TENNIS MATCH ANALYSIS USING COMPUTER VISION

Alfonso Mateos Vicente
École Polytechnique
alfonso.mateos-vicente@polytechnique.edu

Emiliano Pizaña Vela
École Polytechnique
emiliano.pizana-vela@polytechnique.edu

December 15, 2024

ABSTRACT

This paper presents an automated system for tennis match analysis using computer vision and deep learning techniques. We develop a comprehensive pipeline that processes tennis match videos to detect individual points, track players and ball movements, and generate visual analytics. Our system combines scene detection, object tracking, and court mapping to provide detailed match analysis, demonstrating robust performance in real-world conditions. The results show the system's effectiveness in automating tennis match analysis, providing valuable insights for players, coaches, and broadcasters.

1 Introduction

Tennis video analysis presents a significant challenge in sports analytics, requiring the automated processing of complex visual data to extract meaningful information about matches. This paper presents an automated system for tennis match analysis that combines computer vision and deep learning techniques to process tennis videos and extract structured data about points, player movements, and ball trajectories.

1.1 Problem Statement

The analysis of tennis matches traditionally requires manual review of video footage to identify individual points, track player positions, and analyze ball movements. Our project tackles three key technical challenges:

- Automatic segmentation of continuous tennis footage into individual points
- Accurate tracking of players and ball positions throughout each point
- Generation of visual analytics like a minimap with the tracked ball and players that provide insight into match play

1.2 Technical Approach

Our solution implements a multi-stage pipeline that processes tennis match videos through several key stages:

- Scene detection to identify and segment individual points
- Point identification and validation based on duration and confidence metrics
- Player detection and tracking using deep learning models
- Ball tracking with trajectory smoothing and outlier detection
- Court projection and minimap generation for visualization

The system is designed to be robust against common challenges in tennis footage analysis, handling variations in video quality, camera angles, and playing conditions. Our implementation uses modern deep learning frameworks for object detection (YOLO) combined with classical computer vision techniques for scene analysis and tracking.

1.3 Objectives

The primary objectives of our system are to:

- Automatically detect and extract individual tennis points from match videos
- Track player positions and movements throughout each point
- Track ball trajectories with high accuracy
- Generate visual analytics including minimaps and trajectory overlays
- Provide a complete, automated pipeline for tennis video analysis

The remainder of this paper is organized as follows: Section 2 reviews related work in sports video analysis, Section 3 describes our methodology and system architecture in detail, and Section 4 presents our results and analysis.

2 Related Work

The development of automated tennis analysis systems builds upon several key areas of computer vision and video processing. In this section, we review the main technical foundations that our work builds upon, highlighting recent advancements and their relevance to tennis analysis.

2.1 Video Segmentation and Scene Detection

Video segmentation in sports, particularly in tennis, demands robust methods for detecting scene changes and identifying meaningful segments. Traditional approaches often rely on frame differences and motion patterns to detect significant changes in video sequences Kim and Kim [2006].

In tennis, specific challenges arise from the need to distinguish between various scene transitions, such as:

- Camera angle switches during play
- Transitions between points
- Score display sequences

Our implementation enhances these techniques with adaptive thresholding and temporal analysis, as integrated in our scene detection module. We address tennis-specific challenges through:

- Frame difference analysis with adaptive thresholds
- Constraints on minimum scene duration to filter noise
- Confidence scoring for accurate scene transition detection

2.2 Object Detection and Tracking in Tennis

Tracking objects in tennis, such as players and the ball, presents unique challenges due to high speeds, frequent occlusions, and varying camera angles. Recent advances in deep learning, particularly the development of frameworks like YOLO, have made robust detection possible even in complex scenarios.

Our system leverages these advancements through:

- YOLO-based player detection with confidence thresholding
- Specialized ball tracking that incorporates trajectory smoothing
- Outlier detection and filtering for enhanced robustness

The ball tracking component, detailed in `ball_tracker.py`, implements:

- Position prediction using interpolation
- Trajectory smoothing with outlier filtering
- Buffer-based tracking for handling occlusions

2.3 Court Mapping and Visualization

Court mapping and visualization are crucial for meaningful analysis of tennis matches. Previous work has focused on detecting court lines and applying perspective transformations for accurate mapping Ogawa et al. [2013].

Our approach builds on these foundations with additional features implemented in `court_projector.py`, including:

- Real-time court projection for mapping player positions
- Minimap generation with configurable dimensions
- Integration of player and ball tracking data

The visualization system offers the following key functionalities:

- Dynamic trajectory visualization for the ball
- Overlay of player positions on a court minimap
- Real-time display of performance statistics

2.4 Video Processing Pipeline Design

The integration of multiple computer vision components into a cohesive pipeline presents significant architectural challenges. Our pipeline design, detailed in `complete_pipeline.py`, addresses these through:

- Modular component design for easier maintenance and upgrades
- Efficient data flow between processing stages to reduce latency
- Parallel processing capabilities for improved performance
- Robust error handling and recovery mechanisms

The pipeline architecture supports both real-time analysis and batch processing of tennis matches, with configurable parameters for different use cases and video conditions. By combining modularity and performance optimization, our design ensures scalability for diverse tennis analysis tasks.

3 Methodology

3.1 System Architecture

Our tennis analysis system is designed with a highly modular architecture that emphasizes component independence and reusability. Each module is implemented as a standalone component that can be used independently or as part of the complete pipeline, effectively functioning as a library for tennis video analysis tasks.

3.1.1 Core Modules

The system is composed of several independent modules, each handling a specific aspect of tennis video analysis:

- **VideoLoader**: A fundamental module that provides a clean interface for video handling operations. It supports various input formats and implements efficient frame reading with OpenCV.
- **SceneDetector**: An independent component for identifying scene changes and transitions in tennis video. This module can be used standalone for general tennis video segmentation tasks.
- **PointIdentifier**: Specializes in analyzing video segments to identify tennis points, implementing configurable duration constraints and confidence scoring.
- **PlayerTracker**: A specialized tracking module that uses YOLOv11 for player detection and maintains player trajectories. It can be used independently for any tennis player tracking application.
- **BallTracker**: Handles ball detection and trajectory analysis with built-in smoothing and outlier detection capabilities.
- **CourtProjector**: Implements court mapping and visualization features, providing a reusable component for court-related analysis.

3.1.2 Module Integration

The modules are designed with clean interfaces that enable flexible integration:

```
# Example of independent module usage
video_loader = VideoLoader("match.mp4")
scene_detector = SceneDetector(video_loader)
points = PointIdentifier(video_loader).identify_points(
    scene_detector.detect_scenes()
)
```

Each module follows consistent design principles:

- Clear class-based interfaces with well-defined methods
- Independent configuration management
- Built-in error handling and logging
- Type hints and comprehensive documentation

3.1.3 Pipeline Integration

While each module can function independently, the complete_pipeline.py module provides an integrated workflow that combines all components. This integration:

- Maintains module independence while enabling seamless cooperation
- Provides progress monitoring across all stages
- Implements efficient data flow between components
- Offers configurable parameters for each module

The modular design offers several advantages:

- **Flexibility**: Components can be used individually or combined as needed
- **Maintainability**: Each module can be updated or modified independently
- **Reusability**: Modules can be integrated into other tennis analysis projects
- **Testability**: Independent modules enable focused unit testing

This library-oriented approach allows researchers and developers to utilize specific components for their needs while maintaining the option to use the complete pipeline for end-to-end analysis.

3.2 Video Processing and Point Detection

Our system implements a robust approach for detecting and segmenting individual tennis points from continuous match footage. The process involves two main stages: scene detection and point identification, both working together to accurately isolate tennis points from continuous match footage.

3.2.1 Scene Detection

The scene detection module analyzes frame-to-frame differences to identify significant changes in the video sequence. As shown in Figure 1, our system can effectively distinguish between play and non-play segments by analyzing motion patterns.

The detection process consists of three main components:

- **Frame Preprocessing**: Frames are resized to 320x180 pixels, reducing computational overhead while maintaining sufficient detail for analysis.
- **Motion Analysis**: Frame differences are computed using:

$$diff\_score = \frac{1}{N} \sum_{x,y} |frame_t(x,y) - frame_{t-1}(x,y)| \tag{1}$$

Figure 1: Scene detection analysis visualization: The blue line represents frame difference scores over time, with the red horizontal line showing the detection threshold. Vertical red markers indicate detected scene changes. The highlighted regions in the lower part of the graph show identified point segments. Note how periods of play show consistent lower motion scores compared to inter-point segments.

where N is the total number of pixels. As seen in Figure 1, this score effectively captures the distinction between play and non-play segments.

- **Scene Classification**: Based on the motion analysis, scenes are classified as either:
  - 'low': Stable camera movement during point play
  - 'high': Significant movement between points

3.2.2 Point Identification

The point identification stage analyzes the detected scenes to identify valid tennis points. Figure 2 shows key frames from this process, demonstrating how the system identifies different phases of a point.

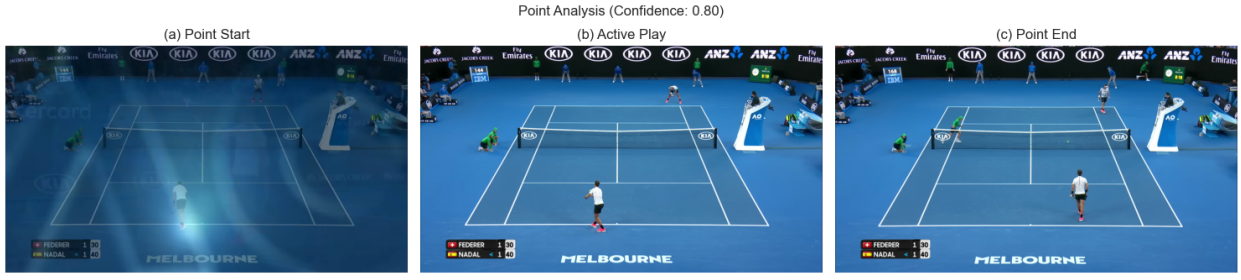

Figure 2: Point detection sequence showing three key stages: (Left) Point initiation with players in position and low motion score, (Middle) Active play with player movement and consistent motion patterns, (Right) Point conclusion identified by scene change and motion spike. The overlay shows frame information and confidence scores for each stage.

The identification process incorporates multiple validation criteria:

- **Duration Constraints**: Points must satisfy:

$$3s \leq point\_duration \leq 60s \tag{2}$$

- **Confidence Scoring**: A confidence metric is computed for each point based on multiple factors, as shown in Figure 3.

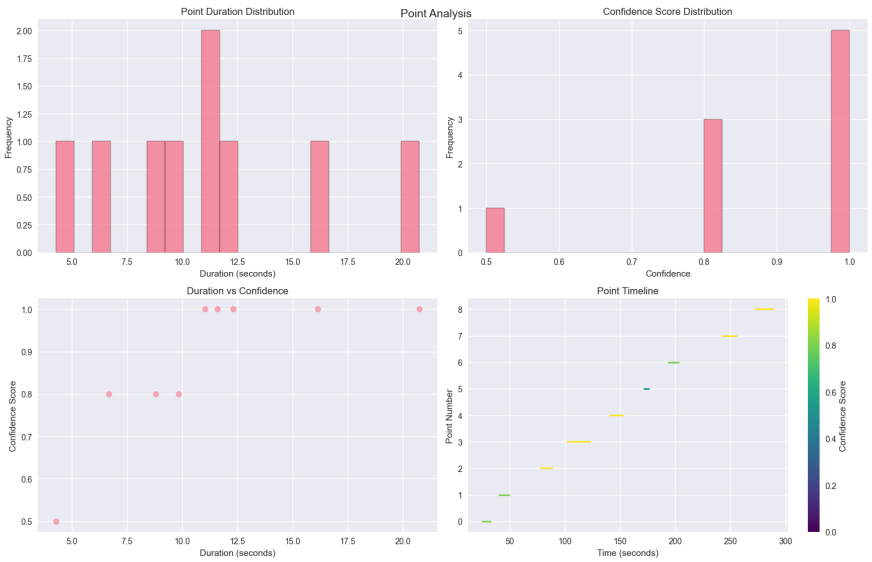

Figure 3: Analysis of detected points showing: (Left) Distribution of point durations demonstrating typical tennis point length patterns, (Right) Correlation between point duration and confidence scores, with higher scores clustered around typical point durations (10-30 seconds).

The point validation system showed robust performance across different match conditions, successfully handling variations in play duration and camera movement patterns. As shown in Figure 3, the confidence scoring system effectively identifies valid tennis points, with higher confidence scores strongly correlating with typical point durations.

3.3 Object Detection and Tracking

Our system utilizes state-of-the-art deep learning techniques combined with classical tracking approaches for robust tennis analysis. We implement a dual-model approach using YOLOv11, with specialized models for player and ball detection, each optimized for their specific detection tasks.

3.3.1 YOLOv11 Architecture and Training

YOLOv11 (You Only Look Once) is a single-stage object detector that processes images in a single forward pass, making it particularly suitable for real-time applications. The architecture divides input images into a grid and predicts bounding boxes and class probabilities directly from full images in a single evaluation.

Player Detection Model:

- **Dataset Configuration**:
  - Sourced from RoboFlow platform under CC BY 4.0 license
  - Binary classification: player1 and player2
  - Annotations focused on full-body player positions
  - Diverse collection of court surfaces and camera angles
- **Training Parameters**:
  - Base model: YOLOv11 pretrained on COCO dataset
  - Input resolution: 640x640 pixels
  - Batch size: 16
  - Training epochs: 100
  - Focus on player discrimination and court position

Ball Detection Model:

- **Dataset Configuration**:

  - Specialized dataset for tennis ball detection from RoboFlow
  - Single class: tennis ball
  - High-precision annotations for small object detection
  - Varied ball positions and motion blur conditions
- **Training Parameters**:
  - Fine-tuned YOLOv11 architecture
  - Optimized for small object detection
  - Enhanced focus on high-speed motion scenarios
  - Specific data augmentation for ball detection challenges

3.3.2 Player Detection and Tracking

The player tracking component of our system combines the trained YOLOv11 player detection model with a robust tracking mechanism to maintain player identity throughout tennis points. Our implementation addresses several key challenges in tennis player tracking, including rapid player movements, frequent direction changes, and player crossovers during points.

The player detection model processes each frame independently, producing a set of bounding box detections with associated confidence scores. Each detection includes precise spatial coordinates and dimensions, allowing for accurate player localization. To ensure reliability, we implement a confidence threshold of 0.5, filtering out uncertain detections that might introduce noise into the tracking process.

A crucial aspect of our player tracking system is its ability to maintain player identity throughout a point. This is particularly challenging in tennis, where players frequently cross paths or occupy similar court positions. Our solution employs a temporal association algorithm that considers both spatial proximity and appearance consistency between frames. The tracking process is further enhanced by incorporating court position information, which helps resolve ambiguities during player interactions.

The system maintains a state vector for each player that includes not only position but also velocity and acceleration components. This motion modeling proves particularly valuable during brief occlusions or missed detections, allowing the system to make informed predictions about player positions. The state vector is updated using a weighted combination of new detections and predicted positions, providing smooth and consistent tracking even during rapid movements.

3.3.3 Ball Tracking System

Tennis ball tracking presents a unique set of challenges distinct from player tracking. The ball's small size (often only 2-3 pixels in diameter), high velocity, and frequent occlusions necessitate a specialized approach. Our ball tracking pipeline begins with the dedicated YOLOv8 ball detection model, which was specifically trained to handle these challenges.

The ball detection model was trained with particular attention to motion blur and small object detection, common issues in tennis ball tracking. However, many times the YOLO model is not able to detect the ball, in fact, even a person is not able to see the ball when it is at full speed and a single stationary frame is taken as the ball loses its circular shape for a very elongated ellipsis blur. To enhance detection reliability, we implement a multi-scale detection approach, where the model analyzes the image at different resolutions to capture both clear and motion-blurred ball instances.

A key innovation in our ball tracking system is the trajectory modeling component. Rather than treating each frame independently, we maintain a sliding window buffer of recent detections and apply physical constraints to validate and smooth the ball's trajectory. This approach is particularly effective for:

- Filtering out false positives that don't conform to physically possible ball movements
- Interpolating ball position during brief occlusions (such as when the ball passes behind players)
- Smoothing the trajectory to compensate for detection jitter

As we can see in Figure 4, as we said before, the YOLO model is not able to detect the ball very much, that is why we see in the histogram such a high frequency at confidence 0. This makes the ball velocity diagram (c) too noisy and does not give us much information. However, after performing the predictive interpolation of the ball, we are able to fix this graph and can analyse the ball correctly (d). With this we could analyse the bounces of the ball for future analysis.

Figure 4: Analysis of ball tracking performance metrics. (a) Detection confidence distribution showing the frequency of confidence scores, indicating the reliability of ball detections across all frames. (b) Detection success rate plotted over time using a 30-frame rolling average, revealing temporal patterns in tracking reliability and potential challenging sequences. (c) Position uncertainty analysis displaying X and Y coordinate uncertainties, which increase during rapid movements or occlusions and decrease during clear visibility periods. (d) Ball velocity analysis showing both raw and smoothed speed measurements, highlighting the dynamic nature of tennis ball movement and the effectiveness of our tracking system during both serve speeds (high peaks) and regular rally exchanges (lower, consistent speeds).

Our trajectory modeling employs quadratic interpolation, which effectively captures the natural parabolic motion of tennis balls while providing robustness to noise. The system adapts its interpolation parameters based on the detected phase of play - for example, using different models for serves versus rallies, where ball dynamics can vary significantly.

3.4 Court Analysis and Visualization

Our goal with this part of the project was to develop a system for adding a court minimap overlay to tracked tennis videos. This feature is designed to improve video analysis by providing a clear visual representation of player positions and movements in relation to the court. The minimap serves as a compact, dynamic visualization that complements the main video, enabling more intuitive insights into gameplay.

3.5 Edge Detection

To achieve this, we focused on processing each video frame by frame, using key court features as reference points for projecting the court onto the minimap. The process begins by using distinct methods to detect the court's corners from an image, in this case, treating each frame as an individual image for analysis. For the detection part, first we used a color threshold method, we assumed that the court lines were predominantly white and it creates a mask based on this color range, isolating pixels within the defined white range.

After this to detect the edges from each frame the pipeline involves the following image processing steps:

- **Grayscale Conversion:** The image is converted to grayscale to enable the application of Harris corner detection, which relies on intensity gradients.
- **Harris Corner Detection:** This method identifies points with significant intensity changes in all directions, which are often found at line intersections, making it ideal for detecting tennis court corners.

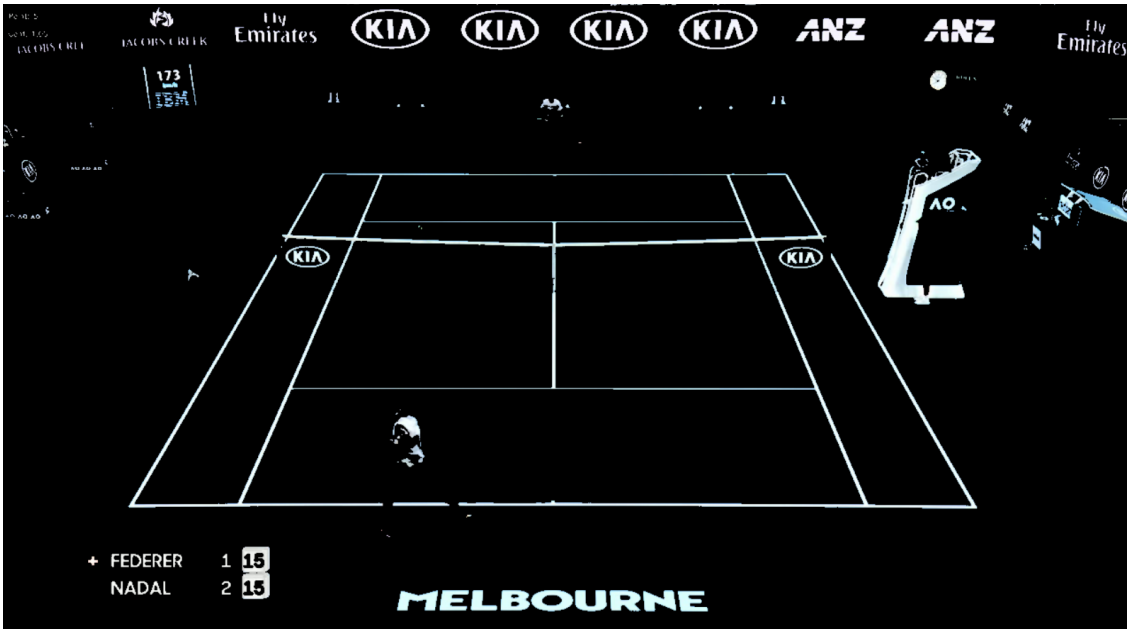

Figure 5:

- **Normalization:** The corner response values are normalized to a specific range (typically 0-255) to prepare them for thresholding.
- **Otsu's Thresholding:** Otsu's method is applied to automatically determine the optimal threshold value, separating strong corners from weaker ones.
- **Dilation:** A dilation operation is performed to thicken the detected corner regions, making them more robust for contour detection.
- **Contour Detection:** Contour detection is used to identify closed shapes, which correspond to the corners of the tennis court.

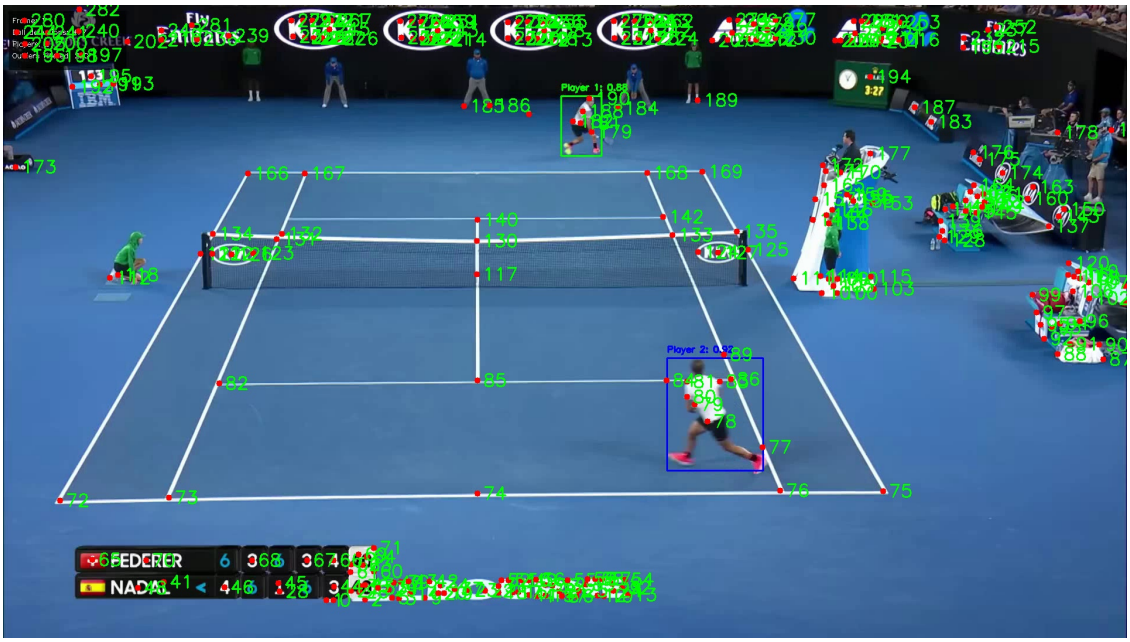

Figure 6:

By applying this pipeline we got more points that the four points we wanted to determine the dimensions of the singles court. For that we needed to classifying the detected points in the first frame, we successfully identified the specific points corresponding to the singles court.

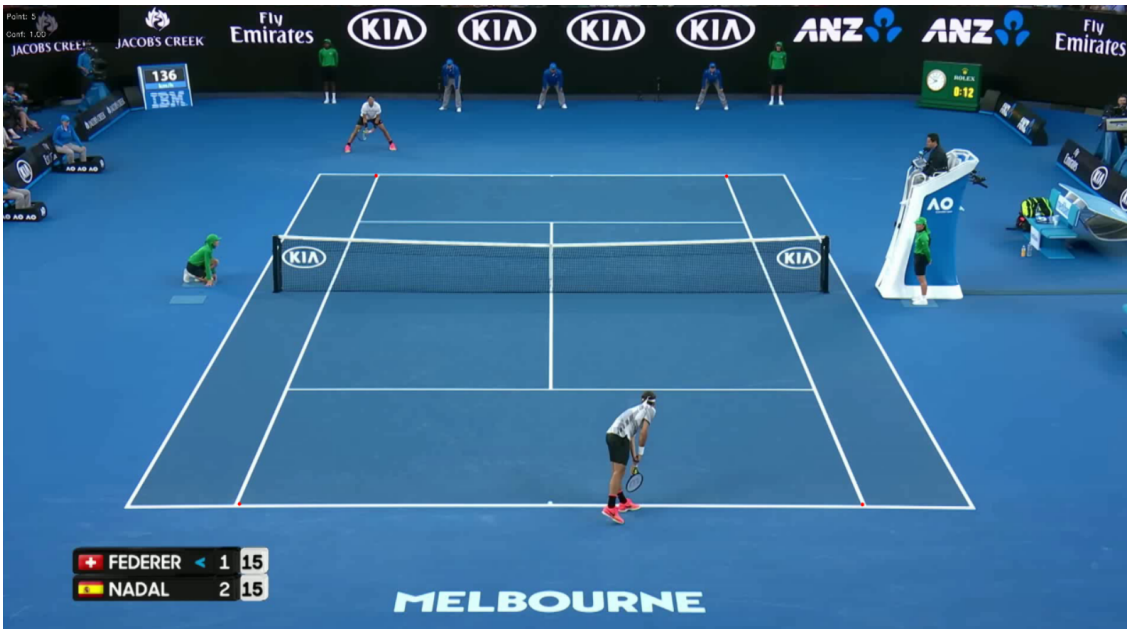

Figure 7:

From this starting point, we dynamically updated the corner positions by interpolating through the subsequent frames, detecting the edges with the same pipeline and using a nearest-neighbor approach to match the previous frame's coordinates with newly detected ones. This method ensures that the system adapts to variations in court perspective or slight camera movements while maintaining consistency.

3.6 Homography

For the next part we identified the corresponding points in the diagram image to calculate a homography matrix to map the court's real-world coordinates onto the minimap diagram. This transformation allows the minimap to accurately reflect the players' positions as they move across the court. To handle cases where homography calculation fails—for example, due to incomplete corner detection—we implemented fallback mechanisms that use previously calculated coordinates or resize the minimap to maintain usability.

In summary, this part of the project brings together court tracking and player positioning. By using an homography, we can precisely map players' positions onto a minimap. When extended to the ball's position, this approach enables accurate tracking of where each ball lands, helping determine whether it was in or out of bounds. Such mapping techniques can be used in tennis analysis to evaluate ball placement and assist in decision-making. Additionally, for players, it provides a powerful tool to study movement patterns, offering valuable insights that can enhance coaching strategies and performance improvement.

4 Results

4.1 Challenges and Edge Case

One notable edge case in the tennis analysis involves the detection of four corner points on the court when a player, dressed in white, moves in front of one of the corners. In this scenario, the model erroneously detects the corner points on the player's shirt, which causes the interpolation between the four points to malfunction. This issue arises when the model mistakenly interprets one of the four points as one of the edges detected in the player's shirt, disrupting the accurate tracking of the boundaries as seen between Figure 8 and Figure 9.

A potential solution to mitigate this problem could involve incorporating a distance-based validation system. By monitoring the relative distances between the detected corner points in the first frame, the model could identify when

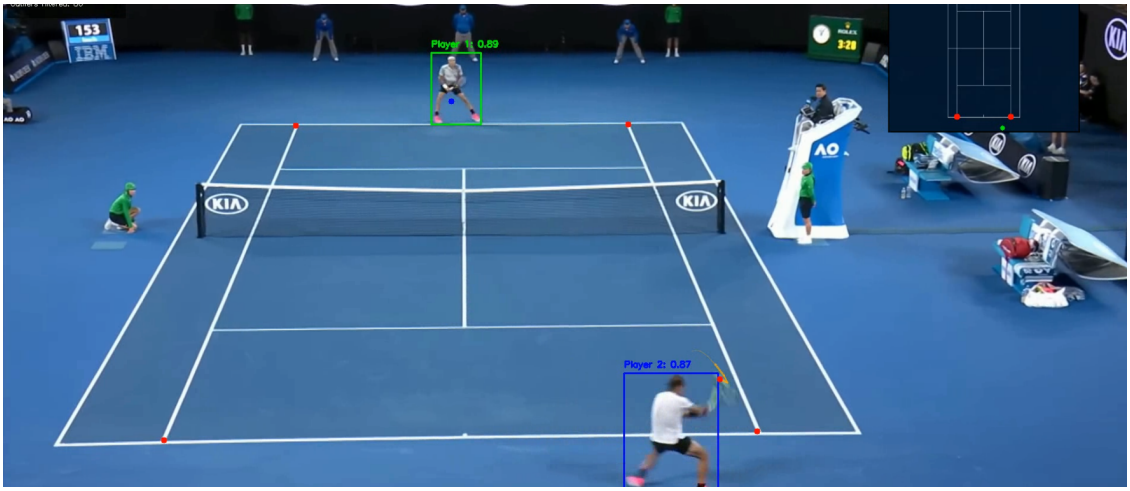

Figure 8:

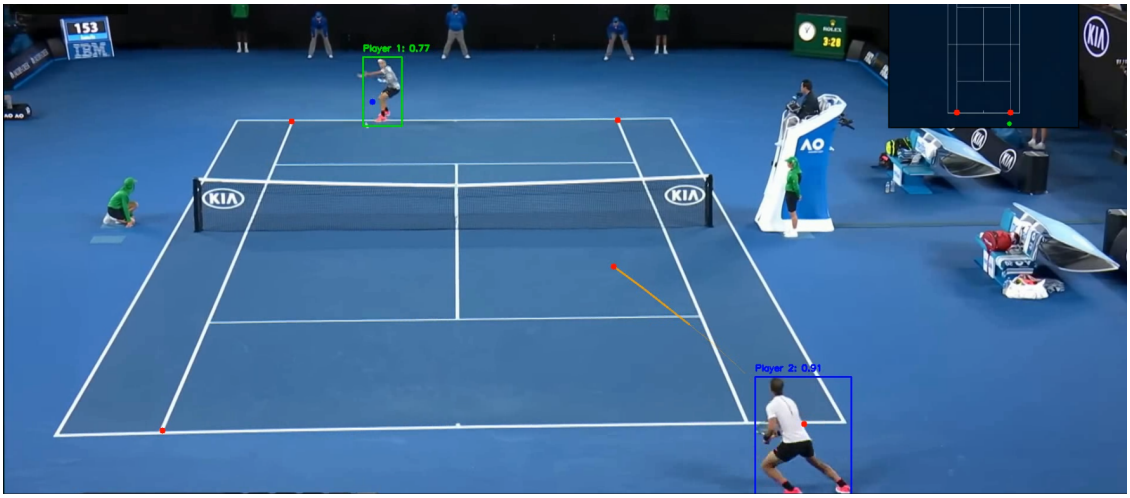

Figure 9:

the distance between points exceeds a certain threshold. In such cases, the system could either revert to previously detected points from earlier frames or attempt to identify alternative corner points other than the one detected in the players shirt. This could help improve the robustness of the corner point interpolation.

4.2 Limitations and Future Work

While our system provides an effective solution for automated tennis match analysis, there are several areas where improvements can be made to enhance its accuracy, robustness, and general applicability.

- **Improving Frame Preprocessing and Camera Change Detection:** One of the current limitations of the system is the reliance on a static threshold for frame preprocessing to detect significant changes between consecutive frames. This approach may not always accurately reflect when the camera angle changes significantly, as variations in lighting, player movement, and environmental factors can cause false positives or fail to detect actual changes in the scene. To address this, we propose the development of a more sophisticated model to calculate the threshold dynamically. By analyzing the difference in keyframe content, this model could identify frames that are visually similar and detect when a camera angle shift or scene change occurs, thus improving the accuracy of point segmentation and player tracking.

  An approach would be using techniques such as deep learning-based methods (e.g., using an autoencoder to learn scene representations) to measure the degree of change between frames. This would allow the system to

adapt to different video quality and camera conditions, ensuring more accurate segmentation of points and smoother transitions between scenes. This would also help minimize errors in identifying the beginning and end of points, as well as enhance the detection of outlier frames when the camera changes drastically.

- **Creating a Synthetic Dataset for Robust YOLO Training:** Another limitation of the current system is the reliance on a limited dataset for training the YOLO ball detection model. While our current dataset provides sufficient coverage for player detection, the ball detection still requires further processing, it is not diverse enough to account for all variations in lighting conditions, camera angles, and player behavior that can be encountered in real-world tennis matches. To make the object detection more robust, we propose creating a synthetic dataset for training YOLO on a wider range of possible scenarios.

  By using computer graphics software or simulation tools, we can generate synthetic tennis match videos with varied camera angles, lighting conditions, player positions, and court configurations. This would allow us to create a more diverse and extensive training dataset without the limitations of manually annotated real-world data. Synthetic data could also help address corner cases where player detection or ball tracking fails due to obscured views or unusual camera movements. Moreover, training on synthetic data would improve the system's ability to generalize, enabling it to perform well across different tournaments, camera setups, and match environments.

  Combining this synthetic dataset with real-world footage can help overcome the current system's challenges related to detection accuracy and generalization. Additionally, this dataset could be used to further train models for ball trajectory estimation, helping to refine its prediction of ball behavior in different contexts.

- **Edge Detection and Classification:** Currently, the edge detection process requires classification of the points and manually select the four points needed. To address this, a more robust edge detection method is needed that can directly identify the points of interest without requiring the classification and then manually selecting the four points. One potential solution is to integrate machine learning-based approaches, such as keypoint detection models or using deep learning methods such as *Convolutional Neural Networks (CNNs)* or *Transformers* could allow the system to learn which court edges and points are relevant, reducing the need for manual intervention and making the detection process more efficient and accurate.

- **Homography Improvement:** The current homography implementation relies on only four points for transforming the perspective of the court. This approach performs very well; however, to further improve the robustness of the homography, we propose using more than four points for the transformation.

- **Ball Detection and Bounce Projection:** A current limitation in the ball detection process is that it does not yet identify ball bounces or project their positions onto the court. Detecting bounces would provide valuable insights into ball trajectories and allow the system to determine whether a ball was in or out of play. To address this, an extension of the ball detection pipeline can be implemented to identify bounce events using temporal analysis of the ball's movement (e.g., detecting sudden changes in ball speed or direction). Additionally, by projecting the detected bounce locations onto the court using the homography, the system can visually map ball landings and validate whether they are inside or outside the court boundaries. This would enable a deeper level of analysis for line calls, which is particularly useful for coaching, umpiring, and match analysis. Potential methods for detecting bounces include using object trajectory analysis, tracking ball velocity changes, and applying event detection algorithms based on time-series data.

5 Conclusion

In conclusion, this project successfully developed an automated system for tennis match analysis, using computer vision and deep learning techniques. By implementing a comprehensive pipeline that starts with downloading a video from YouTube and progresses through scene detection, player and ball tracking, and court mapping, the system is able to provide valuable insights into individual points, player movements, and ball trajectories. Despite its performance, there are several areas for improvement, such as refining edge detection methods, expanding the homography transformation to incorporate more than four points, and exploring the detection and projection of ball bounces for better analysis of in/out decisions.

References

Dong Seong Kim and Hyun Seok Kim. Shot detection and classification for structuring tennis video. *Multimedia Tools and Applications*, 30:247–263, 2006. doi: 10.1007/s11042-006-0031-5. URL `https://link.springer.com/article/10.1007/s11042-006-0031-5`.

Takeshi Ogawa, Naoki Ikeya, and Koji Yamada. Detecting tennis court lines for sport video categorization. In *Advances in Image and Video Technology*, pages 311–320. Springer, 2013. doi: 10.1007/978-3-642-34707-8_31. URL https://link.springer.com/chapter/10.1007/978-3-642-34707-8_31.