Project Report: Tennis Ball Speed Estimation using Computer Vision

Author: Deva Nandan JS

Date: September 8, 2025

1. Introduction & Problem Statement

The objective is to create a system that detects the ball frame-by-frame, tracks its trajectory, and calculates its speed in km/h. This technology is valuable for coaching, broadcasting, and fan engagement. The resulting pipeline is a functional proof-of-concept that demonstrates the viability of this approach and provides a foundation for future improvements.

2. Tools and Technologies Used

The project utilized a stack of open-source tools within a hybrid development workflow to maximize efficiency.

- **Language & Libraries:** Python 3.13.7 with OpenCV for video processing, NumPy for numerical operations, Pandas for data handling (CSV export), and Matplotlib for plotting.
- **Object Detection:** A custom-trained Ultralytics YOLOv8s model was used for its balance of speed and accuracy.
- **Object Tracking:** SORT was chosen for its lightweight and efficient tracking performance.
    - **Development Environment:** A hybrid approach was used:
        - **Google Colab:** For GPU-accelerated training of the YOLOv8s model.
        - **VS Code:** For local development, debugging, and execution of the main pipeline.

3. Approach and Methodology

The solution is a multi-stage pipeline that processes video to produce speed analytics.

3.1. Object Detection

Accurate ball detection is critical. A custom-trained YOLOv8s model was used because generic models are less effective for small, fast-moving objects with motion blur. Training on a specific dataset provides higher accuracy. To improve detection, each frame was preprocessed:

1. **Upscaling:** The frame's resolution was doubled to increase the ball's pixel count for the model.
2. **CLAHE:** This contrast enhancement technique makes the ball more distinct from the background.

3.2. Object Tracking

The SORT algorithm was chosen to build the ball's trajectory. Its speed and simplicity are ideal for single-object tracking, avoiding the unnecessary overhead of more complex trackers like DeepSORT. The tracker's max_age was tuned to handle brief detection failures and maintain a continuous path.

3.3. Trajectory Refinement and Speed Calculation

Raw tracking data requires filtering to ensure accurate speed calculation.

- **Data Smoothing:** A polynomial interpolation filled gaps from missed detections, and a Savitzky-Golay filter smoothed the trajectory to reduce jitter.
- **Scaling and Speed Estimation:** A static pixel-to-meter ratio was established by assuming the court width (10.97m) corresponds to the video frame's width. Speed was then calculated using the distance formula on smoothed coordinates and the video's FPS.

4. Results and Outputs

The pipeline was executed on a sample tennis clip, successfully processing it and yielding the following quantitative results:

- **Frames with Ball Detected:** 198
- **Average Ball Speed:** 18.24 km/h
- **Peak Ball Speed:** 190.84 km/h

The system produced all required deliverables: an annotated video, a CSV file with coordinate data, and a plot visualizing the trajectory and speed profile.

5. Limitations and Future Improvements

This proof-of-concept has clear areas for improvement.

- **Current Limitations:**
    - **Model Generalization:** The model is biased to its training data and may perform poorly on different court types or lighting.
    - **Detection Challenges:** Accuracy suffers from extreme motion blur and when the ball is small in the frame.

- ◦ **Inaccurate Scaling:** The static pixel-to-meter conversion does not account for perspective distortion, which is a major source of error.

- **Future Improvements:**
  - ◦ **Better Detection Model:** Improve the detector by expanding the training dataset with more varied footage and by using larger models like YOLOv8m or YOLOv8l.
  - ◦ **Advanced Camera Calibration:** Implement a homography transformation by mapping court lines to their real-world dimensions. This would correct for perspective and yield more accurate distance calculations.
  - ◦ **3D Trajectory Reconstruction:** For the highest accuracy, use stereo cameras or depth estimation to reconstruct a 3D path, enabling analysis of metrics like spin.