Cricket Ball Tracking System

Samrajnee Ghosh, 2023CSY7547

1 Introduction

This report describes the complete cricket ball tracking system developed using a YOLOv8 object detector combined with a Kalman Filter + Hungarian Algorithm multi-object tracking (MOT) module. The system processes cricket match videos recorded from a single, fixed camera and outputs per-frame ball detections, tracking IDs, trajectory overlays, and processed evaluation files. The goal was to reliably detect and track a fast-moving cricket ball under varying lighting conditions, occlusions, blur, and camera distance.

2 System Overview

The system consists of five core components:

1. **Bounding Box to train YOLOv8** — A custom yolo_label.py file that produces bounding boxes on frames of videos which could be used for supervised fine-tuning of the YOLOv8 model
2. **Ball Detection using SFT YOLOv8** — A custom-trained YOLOv8 model identifies the cricket ball in each frame with bounding boxes.
3. **State Estimation using a Kalman Filter** — A constant-velocity motion model estimates the ball's position during partial or full occlusions.
4. **Data Association via Hungarian Assignment** — The tracker assigns new YOLO detections to existing Kalman predictions, ensuring temporal consistency.
5. **Label Assistant** — Makes JSON file to report frame-wise centroid and visibility of the ball.

The output includes per-frame ball location, a visibility flag, and annotated video with trajectory overlays.

3 Modeling Decisions

3.1 YOLOv8 Detector

YOLOv8 was selected due to:

- High speed suitable for real-time inference.
- Strong performance on small objects through anchor-free design.
- Ease of custom training with limited cricket-specific datasets.

Key design details:

- **Model size:** YOLOv8s (small) offered the best tradeoff between speed and accuracy.

- **Augmentations:** Cropping was critical for realism and to avoid noise.
- **Label format:** YOLO format (frame, x_{center}, y_{center}, visibility). Only one class: “ball.”

3.2 Kalman Filter Tracking

A Kalman filter with a constant-velocity motion model was implemented. The state vector: [x = [u, v, s, r, , ,]$\hat{\text{T}}$] where (u, v) are box center coordinates, s is scale, and r is aspect ratio.

Benefits:

- Smooths noisy YOLO detections.
- Predicts position when the ball is blurred or momentarily invisible.
- Reduces flicker in trajectory visualization.

3.3 Hungarian Assignment

The Hungarian algorithm (also called Munkres assignment) ensures robust alignment between YOLO bounding boxes and Kalman predictions.

We use IoU-based cost: [cost = 1 - IoU(prediction, detection).]

Only matches with IoU > 0.45 are accepted to allow for rapid ball motion between frames.

4 Assumptions

The system assumes:

- Fixed, non-moving camera.
- Only one cricket ball is present.
- Ball remains within the frame for the majority of sequence.
- Training and inference videos share similar camera angles and field conditions.
- Ball has only one colour in a video - either white or red

5 Example Output

Below are example per-frame outputs stored during inference:

Figure 1: A descriptive caption for your image.

5.1 Frame-level CSV Format

```
frame_index, x, y, visible
86, 912, 435, 1
87, 935, 447, 1
88, 958, 462, 1
89, -1, -1, 0
```

5.2 Annotated Trajectory Video

Frames contain:

- Bounding box for each detection.
- Kalman-smoothed position.

6 Conclusion

The cricket ball tracking system integrating YOLOv8 detection with Kalman filtering and Hungarian matching provides robust performance under challenging conditions such as motion blur, occlusion, and lighting variation. The iterative improvements significantly enhanced stability and reliability, making the system suitable for analytics, coaching tools, and automated highlight generation.