Report: Cricket Ball Detection and Tracking Pipeline

Pranav Agarwal, IIIT Hyderabad

Roll no .: 2021113018

Data Collection and Preparation

Training data was collected from multiple publicly available sources using Roboflow. The dataset provided by EdgeFleet was intentionally excluded from training, as it was visually similar to the test videos and could have led to data leakage. The collected dataset primarily consisted of low-resolution images, with most images having a resolution of approximately 640 pixels on the longer side.

In total, around 1,000 image–label pairs were used for training, and approximately 350 image–label pairs were reserved for validation. All training experiments and metrics were logged and tracked using Weights & Biases (W&B) to ensure reproducibility and systematic comparison across configurations.

Model Training

Followin configurations of trained YOLOv11 models were finetuned, - YOLOv11s and YOLOv11m. Experiments were conducted with different input image sizes (830, 960, and 1080) to evaluate the trade-off between spatial resolution and detection performance. Each configuration was trained for a maximum of 100 epochs, with early stopping implicitly governed by convergence behavior observed in validation metrics.

WandB dashboard (https://wandb.ai/pranav_ag/cricket-ball-detection/workspace?nw=nwuserpranav_ag) showing training logs and experiments.

Inference and Tracking Pipeline

The inference pipeline is implemented using a trained YOLOv11 model and operates on video input frame by frame. For each frame, the model performs object detection with a confidence threshold of 0.25. When multiple detections are present, the detection with the highest confidence score is selected, and its bounding box centroid is computed to represent the ball location.

To improve temporal consistency and robustness, additional logic is applied beyond raw detection:

- **Outlier Rejection:** A new detection is validated by comparing its distance from the previous accepted detection. If the spatial jump exceeds a predefined threshold, the detection is rejected as an outlier.

- **Interpolation:** If detections are missing for a short sequence of frames, linear interpolation is applied based on the estimated velocity from previous detections, provided the gap is within an acceptable range.
- **Trajectory Management:** Accepted detections are stored in a fixed-length buffer, which is used to render the ball trajectory over time.

The pipeline outputs two artifacts:

1. A `CSV` file containing per-frame annotations (frame index, centroid coordinates, and visibility flag).
2. A processed `.mp4` video with the detected ball centroid and its trajectory overlaid.

Observations and Results

Overall, the trained models achieved reasonable detection performance, with an average precision of approximately 0.6 for ball detection. The pipeline performed well when detecting white balls but showed noticeably weaker performance on red balls. This imbalance is primarily attributed to the dataset composition, which contained a significantly larger proportion of white ball samples compared to red ones, leading to biased learning during training.