ECE-GY 6143 Project

qin xuchuan(xq730), yang huancheng(hy3281)

December 2025

1 Project Overview

1.1 Project Brief

This project implements a **video-based basketball shooting form analyzer**. Given a shooting video as input, the system applies YOLOv8 pose estimation to extract human keypoints frame by frame, segments the motion into meaningful phases, and computes interpretable biomechanical metrics. It then produces a final score (0–100), a list of detected issues supported by quantitative evidence, actionable coaching suggestions, and an annotated output video/report for easy review.

1.2 Problem Statement

Shooting-form assessment is often performed through manual observation, which can be subjective, difficult to quantify, and hard to reproduce consistently across sessions. **The goal of this project** is to build an accessible tool that requires only a regular video (no wearable sensors) yet can **automatically analyze the shooting motion**, identify common technique problems, and deliver objective, explainable feedback to help players improve efficiently.

2 Code Structure and Pipeline

2.1 Entry Point and Orchestration

The repository is driven by `main.py`, which acts as a simple command-line interface (CLI). It scans the working directory for input videos, prompts the user to select a target clip and a YOLOv8 pose model size, and then calls the core analysis pipeline (`ShootingAnalyzer`). After processing, it prints a concise summary (score, issues, key metrics) and saves the corresponding output artifacts.

2.2 Core Technique and Model Usage

The system is built around pose estimation from monocular video. It uses Ultralytics YOLOv8 Pose pretrained weights (`.pt`) to infer human keypoints on each frame. OpenCV is used for video decoding/encoding and for generating the annotated output video, while NumPy is used for numerical computations and metric aggregation.

2.3 Training Strategy

This codebase does not include a training loop. Instead, it relies on pretrained YOLOv8 Pose models for keypoint extraction and focuses the project contribution on downstream motion understanding: phase segmentation, metric design, scoring, and feedback generation.

2.4 Analysis, Metrics, and Evaluation Logic

After keypoints are extracted over time, the analyzer derives phase-aware motion indicators (e.g., knee flexion characteristics, elbow–wrist alignment ratios, elbow tucked ratio during loading, hip vertical displacement, and other pose-derived kinematic proxies). These metrics are then combined into (1) an overall score on a 0–100 scale, (2) a structured list of detected issues with quantitative evidence, and (3) actionable coaching suggestions. Evaluation in this project is therefore metric-driven and explainable, rather than a dataset-level detection benchmark (e.g., mAP).

2.5 Outputs and Reporting

For each analyzed clip, the pipeline produces an annotated output video (filename contains `_motion_analyzed`) and saves both machine-readable and human-readable reports (JSON and TXT). The reports contain the overall score, detected issues, computed key metrics, and phase segmentation indices, enabling users to trace each conclusion back to measurable motion evidence.

2.6

Repository Layout (Illustrative)

```
shooting-motion-analyzer/
|-- main.py
|-- README.md
|-- requirements.txt
|-- sho_report.json
|-- sho_report.txt
|-- weights/
|   |-- yolov8m-pose.pt
|   |-- yolov8l-pose.pt
```

```
|   |-- yolov8x-pose.pt
|   |-- yolov8n-pose.pt
|   |-- yolov8n.pt
|   |-- yolov8s.pt
|   `-- yolov8x.pt
`-- videos/
    |-- example.mp4
    `-- example_motion_analyzed.mp4
```

3 Approach and Evaluation

3.1 Approach

Our approach is an inference-driven pipeline for shooting-form assessment from a single input video. First, we run a pretrained YOLOv8 Pose model to estimate human keypoints for each frame. We then treat the resulting keypoint trajectories as a motion time series and segment the shot into phases (e.g., setup, loading, release, follow-through). Next, we compute a set of interpretable, phase-aware kinematic proxies from the keypoints, such as knee flexion characteristics, elbow–wrist alignment measures, elbow tucked ratio during the loading phase, and hip vertical displacement. Finally, we map these metrics to an overall score and a set of issue detections, each paired with quantitative evidence and actionable coaching recommendations.

3.2 Evaluation

Evaluation in this project is metric-driven and explainable rather than benchmark-driven (e.g., mAP on a labeled dataset). For each analyzed clip, we report: (1) a final performance score on a 0–100 scale, (2) a ranked list of detected issues with supporting evidence (the specific metric values that triggered the diagnosis), and (3) the underlying computed metrics and phase boundaries to ensure traceability. In the provided example report, the system identifies common form problems such as insufficient elbow tuck during loading and insufficient vertical drive, and justifies these findings using the corresponding alignment ratios and hip displacement measurements.

4 Example Screenshot

Figure 1 (Input video frame). This figure shows a representative frame from the raw input clip used in our demo. The shooter is captured in a typical pre-release posture, and this unmodified video serves as the only input to the system (no wearable sensors or manual annotations). The goal is to analyze shooting mechanics directly from this monocular recording.

Figure 1: Input video screenshot

Figure 2 (processing progress). This screenshot captures the interactive command-line workflow in the development environment. The user selects the input video and the YOLOv8 pose model size, after which the analyzer initializes the pose and ball models and processes the video with a visible progress indicator (resolution, FPS, frame count, and processing speed).

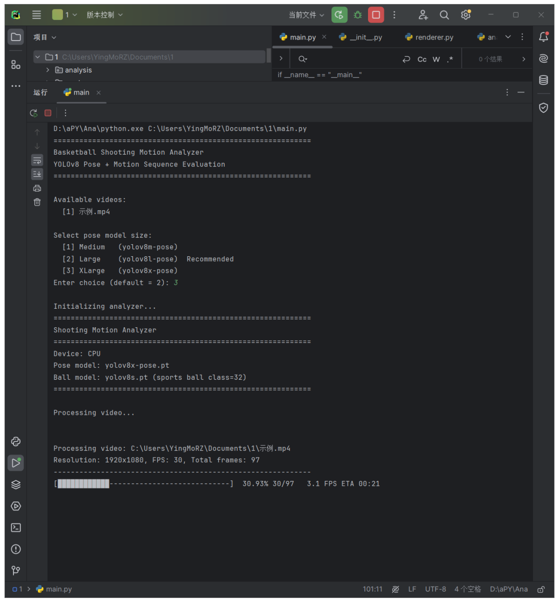
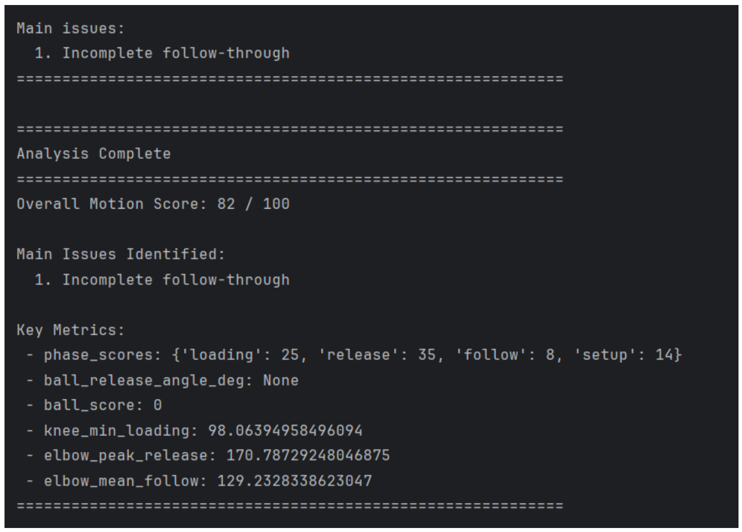


Figure 2: Project processing terminal

Figure 3 (Annotated video during pose tracking). This frame is taken from the generated `_motion_analyzed` output video while the system is running pose estimation. The overlaid skeleton and joint connections visualize the detected keypoints in real time, demonstrating how the model tracks the shooter's body configuration frame by frame to form a clean motion time series for downstream scoring.

Figure 3: Output video screenshot

Figure 4 (Terminal results summary). This screenshot presents the end-to-end analysis summary printed by the program after processing the clip. It reports the overall motion score (e.g., 82/100), highlights the main issue detected (in this example, incomplete follow-through), and lists key quantitative metrics computed from the pose sequence, which provide traceable evidence behind the diagnosis.



```
Main issues:
  1. Incomplete follow-through
============================================================

============================================================
Analysis Complete
============================================================
Overall Motion Score: 82 / 100

Main Issues Identified:
  1. Incomplete follow-through

Key Metrics:
 - phase_scores: {'loading': 25, 'release': 35, 'follow': 8, 'setup': 14}
 - ball_release_angle_deg: None
 - ball_score: 0
 - knee_min_loading: 98.06394958496094
 - elbow_peak_release: 170.78729248046875
 - elbow_mean_follow: 129.2328338623047
============================================================
```

Figure 4: Output terminal

Figure 5 (Final score and coaching tip overlay). This figure shows the final result screen rendered into the output video. The system displays the overall score and a short coaching tip summarizing the primary correction target, allowing users to quickly understand the key takeaway without reading the full report.

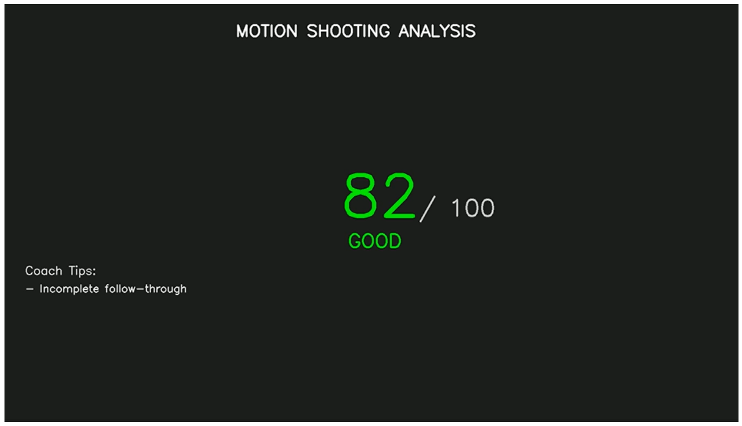


Figure 5: Output score screenshot

5 Project Summary

In this project, we built a practical, video-based basketball shooting motion analyzer that provides objective and explainable feedback from a single monocular clip. The system leverages pretrained YOLOv8 Pose to extract human keypoints on every frame, converts the keypoints into a structured motion sequence, and then evaluates the shot using phase-aware, interpretable kinematic proxies (e.g., joint angles and alignment-based measures). Based on these metrics, the pipeline outputs an overall performance score (0–100), identifies the most critical form issues with quantitative evidence, and generates actionable coaching tips. To improve usability, it also produces a motion-analyzed video with visual overlays and a structured report so users can quickly review results and track improvements across sessions. In our demonstration run, the tool produced a score in the low 80s and flagged incomplete follow-through as the primary correction target, illustrating how the system translates pose trajectories into concrete training feedback. Overall, the project demonstrates an accessible and reproducible workflow for shooting-form assessment without specialized sensors, while leaving room for future extensions such as more robust ball-release detection, multi-person filtering, and data-driven calibration of scoring thresholds.