**Introducing: AceVision**

by: Ibrahim Masa’deh

Contents

[**Chapter 1: Introduction** 4](#_Toc195605413)

[1.1 Purpose of the Project 4](#_Toc195605414)

[1.2 Problem Statement 4](#_Toc195605415)

[1.3 Project Scope 4](#_Toc195605416)

[1.4 Target Audience 4](#_Toc195605417)

[1.5 Document Overview 4](#_Toc195605418)

[**Chapter 2: Background and Literature Review** 4](#_Toc195605419)

[2.1 Background 5](#_Toc195605420)

[2.2 Related Work 5](#_Toc195605421)

[2.3 Theoretical Foundations 5](#_Toc195605422)

[**Chapter 3: Solution Design** 5](#_Toc195605423)

[3.1 System Architecture 5](#_Toc195605424)

[3.2 Component Descriptions 5](#_Toc195605425)

[3.3 Data Flow Diagram 6](#_Toc195605426)

[3.4 Running The Program 6](#_Toc195605427)

[**Chapter 4: Data Analysis and Results** 6](#_Toc195605428)

[4.1 Data Sources 6](#_Toc195605429)

[4.1.1 Local Data 6](#_Toc195605430)

[4.1.2 External Sources 7](#_Toc195605434)

[4.2 Player and Ball Detection 7](#_Toc195605437)

[4.3 Court Keypoints 7](#_Toc195605439)

[4.4 Pose and Movement Analysis 8](#_Toc195605440)

[4.5 Results Summary 8](#_Toc195605442)

[**Chapter 5: Conclusions and Future Work** 8](#_Toc195605443)

[5.1 Conclusions 8](#_Toc195605444)

[5.2 Limitations 8](#_Toc195605445)

[5.3 Future Work 9](#_Toc195605446)

# Chapter 1: Introduction

## 1.1 Purpose of the Project

**AceVision** is a prototype built using computer vision techniques to evaluate tennis player performance by analyzing body posture and motion. The goal is to provide athletes and coaches with automated, objective feedback on incorrect stances and court positioning.

## 1.2 Problem Statement

Traditional methods of evaluating tennis players often rely on subjective human judgment, which can be inconsistent and error-prone. Additionally, evaluating each player's technique can be time-consuming, especially during matches or high-volume training sessions. AceVision addresses this by using advanced computer vision algorithms to deliver accurate, repeatable feedback on player movement and posture, freeing up coaching time for more advanced training.

## 1.3 Project Scope

**AceVision** is designed to analyze pre-recorded tennis match videos, focusing on:

* Detecting players and tennis ball across video frames.
* Analyzing joint angles and posture features (e.g., knee flexion, torso lean).
* Assessing whether players are positioned correctly on the court.
* Logging performance issues and suggesting improvements.

## 1.4 Target Audience

This documentation is intended for:

* Developers working in sports analytics and pose estimation.
* Tennis coaches and athletes interested in automated feedback systems.
* Tennis players who want to identify and improve weaknesses in their gameplay.

## 1.5 Document Overview

* **Chapter 2**: Reviews background technologies and relevant literature.
* **Chapter 3**: Describes the software architecture and individual components.
* **Chapter 4**: Discusses input data, processing methods, and outcomes.
* **Chapter 5**: Summarizes findings and proposes directions for future work.

Chapter 2: Background and Literature **Review**

## 2.1 Background

The evolution of computer vision has made it feasible to analyze human movement with considerable accuracy. Tools like **YOLO** **(You Only Look Once)**, **MediaPipe**, and **PyTorch** facilitate real-time object detection; pose estimation, and classification, particularly useful in applications like sports performance monitoring.

## 2.2 Related Work

Several systems have been developed to analyze athletic motion using deep learning. For instance, **golf swing analysis**, **basketball shooting form evaluation**, and **soccer player tracking** have all leveraged pose estimation to provide actionable feedback. **AceVision** applies similar approaches to tennis.

## 2.3 Theoretical Foundations

* **Object Detection**: **YOLOv8 [1]** is used to identify players and the tennis ball in each frame.
* **Pose Estimation**: **MediaPipe [2]** extracts key body joints for motion analysis.
* **Court Mapping**: A custom model developed in **PyTorch [3]** detects tennis court keypoints, enabling context-aware analysis**.**
* **Movement Analysis**: A rule-based approach is used to analyze player movement by evaluating joint angles and posture.

# Chapter 3: Solution Design

## 3.1 System Architecture

**AceVision** follows a modular pipeline:

1. Input a pre-recorded tennis video.
2. Use **YOLOv8** to detect players and the tennis ball.
3. Detect court keypoints using a **PyTorch** model.
4. Apply **MediaPipe** for pose estimation.
5. Analyze **player positioning** and postural mechanics.
6. Generate **feedback** based on movement assessment.

## 3.2 Component Descriptions

* **Player Detection**: Conducted using **yolov8x.pt** to identify players frame by frame.
* **Ball Detection**: A custom-trained **YOLOv8** model is used to track ball trajectory.
* **Court Keypoint Detection**: Implemented via a **PyTorch** model that outputs key tennis court coordinates.
* **Pose Estimation**: **MediaPipe** captures body landmarks essential for posture evaluation.
* **Feedback Generation**: The system compares measured **joint angles** with ideal values and flags deviations.

## 3.3 Data Flow Diagram

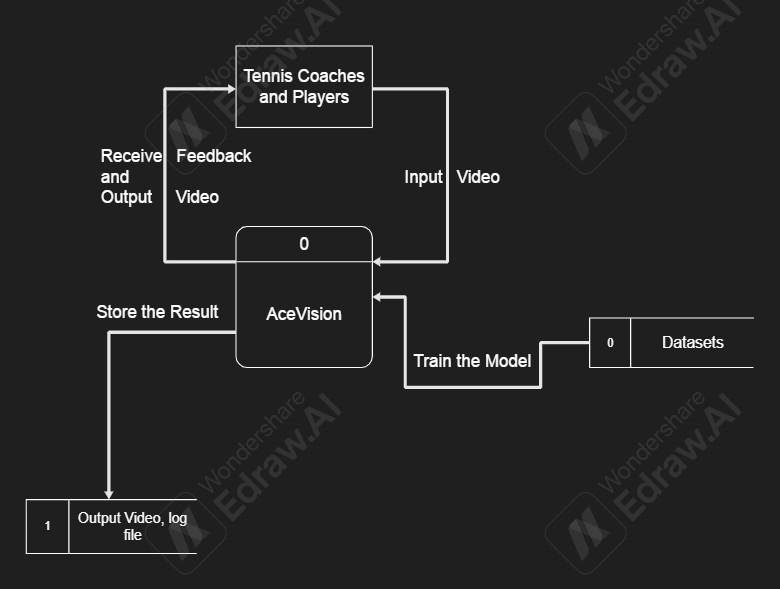


Figure 1.0

## 3.4 Running the Program

To run the system, users need to follow these instructions provided in the **README.md** file. It outlines:

* Creating and activating virtual environment.
* Installing dependencies using **pip install –r requirements.txt**
* Running the main script using **python main.py**

# Chapter 4: Data Analysis and Results

## 4.1 Data Sources

### 4.1.1 Local Data

* Input videos are stored in the **input\_videos/** directory, see figure 2.1

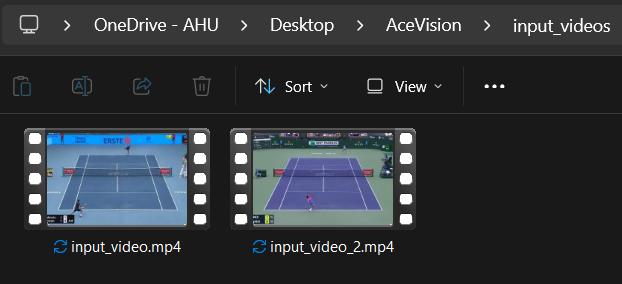


Figure 2.1

* Detections (player and ball) are saved as **.pkl** files for faster testing and reproducibility.
* Output results, including videos and logs, are stored in organized folders under **outputs/** see figure 2.2

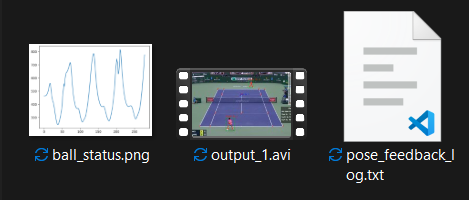


Figure 2.2

### 4.1.2 External Sources

* **Input Videos**: The input videos were taken from **YouTube**, and then the video clips were trimmed for use in the project **[4][5]**.
* **Tennis Ball Dataset [6]:** A custom dataset was created for tennis ball detection, specifically tailored to the needs of the project.
* **Court Line Detection:** The court line detection code was developed using a publicly available **dataset** **[7]** from a **GitHub** repository **[8]**, which was then integrated into the system.

## 4.2 Player and Ball Detection

## The YOLOv8x model provided accurate player detection in most video frames, see figure 2.1

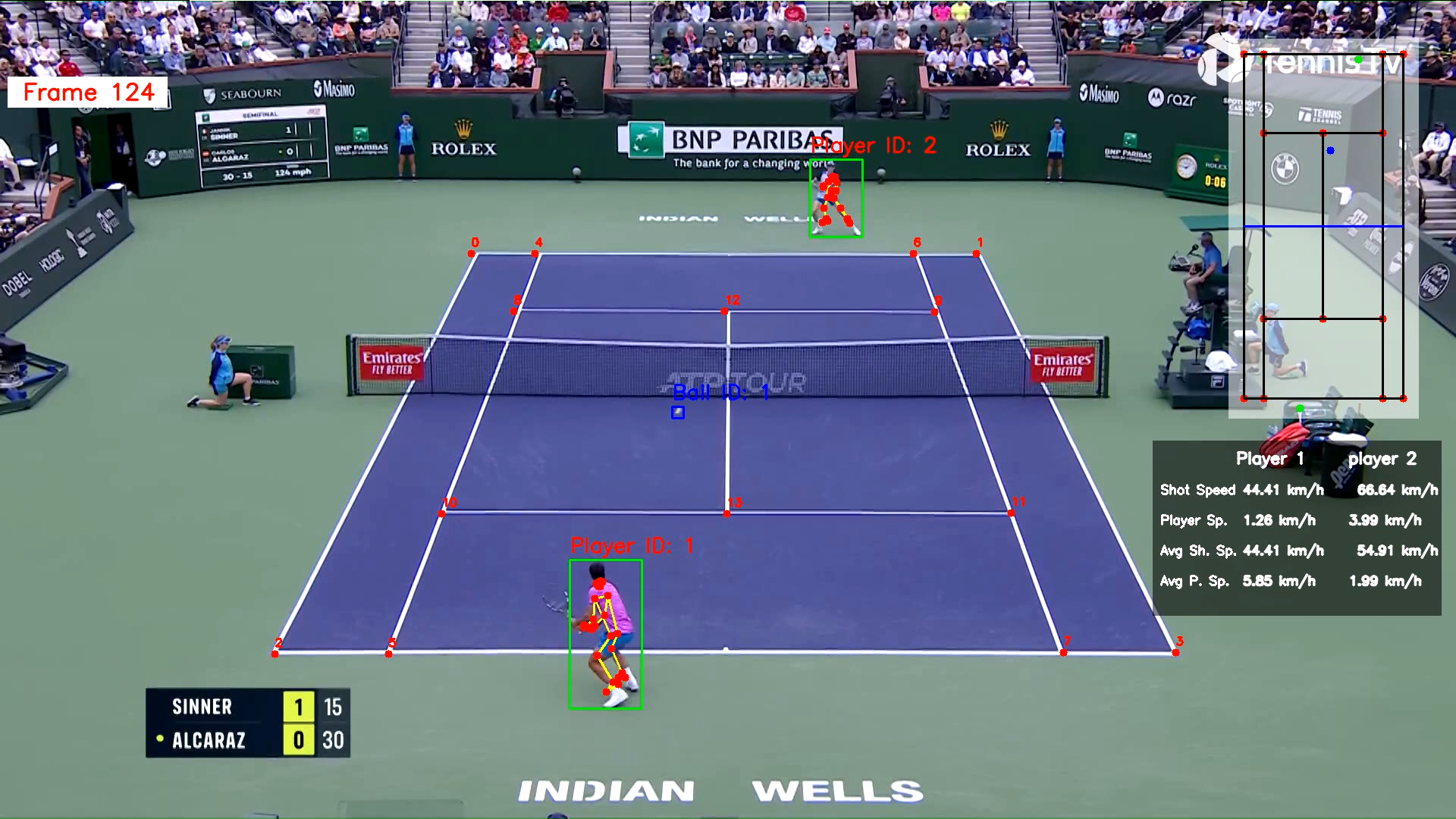


Figure 2.1

* The custom **YOLOv8** model for ball tracking demonstrated stable and consistent results.

## 4.3 Court Key-points

* For reduced computational complexity, court key-points were detected only once from the first frame. While the model is capable of detecting court key-points across different frames, relying on the initial frame allows the system to maintain sufficient accuracy while minimizing redundant computations. If key-point detection were to be extended frame by frame, only the code implementation would need to be adjusted—no architectural changes are required. See figure 2.1

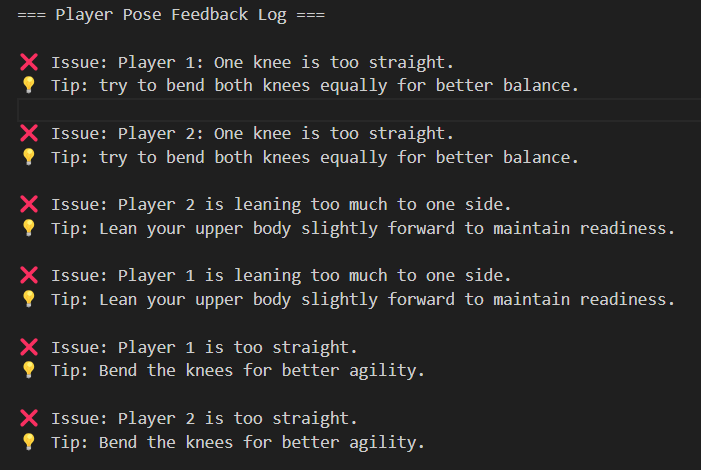
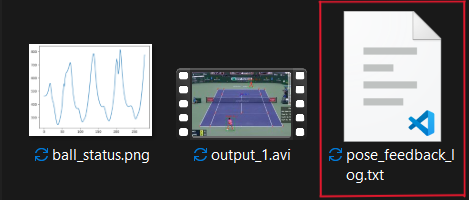
## 4.4 Pose and Movement Analysis

## The system measured knee angles, torso lean, and foot placement.

* Poses were evaluated against rules like:
  + Knee angle should be less than 170° for agility.
  + Torso should lean forward slightly for readiness.
* Players deviating from these criteria were flagged and logged.

## 4.5 Results Summary

* The rule-based system flagged incorrect postures based on predefined heuristics, providing practical assessments of player form.
* The system generated comprehensive output visualizations and feedback. The video included bounding boxes around detected players and the ball, real-time statistics such as average player speed and ball speed, and a mini-court overlay for spatial context. Additionally, ball hit positions were plotted throughout the match.
* A separate log file was also produced, detailing incorrect postures and providing corresponding recommendations for improvement.
* Text-based comments were rendered to assist in reviewing performance.



# Chapter 5: Conclusions and Future Work

## 5.1 Conclusions

**AceVision** validates the application of computer vision in sports performance evaluation. The fusion of object detection, pose estimation, and biomechanical analysis resulted in a system capable of providing detailed movement feedback.

## 5.2 Limitations

* High-resolution cameras were not used to capture the videos, which may lead to motion blur and misdetections. As a result, the accuracy of pose estimation can be negatively affected.
* The system is tuned for doubles tennis and might not generalize to singles matches or other sports.

## 5.3 Future Work

* Add support for real-time input from live cameras.
* Improve model generalization across lighting, player types and players' number.
* Extend posture analysis rules for broader feedback.
* Build a web interface to enhance accessibility.
* Train on a larger, more diverse dataset for better performance.
* Analyze the players' posture using a machine learning model instead of a rule-based approach, to improve flexibility and allow for data-driven assessments of pose correctness.
* Augmented Reality-Based Posture Feedback
  + As a potential future enhancement, the system can be extended to include an Augmented Reality (AR) module that provides real-time visual posture corrections. This feature would overlay feedback elements—such as ideal joint angles, corrective lines, or guidance arrows—directly on the live video feed of the player. By visually comparing their posture to the ideal one, players can make immediate adjustments during practice sessions. This addition would make the training more interactive, intuitive, and accessible, particularly for solo players or those working with remote coaches. Integrating AR technology not only enhances user engagement but also improves the effectiveness of the feedback loop without the need for textual interpretation or post-session analysis.

**References**

[1] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," arXiv preprint arXiv:2408.15857, Aug. 2024. [Online]. Available: <https://arxiv.org/abs/2408.15857>

[2] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, Jun. 2019. [Online]. Available: <https://arxiv.org/abs/1906.08172>

[3] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32 (NeurIPS 2019), 2019, pp. 8024–8035. [Online]. Available: <https://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>

[4] Australian Open TV, “Novak Djokovic v Carlos Alcaraz Extended Highlights | Australian Open 2025 Quarterfinal”, YouTube, Jan. 21, 2025. [Online]. Available: <https://www.youtube.com/watch?v=Up7jxaRYt1o>

[5] Tennis TV, “THRILLING Jannik Sinner vs Carlos Alcaraz Match | Indian Wells 2024 Highlights”, YouTube, Mar. 17, 2024. [Online]. Available: <https://www.youtube.com/watch?v=fTKWoPfkm1M&t=15s>

[6] I. Masa’deh, “tennis ball tracker”, Roboflow, Mar. 15, 2025. [Online]. Available: <https://universe.roboflow.com/computervision-lp8us/tennis-ball-tracker-bywbk/dataset/2>

[7] kosolapov.aetp, “tennis\_court\_det\_dataset.zip”, Google Drive, Jun. 16, 2023, [Online]. Available: <https://drive.google.com/file/d/1lhAaeQCmk2y440PmagA0KmIVBIysVMwu/view>

[8] yastrebksv, “TennisCourtDetector”, GitHub, Mar. 15, 2024. [Online]. Available: <https://github.com/yastrebksv/TennisCourtDetector>