Computer Vision Applications for Soccer Analysis

Joseph Farell
Northeastern University
Campus - Portland, Maine
Portland, USA
farrell.jo@northeastern.edu

Harshil Bhojwani
Northeastern University
Campus - Portland, Maine
Portland, USA
bhojwani.h@northeastern.edu

Pryianka Gujar
Northeastern University
Campus - Boston, Massachusetts
Boston, USA
gujar.p@northeastern.edu

Leonardo DaGraca
Northeastern University
Campus - Portland, Maine
Portland, USA
dagraca.l@northeastern.edu

***Abstract*—The project aimed at developing a system to track ball and player movements in sports using the You Only Look Once (YOLO) model architecture. Inspired by existing applications of YOLO in sports analysis the project team seeks to extend its utility in soccer. The project involves utilizing the YOLO model with sport-specific data to enable distinguishing of teams, tracking of players, referees, and the ball. Additionally, with the help of OpenCV methods, we implemented classical CVPR techniques. In this case, the Kalman Filter to predict the position of the ball after every frame of the video. This project highlights the capabilities of combining deep neural networks and classical computer vision techniques. The latest iteration, YOLO8, released in January 2023, serves as the foundation for this project. By leveraging the strengths of YOLO and customizing it for soccer analysis, the project aims to contribute to the advancement of sports analytics and enhance understanding of player and ball movements.**

I. Introduction

In the current era, where the advancement of Convolutional Neural Networks (CNNs) and Deep Learning Models is at the forefront of technological innovation, their applications extend beyond research and development to commercial utilization. This trend is particularly evident in the world of sports, where renowned sports leagues have widely embraced these technologies for both statistical analysis and tactical insights. Examples include Hawkeye in cricket and tennis, as well as VAR (Video Assistant Referee) in soccer, which are prominent Computer Vision Applications employed in real-world scenarios.

For our project, our objective was to develop a similar soccer analysis tool. We utilize a pre-trained CNN, trained on a diverse range of data, to perform soccer analysis by detecting players and the soccer ball. Furthermore, we employed a Kalman filter to refine the analysis, deriving a ratio of RGB color space for the bounding boxes of players to distinguish between teams and referees.

We did plan to include more tracking tools, including real-time ball trajectory tracking and calculating team ball possession, but faced some limitations which will be elaborated further upon in our report.

II. Related Work

- **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection.**

This was the first paper in 2016 introducing YOLO (You Only Look Once), a neural network predicting bounding boxes of objects and their class probabilities of the entire image feature set in one evaluation. Network-based on being able to process 45 frames per second being able to detect the general representation of the object far superior to DPM and R-CNN at the time.

The project focused on soccer analyses to detect all the players, and referees, including the soccer ball for every single frame. YOLO seemed the most efficient architecture as it could detect all bounding boxes as it is being trained on the entire image optimizing its detection performance. The pre-trained YOLO model being specifically trained towards the generalization of the objects it gave us and having one of the highest mean average precision compared to other models made it ideal for our project needs.

- **Rodrigues, J., Cardoso, P. J. S., Vilas, T., Silva, B., Rodrigues, P., Belguinha, A., & Gomes, C. (2014). A Computer Vision Based Web Application for Tracking Soccer Players. In Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access (pp. 450-462).**

Paper-based on a Computer vision-based web application, this is a far more different analysis tool than ours and is meant to be more statistical than providing or tracking live positions. The web app is based on 5 different modules to do gameplay analyses for the team, using the stadium cameras, track every individual player attribute, track tactical approach for the team players, and finally make the data presentable and readable on the app. This paper was used more as a reference for our approach. The paper was released in 2014 and did not make use of YOLO. It used a novel detection system where rather than tracking players every frame, they based their program on tracking the distance between each player and each frame. Ideologically this paper helped us to integrate certain ideas in our projects and ideas to improve upon our implementation.

- **Cai, Z., Gu, Z., Yu, Z. L., Liu, H., & Zhang, K. (2014). A real-time visual object tracking system based on Kalman filter and MB-LBP feature matching. Multimedia Tools and Applications, 75(February 2016), 2393–2409.**

Paper on introducing the development of the Kalman Filter which used Modified Multi-Scale Block Local Binary Patterns (MB-LBP) feature set to be able to characterize the tracked object and predict its location of the object. It's tracking based on the most current image patch of the target. Realizing that a pre-trained network does not do an efficient job towards tracking the ball, a Kalman filter was implemented on the soccer ball, to predict the ball to be tracked every frame.

III. Methods

1) The YOLOv8 network, pre-trained on the COCO dataset, serves as a robust starting point for our soccer analysis project. Leveraging its pre-trained capabilities, we can efficiently detect and monitor players and the soccer ball on the field. This pre-trained model comes equipped with the ability to identify objects across 80 classes, including crucial elements for our task such as individuals (person class) and the ball (sports ball class). By building upon this foundation, we tailor the model specifically for soccer analysis, refining its detections and bounding box predictions to enhance its relevance and accuracy within the context of the sport.

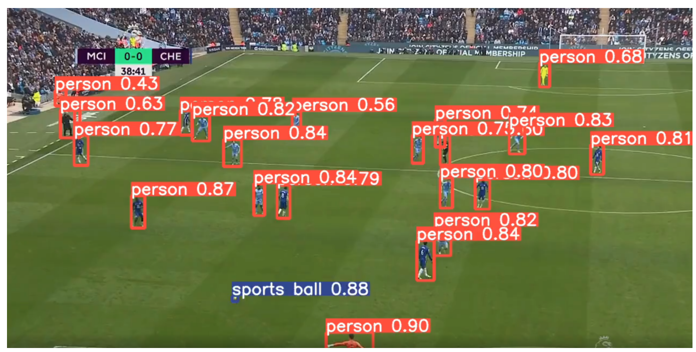

Fig. 1. YOLO Default Output

2) Our first task was to address the ball. The pre-trained model is capable but not efficient in being able to track and monitor the soccer ball for every frame. This is likely due to the small size of the ball, less than optimal resolution of the video example, and object occlusions. Following Cai et al. we decided to implement a Kalman filter to predict the ball position in frames where it was not detected. This helped us achieve superior results in tracking the ball.

   Below (figure 2) displays an example of a processed frame where the ball was accurately detected. As displayed, when the ball is detected we indicate and highlight it in blue.

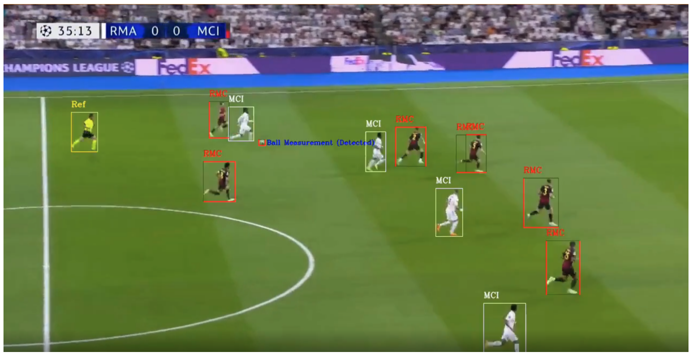

Fig. 2. Ball Detected Example Frame

When the ball is not detected we display the bounding box as predicted by the Kalman filter. For differentiation, we display this in green.

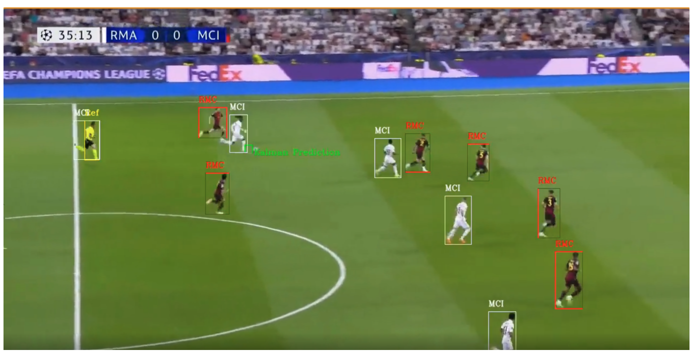

Fig. 3. Ball Not Detected - Kalman Filter Example Frame

The Kalman filter performed well in most of the clips we tested it on. However, an aspect of the Kalman is that the error will continue to increase when true detections are not made in multiple subsequent frames. This was observed in our testing. Again, following Cai et al. we capped the number of consecutive frames without a true detection at $n = 10$, turning the Kalman filter off and ceasing prediction if this criterion was met.

Figure 4 shows an example frame of what is displayed when the Kalman filter is turned off and the ball is lost. We turned the Kalman back on when a true detection was made. A limitation of the Kalman filter we experienced was when there would be errant (false) ball detection it could impact the actual location of where the ball was located. This was min-

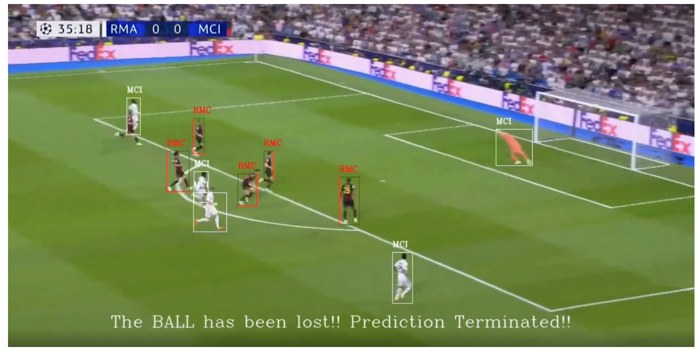

Fig. 4. Ball Lost Example Frame

imal and was quickly corrected when a true detection was made.

3) Our next task was to differentiate the various classes of people in the frames (teams and referees). Our approach took advantage of color variations within the bounding boxes between the classes. The approach that yielded the best results was to pre-define BGR color space ranges of the jersey colors and shirt color worn by the referee. We then generated 3 thresholded bounding boxes for each detection where the pixels within range (of the 3 predefined colors) remained and all the other pixels were turned black.

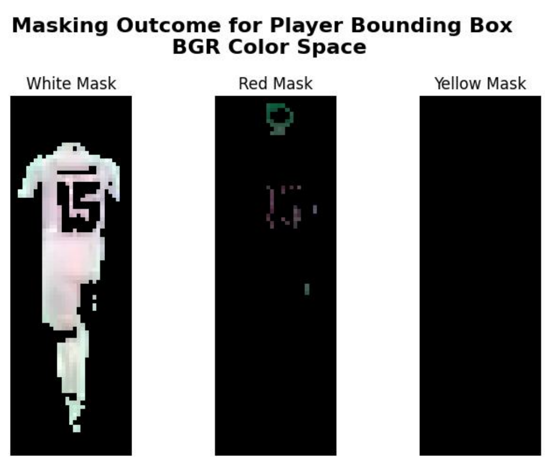

Fig. 5. Bounding Mask: White Jersey

The above figure displays the 3 masked images for an example bounding box. In this example the dominant color is white.
The above figure displays the 3 masked im-

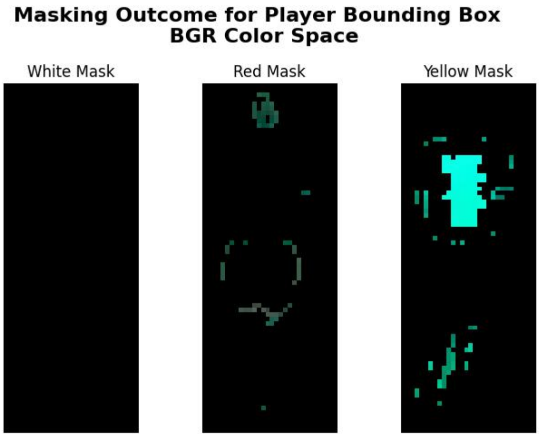

Fig. 6. Bounding Mask: White Jersey

ages for an example bounding box. In this example the dominant color is yellow.

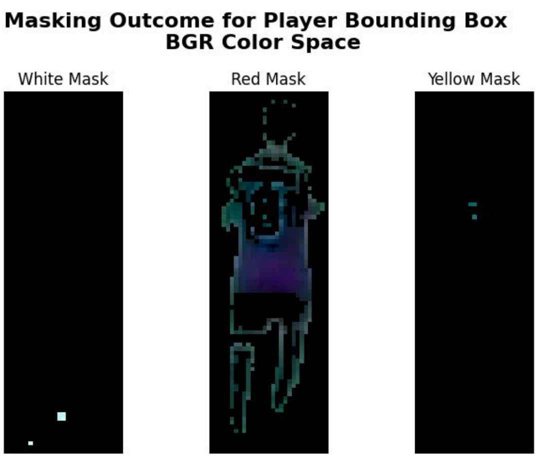

Fig. 7. Bounding Mask: White Jersey

The above figure displays the 3 masked images for an example bounding box. In this example the dominant color is red.

Next, we iterated over each of these 3 thresholded images (3 for each bounding box) and computed a ratio of non-black pixels to total pixels. This generated an array of 3 ratios for each bounding box. We then used the max ratio to classify the person in the bounding box.

As demonstrated in the example video this approach worked rather well. An obvious limitation is that this is not dynamic and requires pre-knowledge of the colors worn by the people of interest and BGR values associated with those colors.

IV. Experiments and Results

Throughout developing this project we conducted numerous experiments, most of which resulted in success and some of them gave us rich learning.

One of these experiments was related to different ways to differentiate the people in the frame. While we settled on the method described above, a related method that we tested and conclude may actually be more effective in most instances is using HSV space rather than BGR space.

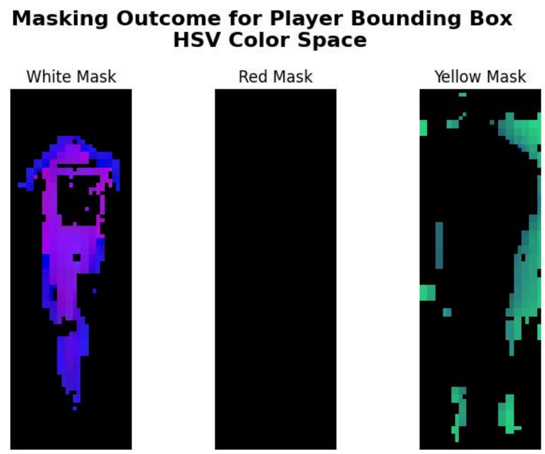

Fig. 8. Bounding Mask: White Jersey

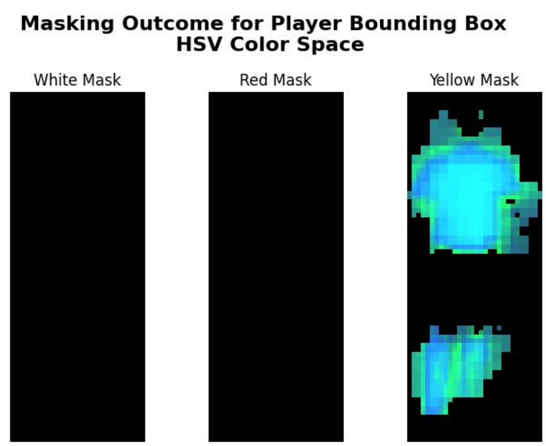

Fig. 9. Bounding Mask: Yellow Jersey

The above two figures display the HSV space outcome of the color masking. These are same the three detection displayed in BGR masking. White and yellow (the true colors) are very well detected.

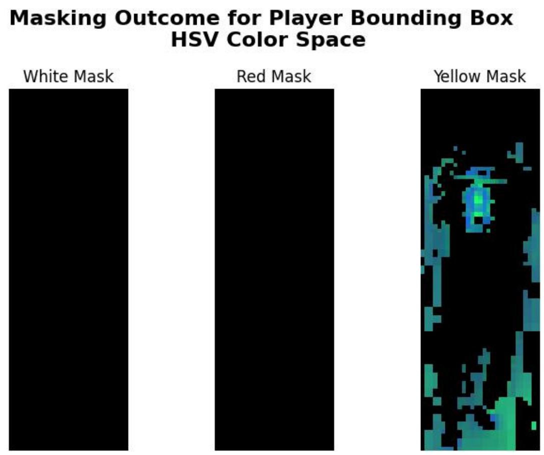

Fig. 10. Bounding Mask: Red Jersey

Figure 10 displays the failing HSV space on the example video. In HSV space it proved difficult to define a color range that would detect the red jerseys. This is likely because the jerseys were not truly red and were not at all a homogeneous color.

We attempted two other methodologies to differentiate the teams and referees. The first was HSV histogram matching. We hypothesized that we could generate a feature vector from the HSV histograms of the person bounding boxes and if we had a baseline reference we could use cosine distance (long feature vector) and assign the classes based on proximity. It proved challenging to dynamically extract the baseline and this strategy was eventually scrapped.

The second methodology we tested utilized KMeans clustering to find the dominate colors within the bounding boxes. The idea was to fit KMeans on bounding boxes to separate the two teams and the referee. This would make the program dynamic and more robust. We experienced glitches and sub-optimal performance leading to us utilizing the color space method over this one. However, we believe is this potentially better solution than the current implementation.

An attempt to train the model from scratch was made with the aim to define our own classes which will be far less than what the pre-trained model outputs for a better optimization and superior accuracy, but due to the computational limitations, sparse annotated dataset, the desired results to detect the ball and players were far less accurate than what was initially anticipated, the confusion matrix post-training denoted the same, leading us to rely on the pre-trained model instead.

Being close to implementing the ball trajectory implementation, encountering a significant setback arises from the discrepancy in ball tracking. As using a pre-trained model, its performance in detecting the ball falls short. This issue resulted as either the model failing to detect the ball entirely or generating numerous false positives, misidentifying unrelated objects as the ball. Consequently, the output failed to meet our expectations, leading us to abandon the implementation altogether.

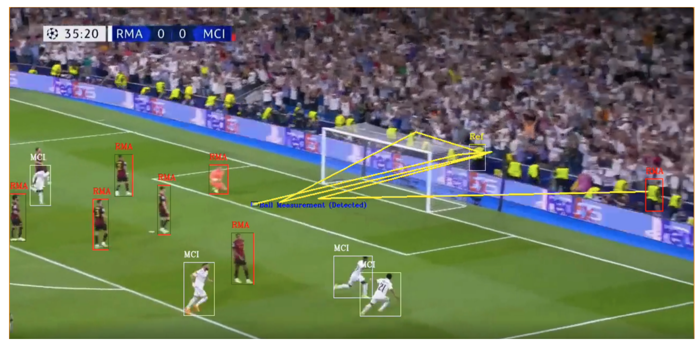

Fig. 11. Ball trajectory results

In the above example, we observe a notable deficiency in the ball trajectory analysis. The model is struggling with excessive false positive soccer ball detections, leading to a cluttered output. Moreover, the trajectory representation in 2D for a frame extracted from 3D space lacks the necessary calibration, thus yielding inaccuracies, particularly evident in longer trajectories.

As a part of our project, we were trying to estimate ball velocity. We started by tracking the ball in each frame and calculated the distance travelled by the ball. While we succeeded in calculating the ball velocity in pixels per sec, however, the bigger challenge was to translate this into real life length metrics. The lack of reference points on the field, movement of camera, and changes in camera magnification made this task too complex. Hence, after consulting with Prof. Bruce Maxwell, we concluded that it was outside the scope of this project decided to exclude it.

V. Discussion and Summary

We have successfully implemented a sports analysis system. The system processes a video, identifies the players, tracks them, identifies and tracks the ball, differentiates between the two teams and the referee, and annotates all this on the video. We use YOLOv8 to identify the players and the ball. The ball is tracked using Kalman filter. We use BGR color space to differentiate the teams and referee.

The major findings of this project were that although YOLO performs well on object detection, it does not do that well in object tracking. A Kalman filter when implemented on top of YOLO makes a robust object tracking system. Although it is a general assumption that HSV color space gives better results than BGR, in this project we found that BGR color space does a much better job at separating the teams and referee.

VI. References

1) Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection.
2) Rodrigues, J., Cardoso, P. J. S., Vilas, T., Silva, B., Rodrigues, P., Belguinha, A., & Gomes, C. (2014). A Computer Vision Based Web Application for Tracking Soccer Players. In Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access (pp. 450-462).
3) Cai, Z., Gu, Z., Yu, Z. L., Liu, H., & Zhang, K. (2014). A real-time visual object tracking system based on Kalman filter and MB-LBP feature matching. Multimedia Tools and Applications, 75(February 2016), 2393–2409.
4) Kuan, C. (2023). Football players tracking - identifying players' team based on their jersey colors using OpenCV. Towards Data Science. https://towardsdatascience.com/football-players-tracking-identifying-players-team-based-on-their-jersey-colors-using-opencv-7eed1b8a1095
5) SkalskiP (2022) Supervision. GitHub Repository. https://github.com/roboflow/supervision/blob/develop/LICENSE.md
6) Roboflow (20223) Track and count objects using YOLOv8 ByteTrack and Supervision. YouTube. https://www.youtube.com/watch?v=OS5qI9YBkfk\&t=608s\&ab_channel=Roboflow
7) TUND-USA (2023) Real Madrid vs Manchester City — UEFA Champions Leahue 2022/23 Semis — TUDN https://www.youtube.com/watch?v=JPOgqHljHOM&ab_channel=TUDN-USA