomechanical analysis of basketball shooting technique. Across both controlled NBA footage and real-world university testing, the pipeline consistently extracted 2D pose keypoints, performed dynamic time warping (DTW) alignment, and generated joint-specific feedback based on quantitative comparisons with expert reference sequences. While not all subsystems operated with complete stability, the overall system consistently produced actionable feedback and demonstrated a high degree of autonomy in processing previously unseen footage.

The inclusion of fallback mechanisms played a critical role in maintaining system robustness. Modifications to the shooter-locking function allowed for correct player identification even when the basketball was not detected, with the addition of wrist-velocity-based release frame estimation ensuring that key biomechanical phases could still be identified when ball tracking did not perform correctly. These design decisions contributed to the system's ability to handle real-world footage, where conditions often deviated substantially from the controlled domain used during initial model training.

5.2 Relation to Existing Literature and Systems

Compared to existing systems or frameworks that analyse sports movements based on marker-based motion capture systems, this project demonstrates the feasibility of generating biomechanical insights from stand-alone video input exclusively. Marker less motion capture has previously been explored in laboratory contexts; however real-world deployment often encounters significant variability in video quality, resolution, and screen pollution. This system demonstrates a low-cost, accessible alternative that lowers technical barriers for athletes, coaches, and researchers alike.

The framework proposed by (Chen and Lu 2022) cannot be ignored during evaluation of the system documented in this report, due to the undeniable similarities the approaches share (combination of 3D pose estimation and dynamic time warping) to align student and teacher basketball shooting sequences. While their work focuses exclusively on pose-based analysis, this project extends that framework by integrating ball detection, automated shooter identification, and a fully deployed user interface for end-to-end feedback generation. The design choices made in this system also introduce greater robustness for real-world footage through adaptive fallback logic that allows operation even when certain detections fail.

5.3 Practical Applications

Despite its technical limitations, the system remains highly applicable to isolated basketball shooting scenarios such as free-throw practice sessions. By providing joint-specific metrics alongside actionable recommendations, the platform offers athletes and coaches an additional data-driven viewpoint, that can be used in conjunction with conventional coaching methods. The system's ability to automatically track the shooter, identify key shooting phases, and evaluate form using expert reference sequences introduces an objective framework for both skill development and long-term performance tracking.

Beyond individual player assessment, the system could be extended to support remote coaching, team-wide benchmarking, or integration into athlete development platforms that track improvements across extended training periods.

5.4 Limitations Framed as Research Challenges

While the system's core pipeline proved functional across multiple video scenarios, several underlying technical challenges remain. First, the custom-trained YOLOv11 ball detection model exhibited limited applicability beyond its original NBA training data or similar video conditions. Variations in lighting, camera angle, and ball scale contributed to detection failures on university-collected footage, preventing full trajectory analysis.

Secondly, the 3D pose lifting module operating on isolated frames demonstrated sensitivity to a one frame temporal window. This often produced physically implausible skeletal distortions, thereby limiting the usefulness of the 3D reconstructions for detailed analysis.

These outcomes reflect broader challenges in monocular 3D lifting, particularly when lacking temporal context.

Finally, while the implemented fallback mechanisms allowed the system to adapt to free-throw scenarios, their current design may struggle under more dynamic, multi-player game contexts where occlusions, motion complexity, and defensive interactions further increase detection ambiguity.

Whilst these challenges have been drawn exclusively from the design and implementation of this project, identified limitations align with previously documented problems in the existing literature. Similar instability in 3D lifting and concerns surrounding dynamic multi-person scenarios were also observed by (Chen and Lu 2022) in their basketball shooting analysis framework.

5.5 Future Work

Several clear directions for future enhancement of the system have emerged throughout this project:

- Expanding the teacher reference functionality to allow user generated keypoints to be added to the teacher library would enable fully customisable benchmarking and personalisation of comparisons.
- Replacing single frame 3D lifting with temporal sequence models would likely stabilise 3D predictions and mitigate the distortions seen in current reconstructions.
- Retraining the YOLOv11 object detection model on a more diverse set of video footage could improve the robustness of ball detection across varying video conditions.
- Assuming that models efficient enough are possible and implementable, the pipeline could be fully extended to a real-time framework that acts based on live camera input, further supporting live coaching sessions and practical use in real scenarios

6 Conclusion

This project successfully developed and evaluated an AI-powered basketball shot analysis system capable of extracting biomechanical insights from video footage. Through the integration of 2D pose estimation, custom-trained object detection, and dynamic time warping-based motion alignment, the system offers an automated solution for performance assessment in basketball free-throw shooting.

The implementation of contingency mechanisms for shooter locking and phase detection allowed for the system to remain operational even under challenging video conditions encountered in real-world data with university participants. Although the 3D pose lifting module displayed instability due to its reliance on single-frame lifting, the 2D keypoint tracking and angle analysis remained sufficiently secure enough to support meaningful feedback generation regardless.

The system's ability to generate actionable feedback through joint angle and angle sequence comparisons demonstrates its potential value for both athletes and coaches. By reducing reliance on subjective assessment, the platform provides an objective, quantitative based evaluation that could be used to supplement traditional coaching.

While several limitations were identified, these challenges align with already documented constrains in broader literature on marker-less motion capture in sports or AI systems, and do not necessarily reflect shortcomings solely of the framework structure, but rather problems yet to be solved in the general wider context of systems such as this one.

The work completed in this computing project represent a strong proof of concept for AI-assisted basketball coaching, while also identifying multiple pathways for future refinements/advancements. These include incorporating temporal 3D lifting models, expanding training datasets to improve detection generalisability, enabling user-customisable teacher keypoints, and exploring pathways towards real-time deployment.

Ultimately, this project contributes to the growing body of research demonstrating the feasibility of low-cost, accessible, AI-powered sports analysis tools that provide detailed biomechanical feedback, even outside of specialised laboratory environments.

References

Alqahtani, J. 2019. Security framework for cloud computing. *Iarjset* 6(12), pp. 27–31. doi: 10.17148/iarjset.2019.61207.

Baumgartner, T., Paaßen, B. and Klatt, S. 2023. Extracting spatial knowledge from track and field broadcasts for monocular 3d human pose estimation. *Scientific Reports* 13(1). doi: 10.1038/s41598-023-41142-0.

Bin, L. and Xu, X. 2021. Application of artificial intelligence in basketball sport. *Journal of Education, Health and Sport* 11(7), pp. 54–67. doi: 10.12775/jehs.2021.11.07.005.

Chen, Y. and Lu, H. 2022. Basketball Shooting Analysis via 3D Pose Estimation. Available at: https://web.stanford.edu/class/cs231a/prev_projects_2022/lCS231a_Project_Final_Report.pdf.

Claudino, J.G., Capanema, D. d. O., Souza, T.V. d., Serrão, J.C., Pereira, A.C.M. and Nassis, G.P. 2019. Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports: a systematic review. *Sports Medicine - Open* 5(1). doi: 10.1186/s40798-019-0202-3.

Demenius, J. and Kreivytė, R. 2017. The benefits of advanced data analytics in basketball: approach of managers and coaches of lithuanian basketball league teams. *Baltic Journal of Sport and Health Sciences* 1(104), pp. 8–13. doi: 10.33607/bjshs.v1i104.11.

Erabati, G.K., Gonçalves, N. and Araújo, H. 2020. Object detection in traffic scenarios - a comparison of traditional and deep learning approaches. *Computer Science &Amp; Information Technology*. doi: 10.5121/csit.2020.100918.

Ghezelseflou, H. and Choori, A. 2023. Athlete perspectives on ai-driven coaching technologies: a qualitative inquiry. *AI and Tech in Behavioral and Social Sciences* 1(1), pp. 4–11. doi: 10.61838/kman.aitech.1.1.2.

Hammes, F., Hagg, A., Asteroth, A. and Link, D. 2022. Artificial intelligence in elite sports—a narrative review of success stories and challenges. *Frontiers in Sports and Active Living* 4. doi: 10.3389/fspor.2022.861466.

Hwang, T., Kim, J. and Kim, M. 2023. A distributed real-time 3d pose estimation framework based on asynchronous multiviews. *KSII Transactions on Internet and Information Systems* 17(2). doi: 10.3837/tiis.2023.02.015.

Ji, Y., Zhang, J., Shi, Z., Liu, M. and Ren, J. 2018. Research on real – time tracking of table tennis ball based on machine learning with low-speed camera. *Systems Science & Control Engineering* 6(1), pp. 71–79. doi: 10.1080/21642583.2018.1450167.

Joshi, S. 2023. A systematic review of machine learning applications in cricket. *International Research Journal of Modernization in Engineering Technology and Science*. doi: 10.56726/irjmets37721.

Khanam, R. and Hussain, M. 2024. YOLOv11: An Overview of the Key Architectural Enhancements. Available at: https://arxiv.org/abs/2410.17725.

Kong, J., Ding, Y., Jiang, M. and Li, S. 2020. Collaborative model tracking with robust occlusion handling. *IET Image Processing* 14(9), pp. 1701–1709. doi: 10.1049/iet-ipr.2019.0827.

Li, H. and Wang, C. 2019. Similarity measure based on incremental warping window for time series data mining. *IEEE Access* 7, pp. 3909–3917. doi: 10.1109/access.2018.2889792.

Li, R. 2022. Cloud monitoring data analysis and simulation based on machine learning. *Machine Learning Theory and Practice* 3(3). doi: 10.38007/ml.2022.030307.

Li, Z. 2024. Accurate tracking method of basketball flight trajectory based on yolo and multi-level data association. *International Journal of High Speed Electronics and Systems* 34(01). doi: 10.1142/s0129156425401895.

Lin, T.-Y. et al. 2014. Microsoft coco: Common objects in context. In: *European conference on computer vision*. pp. 740–755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A.C. 2016. SSD: Single Shot MultiBox Detector. pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

Lotfi, S. and Rebbouj, M. 2021. Machine learning for sport results prediction using algorithms. *International Journal of Information Technology and Applied Sciences (IJITAS)* 3(3), pp. 148–155. doi: 10.52502/ijitas.v3i3.114.

Mane, D. 2024. A metaphor analysis on vehicle license plate detection using yolo-nas and yolov8. *Journal of Electrical Systems* 20(1s), pp. 152–164. doi: 10.52783/jes.761.

Molavian, R., Fatahi, A., Abbasi, H. and Khezri, D. 2023. Artificial intelligence approach in biomechanics of gait and sport: a systematic literature review. *Journal of Biomedical Physics and Engineering* 13(5). doi: 10.31661/jbpe.v0i0.2305-1621.

Naik, B.T. and Hashmi, M.F. 2022. Yolov3-sort: detection and tracking player/ball in soccer sport. *Journal of Electronic Imaging* 32(01). doi: 10.1117/1.jei.32.1.011003.

Nie, Y., Lee, J., Yoon, S. and Park, D.J. 2019. A multi-stage convolution machine with scaling and dilation for human pose estimation. *KSII Transactions on Internet and Information Systems* 13(6). doi: 10.3837/tiis.2019.06.023.

Padala, V.S., Gandhi, K. and Dasari, P. 2019. Machine learning: the new language for applications. *IAES International Journal of Artificial Intelligence (IJ-AI)* 8(4), p. 411. doi: 10.11591/ijai.v8.i4.pp411-421.

Pavllo, D., Feichtenhofer, C., Grangier, D. and Auli, M. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. Available at: https://arxiv.org/abs/1811.11742.

Rathnayake, T., Gostar, A.K., Hoseinnezhad, R., Tennakoon, R. and Bab-Hadiashar, A. 2020. On-line visual tracking with occlusion handling. *Sensors* 20(3), p. 929. doi: 10.3390/s20030929.

Rawashdeh, A., Rawashdeh, B.S. and Shehadeh, E. 2023. The determinants of cloud computing vision and its impact on cloud accounting adoption in smbs. *Human Behavior and Emerging Technologies* 2023, pp. 1–15. doi: 10.1155/2023/8571227.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. 2016. You only look once: unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788. doi: 10.1109/cvpr.2016.91.

Robertson, S. et al. 2023. Challenges and considerations in determining the quality of electronic performance & tracking systems for team sports. *Frontiers in Sports and Active Living* 5. doi: 10.3389/fspor.2023.1266522.

Šajina, R. and Ivašić-Kos, M. 2022. 3d pose estimation and tracking in handball actions using a monocular camera. *Journal of Imaging* 8(11), p. 308. doi: 10.3390/jimaging8110308.

Sithipolvanichgul, J., Chen, C., Land, J. and Ractham, P. 2021. Enhancing user experiences with cloud computing via improving utilitarian and hedonic factors. *Energies* 14(7), p. 1822. doi: 10.3390/en14071822.

Stenum, J., Cherry-Allen, K., Pyles, C., Reetzke, R., Vignos, M. and Roemmich, R. 2021. Applications of pose estimation in human health and performance across the lifespan. *Sensors* 21(21), p. 7315. doi: 10.3390/s21217315.

Syahrial, S., Melinda, M., Junidar, J., Razali, S. and Zulhelmi, Z. 2024. Application of the savitzky-golay filter in multi-spectral signal processing. *Sriwijaya Electrical and Computer Engineering Journal* 1(1), pp. 9–19. doi: 10.62420/selco.v1i1.5.

Terven, J., Córdova-Esparza, D. and Romero-González, J. 2023. A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction* 5(4), pp. 1680–1716. doi: 10.3390/make5040083.

Tripathi, S., Lipton, Z.C., Belongie, S. and Nguyen, T. 2016. Context matters: refining object detection in video with recurrent neural networks. *Procedings of the British Machine Vision Conference 2016*, pp. 44.1-44.12. doi: 10.5244/c.30.44.

Vasileiadis, M., Bouganis, C. and Tzovaras, D. 2019. Multi-person 3d pose estimation from 3d cloud data using 3d convolutional neural networks. *Computer Vision and Image Understanding* 185, pp. 12–23. doi: 10.1016/j.cviu.2019.04.011.

Wang, H. and Zheng, Q. 2024. Improvement and application of hale’s dynamic time warping algorithm. *Symmetry* 16(6), p. 645. doi: 10.3390/sym16060645.

Wang, Z. and Piccardi, M. 2016. A pair hidden markov support vector machine for alignment of human actions. *2016 IEEE International Conference on Multimedia and Expo (ICME)*. doi: 10.1109/icme.2016.7552933.

Wu, W., Xu, M., Liang, Q., Li, M. and Peng, Y. 2020. Multi-camera 3d ball tracking framework for sports video. *Iet Image Processing* 14(15), pp. 3751–3761. doi: 10.1049/iet-ipr.2020.0757.

Zhang, J., Wu, Q., Zheng, R., Zhu, J., Zhang, M. and Liu, R. 2018. A security monitoring method based on autonomic computing for the cloud platform. *Journal of Electrical and Computer Engineering* 2018, pp. 1–9. doi: 10.1155/2018/8309450.

List of Figures

List of Listings

List of Tables