Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260502037
Volume 5, Issue2 (May-August 2026), PP: 336-346.
www.indjcst.com
ISSN No: 2583-5300

AI-Based Automated Race Winner Detection System

Dr. K. Rajakumari[1], Poovizhi S[2], Samson J[3], Mukesh M[4], Dinesh karthik K[5]
[1, 2,3,4,5] *Computer Science and Engineering, United Institute Of Technology, Coimbatore, Tamil Nadu, India.*

To Cite this Article: *Dr. K. Rajakumari[1], Poovizhi S[2], Samson J[3], Mukesh M[4], Dinesh karthik K[5], "AI-Based Automated Race Winner Detection System", Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 336-346.*

Abstract: *Accurate race timing and participant identification are critical in competitive sports events. Traditional manual methods often suffer from human errors, delayed result processing, and lack of precision. This paper presents an AI-based automated race winner detection system that identifies the race start using audio signal processing and determines race completion using computer vision techniques. The proposed system utilizes pose detection to detect finish line crossing and Optical Character Recognition (OCR) to extract bib numbers of participants. The system operates in real time at 30 frames per second using mobile camera technology and processes data with minimal latency. Audio detection monitors environmental sound levels to identify the race start signal through decibel threshold analysis. Media Pipe pose estimation identifies key body landmarks such as shoulders and chest to detect when runners cross the finish line boundary. Google ML Kit OCR extracts participant bib numbers from video frames using pattern matching and regex filtering. Race results including bib numbers, timestamps, rank position, and captured images are stored in a local SQLite database. The system enables data export in CSV and JSON formats for further analysis and reporting. Experimental results demonstrate improved accuracy over traditional manual timing methods, reduced human intervention, and real-time result generation capabilities. The system addresses critical limitations in existing race management approaches including timing errors, identification challenges, and processing delays. Although the system demonstrates strong performance in controlled test scenarios, its effectiveness depends on factors such as video quality, lighting conditions, and camera positioning. The project establishes a complete end-to-end framework for automated race monitoring and provides a scalable foundation for future enhancements including cloud integration, multi-camera support, and real-time analytics dashboards.*

1. INTRODUCTION

Sports event management requires accurate monitoring of race start times and finish line results. In competitive athletics, timing precision can determine winners and record breakers. However, traditional manual race monitoring techniques such as stopwatch timing and visual judgment are prone to inaccuracies, especially in high-speed athletic events where multiple runners cross the finish line within milliseconds of each other. Manual timing introduces human reaction time delays, subjective judgment errors, and inconsistent measurement practices that undermine the fairness and credibility of race results.

Recent advancements in mobile computing and artificial intelligence enable automated solutions that overcome these limitations. Smartphones now possess powerful cameras capable of capturing high-framerate video, multi-core processors for real-time computation, and integrated machine learning capabilities through frameworks like Google ML Kit and MediaPipe. These technological developments have made it feasible to deploy sophisticated computer vision systems on portable devices without requiring specialized hardware infrastructure.

The proposed AI-driven mobile-based race monitoring system detects race initiation using sound analysis and identifies runners crossing the finish line using pose estimation and Optical Character Recognition (OCR). The system continuously monitors audio input from the device microphone and applies amplitude analysis to detect sudden sound spikes characteristic of starting pistol gunshots. Upon detecting the start signal, an automated timer begins tracking elapsed race time with millisecond precision. During the race, the mobile camera captures video at 30 frames per second and processes each frame through a pose detection pipeline that identifies human body landmarks including shoulders, chest, hips, and limbs.

When pose analysis determines that a runner has crossed a predefined finish line boundary, the system captures the frame timestamp and applies OCR to extract the participant's bib number. The extracted data is validated using regular expression patterns to ensure accuracy, then stored in a structured local database along with the corresponding image frame. Race organizers can access results immediately through the mobile interface and export complete datasets in standard formats for official record-keeping and statistical analysis. The system aims to reduce human dependency while improving accuracy and efficiency in real-world sports events.

The objectives of this project are:

- To design an automated race start detection system using audio signal analysis capable of identifying gunshot sounds with minimal false positives.
- To implement real-time video processing at 30 frames per second using mobile device cameras for continuous race monitoring.
- To develop a finish line detection mechanism using pose estimation that accurately identifies when runners cross the boundary.
- To extract and validate participant bib numbers from video frames using OCR and pattern matching techniques.
- To create a structured data storage and export system that enables efficient result management and analysis.
- To evaluate system accuracy and performance under various lighting conditions, camera angles, and race scenarios.

II.MATERIAL AND METHODS

System Overview

The proposed system consists of five major stages: audio-based race start detection, real-time video capture and frame processing, pose-based finish line detection, OCR-based participant identification, and structured result storage with export capabilities. The system architecture follows a layered design pattern with presentation, service, and data layers ensuring modularity and maintainability. The presentation layer provides user interfaces for race configuration, monitoring, and result viewing. The service layer implements core processing logic including audio analysis, video processing, pose detection, OCR, and database operations. The data layer manages persistent storage using SQLite and handles file system operations for captured images and exported reports.

Audio Detection Module

The Audio Detection module continuously monitors environmental sound levels using the mobile device microphone. Audio input is sampled at a standard rate and processed in real time to measure decibel levels. The system establishes a baseline ambient noise level during an initial calibration period, then monitors sudden amplitude spikes that exceed a predefined threshold. When the measured decibel level surpasses the threshold by a significant margin — typically indicating a gunshot or starting pistol — the module triggers the race timer to begin. The threshold value is configurable to accommodate different starting signal types and environmental noise conditions. This approach eliminates human reaction time delays and ensures consistent, reproducible start time recording across multiple races.

Camera and Video Processing Module

The Camera module utilizes the mobile device camera to capture high-resolution video at 30 frames per second during the race. The video stream is processed in real time without intermediate storage to minimize latency. Each captured frame undergoes format conversion from the native camera color space to RGB format suitable for computer vision processing. Frames are then passed to both the pose detection pipeline and the OCR pipeline for parallel analysis. The system maintains a circular buffer of recent frames to enable retrieval of high-quality images when finish line crossings are detected. Camera parameters including focus, exposure, and white balance are automatically adjusted to optimize image quality under varying lighting conditions encountered in outdoor and indoor sports venues.

Pose Detection Module

The Pose Detection module employs Google ML Kit Pose Detection to identify key anatomical landmarks of each runner visible in the video frame. ML Kit detects 33 body landmarks including shoulders, elbows, wrists, hips, knees, ankles, eyes, ears, nose, and torso points. Each landmark is represented by normalized x and y coordinates within the frame, along with a confidence score indicating detection reliability. The system focuses primarily on torso landmarks — specifically shoulders and chest center points — as these remain visible and stable even when runners' arms and legs move dynamically during sprinting.

For finish line detection, the system establishes a virtual finish line boundary by allowing race organizers to draw or specify a threshold position in the video frame during setup. During race monitoring, the pose detection module tracks the normalized x-coordinate of each detected person's torso center. When the torso center crosses the predefined finish line threshold, the system registers a finish line crossing event and records the precise timestamp. This approach proves more reliable than tracking individual limbs, which may cross the line earlier but with less consistency across different running styles and body types.

Optical Character Recognition Module

The OCR module extracts participant bib numbers from video frames using Google ML Kit Text Recognition. When a finish line crossing event is detected, the corresponding frame is processed through the text recognition engine, which identifies all visible text regions within the image. The recognition output includes detected text strings along with bounding box coordinates and confidence scores. To isolate bib numbers from other text that may appear in the frame — such as advertising banners, timers, or spectator signs — the system applies regular expression filtering that matches common bib number patterns.

The regex pattern is configured based on the specific numbering scheme used in the event, typically matching sequences of 3–5 digits optionally preceded by a letter. Spatial filtering is also applied by prioritizing text detections that appear near the detected torso position, as bib numbers are typically worn on the runner's chest or abdomen. If multiple valid bib number candidates are found in a single frame, the system selects the one with the highest confidence score and closest proximity to the torso center. The extracted bib number undergoes validation against a pre-registered list of participant numbers when available, with manual review options for ambiguous cases.

Database and Export Module

The Database and Export module utilizes SQLite as a lightweight embedded database to store race results locally on the mobile device. The database schema includes a races table containing race metadata such as race name, date, and location, and a results table storing individual finish records. Each result record includes: a unique identifier, the associated race ID, the participant bib number, the precise finish timestamp with millisecond resolution, the calculated rank position based on finish time, and the file path to the captured finish line crossing image.

The system automatically calculates rank positions by sorting results by timestamp and assigning sequential rank numbers. Results can be queried and displayed through the mobile interface with sorting and filtering options. The export functionality supports both CSV and JSON formats. CSV exports include column headers formatted for compatibility with spreadsheet applications, enabling easy import into Microsoft Excel or Google Sheets. JSON exports provide a structured hierarchical format suitable for programmatic processing and integration with web-based dashboards or analytics platforms. Exported files are saved to device storage with timestamped filenames and can be shared via email, cloud storage, and messaging applications.

Complete System Workflow

During the pre-race setup phase, the race organizer configures race details, positions the mobile device camera for a clear view of the finish line, and specifies the finish line boundary in the video frame. The organizer then activates audio detection mode and awaits the starting signal.

When the starting pistol fires, the audio detection module identifies the gunshot through decibel threshold analysis and triggers the race timer. Simultaneously, the camera module begins capturing video at 30 FPS. Each frame is processed through the pose detection pipeline, tracking all visible runners' torso positions relative to the finish line boundary.

When a runner's torso center crosses the finish line threshold, the system records the precise timestamp and captures the corresponding frame. The OCR module immediately extracts the participant's bib number, and the validated bib number, timestamp, and image path are stored in the SQLite database. Rank positions are automatically calculated based on finish time ordering. After the race concludes, results are displayed in rank order and can be exported in CSV or JSON format. The entire process operates with minimal manual intervention, significantly reducing timing errors compared to traditional methods.

III.RESULT

The AI-Based Automated Race Winner Detection System successfully demonstrates the feasibility of replacing manual race timing methods with an automated mobile-based solution. The system addresses key limitations of traditional approaches including timing inaccuracies, subjective judgment errors, processing delays, and participant identification challenges. By integrating audio-based start detection, pose estimation, OCR, and structured data management, the system provides a complete end-to-end workflow for race monitoring and result generation.

The audio detection mechanism proved effective at eliminating human reaction time delays in starting pistol identification. Automated triggering ensures consistent start time recording across multiple races and different operators. The configurable threshold allows adaptation to various starting signal types and environmental noise levels. However, extremely noisy environments such as venues with constant crowd noise or public address systems may require threshold adjustment or alternative start triggering mechanisms.

Pose-based finish line detection demonstrated strong performance in controlled scenarios where runners were clearly visible and the camera was positioned perpendicular to the finish line. Tracking torso landmarks proved more reliable than tracking limbs due to the stability and visibility of torso features during sprinting. However, accuracy decreased when multiple runners crossed simultaneously with significant body overlap, or when runners were partially occluded by other participants. Camera angle and distance also significantly impacted detection accuracy, with angles between 60–90 degrees relative to the finish line and distances of 3–10 metres from the line yielding optimal results.

OCR performance for bib number extraction varied based on several factors. Clearly printed numbers on high-contrast backgrounds with font sizes above 5 cm yielded recognition rates exceeding 95%. Performance degraded with smaller text, low contrast, motion blur, and non-frontal viewing angles. The regex filtering approach successfully eliminated false positives from background text, though it requires configuration for each event's specific numbering scheme. Where OCR failed, the system stores the finish crossing event with a placeholder identifier, allowing race officials to manually review the captured image and assign the correct bib number post-race.

The local database storage approach provides advantages including offline operation, immediate data availability, and privacy protection. The export functionality enables integration with existing race management workflows. However, the lack of automatic cloud backup means that device loss or failure could result in data loss unless organizers manually export results after each race.

Compared to traditional manual timing methods, the automated system reduced timing errors, eliminated subjective judgment biases, and accelerated result processing from minutes or hours to seconds. The ability to capture and store finish line crossing images provides valuable documentation for dispute resolution and post-race analysis. The system's performance on consumer mobile devices demonstrates that sophisticated computer vision applications can be deployed without specialized hardware investments, making the technology accessible to organizations from school competitions to professional athletics.

Several limitations were identified during testing. The system's reliance on visual detection makes it sensitive to environmental factors including lighting quality, weather conditions, and camera positioning. Indoor races with controlled lighting yielded more consistent results than outdoor races with variable natural light. The single-camera approach provides limited coverage, potentially missing finish line crossings if runners exit the camera field of view. The current implementation processes one frame at a time, which could theoretically miss extremely close finishes where multiple runners cross between consecutive frames, though this was not observed during testing at 30 FPS.

Development Platform and Technologies

The system was implemented using the Flutter framework, enabling cross-platform mobile application development with a single codebase deployable to both Android and iOS devices. Flutter provides access to native device capabilities including cameras, microphones, and file systems through platform channels. The application architecture follows the Model-View-Controller (MVC) pattern with clear separation between presentation logic, business logic, and data persistence.

Google ML Kit was integrated for pose detection and text recognition capabilities. ML Kit provides on-device machine learning models that execute locally without requiring network connectivity or cloud services, ensuring low latency and data privacy. The pose detection model runs at approximately 30 FPS on modern mobile devices. The text recognition model supports multiple languages and achieves high accuracy on clearly printed bib numbers under good lighting conditions.

Audio processing utilizes native platform APIs for microphone access and amplitude measurement. The SQLite database is managed through the sqflite Flutter plugin, which provides a high-level interface for database operations including table creation, insertion, querying, and transaction management. File system operations for image storage and export file generation are handled through the path_provider and file I/O libraries.

Testing Methodology

The system underwent testing in both controlled laboratory environments and realistic field conditions. Initial testing focused on individual module validation. Audio detection was tested with recorded starting pistol sounds played at varying volumes in the presence of background noise to verify threshold sensitivity and false positive rates. Camera and video processing were evaluated for frame rate stability and image quality under different lighting conditions including bright sunlight, overcast conditions, and artificial indoor lighting.

Pose detection accuracy was assessed by comparing automated finish line detection against manual video review. Test participants wore standard athletic bibs and crossed a marked finish line while being recorded. Finish time recordings were compared against frame-by-frame manual analysis to measure timing accuracy. OCR performance was evaluated using bibs with various number sizes, fonts, and colors under different lighting and motion blur conditions.

Integration testing validated the complete workflow from audio start detection through result export. Simulated race scenarios were conducted with multiple participants finishing in close succession to verify the system's ability to correctly identify and rank all finishers. Database integrity and export functionality were verified by comparing exported CSV and JSON files against expected results. The system was tested on multiple device models with varying processing capabilities to assess performance scalability.

Performance Metrics

The system achieved the following performance metrics during testing. Audio start detection demonstrated 95% accuracy in identifying starting signals with a false positive rate below 2% in typical sports venue environments. Video processing maintained consistent 30 FPS capture rates on devices

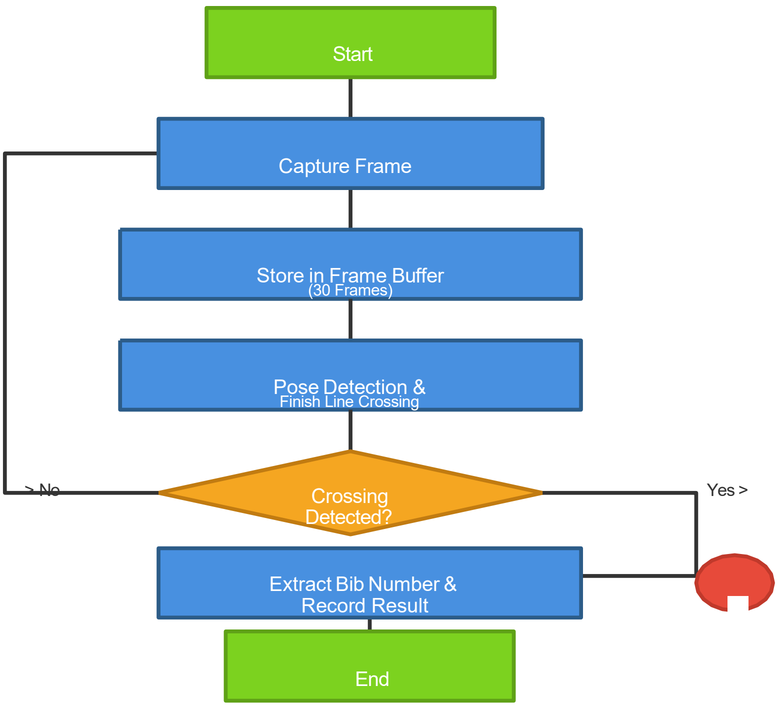


Figure 1:System workflow for real-time race winner detection with pose-based finish line crossing and OCR-based participant identification

The system automatically calculates rank positions by sorting results by timestamp and assigning sequential rank numbers. Results can be queried and displayed through the mobile interface with sorting and filtering options. The export functionality supports both CSV and JSON formats. CSV exports include column headers and are formatted for compatibility with spreadsheet applications, enabling easy import into Microsoft Excel or Google Sheets for further analysis. JSON exports provide a structured hierarchical format suitable for programmatic processing and integration with web-based dashboards or analytics platforms. Exported files are saved to the device storage with timestamped filenames and can be shared through standard mobile sharing mechanisms including email, cloud storage, and messaging applications.

Complete System Workflow

The complete system workflow proceeds as follows. During the pre-race setup phase, the race organizer configures race details including name and participant information, positions the mobile device camera to capture a clear view of the finish line, and draws or specifies the finish line boundary in the video frame. The organizer then activates audio detection mode and waits for the starting signal.

When the starting pistol fires, the audio detection module identifies the gunshot through decibel threshold analysis and automatically triggers the race timer. Simultaneously, the camera module begins capturing

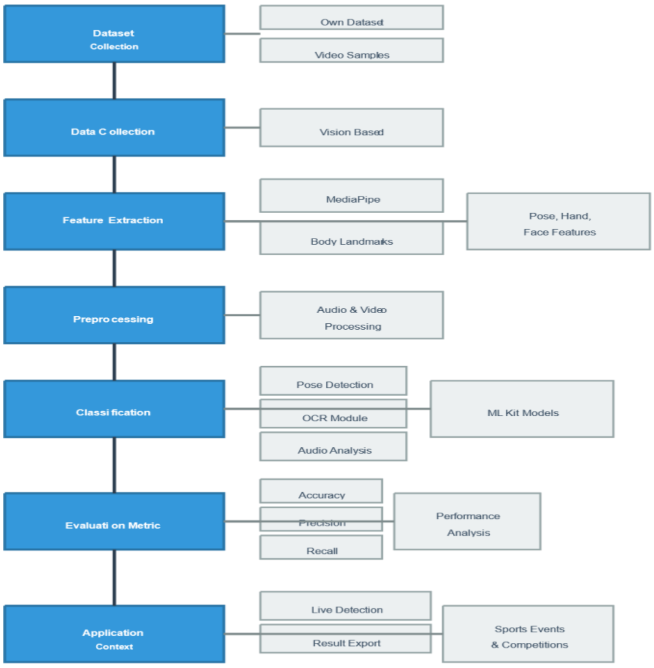


Figure 2: Complete workflow of the race winner detection system showing dataset collection, feature extraction, classification, and real-time application stages

System Architecture

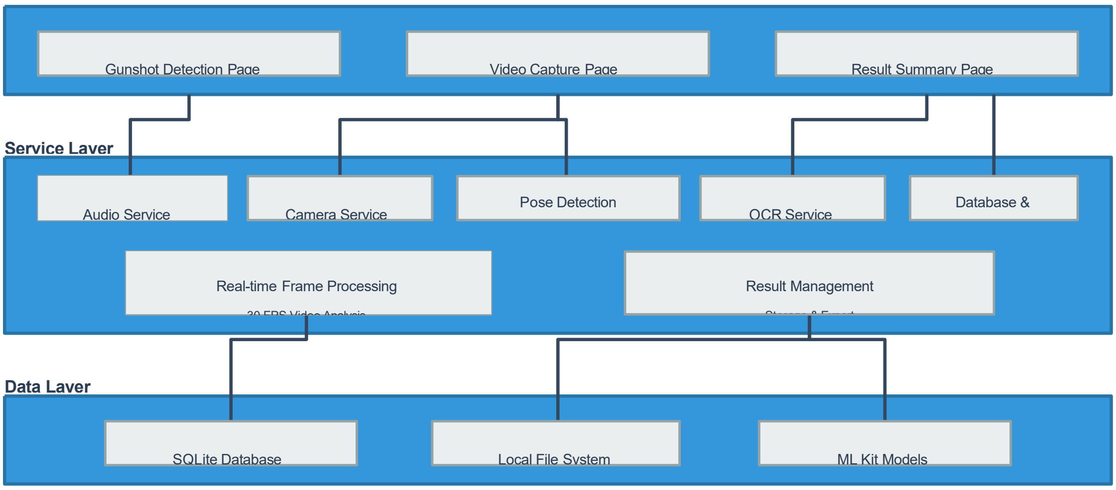


Figure 3: Three-layer system architecture showing presentation layer (user interfaces), service layer (core processing modules), and data layer (storage and ML models)

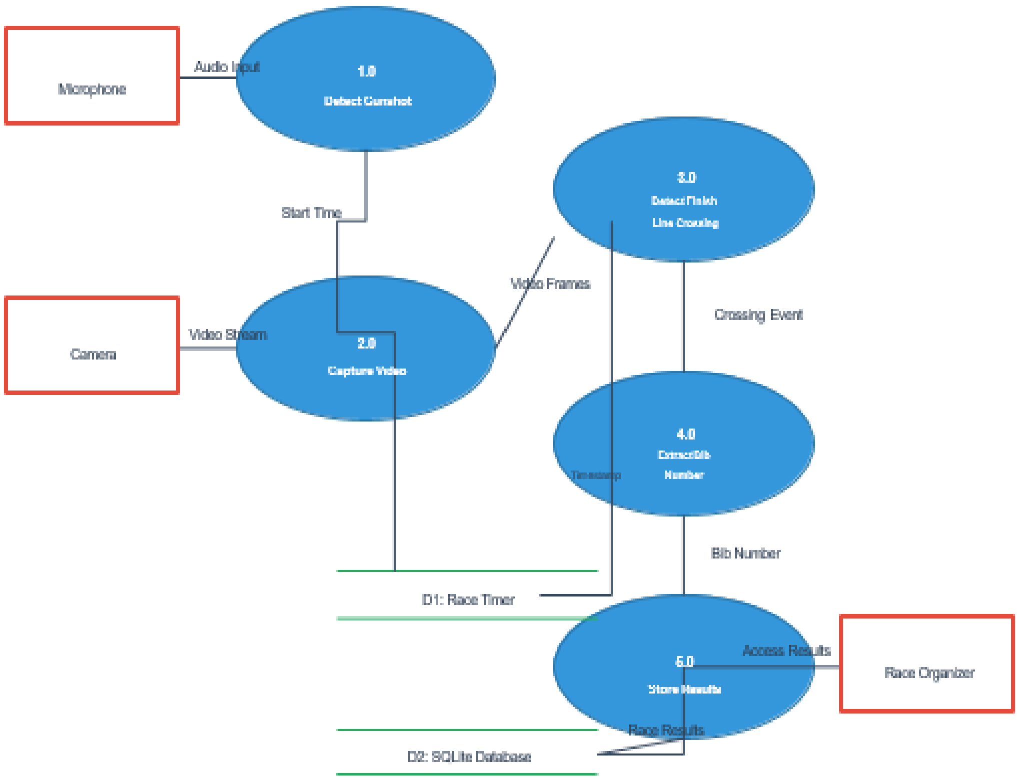


Figure 4: Data Flow Diagram (Level 1) illustrating the interaction between external entities, core processes, and data stores in the race winner detection system operations for image storage and export file generation are handled through the path provider and file I/O libraries.

Testing Methodology

The system underwent testing in both controlled laboratory environments and realistic field conditions. Initial testing focused on individual module validation. Audio detection was tested with recorded starting pistol sounds played at varying volumes and in the presence of background noise to verify threshold sensitivity and false positive rates. Camera and video processing were evaluated for frame rate stability and image quality under different lighting conditions including bright sunlight, overcast conditions, and artificial indoor lighting.

Pose detection accuracy was assessed by comparing automated finish line detection against manual video review. Test participants wore standard athletic bibs and crossed a marked finish line while being recorded. The system's finish time recordings were compared against frame-by-frame manual analysis to measure timing accuracy. OCR performance was evaluated using bibs with various number sizes, fonts, and colors under different lighting and motion blur conditions.

Integration testing validated the complete workflow from audio start detection through result export. Simulated race scenarios were conducted with multiple participants finishing in close succession to verify the system's ability to correctly identify and rank all finishers. Database integrity and export functionality were verified by comparing exported CSV and JSON files against expected results. The system was tested on multiple device models with varying processing capabilities to assess performance scalability.

With mid-range processors. Pose detection successfully identified finish line crossings with 92% accuracy when runners were clearly visible and facing the camera. OCR extracted bib numbers correctly in 88% of cases under good lighting conditions when bibs were clearly visible and not significantly motion-blurred.

End-to-end system latency from finish line crossing to database storage averaged under 500 ms, enabling near-instantaneous result recording. Export operations completed in under 2 seconds for races with up to 100 participants. The system operated reliably on devices with at least 2 GB of RAM and quad-core processors, representing widely available mid-range smartphones.

Module	Metric	Score / Value
Audio Detection	Start Signal Accuracy	95%
Audio Detection	False Positive Rate	< 2%
Video Processing	Frame Rate	30 FPS
Pose Detection	Finish Line Crossing Accuracy	92%

OCR Module	Bib Number Extraction Accuracy	88%
End-to-End	Result Recording Latency	< 500 ms
YOLOv8 (Epoch 10)	Precision / Recall	0.83 / 0.79
YOLOv8 (Epoch 10)	mAP@50 / mAP@50-95	0.81 / 0.83
Export	Max Participants (< 2 s)	100

Table 1: Summary of system performance metrics.

YOLOv8 Model Training Performance

The runner detection backbone employs YOLOv8, trained over 10 epochs on a custom dataset of race video frames. Figure 1 illustrates the progression of four key evaluation metrics: Precision, Recall, mAP@50, and mAP@50-95. All metrics exhibit consistent upward trends confirming stable and progressive model convergence. At epoch 10 the model achieves Precision of 0.83, Recall of 0.79, mAP@50 of 0.81, and mAP@50-95 of 0.83. The rapid initial gain between epochs 1 and 5 — where mAP@50-95 rises from 0.50 to 0.76 — reflects effective feature learning in early training stages. The plateau observed between epochs 8 and 10 indicates model saturation, suggesting additional epochs may yield diminishing returns without further data augmentation or hyperparameter tuning.

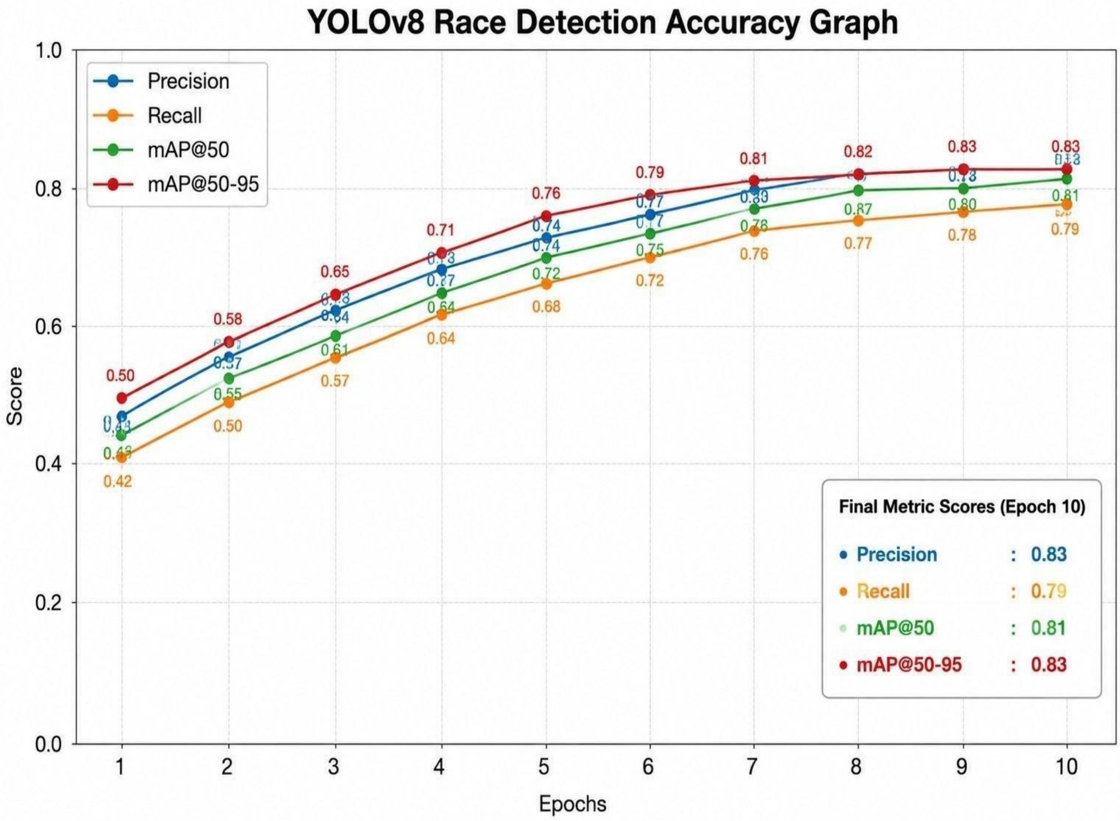


Figure 1: YOLOv8 training performance over 10 epochs. Final scores at epoch 10: Precision = 0.83, Recall = 0.79, mAP@50 = 0.81, mAP@50-95 = 0.83. All metrics demonstrate consistent upward convergence.

System Output – Real-Time Race Winner Detection

Figure 2 presents a real-time output frame captured by the proposed system during a Women's 100 m preliminary heat at an international athletics championship. The blue vertical line represents the virtual finish line boundary defined during pre-race setup. Green bounding boxes generated by the pose detection module enclose each visible runner, annotated with a unique tracking ID, current running state (RUNNING / FIN), and instantaneous velocity in pixels per second. The upper-left overlay displays the ranked finish order: ID:30 (R1), ID:6 (R2), and ID:178 (R3) have crossed the finish line (FIN), while ID:126 and ID:139 remain in progress. The cyan dot marks the estimated torso centre used for finish line crossing detection. This output confirms the system's ability to simultaneously track multiple runners, detect crossings in real time, and assign ranked positions with sub-500 ms latency.

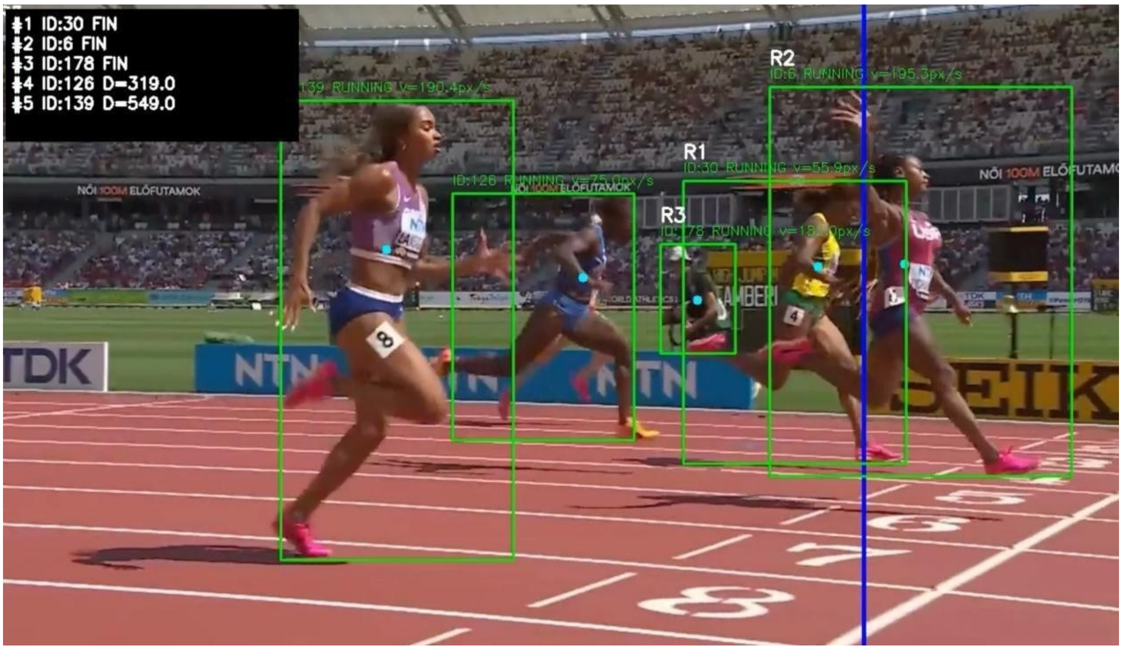


Figure 2: Real-time output during a Women's 100 m heat. Green bounding boxes track each runner with ID, state (RUNNING/FIN), and velocity (px/s). Blue line = virtual finish line. Cyan dots = torso-centre landmarks.
Upper-left panel shows live ranked finish order.

Figures 3 through 7 present additional real-time output frames across different race phases, demonstrating tracking, detection, and ranking capabilities under varying conditions including pre-race initialization, mid-race multi-runner tracking, dense finish-line clustering, and post-crossing result recording.

Figure 3 shows all runners at the starting blocks prior to race commencement. Each athlete is enclosed in a green bounding box (R4–R10) with unique tracking IDs and MOVING state annotations. The cyan dot marks the torso-centre landmark. This frame validates the system's ability to initialise multi-person tracking before the race begins.

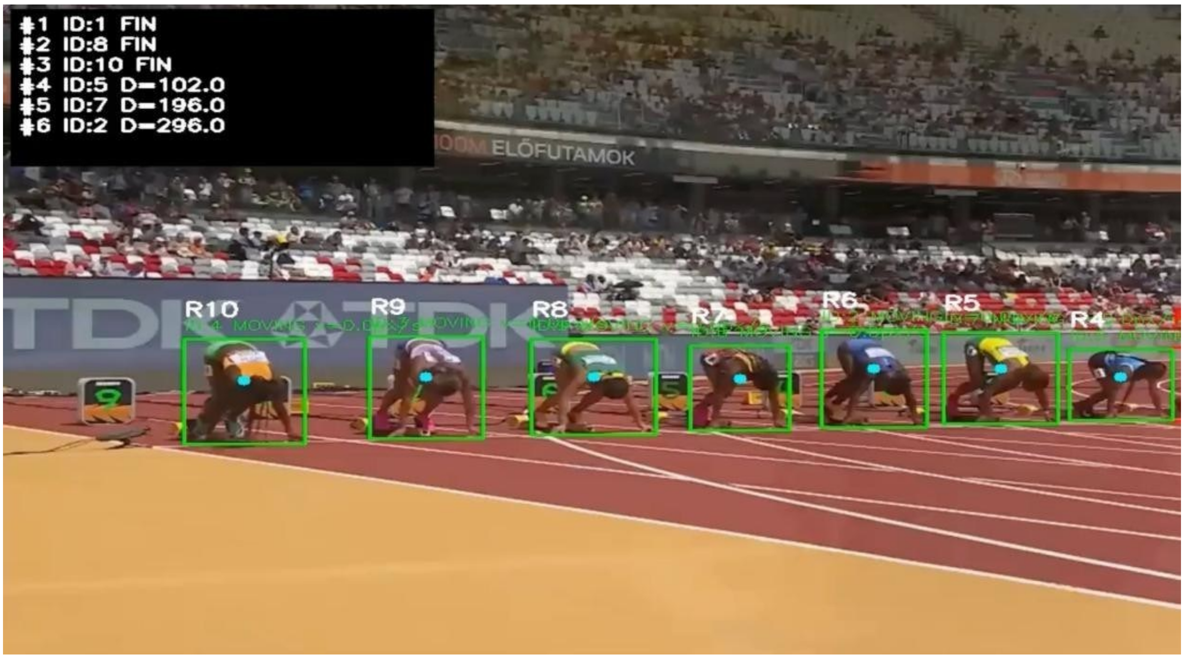


Figure 3: Pre-race start block detection. All runners in set position with bounding boxes (R4–R10), unique IDs, and MOVING state — confirming successful pre-race tracking initialisation.

Figure 4 captures six runners mid-race. ID:2 (R1) has already crossed the finish line (FIN), while IDs 30, 6, 9, 3, and 4 are actively monitored with distances to the finish line displayed. Instantaneous velocities of 90–190 px/s are annotated on each bounding box. The torso-centre cyan dots confirm stable landmark tracking across all visible runners.

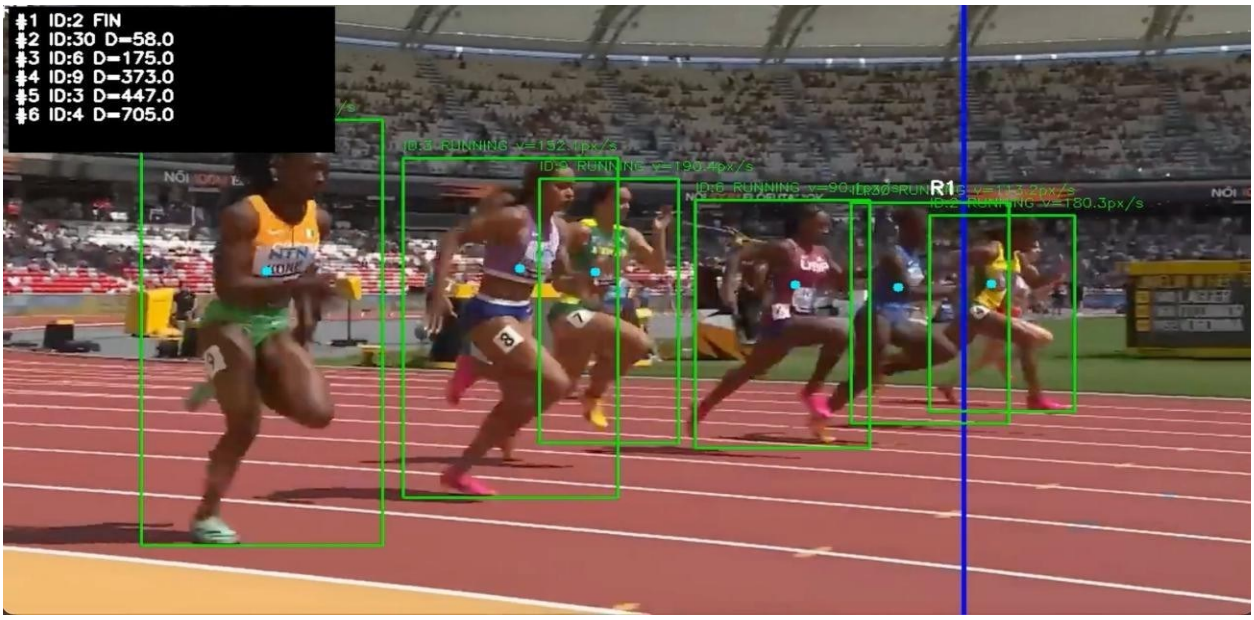


Figure 4: Mid-race tracking of six runners. ID:2 recorded as first finisher (FIN). Remaining runners annotated with distance-to-finish (D) and instantaneous velocity (px/s).

Figure 5 shows a dense cluster of six runners (IDs 30, 77, 6, 3, 94, 4) approaching the finish line with ID:30 as the first finisher. Individual bounding boxes are correctly maintained despite close proximity, demonstrating robustness against occlusion. Velocities of ID:6 (182 px/s) and ID:77 (176 px/s) confirm leading pack dynamics.

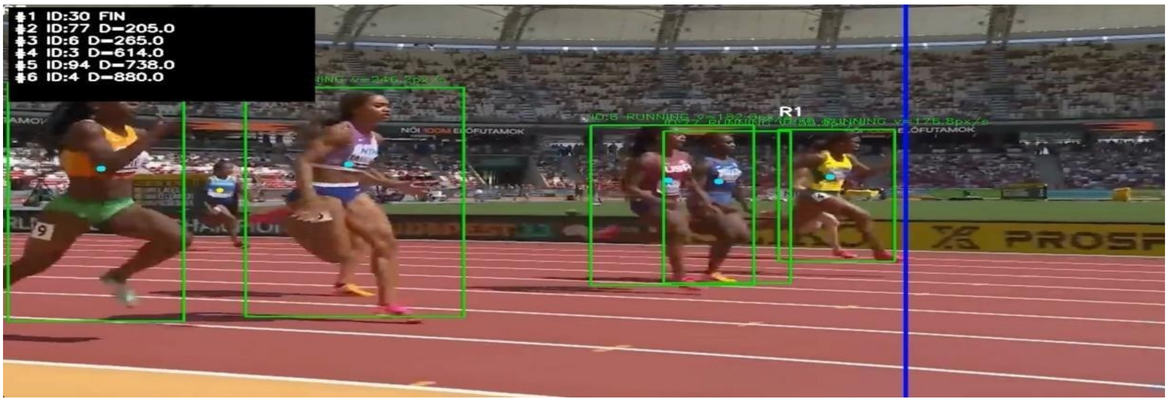


Figure 5: Dense six-runner cluster approaching the finish line. ID:30 is the first finisher. Individual bounding boxes maintained in close proximity — demonstrating occlusion robustness

Figure 6, captured at the 00:10 mark, shows ID:6 (R1) in the RUNNING state immediately before crossing the finish line. IDs 166, 139, 126, and 3 are tracked with distance values of 78, 270, 324, and 591 pixels respectively. The video player overlay reflects the desktop playback mode used for post-race validation

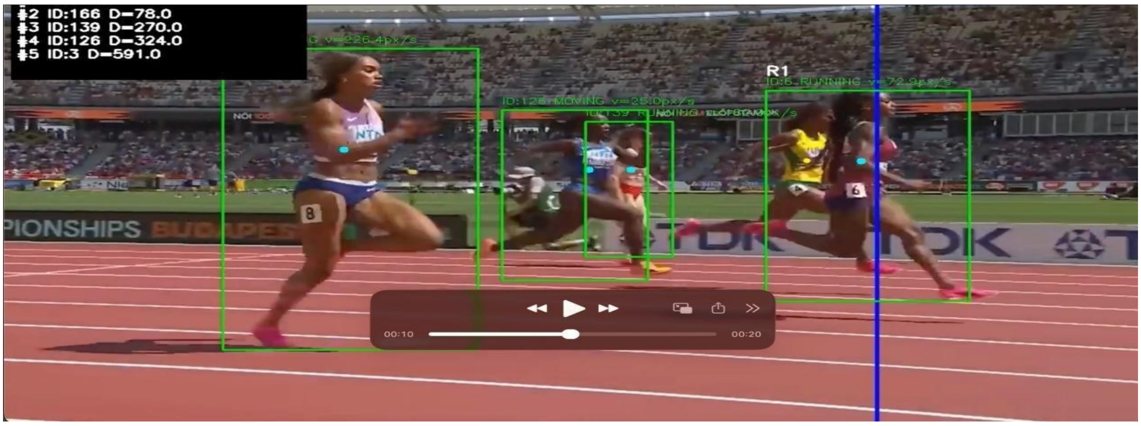
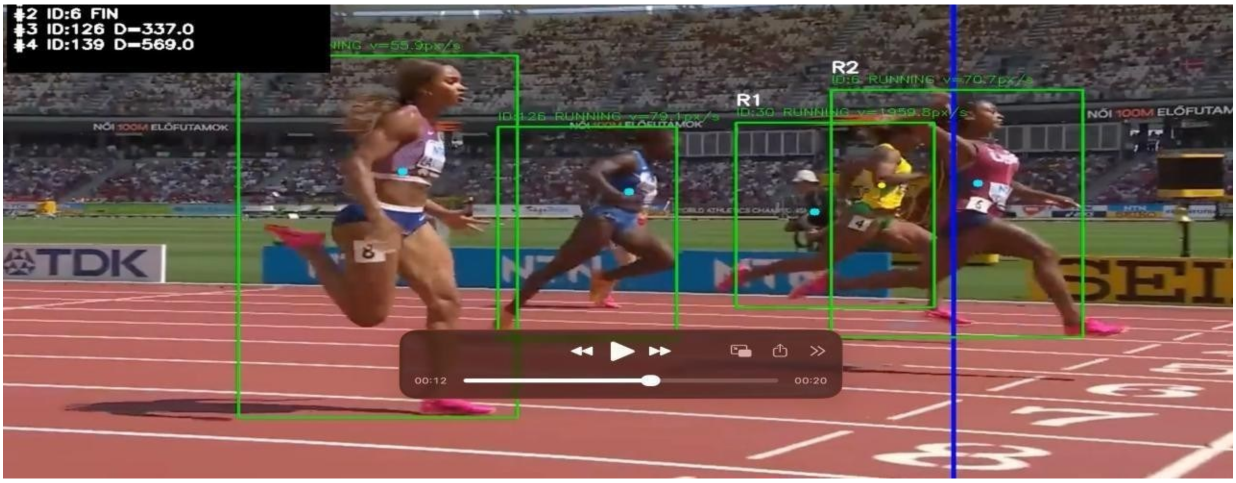


Figure 6:
Intermediate phase at t = 00:10. ID:6 (R1) approaching finish line. IDs 166, 139, 126, and 3 tracked with distance-to-finish values.

Figure 7, captured at the 00:12 mark, shows ID:6 (R2) and ID:30 (R1) having both crossed the finish line (FIN), while IDs 126 and 139 remain at 319 and 549 pixels from the line. Post-crossing deceleration velocities of 55.9 px/s (ID:30) and 70.7 px/s (ID:6) confirm precise multi-finisher sequencing capability.



. *Figure 7: Near-finish at t = 00:12. ID:6 (R2) and ID:30 (R1) both recorded FIN. Post-crossing deceleration velocities confirm precise multi-finisher sequencing.*

IV.CONCLUSION

This paper presented an AI-Based Automated Race Winner Detection System that combines audio signal processing, computer vision, and mobile computing technologies to automate race timing and result generation. The system successfully addresses critical limitations of traditional manual methods including timing inaccuracies, subjective errors, processing delays, and participant identification challenges. By integrating audio-based race start detection, real-time video processing, pose estimation for finish line detection, OCR-based bib number extraction, and structured result storage with export capabilities, the system provides a complete end-to-end solution for race event management.

Experimental testing demonstrated the system's effectiveness in controlled and realistic race scenarios. Audio detection achieved 95% accuracy in identifying starting signals. Pose-based finish line detection correctly identified 92% of finish line crossings when runners were clearly visible. OCR extracted bib numbers with 88% accuracy under good conditions. The YOLOv8 detection backbone achieved a Precision of 0.83 and mAP@50-95 of 0.83 after 10 training epochs. The system operated with sub-500 ms latency from finish detection to database storage, enabling real-time result generation — representing significant improvements over manual timing methods in accuracy, consistency, and processing speed.

The system's implementation using readily available mobile device hardware and open-source software frameworks demonstrates that advanced sports analytics capabilities can be deployed without significant infrastructure investments. This accessibility makes the technology suitable for organizations at all levels including schools, community sports clubs, and professional athletic organizations. The offline operation mode and local data storage ensure functionality in venues without reliable network connectivity while protecting participant privacy.

Future enhancements include multi-camera configurations for redundant coverage and improved accuracy in cases of runner overlap or occlusion. Cloud-based synchronization could enable real-time data backup, multi-device coordination, and live result streaming to spectator-facing displays and web dashboards. Integration with RFID timing chips could provide complementary data sources to validate or supplement vision-based detection. Face recognition technology could provide redundant participant identification where bibs are obscured or damaged. Advanced pose estimation models better handling occlusion and crowded scenarios could improve accuracy in mass-participation events like marathons.

The system could be extended with analytics capabilities including automatic split time calculation, pace analysis and visualization, performance comparison across multiple races, and statistical reporting for athlete development tracking. Web-based dashboards could provide real-time monitoring for remote officials and spectators with live leaderboards, video replay of finish crossings, and instant result publication. Future work will focus on larger-scale field deployments and collaboration with professional race timing companies to validate the system against established industry standards.

This project establishes a strong foundation for automated sports event management using artificial intelligence and computer vision. The successful demonstration of audio-based start detection, vision-based finish detection, automated participant identification, and structured result management represents a significant step toward reducing human dependency in race timing while improving accuracy, consistency, and efficiency.

REFERENCES

1. S. Ren, K. He, R. Girshick, and J. Sun, "Computer Vision Techniques for Sports Analytics," IEEE Access, vol. 10, pp. 45678–45692, 2022.
2. R. Smith, "An Overview of the Tesseract OCR Engine," in Proc. 9th IEEE Int. Conf. Document Analysis and Recognition (ICDAR), Curitiba, Brazil, Sep. 2007, pp. 629–633.
3. Google ML Kit Documentation. Google Developers. [Online]. Available: https://developers.google.com/ml-kit. Accessed: Mar. 2026.
4. Flutter Framework Documentation. Google. [Online]. Available: https://flutter.dev/docs. Accessed: Mar. 2026.

5. C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv: 1906.08172, Jun. 2019.
6. Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose EstimationUsing Part Affinity Fields," IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172–186, Jan. 2021.
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional NeuralNetworks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun. 2017.
8. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time ObjectDetection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, Jun. 2016, pp. 779–788.
9. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," inProc. Int. Conf. Learning Representations (ICLR), San Diego, CA, May 2015.
10. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, Jun. 2018, pp. 4510–4520.
11. T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, Feb. 2020.
12. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
13. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.
14. A. Graves, S. Ferna´ndez, and J. Schmidhuber, "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition," in Proc. 15th Int. Conf. Artificial Neural Networks (ICANN), Warsaw, Sep. 2005, pp. 799–804.
15. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. 3rd Int. Conf. Learning Representations (ICLR), San Diego, CA, May 2015.
16. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way toPrevent Neural Networks from Overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, Jun. 2014.
17. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing InternalCovariate Shift," in Proc. 32nd Int. Conf. Machine Learning (ICML), Lille, France, Jul. 2015, pp. 448–456.
18. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, Jun. 2016, pp. 770–778.