KEC Conference 2026

ISSN: 3021-940X (Print) DOI: 10.3126/injet.v3i2.95782

Enhancing Tennis Player Tracking Accuracy Using a Vision-Based Framework with YOLOv8, Adaptive Kalman Filtering and Homography-Based Court Mapping

Bivedh Nhuchhe Pradhan[1*], Kritika Joshi[2], Kushal Shrestha[3], Lalit Joshi[4]

[1]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur and Nepal, bivedhnhp@gmail.com*

[2]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur and Nepal kritika11joshi@gmail.com*

[3]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur and Nepal shresthakushal125@gmail.com*

[4]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur and Nepal lalitjoshi2062@gmail.com*

Abstract

Manual player monitoring in sports is error-prone, and existing automated systems are economically constrained by the need for high-speed cameras. This paper presents a cost-effective tennis player tracking system leveraging YOLOv8, which achieves 94.90% training accuracy and 97.87% validation accuracy. The system employs a vision-based framework combining a key point-based approach for court landmark detection, Homography transformation to map image coordinates to real-world court positions, and a Kalman filter for robust tracking during fast movements and occlusion. Together, these components enable quantitative analysis including trajectory visualization, distance coverage, speed estimation, and heatmap generation.

Keywords: Tennis, Player tracking, YOLOv8, Kalman Filter, Homography, Heatmap

1.Introduction

Recognizing human actions in sports through video analysis has become a significant focus within the field of computer vision and deep learning as it greatly contributes to identifying players' movements, supporting performance evaluation and helps coaches in strategy planning and training. Despite various significance, recognizing the actions of players in sports has great challenges due to various factors like players movement, speed, and so on.

For sports like tennis, analyzing the actions of players through video analysis comes as a great challenge. Subtle movement between each stroke and speed of players makes the analysis harder for human eyes, hence object tracking can be used for tracking the players to see their performance in games. While traditional object tracking methods often rely on hand designed features which makes it difficult to handle complex scenes like occlusion, and changes in the appearance of objects, deep learning has shown remarkable ability in handling large-scale data and complex pattern recognition tasks [1].

So in order to help with the tracking of players movement in the sports field, using YOLO comes as a great help due to YOLO being an object detection architecture that frames object detection as a single regression problem, directly predicting bounding boxes and class probabilities from an input image in one forward pass of neural network. At the same time YOLOv8 is the iteration of the YOLO family of object detection models which are known for their speed and accuracy since it builds upon the success of its predecessors while introducing several key innovations that push the boundaries of real-time object detection.

Due to the updated backbone in YOLOv8, it has faster training speeds, reduced memory usage, and improved precision-recall tradeoffs while also having a built-in support for Ultralytics training tools and Python API, which makes it easier to implement, evaluate and deploy. YOLOv8 uses an anchor-free design, decoupled heads (separate branches for classification and regression), and a more advanced backbone based on CSP and C2f modules instead of a handcrafted anchor boxes and MaxPool layers for down sampling like its previous versions.

* *Corresponding author*

2. Related Research

Recent studies have examined the use of deep learning techniques to improve the accuracy of sports image classification across a wide variety of sports categories. In one study, several well-known architectures such as ResNet-50, EfficientNet-B7, DenseNet-121, and YOLOv8 were evaluated using a dataset consisting of 12,500 sports images representing 110 different categories. Although all of the models demonstrated good classification capability, YOLOv8 achieved the highest overall performance among them. Other research has also concentrated specifically on the YOLOv8 architecture and reported strong results, reaching about 94.90% training accuracy and 97.87% validation accuracy. The model's ability to detect and classify objects within an image at the same time enables it to identify important elements such as players, balls, and sports equipment, which contributes to a more accurate understanding of sports scenes. These findings indicate that detection-based approaches, particularly YOLOv8, can outperform many traditional deep learning models in sports image classification tasks.[2]

Recent advancements in AI have driven extensive research on YOLO-based models for sports analytics. Wu and Xiao demonstrated that an optimized lightweight YOLO framework improves inference speed in embedded systems while maintaining acceptable accuracy. Li and Zhao compared YOLO variants (YOLOv5, YOLO-X, YOLOv7) and found YOLOv7 achieves state-of-the-art performance due to better feature aggregation and training optimization, making it suitable for dynamic sports. Vidor et al. combined YOLO with DeepSORT tracking for tennis ball monitoring, showing improved trajectory estimation and object continuity. Glaspey further showed that YOLOv8 outperforms Mask R-CNN in real-time tennis ball detection with precise localization. Overall, YOLOv7 and YOLOv8 represent state-of-the-art solutions; however, most studies focus mainly on detection accuracy and lack adaptive tracking and court-coordinate mapping. Therefore, an integrated framework combining YOLOv8 with adaptive motion modeling and homography-based spatial mapping is necessary for robust tennis player tracking in real-world scenarios.[3]

This work explores an improved adaptive Kalman filter for tracking moving targets in sports videos. Recent studies enhance tracking accuracy by integrating multiple features and refining filtering techniques. One approach used a multi-feature fusion Kalman filter combining color histogram and LBP texture features, achieving about a 55% improvement in tracking accuracy on 600 video frames. It improved precision by adaptively weighting features and refining target position estimates, while remaining robust under occlusion, complex backgrounds, and varying lighting. Compared to a conventional single-feature Kalman filter (~20-pixel deviation), the proposed method reduced deviation to about 9 pixels. Overall, these findings emphasize that adaptive filtering combined with multi-feature fusion significantly improves accurate and reliable target tracking in complex sports videos.[4]

3. Research Gap

Although YOLOv8 provides a strong object detection performance in sports videos, its performance degrades under some challenging real-life conditions like motion blur, occlusion, and missing frames which are quite commonly seen in tennis matches. Some existing studies which use YOLO along with tracking methods like Kalman filter and homography based court mapping still struggle with maintaining precision in fast-moving tennis match videos. So, designing a more reliable system which can track players accurately under all challenging criteria like low quality video or any other commonly experienced real-life problems is required.

4. Methodology

This section describes the complete experimental methodology used in this study. It begins with the collection and preprocessing of tennis match videos, including frame extraction, annotation, and conversion of the dataset into the required format for training. The methodology then explains the implementation of the YOLOv8 models used for player detection and court key point detection. After detection, a Kalman filter–based tracking approach combined with the Hungarian algorithm is applied to maintain consistent player identities across frames. Homography transformation is used to map image coordinates to real-world court coordinates for accurate spatial analysis. Finally, the system computes player movement statistics such as distance covered, speed estimation, and heatmap visualization to analyze player performance.

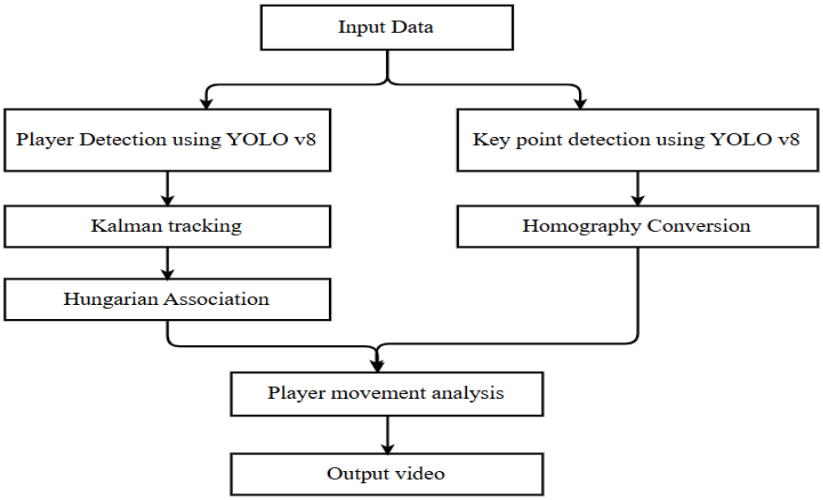

Figure 1. Model Architecture

4.1 Dataset Description

The player detection dataset was sourced from Roboflow, comprising 4,166 high-resolution images spanning diverse court surfaces (clay, hard, grass), camera angles, lighting conditions, and player appearances. Since this is a single-class detection task, class imbalance is not a concern. The dataset was split into training (72%, 3,000 images), validation (17%, 691 images), and test (11%, 475 images) subsets [5]. Although the dataset size may appear limited, this is justified by the transfer learning approach adopted. The model was initialized with pre-trained COCO weights, which already encode generalizable visual features of edges, textures, and human body contours learned across 118,000+ images [6]. Tennis key point dataset was sourced from GitHub user 'Sergey Kosolapov' containing 8,841 images with 14 court keypoints annotated in JSON format. Dataset split into training (75%, 6630 images) and split (25%, 2211 images).

4.2 Adaptive Kalman filter

The object tracking in the proposed system was performed to track players throughout the match using a Kalman filter based tracking algorithm. Initially, the players were detected in each video frame using the YOLOv8 object detection model. The Kalman filter was then employed to estimate and track the state of each detected player over time, allowing robust tracking even in the presence of temporary occlusions or missed detections [7]. Each player was represented by an 8-dimensional state vector:

$$\mathbf{x} = [u,\ v,\ a,\ h,\ \dot{u},\ \dot{v},\ \dot{a},\ \dot{h}]^\top \quad \text{Equation 1}$$

where (u, v) is the bounding-box center, the aspect ratio, h the height, and the dotted terms their frame-to-frame velocities. YOLOv8 detections are converted as $u = (x_1 + x_2)/2$, $v = (y_1 + y_2)/2$, where (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right corners.

4.3.1 Filter Dynamics

A constant-velocity model with dt = 1 frame gives the state-transition and measurement matrices which reads only the four position states, since YOLO provides no velocity information. The standard predict–update recursion is

$$\hat{\mathbf{x}}_{k|k-1} = F\hat{\mathbf{x}}_{k-1|k-1}, \qquad P_{k|k-1} = FP_{k-1|k-1}F^\top + Q_k \quad \text{Equation 2}$$

The error covariance was updated in Joseph form to preserve symmetry under finite-precision arithmetic:

$$P_{k|k} = (I - K_k H)\, P_{k|k-1}\, (I - K_k H)^\top + K_k R K_k^\top$$

Equation 3

4.3.2 Parameter Initialization

The initial covariance is set as

$$P_0 = \mathrm{diag}(\underbrace{100,\ 100,\ 100,\ 100}_{\text{position}},\ \underbrace{5000,\ 5000,\ 5000,\ 5000}_{\text{velocity}})$$

Equation 4

A position uncertainty of 100 corresponds to a ±10-pixel spatial tolerance, matching the typical YOLOv8 localization error at 1080p. Velocity uncertainty is set 50× higher because no velocity estimate is available at initialization, following the standard SORT tracker convention. The measurement noise matrix

$$R = \mathrm{diag}(1.5,\ 1.5,\ 2.5,\ 6.0)$$

Equation 5

was fitted to the variance of YOLO outputs on stationary ground-truth boxes: center jitter is below 2 px (1.5), height varies moderately with pose (2.5), and aspect ratio carries the largest variance (6.0) due to racket occlusion, arm extension, and motion blur.

4.3.3 Adaptive Process Noise

Upon receiving detection z_k, the innovation $y_k = z_k - H x_{k|k-1}$ drives an adaptive scaling factor

$$\alpha_k = \max\left(1,\ \frac{\mathbf{y}_k^\top \mathbf{y}_k}{\mathrm{tr}(\mathbf{S}_k)}\right), \qquad \mathbf{S}_k = H P_{k|k-1} H^\top + R$$

Equation 6

so that $Q_k = \alpha_k Q_0$ expands automatically during abrupt manoeuvres. A hard-coded multiplier additionally scales Q_0 by 3.0 on detected direction reversals and by 1.2 otherwise. The base noise matrix

$$Q_0 = \mathrm{diag}(20,\ 20,\ 0.01,\ 3,\ 8,\ 8,\ 0.001,\ 1.5)$$

Equation 7

was set to the measured per-frame variance of each state on sample footage. Position entries (20) match the 15–25 px displacement of players moving at 3–5 m s⁻¹ at broadcast resolution; velocity entries (8) cover observed accelerations; aspect-ratio and size-velocity entries are near zero as these quantities remained stable across most frames.

4.3.4 Gating and Velocity Damping

Detections beyond a Mahalanobis distance of 5.0 (≈ two standard deviations in four-dimensional measurement space) trigger a proportional gain boost capped at 8.0 to prevent the filter from fully discarding its prediction on a single anomalous detection. Predicted velocities are additionally damped each step to contain drift:

$$\text{lateral decay} = \begin{cases} 0.97 & \text{normal motion} \\ 0.85 & \text{direction reversal} \end{cases}, \qquad \text{size decay} = 0.90 \qquad \text{miss decay} = \begin{cases} 0.92 & \text{lost} \le 8 \text{ frames} \\ 0.75 & \le 20 \text{ frames} \\ 0.50 & > 20 \text{ frames} \end{cases}$$

The Kalman filter parameters were determined through empirical tuning on sample broadcast footage, with each parameter varied across candidate ranges and evaluated based on tracking continuity, identity switch rate, and false track frequency. The initial position covariance of 100 was selected after lower values (10–50) produced excessive filter rigidity causing track loss during fast movements, and higher values (200+) introduced positional drift during stationary play. The Mahalanobis gating threshold of 5.0 follows established SORT tracker conventions and was empirically confirmed as the optimal balance between rejecting spurious detections and maintaining continuity under occlusion. Adaptive scaling multipliers were tuned to observed tennis player motion characteristics at broadcast resolution, where per-frame displacements of 15–25 pixels and abrupt directional changes are well-documented. Formal ablation studies with retained

metrics and systematic comparison against standard baselines such as SORT and DeepSORT are acknowledged as directions for future work.

4.4 Data Association: Hungarian Algorithm

At each frame, the Hungarian algorithm finds the optimal one-to-one matching between the n predicted Kalman tracks T = {T₁, …, T□} and the m YOLO detections $\mathcal{D}$ = {D₁, …, D□} by solving the linear assignment problem:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} C_{i,j}\, x_{i,j}, \quad \sum_{j} x_{i,j} \leq 1, \quad \sum_{i} x_{i,j} \leq 1$$

Equation 8

Each binary variable x_{i}□ ∈ {0, 1} indicates whether track i is assigned to detection j. The row and column constraints enforce at most one assignment per track and per detection.

4.4.1 Cost Matrix

A cost matrix C ∈ R^{n×m} is built for every track–detection pair. All entries are initialized to 1000.0 (effectively infinite), so unevaluated pairs are never selected. Three branches set the actual cost depending on spatial overlap:

$$\text{cost} = \begin{cases} 0.6(1 - \text{IoU}) + 0.2\, d_n + 0.1(1 - s_r) + 0.1(1 - a_r) & \text{if IoU} > 0.1 \\ 0.7 + 0.1\, d_n + 0.1(1 - s_r) + 0.1(1 - a_r) + 0.2\, \max(0, -\text{GIoU}) & \text{if } d_n < 3.0 \\ \min(1.0 + 0.2\, d_n,\ 2.0) & \text{otherwise} \end{cases}$$

Equation 9

Here d□ is the center distance normalized by detection height, s_r is the height ratio, and a_r is aspect-ratio consistency.

4.4.2 Cost Modifiers

The raw cost is then adjusted for track age, missed frames, and detection confidence:

$$\text{cost} = \frac{\text{cost} \times (1 + f_m)}{1 + f_a} \div \max(\text{conf},\ 0.3)$$

Equation 10

The missed-frames factor f□ is capped at 0.5 (max 1.5× penalty) to prevent long-lost tracks from accumulating costs that block re-association. The age reward f_a is capped at 0.2—larger rewards unfairly favored stale tracks in earlier versions. The confidence floor of 0.3 stops weak detections from artificially inflating match priority.

4.4.3 Acceptance Thresholds

A state-dependent threshold τ rejects assignments whose cost exceeds it. The base threshold was raised from 0.8 to 1.0 to align with the updated cost scale. Confirmed strong tracks (>10 hits) receive a threshold that grows with lost frames, reaching a maximum of 3.0 at 12 or more lost frames—directly solving the re-identification problem where a returning player would otherwise spawn a new track instead of re-attaching to their original identity. Unconfirmed tracks are held to a tighter 0.8 to suppress false positives before a track is established.

4.4.4 Unmatched Tracks

Detections with no accepted match spawn new tracks. Unmatched existing tracks are retained across frames rather than immediately deleted to survive short occlusions. Tracks that exceed the maximum allowed loss streak are then removed.

4.5 Tennis Court Analysis

4.5.1 Hartley Normalization

Real world co-ordinates of tennis court used were full doubles court width = 10.97m, full court length baseline to baseline = 23.77m, singles court width = 8.23m, width of each double's alley = 1.37m, distance from baseline to service line = 6.4m, half singles width from singles sidelines = 4.115m. A 300×600 pixel mini-court canvas was setup. Applying T to any point (x, y, 1) produced the normalized coordinate s×(x-cx), s×(y-cy) in a single matrix multiplication, which made it efficient to normalize all 14 key points simultaneously by stacking them into a matrix and multiplying by T in one operation.

$$T = \begin{bmatrix} s & 0 & -s \times c_x \\ 0 & s & -s \times c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.003378 & 0 & -2.162 \\ 0 & 0.003378 & -1.520 \\ 0 & 0 & 1 \end{bmatrix}$$ Equation 11

4.5.2 Direct Linear Transform

Homography was computed with the normalized coordinates using a direct linear transform algorithm. For each pair of source points (xs, ys) and destination points (xd, yd), the homography constraint yielded two linear equations as shown in matrix form [8].

$$x_d^{(i)} \left(h_{31} x_s^{(i)} + h_{32} y_s^{(i)} + h_{33}\right) = h_{11} x_s^{(i)} + h_{12} y_s^{(i)} + h_{13}$$ Equation 12

$$y_d^{(i)} \left(h_{31} x_s^{(i)} + h_{32} y_s^{(i)} + h_{33}\right) = h_{21} x_s^{(i)} + h_{22} y_s^{(i)} + h_{23}$$ Equation 13

4.5.3 RANSAC

Some keypoints were partially occluded while some were detected with low confidence or slightly mislocated due to motion blur or broadcast compression. Even a single key point missing was corrupting the DLT solution since it fits all points equally. This problem was handled using RANSAC, which was applied as an outer loop around DLT computation.

For 30 iterations:

- Randomly, 4-point correspondences were selected from the 14 available points.
- Homography matrix (H) was computed using DLT on 4 points.
- After that all source points were projected through H.
- Inliers were counted which are the points where reprojection error < 8 pixels.
- If the inliers count exceeded the current best, that H was stored as best candidate.
- Final H was recomputed using DLT on all inliers of the best candidate [9].

4.5.4 Temporal Smoothing

An exponential moving average was applied to the homography estimates across consecutive frames where the smoothed homography was computed by assigning 85% weight to the previously computed homography and 15% to the newly computed homography which allowed homography to adapt gradually to genuine changes in camera angles while filtering out high frequency noise from keypoint detector.

4.6 Player Stats Analysis

Player speed was estimated using a sliding window approach over a fixed interval of 5 frames rather than consecutive frame differences. This approach was more robust to detect noise since single frame displacements were susceptible to bounding box jitter even when the player was stationary. As for distance estimation, total distance covered was accumulated incrementally in each frame. Since the speed window measurement was computed for every frame but covers 5 frames of movement, the full window distance was

divided by 5 before being added to the cumulative total. Positions where the player was not detected due to occlusion, camera cuts, or detection failures were filled using linear interpolation between the nearest valid positions before and after the gap.

4.7 Training process

The player detection model was trained using YOLOv8m initialized with pre-trained COCO weights and fine-tuned on approximately 3,000 domain-specific tennis images. Training was conducted on a single NVIDIA T4 GPU under the Kaggle computing environment. The input resolution was fixed at 640×640 pixels with a mini-batch size of 16. The model was optimized using the Adam solver with a cosine annealing learning rate schedule over 100 epochs, with early stopping configured at a patience of 25 epochs to prevent over-training.

Table 1. Training hyper parameters of the player detection model

Parameter	Value
Base Model	YOLOv8m
Pre-training	COCO (80 classes, 118k images)
Input resolution	640 x 640
Batch size	16
Epochs	100
Patience	25
LR schedule	Cosine annealing
Flip augmentation	0.5
HSV hue jitter	0.015
HSV saturation jitter	0.7
Rotation	±10°
Mosaic composition	1.0
MixUp blending	0.1

To address cross-dataset generalization, the trained model was additionally tested on an independent two-minute broadcast video sourced from YouTube, featuring match footage entirely unseen during training. The model demonstrated consistent and reliable detection throughout. The primary failure case was partial occlusion caused by on-screen score overlays, which occasionally reduced detection of confidence in a known limitation of single-stage detectors under occlusion. During rapid directional changes, detection confidence also dropped due to motion blur, which was mitigated through a Kalman filter-based tracking mechanism that maintained player position continuity across low-confidence frames. Overall, cross-video evaluation confirmed effective generalization to unseen broadcast footage, with failures limited to physically justified edge cases rather than systematic generalization breakdown.

5. Result and Analysis

5.1 Model evaluation

The trained model was evaluated on 691 fully held-out validation images.

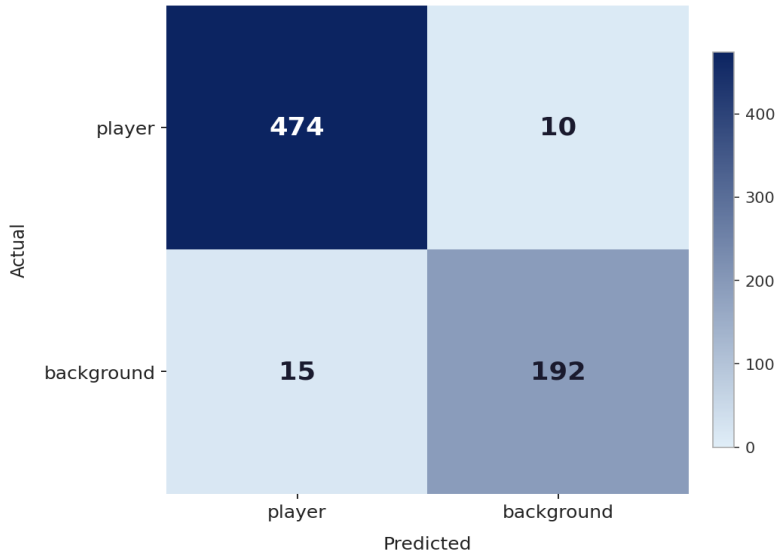

Figure 3. Confusion matrix

Validation metrics:

Table 2. Player detection evaluation metrics

Metric	Value
Precision	96.9%
Recall	97.9%
F1 Score	97.4%
Accuracy	96.4%
mAP@50	98.1%
mAP@50-95	84.7%

Table 2 confirms strong overall detection performance. The precision of 96.9% and recall of 97.9% are nearly equal, yielding an F1 score of 97.4% and confirming a well-balanced detector. The mAP@50 of 98.1% indicates near-perfect detection at the standard IoU threshold, while mAP@50-95 of 84.7% validates robust spatial localization quality under strict overlap criteria. The absence of overfitting is evidenced by these strong held-out validation metrics as a model that memorized training samples would fail to maintain such performance on 691 completely unseen images. The near-identical precision and recall further confirm generalizable learned representations rather than training distribution bias. The high recall of 97.9% is particularly critical for downstream Kalman filter tracking integration, where missed detections propagate directly as track losses.

5.2 Player Tracking Evaluation and Benchmarking

To evaluate the robustness of the proposed tracking pipeline, the system was tested on 30 second tennis match clip (750 frames at 25 fps). Since frame level ground truth trajectory annotations were not available for this study, standard MOT benchmark metrics (MOTA, MOTP, IDF1) could not be computed directly. Instead, proxy tracking metrics were derived from the internal state of the tracker across all processed frames. To benchmark the proposed system, a comparative evaluation was conducted against two widely used baseline tracking algorithms SORT and DeepSORT, which extends SORT with appearance-based reidentification features. All systems were evaluated on the same tennis match clip using identical YOLO detection inputs.

Table 3. Player tracking evaluation metrics and bench marking

Metric	Proposed System	SORT	DeepSORT
Total Frames	750	750	750
Total Track ID	2	6	3
Extra Spawns	0	4	1
Fragmentation Rate	0.0000	0.0053	0.0013
Track Stability Score	1.00	0.9467	0.9867
Both Players Visible	72.0%	64.67%	92.8%
Any Player Visible	99.6%	99.73%	99.73%
Worst Occlusion Survived	24 frames	1 frame	1 frame

It was observed that SORT produced 4 extra identity spawns while DeepSORT produced 1, confirming that general purpose trackers are more prone to identity switches in the constrained two player tennis environment. The most notable advantage of the proposed system was its occlusion handling. It maintained track identity through occlusions of up-to 25 consecutive frames, compared to just 1 frame for both SORT and DeepSORT. This was due to adaptive Kalman filter dynamically adjusting process noise during sudden movements and applying velocity decay during occlusion, enabling continuous state prediction without a matching detection.

5.*3* *Court Keypoint Detection Validation Result*

Court keypoint detection achieved the following results.

Table 6. Keypoint detection metrics

Metric	Value
mAP	97.32%
mAP50	97.48%
mAP75	97.34%

5.4.2 Mean Pixel Error (MPE) and Normalized Key-point Error (NKE)

MPE measures the average Euclidean distance (in pixels) between predicted key-points and ground-truth key-points.

$$MPE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i^{pred} - x_i^{gt})^2 + (y_i^{pred} - y_i^{gt})^2} \quad \text{Equation 14}$$

where, N = number of visible key-points. Pixel error alone was resolution-dependent, so normalization was required, MPE divided by D where, D = reference distance image diagonal in pixels computed from top-left and bottom right ground truth corners. MPE was found to be 3 pixels and NKE was found to be 0.00375.

5.4 Final Output

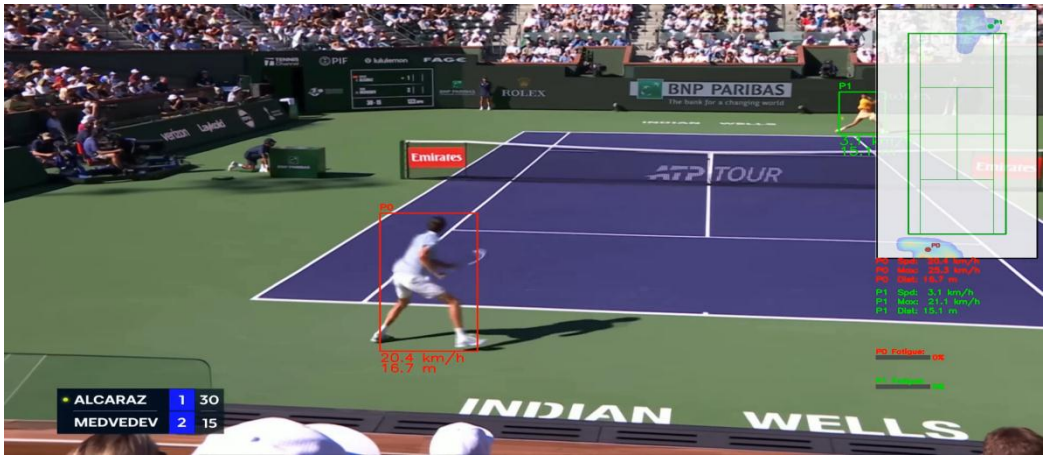

Figure 4. Final output

6. Conclusion

The integration of YOLOv8 and Adaptive Kalman Filtering facilitates the transition from raw video to high-level analytics. Identity consistency is maintained via the Hungarian Algorithm, which solves the global assignment problem between predicted states and new detections. This ensures stable tracking data throughout the match duration.

By applying a Homography transformation, the system maps image coordinates to a top down 2D plane. This enables the calculation of real-world kinematics (speed/acceleration) and the generation of spatial heatmaps and tactical minimaps. Ultimately, this framework converts video sequences into a quantitative dashboard for professional performance evaluation

7. Acknowledgement:

We would like to express our sincere gratitude to our supervisor, Er. Nirajan Acharya, for his valuable guidance, encouragement, and continuous support throughout the course of this research. His insights and feedback have been instrumental in sharing the direction and quality of our work.

We also affirm that all authors have contributed equally to the research and preparation of this paper.

References

[1] Z. Jin and H. Yang, “Real time object tracking using deep learning and opencv,” *Applied and Computational Engineering*, vol. 35, pp. 272–*279*, 2024.

[2] M. H. Rahman, A. Mohiul Islam, A. I. Hasan, M. Uddin, A. Ahmed, A. A. Miazee, and Y. Hossain, “Yolov8 image processing for evaluation of stability algorithms based on neural networks: A sports use case,” in *International Conference on Inventive Communication and Computational Technologies*. Springer, 2024, pp. 613–622.

[3] A. Bista, A. Chaudhary, D. K. Chhetri, and U. Sapkota, "Tennis Ball Analysis System using Advanced YOLO," Project Report, Dept. Comput. Sci. Eng., CMR Inst. Technol., Bengaluru, Karnataka, India, 2025.

[4] G. Quanan and X. Yunjian, “Kalman filter algorithm for sports video moving target tracking,” in *2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI)*. IEEE, 2020, pp. 26–30.

[5] B. Dwyer, J. Nelson, and J. Solawetz, "Roboflow," 2022. [Online]. Available: https://roboflow.com

[6] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, and D. Ramanan, "Microsoft COCO: Common objects in context," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 740–755.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," Trans. ASME—J. Basic Eng., vol. 82, no. 1, pp. 35–45, Mar. 1960.

[8] K. I. Sainan, M. Mohamad, Z. Mohamed, S. B. Saari, M. F. Mat, and N. S. Khusaini, "Athletes tracking using homography method: a preliminary study," International Journal of Engineering & Technology, vol. 7, no. 4, pp. 6–10, 2018.

[9] J. M. Martinez-Otzeta, I. Rodriguez-Moreno, I. Mendialdua, and B. Sierra, "RANSAC for robotic applications: A survey," Sensors, vol. 23, no. 1, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/327