Cricket Ball Centroid Detection and Trajectory Tracking in Broadcast Videos

1 Abstract

This project presents an end-to-end system for detecting the centroid of a cricket ball in broadcast match videos captured from a single static camera. The system produces per-frame centroid annotations and a processed output video with the ball trajectory overlayed.

Multiple modelling strategies were explored, including pretrained object detection, spatial filtering, hybrid detection-tracking methods, physics-based constraints, and domain-specific fine-tuning. The final system leverages a fine-tuned YOLOv8 model combined with spatial and temporal constraints to improve robustness under varying lighting, occlusion, and compression conditions.

2 Problem Statement

Given a cricket video from a static broadcast camera, the system must:

- Detect the cricket ball centroid in each visible frame.
- Output a per-frame annotation file containing:
  - Frame index
  - x-centroid
  - y-centroid
  - visibility flag
- Generate a processed video with trajectory overlay.
- Provide reproducible training and inference pipelines.

3 Challenges

The task presents several technical challenges:

- Very small object size in broadcast resolution.

- Motion blur during fast bowling.
- White ball under bright lighting.
- Confusion with scoreboard logos and circular pitch markings.
- Occlusion by batsman pads or bat.
- Domain shift between static product images and broadcast footage.

4 Baseline Approach: Pretrained YOLO

4.1 Direct Detection Using COCO Pretrained Model

Initially, YOLOv8 (pretrained on COCO) was used with the "sports ball" class to detect cricket balls directly.

Observations:

- Frequent false positives from scoreboard logos.
- Poor white ball detection.
- Inconsistent detection due to small object size.

Figure 1: Example baseline detection failure (replace with your image)

5 Spatial Filtering via Center ROI

To reduce false positives from static overlays and logos, detection was restricted to a central Region of Interest (ROI) corresponding to the pitch area.

5.1 Method

- Cropped center 50–60% of the frame.
- Ran YOLO only inside ROI.
- Mapped detections back to full-frame coordinates.

Impact:

- Reduced scoreboard-related false detections.
- Improved stability during delivery phase.
- Still failed under occlusion and white ball scenarios.

6 Hybrid Detection + Tracking

To improve temporal consistency, a hybrid system was introduced.

6.1 Tracking Module

- YOLO used for initial detection.
- CSRT tracker maintained object identity across frames.

6.2 Physics-Based Constraints

Additional validation mechanisms included:

- Maximum displacement threshold.
- Minimum velocity requirement.
- Static object rejection.
- Bounding box area consistency.

Outcome:

- Reduced trajectory jumps.
- Still unreliable when ball visually resembled logos.

7 Failure Analysis

Through experimentation, it was observed that:

> The COCO sports ball class does not adequately represent small cricket balls in broadcast footage.

Most training images for the COCO sports ball class contain:

- Large basketballs or footballs
- Clear foreground objects
- Minimal compression noise

Broadcast cricket footage contains:

- Tiny (10–25 pixel) balls
- Motion blur
- Complex background textures
- Compression artifacts

This domain mismatch significantly impacted detection robustness.

8 Fine-Tuning Strategy

8.1 Dataset Preparation

A cricket-specific dataset was obtained containing annotated red and white cricket balls. The dataset was structured as:

- Train split
- Validation split
- Test split

8.2 Training Configuration

- Model: YOLOv8l
- Input resolution: 1280
- Epochs: 50
- Batch size: 8
- Optimizer: Default YOLO configuration

Training was initialized from pretrained weights and fine-tuned on the cricket dataset.

9 Training Results

9.1 Confusion Matrix

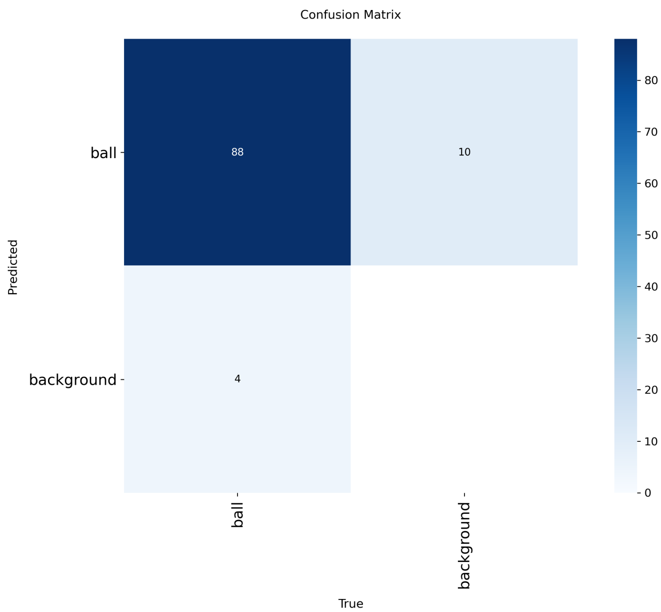

Figure 2: Confusion Matrix (replace with your image)

The confusion matrix indicates strong classification performance with high true positive rates and limited false negatives.

9.2 Precision-Recall Curve

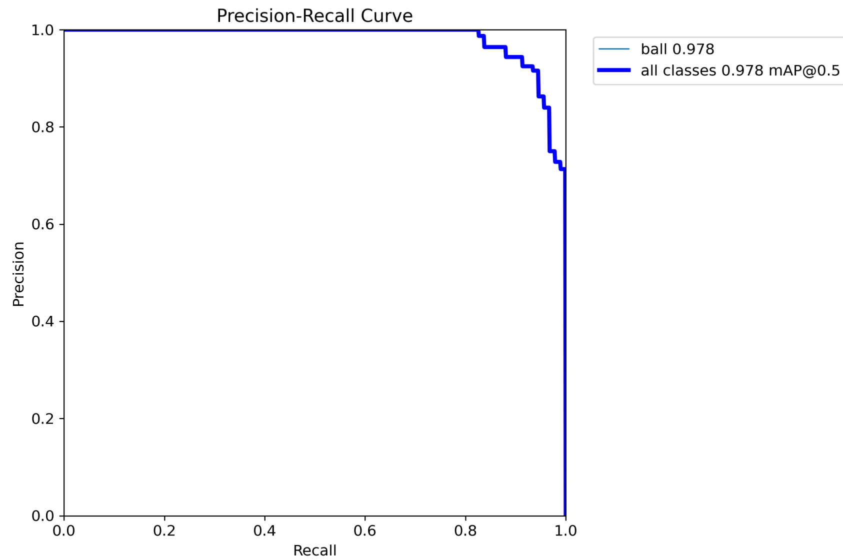

Figure 3: Precision-Recall Curve (replace with your image)

The model achieved:

- mAP@0.5 = 0.978
- High precision at moderate confidence thresholds
- F1 score peak around confidence ≈ 0.47

9.3 Confidence Curves

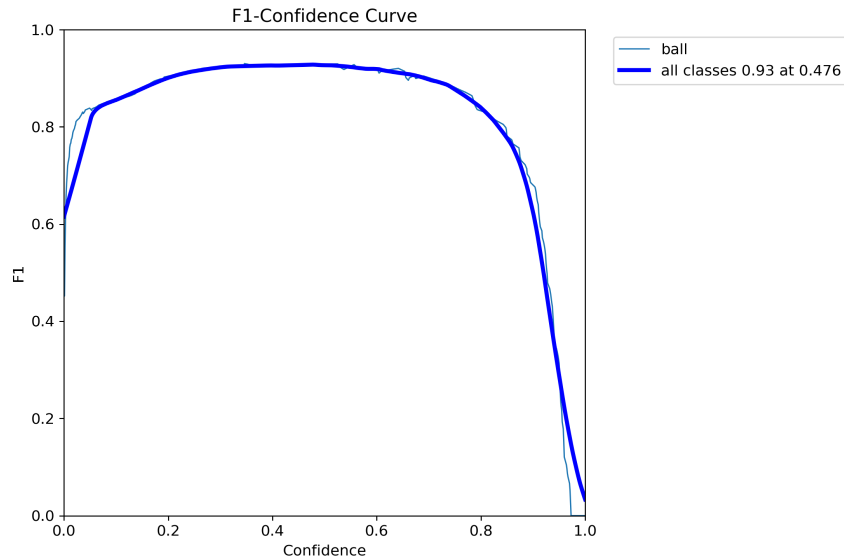

Figure 4: F1 vs Confidence Curve (replace with your image)

The optimal operating confidence threshold was selected around 0.35–0.45 based on F1 analysis.

10 Final Inference Pipeline

The final deployed system uses:

- Fine-tuned YOLOv8 model
- Center ROI cropping
- Bounding box area filtering
- Distance gating between consecutive frames
- Moving average smoothing

10.1 Final Architecture

```
Input Video
      ↓
Center ROI Extraction
      ↓
Fine-Tuned YOLO Detection
      ↓
```

```
Area Filtering
      ↓
Distance Gating
      ↓
Trajectory Smoothing
      ↓
Video + CSV Output
```

11 Results Comparison

Method	Red Ball	White Ball	False Positives
Pretrained YOLO	Moderate	Weak	High
ROI + Hybrid Tracking	Moderate	Moderate	Medium
Fine-Tuned YOLO	Strong	Strong	Low
Fine-Tuned + ROI (Final)	Strong	Strong	Very Low

Table 1: Comparative Performance Across Iterations

12 Discussion

The experiments demonstrate that:

- Heuristic filtering alone cannot overcome domain mismatch.
- Domain-specific fine-tuning significantly improves robustness.
- Spatial constraints (ROI) effectively reduce static false positives.
- Temporal gating improves trajectory stability.

13 Conclusion

This project followed an iterative engineering approach:

1. Direct pretrained detection.
2. Spatial ROI filtering.
3. Hybrid tracking with physics constraints.
4. Domain-specific fine-tuning.
5. Final clean inference pipeline.

Fine-tuning proved to be the most impactful improvement, enabling reliable detection of both red and white cricket balls in broadcast footage.

The final system is reproducible, modular, and extensible for sports analytics applications.

14 Future Work

Future improvements could include:

- Larger broadcast-specific dataset.
- Multi-scale training for tiny object enhancement.
- Optical flow integration.
- Transformer-based tracking models.