Foosball Tracker

Roberto De Paoli

Dipartimento di Ingegneria e Scienze dell'Informazione
Università di Trento

Project for the course "Signal Image and Video"
Professor: Prof. Andrea Rosani

January 2025

Abstract

The report discusses the development of the Foosball Tracker system, which applies the image processing techniques learned in the "Signal Image and Videos" course. The primary goal is to create a tracking system for players and the ball on a foosball table. The system addresses challenges like occlusions, varying lighting conditions, and motion blur, showcasing the practical application of computer vision techniques.

Contents

1 Introduction

The Foosball Tracker project aims to design and implement an innovative system for tracking and analyzing table soccer games. Table soccer, also known as foosball, is a widely played game that simulates soccer on a small table. It is often enjoyed both recreationally and in competitive settings. However, despite its popularity, there is a lack of automated systems that can accurately track player and ball movements, analyze game dynamics, and provide real-time feedback.

The primary motivation behind this project is to explore and test the potential of image processing techniques for automating the tracking of foosball games. By using advanced computer vision algorithms, the system aims to detect and follow the movements of players and the ball, correct perspective distortions in the captured images, and provide useful analytics to improve player performance and strategy. The focus is on experimenting with and applying techniques learned in the "Signal Image and Videos" course, to evaluate their effectiveness in real-world scenarios.

This project leverages several techniques learned in the course, including color filters, median blur, contour detection, and the Hough Transform, to address challenges such as occlusions, motion blur, and noise in the video data. Additionally, the Kalman filter is employed to predict and smooth the trajectories of moving objects. These tools will be used to isolate relevant objects, reduce image noise, and detect geometric shapes like the ball.

The goal of this project is to provide a fully functional system capable of automatically tracking both the players and the ball during a foosball game, as well as correcting the perspective of the camera feed to ensure accurate tracking. The system's performance will be evaluated in terms of its accuracy and reliability in real-world conditions, with potential applications in training and competitive play.

The structure of the report is as follows: Section 2 covers the system design, including hardware and software components. Section 3 describes the perspective correction methodology, while Sections 4 and 5 focus on the players and ball detection techniques. Finally, Section 6 presents the conclusions and future work, outlining potential improvements and extensions to the system.

2 Design

The design of the Foosball Tracker system focuses on providing accurate, real-time tracking of both players and the ball during a foosball game. The video feed is captured using a smartphone camera positioned directly above the foosball table, offering a top-down view that simplifies the tracking process. This video feed is processed through three primary software modules: perspective correction, player tracking, and ball tracking. The system is implemented in Python, utilizing libraries such as "OpenCV" for image processing and "NumPy" for numerical operations.

2.1 Code Structure

The code is organized into a modular structure, with each module dedicated to a specific task in the tracking pipeline. This modular design improves maintainability and allows for easy updates or extensions to the system. Below is the general structure of the codebase:

```
foosball-tracker
├── ball
│   ├── contours.py
│   ├── hough.py
│   ├── kalman.py
│   ├── tracker.py
│   └── trajectory.py
├── players
│   ├── biliardino.py
│   ├── tracker.py
│   └── zone.py
├── utils
│   ├── edges.py
│   ├── filters.py
│   ├── forces.py
│   ├── stabilizer.py
│   └── stream.py
├── .gitignore
├── config.json
├── main.py
├── README.md
└── requirements.txt
```

Each module is responsible for a specific aspect of the tracking process:

- **Perspective Correction Module:** This module corrects the perspective distortion introduced by the camera's top-down view, using geometric transformations like homography.
- **Players Tracking Module:** It detects and tracks the players on the foosball table using contour detection and color filtering. The positions of the players are continuously updated.
- **Ball Tracking Module:** This module detects and tracks the ball by processing each frame through two parallel approaches: the Hough Transform, which detects its circular shape, and the findContours function, which identifies its position based on object boundaries. A combination of filters is applied beforehand to isolate the ball, and the Kalman filter is used to predict and smooth its trajectory, ensuring accurate and robust tracking.

3 Perspective Correction

To accurately track the players and the ball on the foosball table, it is essential to correct the perspective distortion caused by the camera's angle. Since the camera is not stably positioned above the table and may be angled, the captured image can exhibit distorted shapes and object positions. This distortion makes it difficult for detection algorithms to correctly identify the shapes and positions of objects on the table. Perspective correction is necessary to ensure the image corresponds to a top-down view, which is crucial for accurate object detection and tracking.

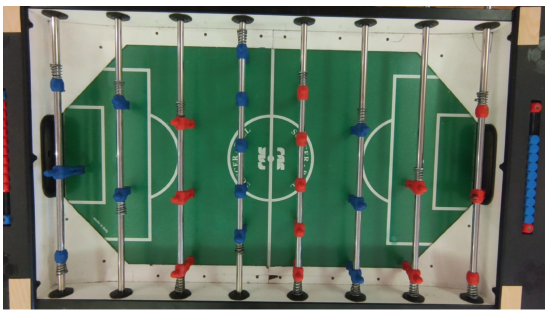
3.1 Approach to Perspective Correction

To address perspective distortion, we apply a perspective transformation to "flatten" the image using a homography matrix. This matrix is computed based on selected reference points from the image and a desired top-down view, which are identified by detecting the corners of the foosball table. To make corner detection more reliable, we place brown rectangles at the four corners of the table. These rectangles serve as clear visual markers for the system, ensuring the correct corner positions are identified, even if the image is slightly distorted or obscured. Once the corners are detected, we compute the homography matrix based on their coordinates in the image and map them to the desired top-down view. Using OpenCV, we then apply the homography matrix to the captured frame, resulting in a corrected image where the objects on the table are displayed in their true relative positions.

3.2 Results of Perspective Correction

After applying the perspective transformation, the objects on the table appear in their accurate positions, significantly improving the accuracy of object detection and tracking. The correction ensures that shapes like the ball, which should be circular, are correctly detected, and the relative positioning of the players is also accurate.

The following figure shows an example of the perspective correction: the left image represents the distorted view captured by the camera, while the right image demonstrates the corrected top-down view.



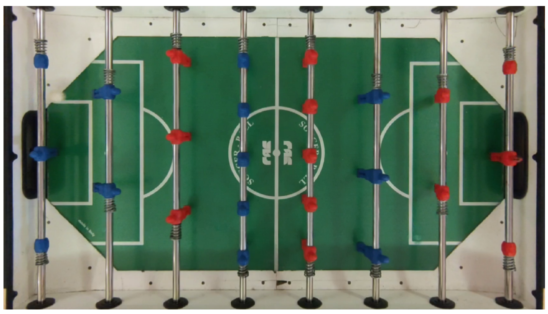


Figure 1: Example of Perspective Correction: Before (left) and After (right)

By correcting the perspective, the system can reliably track the players and the ball, regardless of the camera's angle, thus improving the overall performance of the tracking system.

4 Players Tracking

Tracking the players in a foosball game is a crucial task in the Foosball Tracker system. Accurate players tracking allows the system to monitor the movements of each player in real-time, offering valuable insights into game dynamics. By accurately detecting and following the players, the system can provide information on their positions, movements, and interactions with the ball. This data is useful for game analysis and player improvement, enabling the identification of key moments, such as shots on goal or defensive moves.

The challenges associated with player tracking include fast movements, occlusions, and varying lighting conditions during the game.

4.1 Techniques for Player Detection and Tracking

To detect and track players, the system employs a combination of color filtering and contour detection. Each players set is represented by a distinct color, which allows the system to apply a color filter to isolate the players in the image.

The process starts with applying the color filter to highlight the relevant areas in the frame. Depending on the player's color (e.g., blue or red), a specific filter is chosen to detect the corresponding players. This filtered image is then processed to detect contours, which are the outlines of the objects in the image.

The findContours method in OpenCV is used to identify the contours of the detected objects. After contours are identified, they are filtered based on their size and shape to ensure that only relevant objects (players) are tracked. This allows the system to focus on the areas of the frame that correspond to the players, ignoring irrelevant background elements. Figures below illustrate two separate steps of this process. The first image shows an example of a blue color filter applied to isolate players. The second image demonstrates the raw contours detected in a frame, which include all objects identified before filtering:

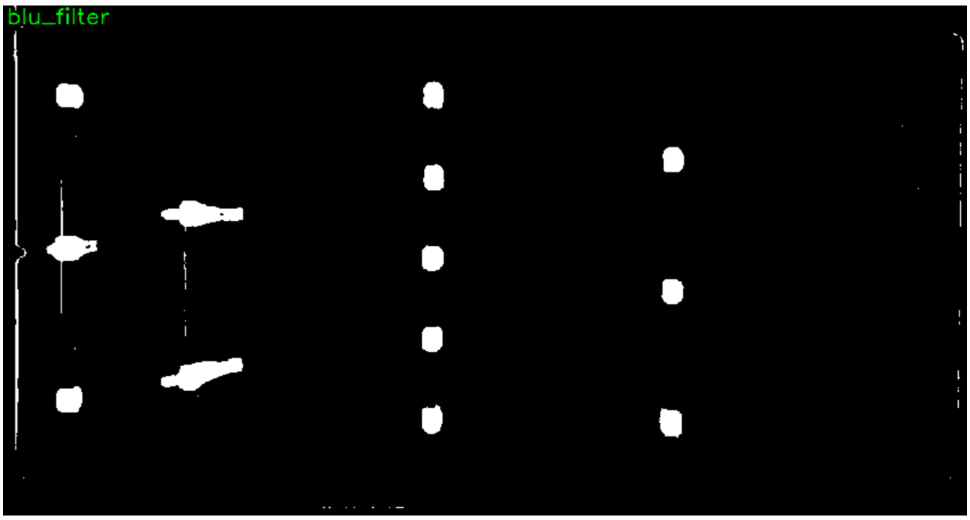


Figure 2: Example of a blue color filter applied to isolate players.

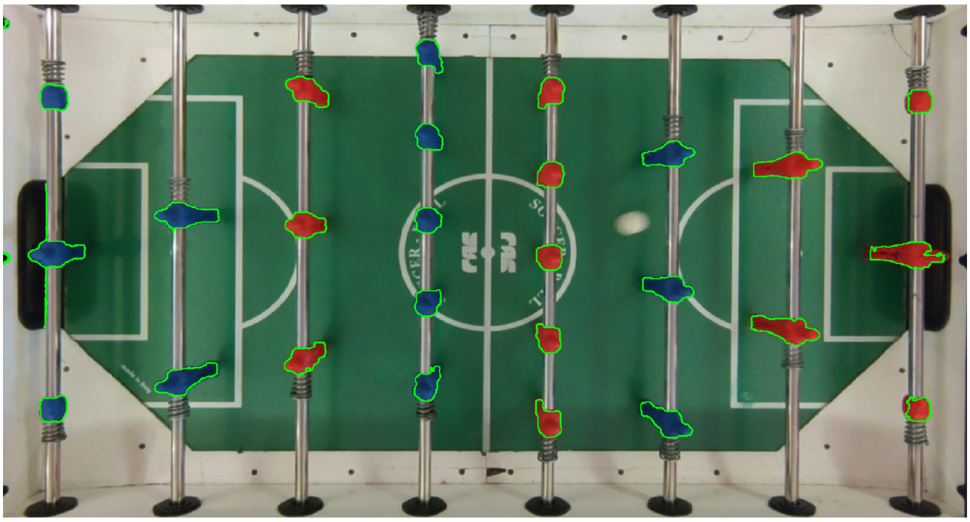


Figure 3: Raw contours detected in the frame.

4.2 Defining the Structure of the Foosball Table

The system is designed around a clear structure representing the foosball table, with zones corresponding to different areas of the field, such as attack, defense, and midfield. Each player is represented by a set of "pitotini" (player markers), which are dynamically updated during gameplay.

The structure of the foosball table is divided into two teams, each with different colored stecche (rods). The "stecche" are categorized into four main groups: portiere (goalkeeper), difensore (defender), centrocampo (midfield), and attacco (attack). Each group contains a specific number of pitotini, which correspond to the players in each role. The system updates and tracks the position of each pitotino as they move across the table.

By maintaining the structure of the foosball table and continuously tracking the players' positions, the system can analyze the game in real-time, providing valuable insights into player movements and strategies.

4.3 Results of Player Tracking

The player tracking system demonstrates good performance in most scenarios, effectively tracking players under various conditions. By leveraging a combination of color filtering, contour detection, and algorithmic position updates, the system can follow each player's movements with a high degree of accuracy, enabling continuous monitoring and analysis of game dynamics.

However, the system is not without limitations. Tracking accuracy decreases when the rods are highly inclined, as this causes light reflections that interfere with the detection process and increase the difficulty of distinguishing individual contours. These situations can lead to occasional errors in player position updates.

The figure below illustrates an example of successful player tracking, highlighting the system's capability under standard conditions:

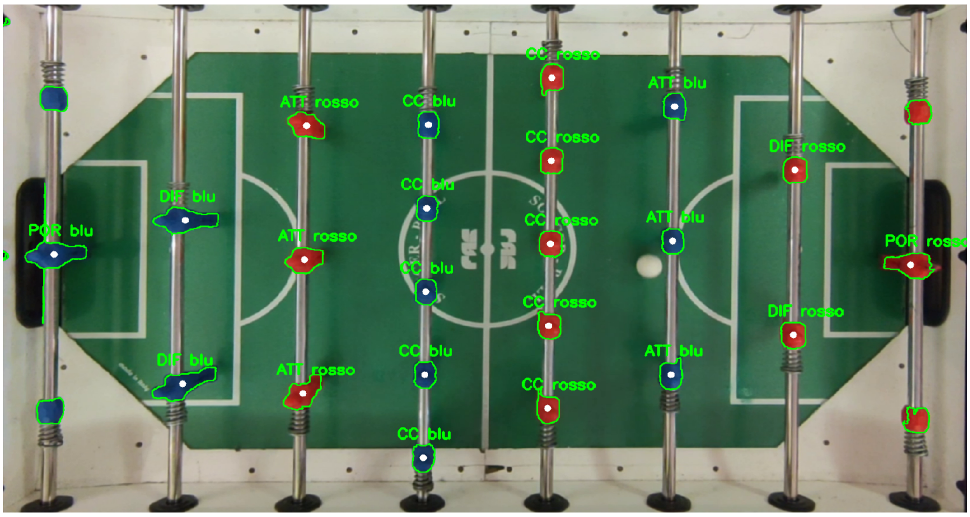


Figure 4: Example of Player Tracking: Accurate tracking under standard conditions.

Despite its challenges, the system reliably detects key game events, such as shots on goal, player interactions, and defensive maneuvers, offering valuable insights for player improvement and strategic decision-making.

5 Ball Tracking

The detection and tracking of the ball on a foosball table present several challenges due to the dynamic and unpredictable nature of the game. The ball moves quickly, changes direction abruptly, and can be partially occluded by players, making it difficult to track with high precision. Moreover, the lighting conditions and reflections on the table surface can obscure the ball, adding noise to the detection process. These challenges require robust and responsive techniques to ensure reliable tracking of the ball during the game.

5.1 General Process and Approach

The ball tracking process is based on a parallel approach, where the image is processed simultaneously by two different techniques, each returning a set of estimated coordinates. The first technique employs the Hough Transform to detect circular shapes, and the second utilizes the findContours function to extract contours of potential objects. These two methods provide two different estimations of the ball's position. The results are then combined by an algorithm that merges the information and provides a more accurate estimate of the ball's location. The final step involves passing the combined estimates to a Kalman Filter, which updates the ball's position and provides a prediction of its trajectory.

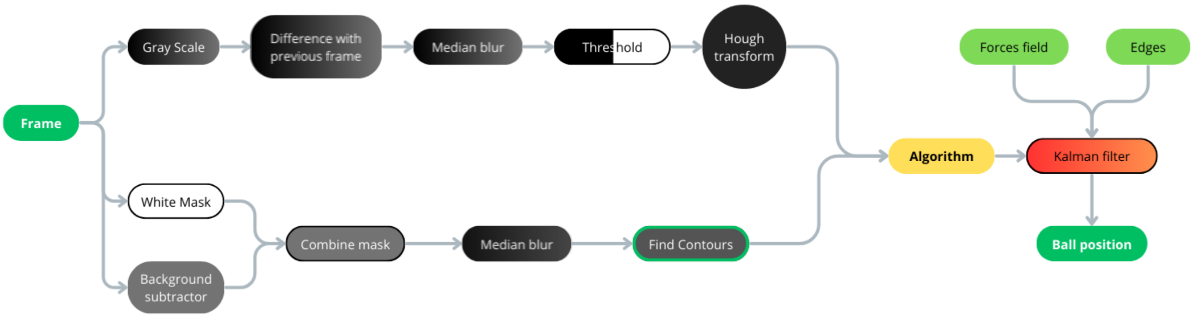


Figure 5: Scheme block of the frame processing to obtain the ball coordinates.

5.2 Hough Transform for Ball Detection

The process of extracting coordinates using the Hough Transform involves several steps aimed at detecting circular shapes corresponding to the ball. The following steps outline the procedure:

- **Grayscale Conversion:** The frame is converted to grayscale to simplify the image and reduce the complexity of processing.
- **Differencing with Previous Frame:** A differencing step is applied to highlight moving objects by subtracting the current frame from the previous one.
- **Median Blur:** A median blur is applied to reduce noise and smooth the image, making it easier to detect the ball.
- **Thresholding:** A threshold is applied to isolate areas of high intensity, corresponding to the ball's location.
- **Hough Transform:** The Hough Transform is then used to detect circular shapes within the processed image, identifying the ball's position.

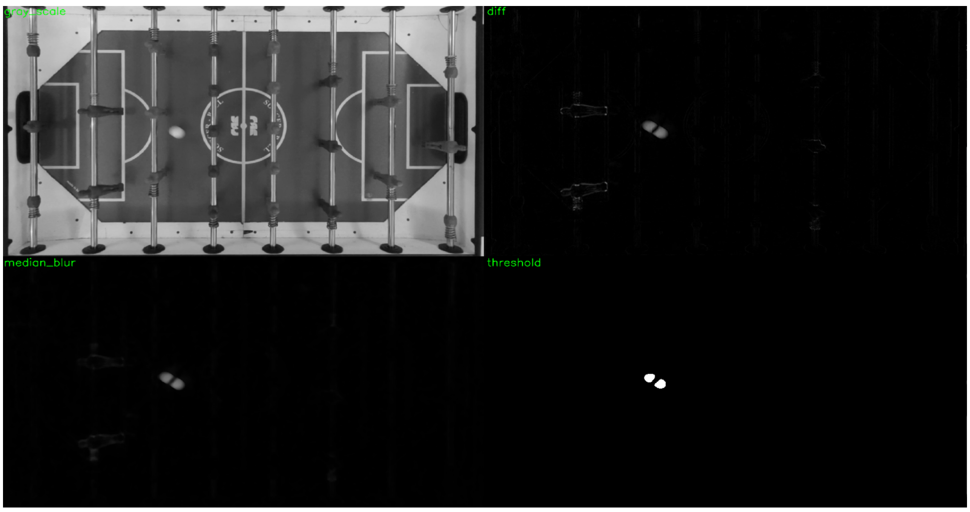


Figure 6: Processing steps for the Hough Transform.

This series of steps is particularly useful for detecting the ball when it moves at high speeds, ensuring accurate tracking despite its rapid motion.

5.3 Contours for Ball Detection

The findContours method is another approach used to detect the ball's position. The steps involved in this process are:

- **White Mask and Background Subtraction:** The image is processed to create a mask that highlights areas where the ball is likely located. This is done using a combination of a white mask for the ball and a background subtractor for the rest of the image.
- **Median Blur:** A median blur is applied to smooth the image and reduce noise, which helps in clearer contour detection.
- **Find Contours:** The findContours method is used to detect the boundaries of objects in the image. It focuses on circular objects, which are likely to correspond to the ball.

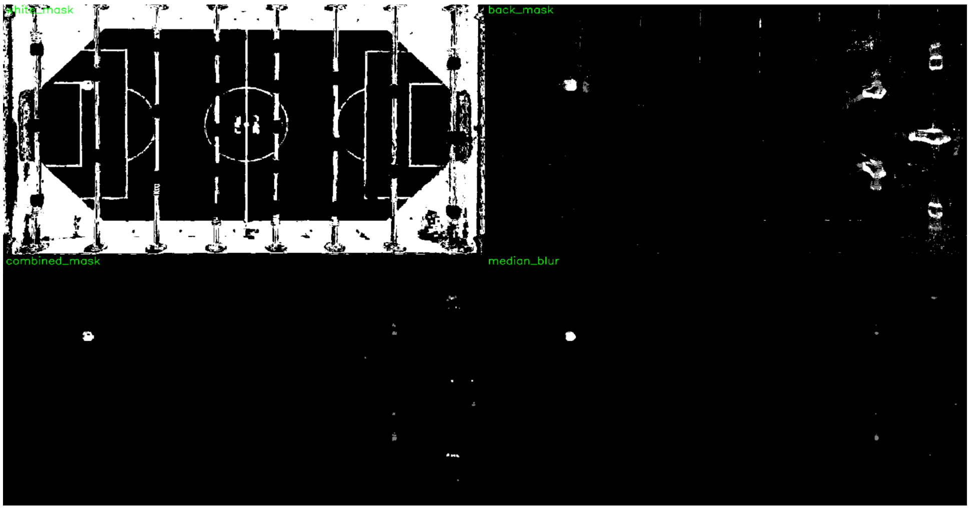


Figure 7: Processing steps for the Contours analysis.

This method works well when the ball is clearly separated from the background and other objects.

5.4 Kalman Filter for Ball Tracking

To maintain accurate tracking of the ball, even when it is occluded or temporarily lost, the Kalman Filter is employed. The Kalman Filter is an algorithm that combines predictions based on the ball's previous state (position and velocity) with new measurements obtained from the tracking process. This allows the filter to update the ball's position and correct for temporary inaccuracies in detection.

The Kalman Filter operates in two main stages:

- **Prediction:** The filter predicts the ball's next position based on its current velocity and previous trajectory.
- **Update:** When the ball is detected in a new frame, the filter updates its state by combining the predicted position with the actual detected position.

Additionally, the Kalman Filter takes into account forces acting on the ball, including the effects of gravity and friction, as well as the influence of the table's edges and corners. These forces are modeled separately to improve the accuracy of the ball's trajectory predictions, especially near the edges of the table. This ensures that even if the ball bounces off the table's boundaries or temporarily disappears from view, the system can continue to track its movement with high accuracy.

5.5 Trajectory Analysis and Filtering

Once the ball's position is estimated using the Hough Transform and findContours methods, the trajectory of the ball is analyzed algorithmically to refine the tracking accuracy. The system evaluates the predicted position and compares it with the previous points in the ball's trajectory. If the predicted position deviates significantly from the expected path, or if it is inconsistent with the recent trajectory, the prediction is discarded or adjusted.

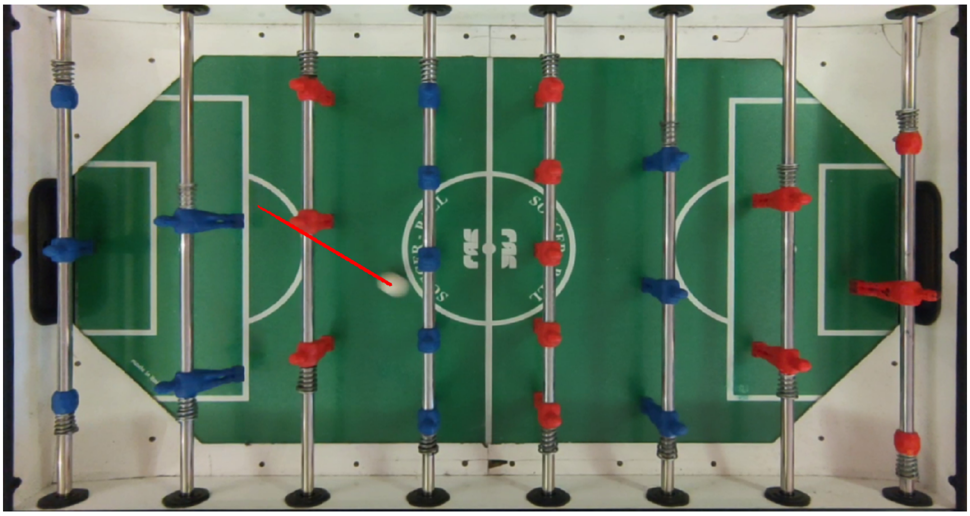


Figure 8: Tracked trajectory as a continuous line through ball detected positions.

This analysis helps eliminate any outliers or erroneous predictions that may arise due to temporary occlusions or noise in the detection process. By applying this filtering step, the tracking system ensures that only plausible trajectories are considered, maintaining accurate tracking of the ball's movement over time. The algorithm effectively handles sudden changes in direction or speed while preserving the consistency of the trajectory, leading to more reliable tracking performance.

6 Conclusion

The Foosball Tracker project has been instrumental in familiarizing with the tools and techniques covered during the "Signal Image and Videos" course. The system successfully demonstrated the application of various image processing methods such as color filters, median blur, contour detection, Hough Transform, and the Kalman filter to track both players and the ball on a foosball table. These techniques were functional in meeting the tracking objectives, but the system's performance was limited, particularly in unfavorable conditions.

While the system performs well under typical circumstances, its reliability and consistency are compromised in more challenging scenarios. For example, when the rods are highly inclined or the lighting conditions are variable, the system struggles to maintain accurate tracking. Additionally, the system's responsiveness is not optimal for live streaming, as there are delays in processing that affect its real-time application.

Despite these limitations, the project has laid a solid foundation for future enhancements and applications, especially in sports analytics and real-time object tracking.

6.1 Future Developments

Future work can focus on several key areas to enhance the system's performance and expand its applicability:

- **Optimization for Live Streaming:** The system can be optimized for real-time processing to reduce latency and improve its responsiveness. Techniques such as parallel processing and hardware acceleration could be explored to achieve smoother performance during live streaming.

- **Implementation of Advanced Technologies and Neural Networks:** Incorporating more advanced computer vision algorithms and deep learning techniques, such as convolutional neural networks (CNNs), could improve the accuracy and robustness of the tracking system. Neural networks could be used for better object recognition, especially in complex or occluded scenarios.

- **Automatic Referee System:** With further development, the system could be adapted to act as an automated referee, detecting fouls or illegal actions during the game. This would require implementing additional logic for action recognition and decision-making based on predefined game rules.

- **Real-time Commentary Using LLMs:** Integrating the system with large language models (LLMs) could enable the generation of real-time commentary. By analyzing the game dynamics, such as player interactions and ball movements, the system could produce insightful commentary to enhance the viewer's experience.

By addressing these challenges and exploring these future directions, the Foosball Tracker project has the potential to evolve into a more sophisticated and versatile system, suitable for a wider range of applications in sports analytics, gaming, and real-time automated systems.