Cricket Ball Trajectory Modelling Report

Kandi Jayanth Reddy

December 19, 2025

1 Introduction and Problem Statement

Detecting and tracking a cricket ball from broadcast-style video is challenging due to the ball's small size, high speed, motion blur, and frequent occlusions. This project aims to build a robust end-to-end computer vision pipeline that accurately detects the ball and reconstructs a physically consistent trajectory from a single fixed camera.

2 Modelling Decisions and Configuration

To reliably detect a small, fast-moving object in high-resolution footage, several key deviations from standard YOLO defaults were made.

2.1 Model Architecture

Decision: Switched from YOLOv8n (Nano) to YOLOv8s (Small).

Reasoning: While YOLOv8n is optimized for lightweight, real-time tasks, it lacked sufficient representational capacity to resolve a cricket ball occupying only a few pixels in large-field video frames. YOLOv8s provides a better balance between model capacity and inference speed, improving small-object recall without excessive computational overhead.

2.2 Input Resolution

Decision: Increased input resolution (imgsz) from the default 640px to 1280px.

Reasoning: The source videos had resolutions of approximately 2000×1000 pixels. Downscaling to 640px caused the ball to become too small for reliable detection. Increasing the input resolution preserved critical spatial details necessary for detecting the ball.

2.3 Training Hyperparameters

- **Optimizer:** AdamW was used instead of SGD, as it offers more stable convergence for smaller, task-specific datasets.

- **Regularization:** Dropout of 0.05 was applied to reduce overfitting, particularly due to the repetitive nature of video frames.
- **Learning Rate:** The initial learning rate was set to 0.001, with a final learning rate fraction (lrf) of 0.01.
- **Augmentation:** Default YOLO augmentations were enabled to improve robustness against lighting variations and motion blur.

3 Issues Faced and Fallback Logic

Raw frame-wise detections were insufficient to produce a reliable trajectory, necessitating a shift from naive point linking to global trajectory optimization.

3.1 Issue 1: Long-Range Interpolation Errors

Problem: Simply connecting detected centroids across frames resulted in erratic trajectories. When detections were missed for several consecutive frames, straight-line interpolation produced unrealistic jumps through empty space. See the below picture for reference.

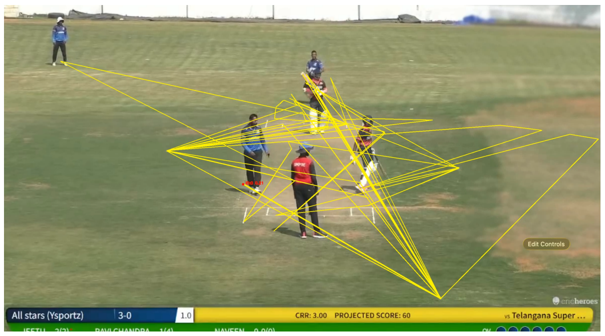

Figure 1: The high confidence points gives unrealistic trajectory

Fix: Instead of selecting all the highest-confidence detections, check the present point can be followed from previous point. But, the selection of first detectoin of ball is not fixed and could be random. So, I decided to select the valid set of trajectory ppoints from all high confidence points seems to solve the problem.

3.2 Issue 2: False Positives and Greedy Pathing

Problem: The detector occasionally misclassified objects such as shoes or helmets as the ball. A greedy nearest-neighbor linking strategy often latched onto these false positives, resulting in invalid short trajectories.

Fix (Dynamic Programming): The problem was reformulated as follows: given multiple candidate detections across time, select the subset that forms the most physically plausible trajectory.

Assumption: A cricket ball approximately follows projectile motion. Therefore, a valid trajectory should maximize detection confidence while minimizing abrupt, physically implausible changes in velocity or direction.

Implementation: A dynamic programming (DP) approach was used to globally optimize the trajectory selection. This ensured temporal consistency and physical plausibility, rather than locally optimal but globally incorrect greedy solutions.

4 Model Performance and Outputs

The final model configuration (`cricket_ball_v25/weights/best.pt`) was evaluated on a held-out test dataset.

4.1 Quantitative Metrics

- **mAP@0.5:** 0.659
- **mAP@0.5–0.95:** 0.187
- **Inference Speed:** Approximately 38.9 ms per image at 1280×1280 resolution, enabling near real-time processing.

4.2 Qualitative Results

The model successfully distinguishes the cricket ball in complex scenes. Inference logs indicate consistent detection of ball instances, including occasional false positives that are later filtered by the trajectory optimization module. Validation predictions with bounding box overlays were saved to `/content/runs/detect/val4`, enabling visual verification of detection quality.