Cricket Ball Trajectory Detection & Tracking – Technical Report

1. Project Overview

This project focuses on detecting and tracking a cricket ball in broadcast-style cricket videos using a YOLO-based object detection pipeline. The system generates both a visual trajectory overlay and structured per-frame annotations suitable for analysis or downstream learning tasks.

2. Model Selection and Training Decisions

A YOLO object detector was selected due to its strong performance on small objects, real-time inference capability, and ease of deployment. The model was fine-tuned on a custom cricket ball dataset, with a single target class (cricket ball).

- • YOLO architecture used for efficient small-object detection
- • Single-class training to reduce class confusion
- • High-resolution inference (1280×1280) to preserve ball details

3. Assumptions Made

- • Only one cricket ball is visible per frame
- • The ball appears predominantly within the pitch region
- • Input video is recorded from a relatively fixed camera angle

4. Fallback Logic and Robustness Handling

To ensure stability in real-world videos, several fallback mechanisms were introduced. If the ball is not detected in a frame, the system records the ball as not visible while maintaining historical trajectory information.

- • Visibility flag set to 0 when detection is missing
- • Trajectory drawn only using valid historical detections
- • Pitch-based spatial filtering to remove false positives

5. Performance Issues Encountered

Several performance and accuracy issues were observed during development, particularly when running inference on CPU-based systems.

- • Low FPS due to high-resolution inference
- • False detections on players and background objects
- • Static false positives when the ball left the frame

6. Improvements and Fixes Applied

- • Restricted detection to central pitch region (20% width ignored on both sides)
- • Increased confidence threshold to reduce false positives
- • Added per-frame CSV logging for debugging and validation
- • Used centroid-based trajectory smoothing via fixed-size queue

7. Output Artifacts

- • Annotated output video with trajectory overlay
- • Per-frame CSV containing centroid coordinates and visibility flag

8. Example Output Description

The output video displays a red bounding box around the detected ball, a blue trajectory line connecting centroids across frames, and a pitch boundary overlay for debugging. The CSV file contains structured data for every frame, enabling quantitative evaluation.

9. Conclusion

This project demonstrates a robust and extensible approach to cricket ball tracking using modern object detection techniques. The combination of spatial constraints, fallback logic, and structured outputs ensures both practical usability and research extensibility.