University of Illinois at Chicago

Sport analysis on a tennis match

Davide Ettori, Antonio Marusic, Francesco Santambrogio

Instructor: Prof. Wei Tang

Course: Computer Vision CS415

Date: December 2024

Contents

1 Abstract

This project introduces an advanced analytics software for tennis matches, designed to automate the analysis of player movements, ball trajectories, and game dynamics. By leveraging computer vision, the system detects and tracks objects in real time, transforming standard video perspectives into a bird's-eye view of the court. This feature enables the generation of heatmaps, providing actionable insights into player positioning and movement patterns. The software aims to support players and coaches by streamlining game analysis, reducing subjectivity, and enhancing strategy development.

2 Course topics and areas of further study

2.1 Personal Views on Computer Vision and Its Applications

Computer vision is a fascinating field of study that bridges the gap between human visual perception and computational understanding. It enables machines to interpret and analyze visual data, making it a cornerstone of modern artificial intelligence. Its applications are transformative across various domains, such as healthcare, where it powers diagnostic tools using medical imaging, autonomous vehicles, which rely on visual data for navigation and decision-making, and entertainment, where it enhances experiences through augmented reality and computer graphics.

The three of us, find the field immensely impactful because of the vast array of real-world problems that were once considered exclusive to human capabilities. With newer and faster machines we are able to process huge amount of data, making more and more computer vision tasks feasible also with real time constraints.

2.2 What we learned in the Course

In this computer vision course we covered several foundational topics that gave us a solid understanding of how machines interpret visual information:

- **Filters:** We explored the fundamental concept of filters and convolution. This tech-

nique are at the basis of image blurring, sharpening and denoising, and convolution is a technique that revolutionised the neural networks for image analysis.

- **Edge and Corner Detection:** We learned how algorithms like the Canny edge detector and Harris corner detector work to identify significant structural elements in images. These methods are critical in identifying objects, boundaries, and points of interest, forming the basis for higher-level tasks.

- **Image segmentation** We explored how computers can identify objects by working under the assumption that similar pixels are likely to belong to the same object. To group these similar pixels, we studied clustering algorithms, which categorize pixels based on their shared features. Additionally, we examined the 2D Gaussian histogram approach, which attempts to locate specific objects in an image by leveraging a model trained on previous instances of those objects. However, we recognized the limitations of this approach—it primarily focuses on color information while neglecting critical factors such as object shapes or the spatial arrangement of pixels.

- **Feature Detection and Matching:** We explored feature detection algorithms, such as SIFT (Scale-Invariant Feature Transform) and how these features can be matched across images for tasks like image stitching and object tracking.

- **Image alignment** Image alignment involves warping one image so that corresponding feature points in two images of the same scene are aligned at the same location.

- **Image transformation** Common transformations include affine (scaling, rotation, translation, and shearing) and projective (homography) transformations.

  We can estimate unknown transformation parameters are determined by solving a linear system derived from point correspondences using the least squares method.

- **Object recognition** Object recognition is the process of identifying specific objects in an image and classifying them into predefined categories. Old approaches used

SVMs, while newer approaches use deep neural networks.

2.3 Next topics to explore

Going forward in our education and work experience we would like to explore more advanced topics in computer vision such as:

- **Deep Learning in Computer Vision:** study state-of-the-art neural network architectures, such as transformers in vision (e.g., Vision Transformers), and how they outperform traditional CNNs in certain tasks.
- **3D Computer Vision** Topics like Structure from Motion (SfM), SLAM (Simultaneous Localization and Mapping), and LiDAR-based vision intrigue us. These methods are pivotal in autonomous systems and AR/VR applications.
- **Real-Time Vision Systems:** As autonomous driving is becoming a reality an important field of work is the acceleration of the computer vision pipeline. We would like to explore state of the art algorithms and develop clever way to accelerate them.
- **Multimodal Systems:** Vision is one of the way we perceive the world. It would be interesting also to integrate computer vision tasks with other modalities such as language processing and audio.
- **Ethics and Fairness in Vision Systems:** As computer vision becomes more prevalent, we are interested in studying its ethical implications, including biases in datasets and models, and how to design systems that are fair and inclusive.

3 Project description

With this project, we developed an analytics software tailored for tennis matches, enabling detailed analysis of player movements, ball trajectories, and game dynamics.

3.1 Goals

The software leverages advanced computer vision techniques to process and analyze the movements of both the ball and the players during a tennis match. Using these techniques, the system can detect and track objects in real time, providing immediate visual feedback on the game. One of the standout features is the ability to transform the standard camera perspective into a top-down, bird's-eye view of the court.

This bird's-eye view serves as the foundation for generating heatmaps that visualize areas of movement on the court. These heatmaps offer valuable insights for players and coaches, highlighting the most and least active regions. By identifying areas where a player is less likely to be positioned, the software aids in strategy development, enabling users to analyze their own gameplay or that of their opponents with greater precision and depth.

3.2 Problem statement

Tennis match analysis is traditionally a manual and time-intensive process, often limited by human accuracy and consistency. Coaches and analysts rely heavily on their observations to evaluate player performance and strategies, which can lead to subjective conclusions. We aim to produce an application that can support the analyst, players and coaches in the process of analyzing the game.

Key challenges addressed by this project include:

1. Accurately detecting and tracking fast-moving objects like the ball, even under challenging conditions such as motion blur.

2. Differentiating between players and assigning identities to their movements on the court.

3. Creating a precise mapping of the court lines and corners to project it to a new point of view from the top.

4. Mapping players and ball to the bird's eye view.

Used dataset We built the whole application around a video of a match between the two famous tennis players, Novak Djokovic and Jannnik Sinner, that is recorded from the bottom of the field, on the side where Sinner plays.

3.3 Implemented features

We implemented the following features.

- Object detection and classification of the two players and the tennis ball.
- Bird eye (from the top) view of the field.
- Tracking of the ball and the players on the bird's eye view
- Heat map of the positions of the players.
- Show the trajectory that the object followed (is not a prediction, is just the points in the video where the object was detected)
- Show the box around a tracked object only for a selected object.

4 Design

4.1 Assumptions

- We assume that we are analyzing a game of tennis with a point of view (POV) from the back.
- We assume that the ball is yellow, like in the tennis standard.
- We assume that the players don't change sides of the pitch.

4.2 Inner working

The application is based on YOLO, a pre-trained convolutional model that is able to perform object detection.

We implemented a main loop that, in the first iteration, finds the homography used to project the field onto a new plane that represents the bird's eye view. The loop then goes on and is responsible for the object detection task and for mapping frame by frame the recognized objects onto the bird's eye view.

In addition, we have two callback functions: when the user moves the mouse cursor on the image and when he taps the keyboard.

First iteration: building birds' eye view of the field During the first iteration, the system builds a top-down view (or bird's eye view) of the pitch. We first identify the four corners of the pitch and then find the homography matrix to map the four points onto the bird's eye pitch.

In more detail, the procedure involves:

1. The input frame is converted to grayscale.
2. Gaussian blur reduces noise and smooths the image.
3. The Canny edge detector identifies edges in the blurred image.
4. Straight lines are detected in the edge-detected image using the probabilistic Hough transform. In figure 1 we see the lines found.
5. The detected lines are separated into horizontal and vertical. To do so we use hard-coded angles (0 for horizontal and ¿65 for vertical) lines, using the assumption that the POV is from the back.
6. Intersections of horizontal and vertical lines are computed. The intersections are marked with blue dots in figure 2.
7. Intersections near the image boundaries are filtered out.
8. The intersection points are clustered using K-means clustering to identify potential court corners or key points. The output of the k-means algorithm is showed in figure 3

9. From the clustered points, we identify the four corners of the court (e.g., top-left, top-right, bottom-left, bottom-right).

10. A homography matrix is computed to map the selected court corners to a predefined rectangle.

11. The frame is warped using the homography matrix to generate a rectified, top-down view of the court.

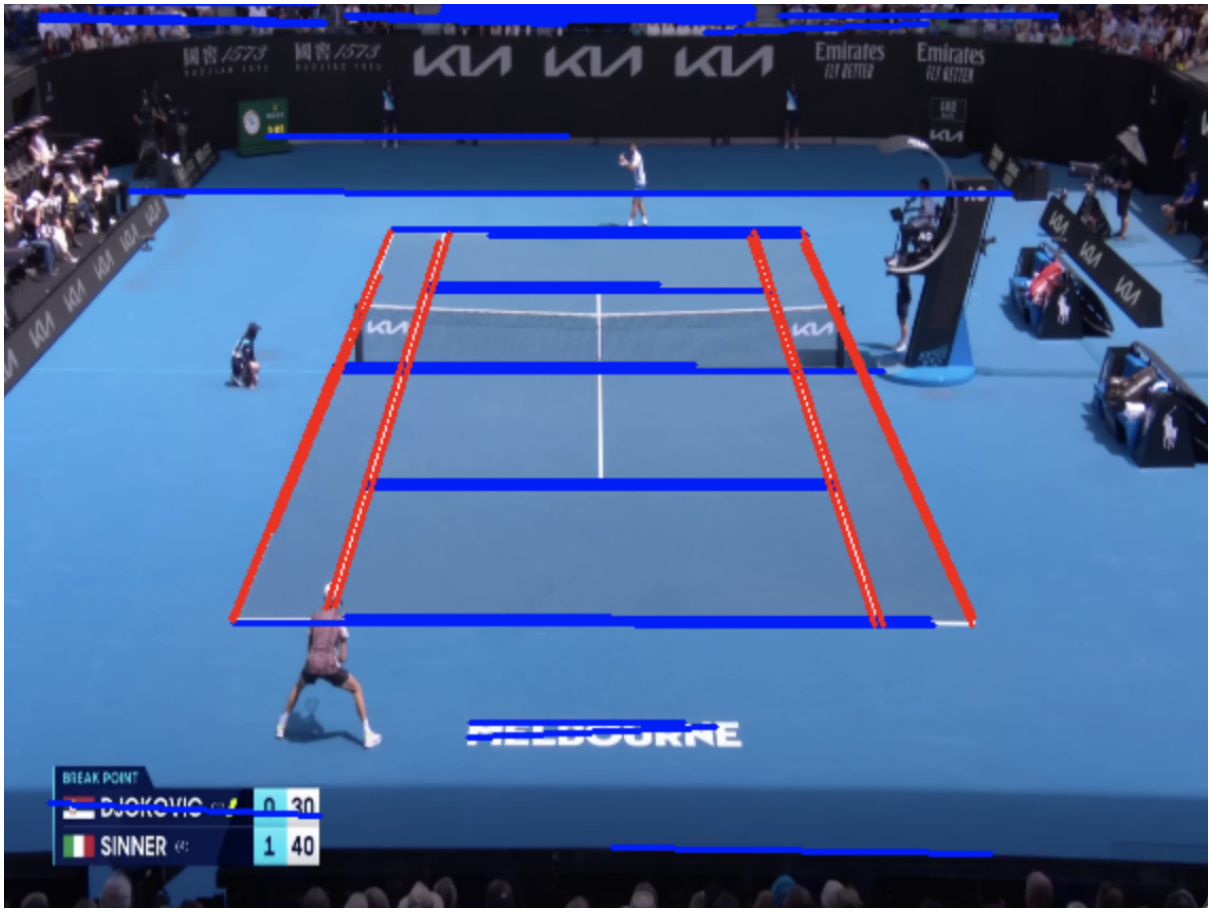

Figure 1: Lines detected to do the rectification.

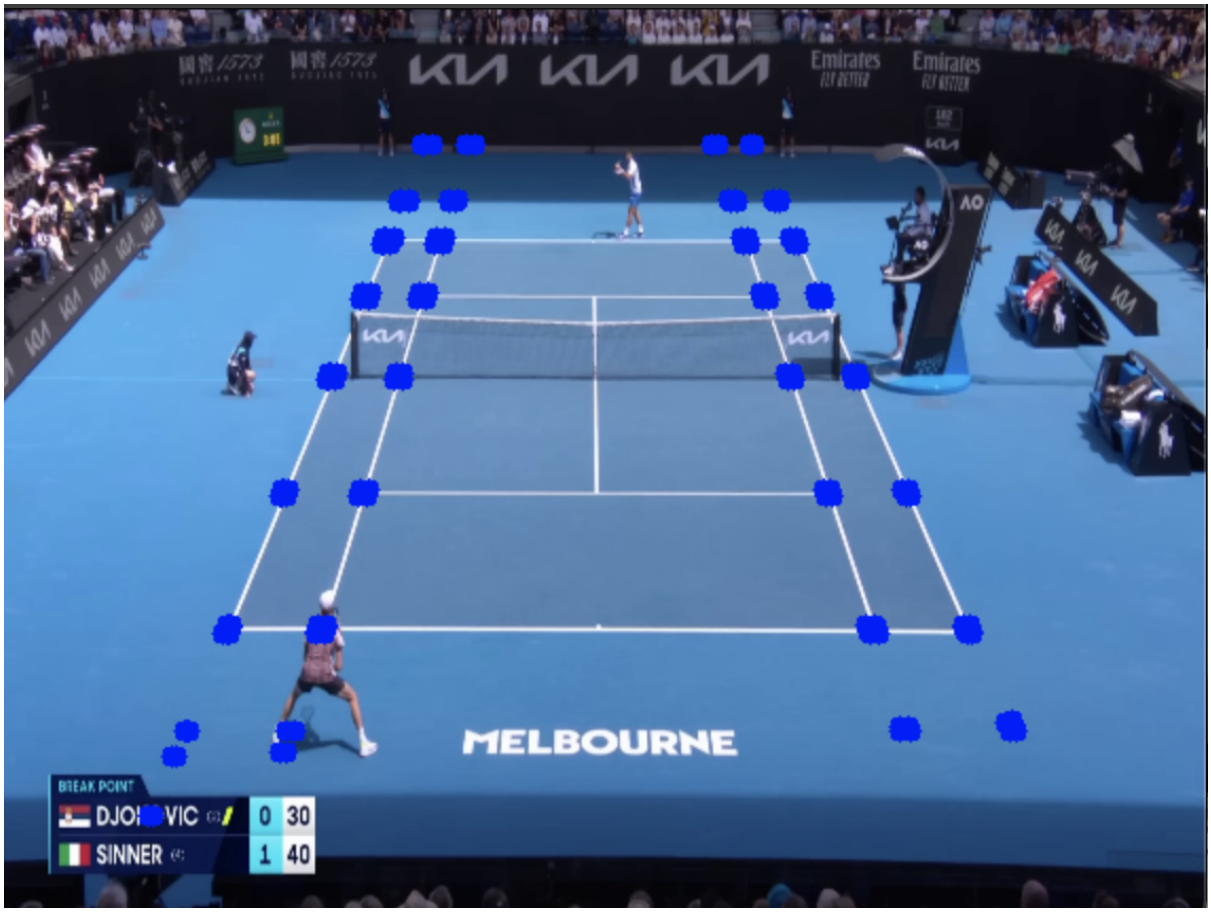

Figure 2: Intersection of the detected lines.

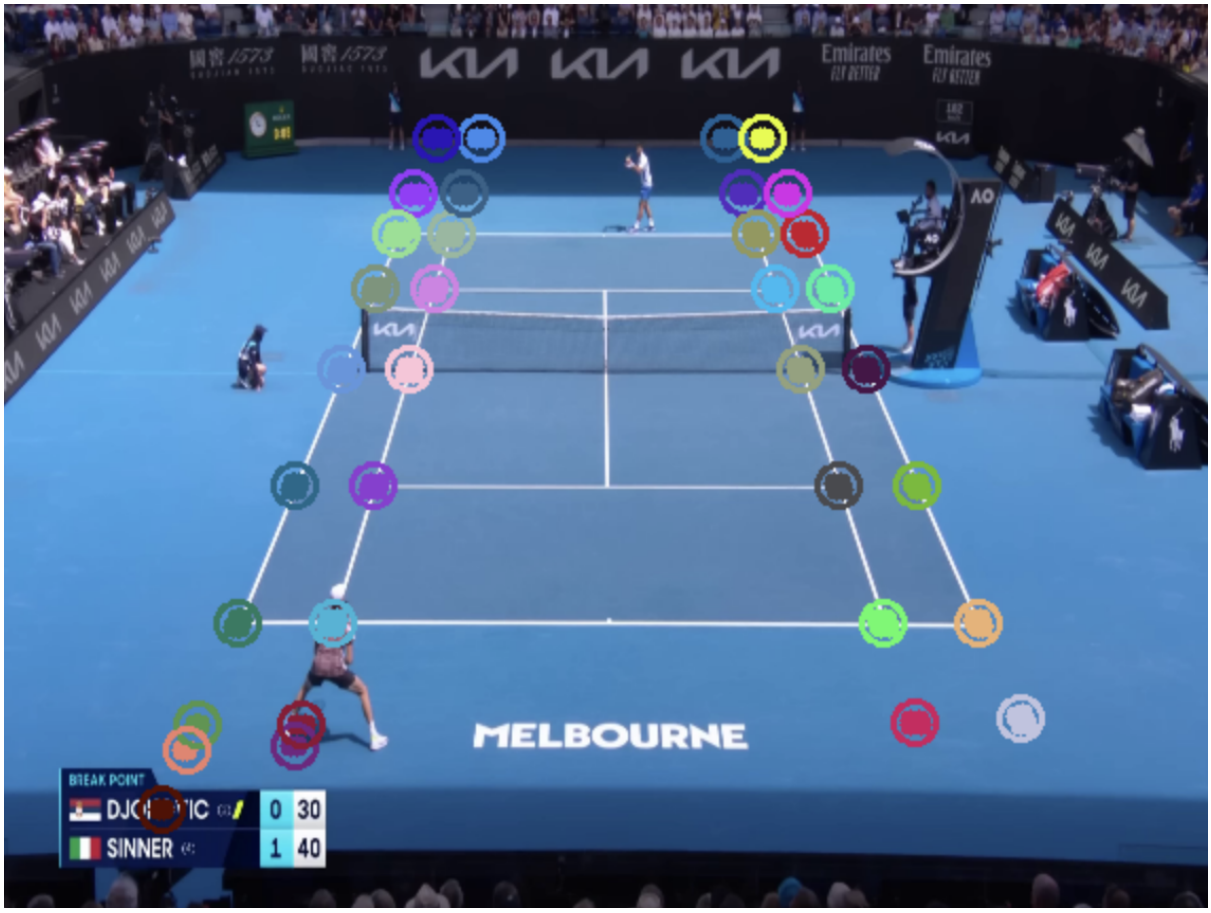

Figure 3: K-means output.

During the video Every frame of the video is rescaled to a smaller dimension for improved performance when using YOLO.

Marking the ball The YOLO model cannot reliably detect the tennis ball in the raw video frames, probably because it is too small. To address this issue, we preprocess each frame by marking the tennis ball with a yellow overlay, which YOLO then recognizes as the tennis ball.

To locate the ball, we use color thresholding, a technique that identifies pixels within a specific color range defined by lower and upper bounds in the RGB color space. To further enhance detection accuracy, we restrict the search to a predefined region of interest (ROI) corresponding to the court area in the video. This ROI is currently extracted as a specific section of the image, ensuring that yellow regions outside the court are excluded from consideration. Each frame is then preprocessed, and once the ball is located, it is

surrounded by a yellow circle. At this point, it becomes much more distinguishable, and YOLO is able to detect it in the newly processed frame.

We also experimented with a 2D Gaussian distribution to locate the ball, but this approach proved less effective. It struggled in scenarios where the camera's shutter speed was insufficient to capture the ball clearly, resulting in significant color distortion that hindered detection. The same problem persists with the color thresholding approach; in fact, when the ball reaches the fastest speeds, usually when flying in the vicinity of the net, the color of the ball sometimes goes out of our predefined range and thus is not detected.

Finding the two players and the ball After the preprocessing described in the previous paragraph, we used the YOLO model to detect the two players and the ball. The model can just detect people, so we use the concept of ROI again to classify the player. In fact, we assume that each player stays on its side of the field if an object is classified as a person and is inside the ROI that is classified either as Djokovic or Sinner.

Showing the ball and players on the bird eye view For each frame we visualize This code visualizes the positions of the tennis ball and the two players (Sinner and Djokovic) on a rectified court view using a homography transformation to map their detected coordinates to the new perspective.

Heatmap We generate a heatmap for each of the three key entities: the ball and the two players. Each heatmap visually represents the distribution of positions occupied by the respective entity on the court, providing insights into their movement patterns and areas of activity during the game.

Commands The user can send commands to the application using the keyboard and the mouse.

1. **press s** the key "s" is used to toggle the display of the trajectory of the selected object. Does nothing if no object is selected.

2. **press f** To select an object the user has to hover on the box of the object with the mouse and press "f".

5 Results

Our application effectively detects and tracks both players and the ball within video frames. Additionally, it performs a perspective transformation of the field, mapping it to a bird's-eye view. This enables precise tracking of players and the ball from a transformed perspective, enhancing analysis and visualization capabilities.

Detection In figure 4 we see how the application can detect the two players and the ball. Also, the referee and the ball boy are detected as people. The tennis ball is enlarged with a yellow circle.

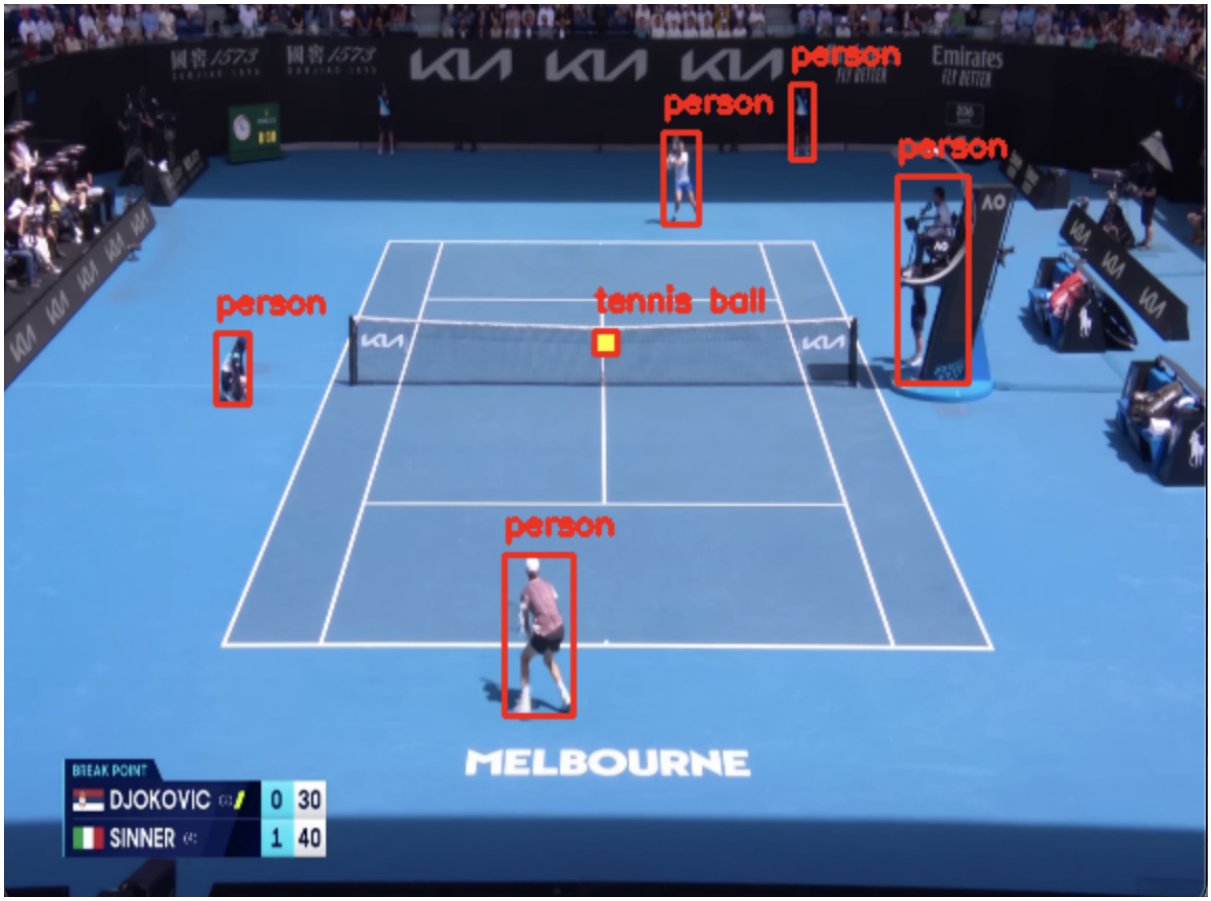

Figure 4: Image of the detected objects.

Shortcomings Unfortunately, we could not detect the ball in every frame due to the fast movement speed and consequent shutter effects in the video. Further work could assess this problem, maybe training a neural network that can detect the tennis ball also in this case.

Tracking In Image 5, we see the user selecting Sinner for tracking and activating the trace feature. The application highlights the player with a bounding box and displays the trajectory of their movement, showing a history of tracked points for better visualization and analysis.

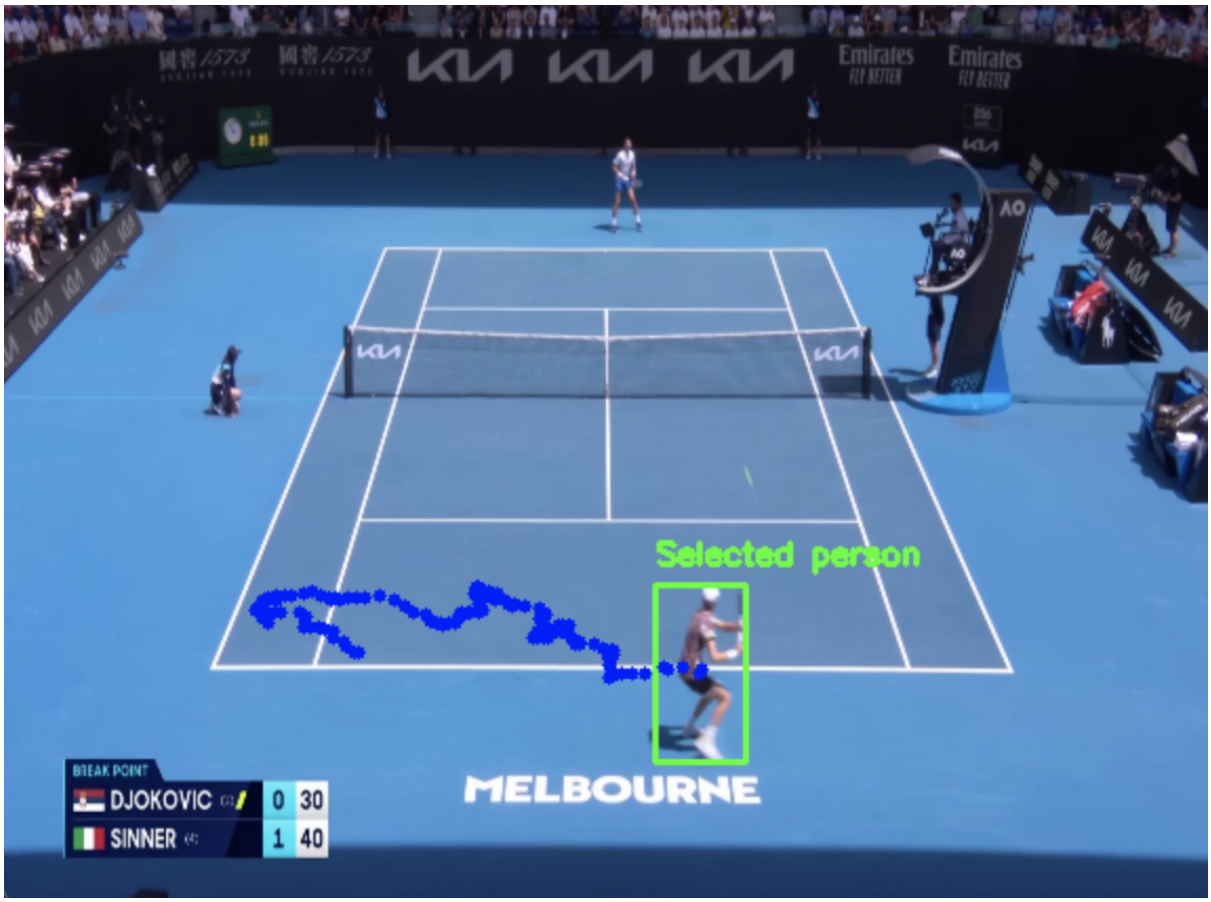

Figure 5: Image of the tracked objects.

Bird eye view In the following figures, we can see the bird's eye view of the scene. The application can detect the lines and the corners.

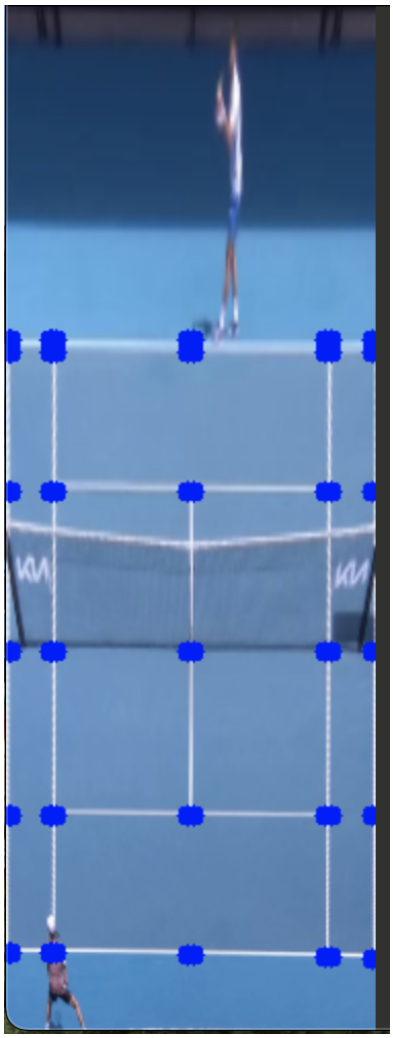

Figure 6: Detected corners.

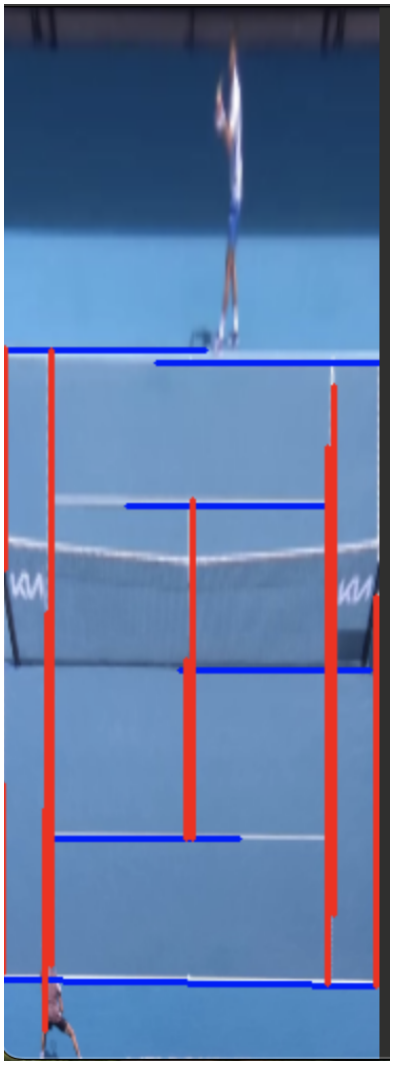

Figure 7: Detected lines.

Figure 8: Detected corners.

Bird eye view In image 9, we can see how the application effectively tracks the ball and the players also on the rectified field. The two players are identified and tracked using different colors.

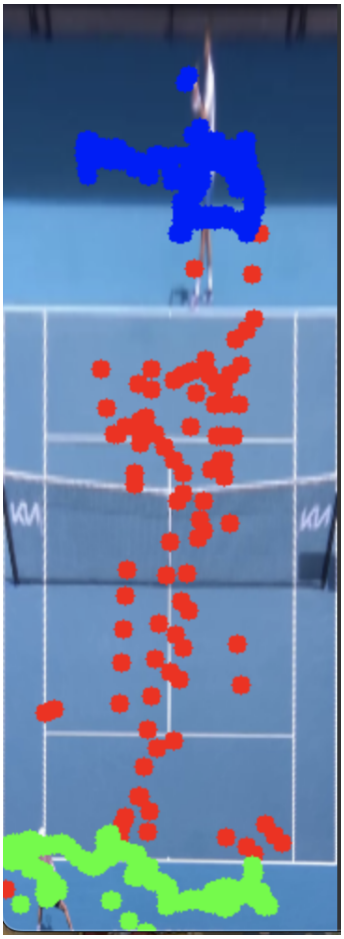

Figure 9: Trajectories on the rectified field.

Heatmaps In figure 10, we see how the heatmaps are printed at the end of the video. From the heatmaps, we can infer useful data, for example the fact that Djokovic played more on the back of the court, while Sinner more on the front.

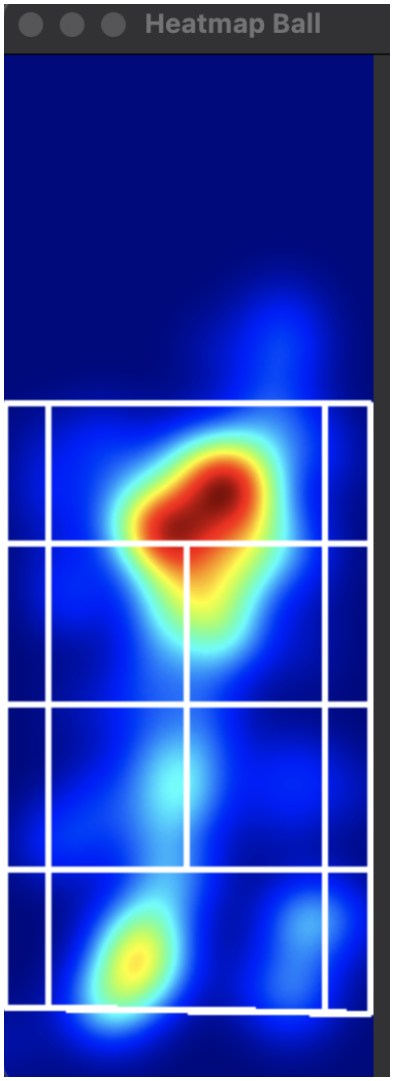

Figure 10: Heat map of the ball.

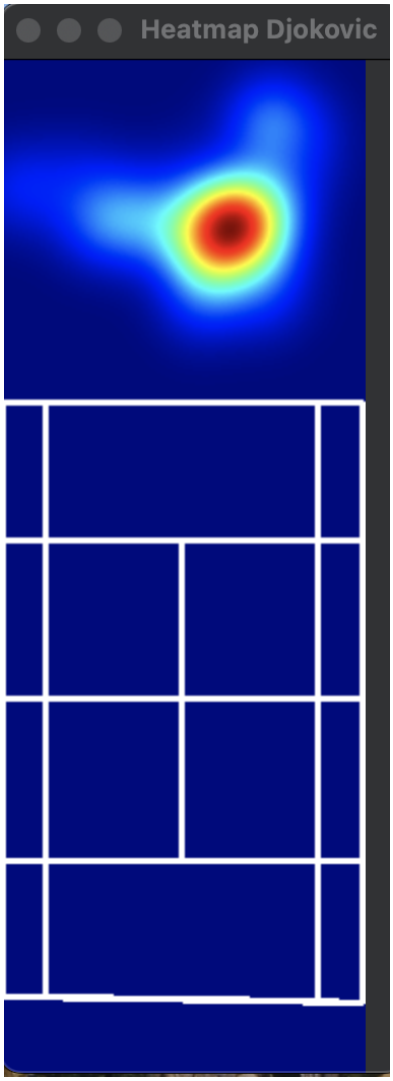

Figure 11: Heat map of Djokovic.

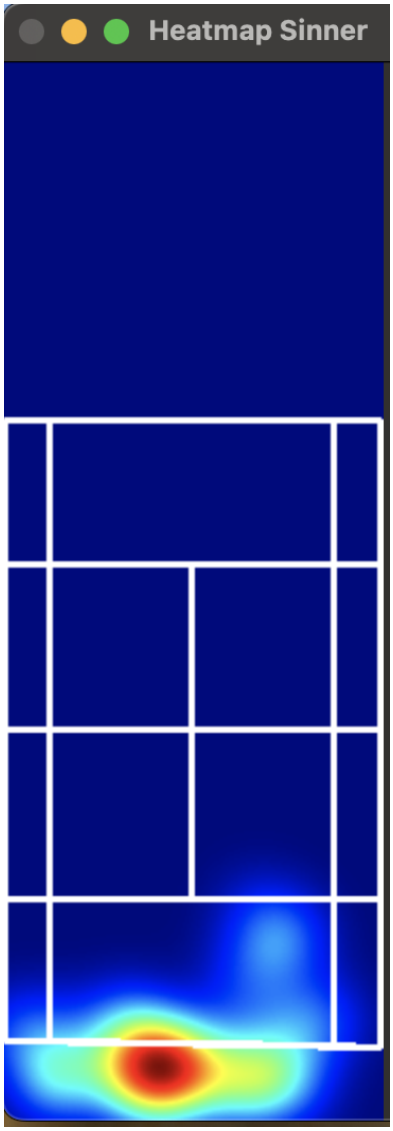

Figure 12: Heat map of Sinner.

6 Resources

To carry out this project, we utilized the following resources:

6.1 YOLO

For object detection and tracking, we employed a state-of-the-art pre-trained model called YOLO. This model excels in classifying and tracking objects in images and achieves real-time performance on videos. More information about YOLO can be found in its official

documentation: YOLO Documentation.

6.2 OpenCV

To process images, perform edge detection, corner detection, compute homographies, and apply blurring, we used the OpenCV library. OpenCV provided efficient tools for implementing various image processing tasks. Additionally, this semester's course slides and assignments were invaluable for understanding and effectively utilizing this library. For more details about OpenCV, refer to its official website: OpenCV Website.

6.3 Course Slides

Slides from the Computer Vision course (CS 415) taught by Professor Wei Tang. They were fundamental in understanding and applying various image analysis techniques, in particular: Hough Transform, Canny Edge Detector, and Image Projection. In general, they also helped us understand which problems to tackle and which solutions to implement for our final project.

7 Future Work

To further enhance the functionality and performance of the application, the following improvements are proposed:

- Achieve real-time performance by leveraging GPU acceleration for computationally intensive tasks.
- Dynamically identify the region of interest (ROI) for ball detection, possibly using data derived from the bird's-eye view transformation of the field.
- Enhance ball detection by integrating a neural network specifically trained to locate the ball in each frame.

- Explore the use of multithreading to optimize performance, such as periodically recalculating the homography in case of slight camera orientation changes or processing frames in batches to improve frame rate.
- Extract additional information, such as the speed of the ball and players, to provide deeper insights into game dynamics.

8 Contributions

The contributions of team members to the project are summarized in the table below:

Member	**Contribution**
Davide Ettori	Implemented player and ball detection, developed the tracking algorithm.
Antonio Marusic	Integrated YOLO in the pipeline, wrote the report.
Francesco Santambrogio	Performed perspective transformation to create the bird's-eye view, worked on homography and edge detection using OpenCV.

Table 1: Summary of contributions by team members.