PROJECT REPORT

Frame Drop / Merge Detection

Submitted in partial fulfillment of
Pragyan NextGen Hackathon 2026
Problem Statement 2

Submitted By

Team Ctrl+Shift

- Shuchi Anush S
- Hareeshwar N K
- Kavin V K
- Harish Y

Institution

Coimbatore Institute of Technology

Submission Date

22/02/2026

Technical Stack

Python · FastAPI · React (Vite) · YOLOv8 · OpenCV · NumPy

Pragyan NextGen Hackathon 2026

1. Introduction

This project addresses temporal inconsistencies in professional cricket broadcast footage, specifically the detection of:

- Frame Drops
- Frame Merges

Rather than relying on unreliable timestamp metadata, the system performs ball-centric motion modeling and visual anomaly detection to identify corrupted frames.

In high-stakes environments such as DRS review and trajectory analytics, even a single corrupted frame can significantly impact interpretation.

---

2. Problem Definition

Frame Drop

A frame drop occurs when one or more frames are missing from the video stream. This results in unnatural discontinuity in object motion.

Frame Merge

A frame merge occurs when two consecutive frames are blended, typically producing blur and unusually high visual similarity between adjacent frames.

Timestamp-based detection is unreliable in broadcast pipelines due to re-encoding and metadata interpolation. Therefore, a motion-grounded detection approach is required.

---

3. System Overview

The system follows a ball-centric anomaly detection pipeline:

Video Input

→ YOLOv8 Ball Detection

→ ROI-Constrained Tracking

→ Motion Prediction (Kalman Model)

→ Drop & Merge Classification

→ Annotated Output + Structured Reports

Each frame is automatically classified as:

- Normal
- Drop
- Merge

---

4. Methodology

4.1 Ball Detection

A fine-tuned YOLOv8 model detects the cricket ball in each frame.

Detection confidence filtering is applied to remove low-probability detections.

4.2 ROI-Constrained Tracking

To reduce false positives from background clutter, detection is restricted to a Region of Interest centered around the predicted ball position.

This ensures spatial consistency and improves detection stability.

4.3 Motion Prediction

A Kalman-based motion model predicts the ball's next position using velocity and positional history.

The predicted position is compared against the detected position to compute motion error.

4.4 Frame Drop Detection

If motion error exceeds an adaptive gating threshold, the frame is classified as a Drop.

Additional drop indicators include:

- Detection absence
- Temporal gaps between valid detections

4.5 Frame Merge Detection

Merge frames are detected using appearance-based metrics:

- Structural Similarity Index (SSIM)
- Laplacian variance (blur detection)
- Low detection confidence filtering

High structural similarity combined with reduced sharpness indicates potential frame blending.

---

5. System Architecture

The system is divided into:

Backend

- FastAPI server
- YOLOv8 inference engine
- Kalman tracking logic
- Drop/Merge detection logic
- Report generation (JSON, CSV)

Frontend

- React (Vite)
- Interactive dashboard
- Video preview
- Timeline visualization
- Motion error graph
- PDF report export

The backend processes video and returns structured analytics consumed by the frontend.

---

6. Technologies Used

- Python
- FastAPI
- React (Vite)
- Ultralytics YOLOv8
- OpenCV

- NumPy
- Recharts
- jsPDF

GPU acceleration is supported when available.

---

7. Output

The system generates:

- Annotated MP4 video (trajectory + anomaly markers)
- JSON frame-level report
- CSV summary report
- Interactive analytical dashboard
- Downloadable PDF report

---

8. Challenges Faced

Background False Positives

Cricket broadcasts contain cluttered backgrounds. ROI-based motion gating reduced false detections.

Motion Blur

High ball velocity causes blur, affecting detection confidence. Merge detection compensates using blur metrics.

Occlusion

Temporary ball occlusions required predictive continuity via motion modeling.

Legitimate Direction Changes

Ball bounce events can appear as discontinuities. Threshold calibration was required to differentiate real motion from corruption.

---

9. Limitations

- Extended occlusion scenarios
- Extreme motion blur
- Multi-ball ambiguity in rare frames

---

10. Future Improvements

- Optical-flow-based merge detection
- Multi-camera fusion
- Real-time broadcast integration
- Edge-optimized deployment

---

11. Conclusion

This project demonstrates that ball-centric motion modeling provides a robust and metadata-independent solution for detecting temporal corruption in sports video streams.

By integrating detection, tracking, motion prediction, and appearance-based analysis into a unified pipeline, the system provides both visual and quantitative validation of frame integrity.
