

Cricket Vision AI: Real-Time Cricket Detection and Analysis System

An advanced, AI-powered system that performs real-time detection, analysis, and counterfactual prediction for cricket matches. Built using deep learning and computer vision, it tracks ball movement, player positions, bat swing angles, and predicts alternative outcomes to enhance cricket understanding and analytics.

Table of Contents

- Overview
- Key Features
- Tech Stack
- Project Structure
- Installation
- Usage
- Sample Results
- Credits
- License

Overview

This system is designed to analyze cricket match footage by detecting various objects and actions in real-time or from a recorded video. It integrates multiple modules into a unified interface capable of:

- Tracking ball trajectory and speed
- Recognizing batting shots
- Analyzing player positions
- Generating visual counterfactuals
- Estimating confidence scores

The system works on both pre-recorded video files and live screen content (e.g., streaming match playback).

Key Features

This system includes **7 powerful AI modules**:

1. Object Detection

Detects and tracks cricket objects: ball, bat, stumps, batsman, bowler, fielders, and umpire using YOLOv8.

2. **Ball Tracking & Speed Analysis**

Visualizes real-time ball trajectory and estimates its speed between frames.

3. **Counterfactual Ball Prediction**

Simulates hypothetical scenarios: what if the ball was slower/faster?

Color-coded trajectory lines indicate variations like *slower*, *slow*, *normal*, *fast*, *faster*.

4. **Bat Swing and Angle Analysis**

Tracks bat motion to visualize the swing path and estimate swing angle.

5. **Shot Execution Detection**

Classifies the shot style (e.g., **Drive**, **Defensive**, **Aggressive**, **Leaving**) using motion and position data.

6. **Player Positioning & Stance Detection**

Identifies whether the batsman is **left-handed** or **right-handed** and tracks key player positions.

7. **Shot Style Vision and Comparison**

Displays side-by-side actual vs hypothetical outcomes based on variations in shot or delivery.

Tech Stack

- **Language:** Python 3.9+
- **Deep Learning Framework:** PyTorch
- **Model:** YOLOv8 (Ultralytics)
- **Vision & Tracking:** OpenCV, NumPy
- **Data:** Custom YOLO-formatted annotations (.csv)
- **Other:** ONNX, Matplotlib (for visualization), Tkinter (for file dialog)

Project Structure

Cricket-Analysis/

```
|— Cricket/YOLO/central.py          # Main launcher script
|— modules/
|   |— Cricket/YOLO/t1.py  # Object detection module
|   |— Cricket/YOLO/t22.py  # Ball speed and trajectory
|   |— Cricket/YOLO/t3.py  # Ball variation predictions
|   |— Cricket/YOLO/b2.py  # Bat motion & angle
|   |— Cricket/YOLO_2/b1.py  # Shot recognition
|   |— Cricket/Yolo_3/1a.py  # Player position. (F = front, R = Rear view)
|   |— Cricket/Yolo_3/5.py  # Actual vs hypothetical output
|— Cricket/asset  # Saved output videos (module output m1,m2, ALL.mp4 is
|                                     joint file ).
|— models.  # YOLOv8 and ONNX models
|   |— Cricket/YOLO/runs/detect/train/weights/best.pt  # object detection
|   |— Cricket/YOLO_2/runs/detect/train/weights/best.pt  # player shot style
|   |— Cricket/YOLO_3/runs/detect/train/weights/best.pt  # left/right hand
|— README.md
```

Usage

Run the main script to start the system:

```
python central.py
```

You will be prompted to choose:

- **Load a Video File:** Select .mp4 or .avi from your system
- **Live Detection on Screen:** Capture and analyze from your screen (like full-screen stream) currently not working

Author

Vraj Alpeshkumar Modi

M.Tech (Computer Science & Engineering), NIT Goa

Email: vrajalpeshkumarmodi@gmail.com

