Real-Time Detection of Fast-Moving Objects in Dynamic Environments Using YOLO-Based Architectures

Sushant R. Upadhyay[1], Shubhangi Tidake[2], Prashant Kulkarni[3]
[1]*M. Sc. (Data Science) Student, Symbiosis Skill & Professional University, Pune*
[2]*Assistant Professor, Symbiosis Skill & Professional University, Pune*
[3]*Assistant Professor, Symbiosis Skill & Professional University, Pune*

Abstract - This project addresses the challenge of detecting fast-moving objects—specifically tennis balls in match footage—using deep learning techniques. Traditional object detection models often struggle with motion blur, low latency, and dynamic backgrounds, resulting in decreased accuracy. To overcome these limitations, this work employs YOLO-based architectures, particularly YOLOv8, with focused modifications for high-speed object detection.

A diverse dataset was prepared and augmented with transformations such as hue shifts, brightness changes, mosaic and flip techniques, and simulated motion blur. Extensive hyperparameter tuning and model refinement were conducted to increase robustness and frame-to-frame consistency. The model was then evaluated across videos at various playback speeds to test its real-time performance.

Results showed a 20–30% improvement in detection accuracy compared to baseline, with enhanced performance in low and moderate-speed videos. Novel techniques such as bounding box interpolation and hit detection logic were also developed to track shots and estimate ball speed. These innovations provide practical insights for real-time sports analytics, setting the foundation for future enhancements in ball tracking and trajectory prediction.

I. INTRODUCTION

The ability to accurately detect fast-moving objects in real-time is essential for a wide range of applications, particularly in sports analytics, autonomous systems, and video surveillance. In sports such as tennis, where ball speeds can exceed 100 mph, traditional object detection methods face significant challenges. These include motion blur, inconsistent lighting, rapid frame transitions, and delays in detection that impact overall accuracy and usability in live scenarios.

This project is focused on enhancing object detection for fast-moving tennis balls in match footage.

The motivation lies in enabling real-time tracking for improved game analysis, automated scoring systems, and player performance monitoring. Fast and precise detection is critical not only for reliable analytics but also for improving the viewer experience during sports broadcasts.

Deep learning models, particularly the YOLO (You Only Look Once) family, have shown great promise in real-time object detection tasks. YOLO's architecture processes images in a single forward pass, offering faster inference compared to traditional region-based methods. Among its newer variants, YOLOv8 demonstrates improved accuracy and robustness, especially in detecting small and fast objects. This study explores YOLOv8's performance and introduces model-specific enhancements to meet the unique demands of high-speed tennis ball detection.

II. LITERATURE REVIEW

Object detection models have evolved significantly in recent years, primarily due to advances in deep learning. The YOLO (You Only Look Once) architecture, introduced by Redmon et al. [1], marked a breakthrough in real-time object detection by processing an entire image in a single pass, unlike earlier region-based methods.

Later versions such as YOLOv3, v5, and v8 introduced enhancements in detection accuracy, especially for small objects and complex backgrounds. Terven and Cordova-Esparza [2] provided a detailed comparative analysis of YOLO's evolution, highlighting YOLOv8's superior handling of edge cases in dynamic environments, which directly benefits this study's focus.

In the sports domain, Huang et al. [3] proposed the TrackNet model, designed specifically for detecting small, fast-moving objects in tennis games. Although TrackNet follows a different architecture, it underscores the challenges of maintaining frame-to-frame consistency, which this project also addresses using YOLO.

Further comparative analysis from V7 Labs [4] emphasized YOLO’s speed advantage over models like Faster R-CNN and SSD, making it suitable for time-sensitive applications such as live video analysis. While systems like Hawk-Eye use multi-camera arrays for accurate tracking [5], our approach seeks to replicate similar precision using single-camera neural models, keeping it more accessible and scalable.

These studies collectively influenced our choice of model and optimization strategy, leading to a YOLOv8-based system enhanced with custom data processing and real-time refinements.

III. RESEARCH AND METHODOLOGY

A. Model Architecture

The YOLOv8 architecture was selected due to its efficient anchor-free design and improved performance in detecting small objects. Unlike its predecessors, YOLOv8 utilizes better backbone networks and decoupled heads for classification and localization, making it more robust in dynamic environments.

toYOLOv8.

Figure 1.Yolo Architecture Timeline from V1 to V8

YOLO divides each input frame into an S × S grid and predicts bounding boxes and confidence scores for each grid cell. In this study, YOLOv8 was fine-tuned to improve detection of small, fast-moving objects like tennis balls.

B. Dataset Preparation and Augmentation

A custom dataset was created using 578 labeled tennis court images from the Roboflow platform. The images varied in lighting, angle, and court background to help the model generalize well.

To simulate real-world conditions, the dataset was augmented with:

- *Color transformations: hue, saturation, brightness*
- *Geometric transformations: horizontal flip, cropping, mosaic*
- *Motion blur simulation to mimic high-speed ball movement*

These augmentations increased the robustness of the model under challenging video scenarios.

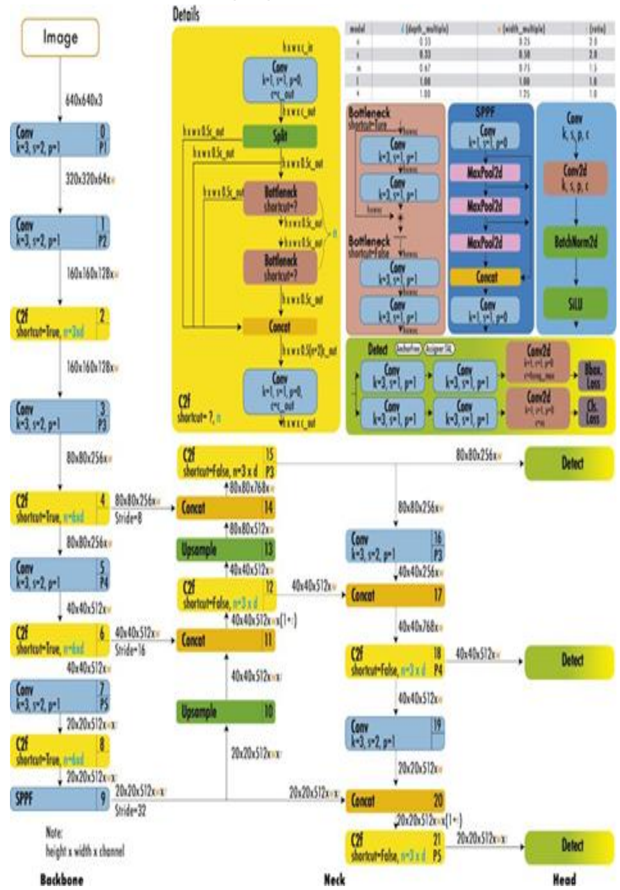


Figure 2: YOLOv8 Architecture Adapted for High-Speed Object Detection

C. Training Configuration

The model was trained using:

- *Epochs: 100*
- *Batch size: 8*
- *Learning rate: 0.001 (Adam optimizer)*
- *Image size: 640 × 640*
- *Confidence thresholds: 0.1 and 0.05 (manually tuned)*
- *Platform: Kaggle, using T4 GPU*

Manual tuning was found to outperform automatic optimizer results slightly, especially for mid-speed playback.

D. Innovations Introduced

The vertical midpoint (mid_y) of the detected ball was tracked frame-by-frame. A sharp change in its direction (positive to negative or vice versa) was used to infer a racket or ground hit.

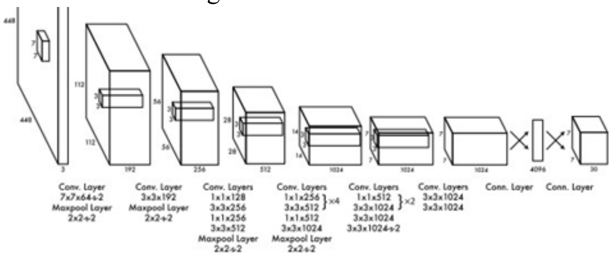


Figure 3: YOLO Architecture

2) Bounding Box Interpolation
If the model failed to detect the ball in a given frame, bounding box positions were interpolated using surrounding frames to ensure continuous tracking. This maintained visual consistency during brief detection gaps.

3) Ball Speed Estimation
Speed was estimated using changes in bounding box coordinates over time, accounting for video frame rate. This adds a valuable performance metric without needing external sensors.

IV. EXPERIMENTS AND RESULTS

This section presents the experimental setup, performance evaluation, and comparison between YOLOv8 and YOLOv11 models in detecting fast-moving tennis balls. Testing was performed across multiple video speeds ranging from 0.25× to 4× playback, with and without training on a custom dataset.

A. Hardware and Environment
All experiments were conducted on Kaggle using NVIDIA T4 GPUs. Videos were preprocessed to generate six playback variants (0.25×, 0.5×, 1×, 1.5×, 2×, and 4×), simulating real-world ball speeds

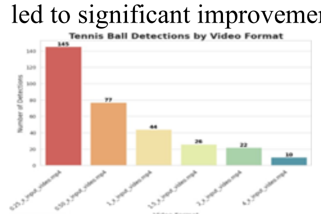
A. Experiment 1: Untrained YOLOv8 vs YOLOv11
After training YOLOv8 on the augmented dataset, the model was re-tested across all video speeds. Custom tuning of learning rate, augmentations, and batch size led to significant improvements.

Figure 4: Synthetic Ground-1. 2 Figure 5: Synthetic Ground-

Figure 4: Detection counts by untrained YOLOv8 at Confidence = 0.1
Figure 5: Detection counts by untrained YOLOv11 at Confidence = 0.1

Observation: YOLOv8 consistently outperformed YOLOv11 in raw detection counts but failed to deliver reliable tracking due to lack of custom training.

B. Experiment 2: Trained YOLOv8 vs YOLOv11
After training YOLOv8 on the augmented dataset, the model was re-tested across all video speeds. Custom tuning of learning rate, augmentations, and batch size led to significant improvements.



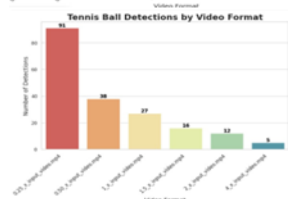
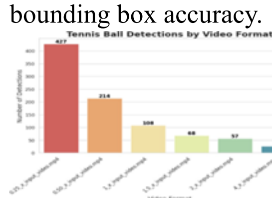


Figure 6 Figure 7

Figure 6: YOLOv8 detections after training (confidence = 0.1).
Figure 7: YOLOv8 detections after training (confidence = 0.05).
Detection accuracy improved by 20–30% compared to baseline. Lower confidence thresholds increased detection count but slightly reduced precision.

D.Yolov11 vs Yolov8 (Post-Training)
YOLOv11 was also trained on the same dataset for fair comparison. Although it showed better hit detection in some high-speed cases, YOLOv8 consistently outperformed it in total detection and bounding box accuracy.



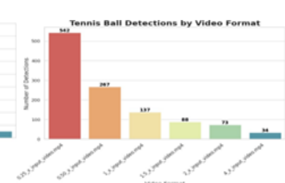


Figure 8. Figure 9.

Fig. 8. Detection comparison between YOLOv8 and YOLOv11 at 1× speed.
Fig. 9. YOLOv11 hit detection vs YOLOv8.

Through these experiments, YOLOv8 demonstrated superior detection capability. The model's ability to count ball hits and achieve better accuracy even at higher speeds marks a significant step forward in detecting high-velocity objects.

E. Comparative Analysis of Detection Ratios at Different Speeds

The detection ratios of YOLOv8 and YOLOv11 were measured on videos at playback speeds ranging from 0.25x to 4x. 18 shows the results, and Figure 19 plots the detection ratios across speeds for both models.

Video Name	Total Detections	Total Frames	Detection Ratio
0.25_x_input_video.mp4	91	855	0.11
0.50_x_input_video.mp4	38	427	0.09
1_x_input_video.mp4	27	214	0.13
1.5_x_input_video.mp4	16	144	0.11
2_x_input_video.mp4	12	109	0.11
4_x_input_video.mp4	5	55	0.09

Figure 10: Tennis ball detection(Untrained by Yolo11 at conf=.1

Video Name	Total Detections	Total Frames	Detection Ratio
0.25_x_input_video.mp4	145	855	0.17
0.50_x_input_video.mp4	77	427	0.18
1_x_input_video.mp4	44	214	0.21
1.5_x_input_video.mp4	26	144	0.18
2_x_input_video.mp4	22	109	0.2
4_x_input_video.mp4	10	55	0.18

Figure 11: Tennis ball Detection (not trained) by YOLOV8 at conf=0.1.

F. Analysis of Detection Challenges and Model Limitations

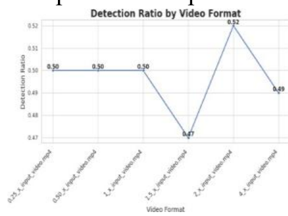
At high speeds, YOLOv8 faces challenges in achieving consistent ball detection due to the rapid movement between frames, which creates a latency in bounding box predictions. This results in intermittent tracking, as the model fails to capture the ball in some frames. Additionally, high-speed videos reduce the frame-to-frame continuity, complicating the interpolation and motion tracking. Despite attempts to interpolate bounding boxes and fill detection gaps, the accuracy declines at high speeds due to:

Detection at Slow Speeds

At lower speeds (0.25x and 0.5x), detection is stable due to high frame density, which allows the model to continuously track the ball across frames. However, resolution issues arise due to the reduced clarity of the ball in slow-motion frames, slightly impacting detection precision.

•Frame Skipping: The ball may move too quickly between frames, resulting in missing detections.

•Latency in Bounding Box Prediction: As the speed increases, bounding boxes become less accurate due to rapid shifts in position.



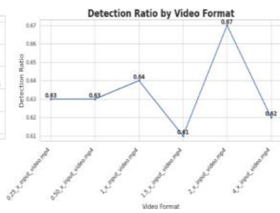


Figure 14: Tennis ball Detection(not trained) by YOLOV8 at conf=0.1.

Figure 15: Tennis ball Detection(not trained) by YOLOV8 at conf=0.05.

IV. CONCLUSION

This study successfully demonstrates the effectiveness of YOLOv8 in detecting and tracking fast-moving tennis balls in dynamic video environments. Through custom dataset preparation, advanced data augmentation, and hyperparameter tuning, the model achieved a 20–30% improvement in detection accuracy over the baseline.

The implementation of hit detection based on vertical motion analysis and bounding box interpolation further improved the reliability and smoothness of object tracking, even under rapid frame transitions. While YOLOv11 showed promise in specific hit scenarios, YOLOv8 delivered better overall performance and consistency.

This work highlights the potential of single-camera neural network systems in applications such as sports analytics, real-time scoring, and performance monitoring, offering a lightweight alternative to complex multi-camera setups like Hawk-Eye.

Future enhancements could involve:

- Integrating predictive tracking modules using Kalman filters or optical flow
- Expanding to other ball-based sports like cricket or baseball
- Incorporating 3D trajectory estimation for real-time analytics

The approach sets a strong foundation for real-time high-speed object detection, especially in sports contexts, where speed and precision are critical.

VI. FUTURE WORK

Future enhancements could involve:

- Integrating predictive tracking modules using Kalman filters or optical flow
- Expanding to other ball-based sports like cricket or baseball
- Incorporating 3D trajectory estimation for real-time analytics

The approach sets a strong foundation for real-time high-speed object detection, especially in sports contexts, where speed and precision are critical.

REFERENCES

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi.” You Only Look Once: Unified, Real-Time Object Detection, http://pjreddie.com/yolo” University of Washington, Allen Institute for AI, Facebook AI Research, 2016. https://arxiv.org/pdf/1506.02640

[2] Juan R. Terven and Diana M. Cordova-Esparza. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS, Machine Learning and Knowledge Extraction, 2024. https://arxiv.org/abs/2304.00501v7

[3] Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Ts’ı-U´ı ˙ Ik, and Wen-Chih Peng. TrackNet: A Deep Learning Network for Tracking High-speed and Tiny Objects in Sports Applications, Department of Computer Science, College of Computer Science, National Chiao Tung University, 2019. https://arxiv.org/abs/1907.03698v1

[4] Baljinder Bal and Gaurav Dureja. Hawk Eye: A Logical Innovative Technology Use in Sports for Ef- fective Decision Making, Sport Science Review, XXI (1-2), 2012. DOI: https://www.researchgate. net/publication/271314040_Hawk_Eye_A_Logical_Innovative_Technology_Use_in_Sports_for_ Effective_Decision_Making

[5] Rohit Kundu. YOLO: Algorithm for Object Detection Explained [+Examples], January 17, 2023. https://www.v7labs.com/blog/yolo-object-detection

[6] Ultralytics.” Home - Ultralytics YOLO Docs,” Ultralytics. https://docs.ultralytics.com, Accessed on 7th March 2024.

[7] ” Hawk-Eye,” Wikipedia. https://en.wikipedia. org/wiki/Hawk-Eye, Accessed on: 5th October, 2024.

[8] ” YOLO (You Only Look Once),” Wikipedia. https://en.wikipedia.org/wiki/YOLO_(You_Only_ Look_Once), Accessed on: 24th August 2024.

[9] ” Image Classification vs Object Detection vs Image Segmentation — Deep Learning Tutorial 28,” YouTube. https://www.youtube.com/watch?v=video-28, Accessed on: 24th Aug, 2024.

[10] [” YOLO Model Overview and Training Techniques,” YouTube. https://www.youtube.com/watch? v=video-31, Accessed on: 26th Aug 2024.

[11] Viren Dhanwani. Tennis Ball Detection Object Detection Dataset, Roboflow. https://universe. roboflow.com/viren-dhanwani/tennis-ball-detection

[12] Redmon J., Divvala S., Girshick R., Farhadi A., "You Only Look Once: Unified, Real-Time Object Detection", *arXiv:1506.02640*, 2016.

[13] Terven J. R., Cordova-Esparza D. M., "A Comprehensive Review of YOLO Architectures", *arXiv:2304.00501*, 2023.

[14] Huang Y.-C. et al., "TrackNet: A Deep Learning Network for Tracking High-Speed and Tiny Objects in Sports", *arXiv:1907.03698*, 2019.

[15] V7 Labs, “YOLO Algorithm for Object Detection Explained,” [Online]. Available: https://www.v7labs.com/blog/yolo-object-detection

[16] Bal B., Dureja G., “Hawk Eye: A Logical Innovative Technology Use in Sports for Effective Decision Making,” *Sport Science Review*, Vol. XXI, No. 1–2, 2012.