



International Journal of Marketing Management

ISSN 2454 - 5007



www.ijmm.net

Email ID: editor@ijmm.net , ijmm.editor9@gmail.com

Identifying and following players in Soccer Track footage

¹ Chirala Sravan Kumar, ² G. Amulya,

¹Assistant Professor, Megha Institute of Engineering & Technology for Women, Ghatkesar.

² MCA Student, Megha Institute of Engineering & Technology for Women, Ghatkesar.

Abstract—

Improving both individual and team performance is possible via analysis of player movement on the field during a match. Current methods frequently use cumbersome GPS trackers strapped to players or a network of several cameras, each of which monitors a different area of the field, to monitor how the football players move about. This research introduces a new method that utilizes cutting-edge object recognition and tracking algorithms in conjunction with a single high-resolution fisheye camera. At all times, the suggested technique gives you the right player identifiers and locations. We tested three object tracking algorithms and investigated many variations of the YOLOv8 algorithm that used different picture pre-processing methods. This method drastically cuts down on system costs, allowing even smaller teams to use precise match data to their advantage.

Keywords

items recognition; items following; players being followed; homographic transformation.

I. INTRODUCTION

Teams in today's sports rely more and more on analytics to help them perform better. It takes a lot of effort and isn't always accurate to manually analyze recorded games. Keep in mind that most TV shows don't record the whole field [1, 2], so you can't get all the info you need from them. On the other hand, professional football leagues use a system wherein a number of cameras are placed strategically around the stadium, each of which captures a different area of the field. This system's accuracy belies the complexity and expense of the infrastructure it requires, which includes not only cameras and

hardware but also advanced algorithms and a lot of computing power [3]. The use of measuring sensors, such as the GNSS [4] or the LPS [5], is another typical strategy. Under perfect circumstances, this method produces very precise results, but it isn't up to snuff when faced with difficult situations, such as indoor matches. Plus, every participant has to have a tracking device, which, especially with high-precision sensors, may be rather cumbersome. The hefty price tag is another issue, since it could be difficult for teams with low funds to purchase such equipment. Therefore, one reasonable issue to ask is whether it's feasible to use a single, reasonably priced sensor to monitor the whereabouts of every player on the field. A high-resolution fisheye camera, which can capture the whole field in a single shot, is the answer to this problem. To identify gamers up to fifty meters away, high resolution is required. In order to capture every participant on the field, a fisheye lens is necessary because of the broad field of view it provides. In the past, high-resolution cameras and fisheye lenses were prohibitively costly. However, their costs have dropped so much in recent years that they are now integrated into inexpensive portable devices like smartphones. The placement of the camera is another important consideration. Right above the field is where you want to be for optimal visibility and comfort. Although this would make player monitoring easier, the necessary infrastructure is often large and unavailable. Methods such as using drones to hover above the football pitch or installing cameras on steel cables above the field are often used [6, 7]. But there are certain downsides to using drones, such their short flight durations and vulnerability to weather, and steel wires are usually reserved for well-equipped stadiums. Consequently, placing the camera close to the football field is the most feasible option [26]. There are some issues with player localization, tracking, and converting their locations from camera coordinates to field coordinates, but overall, it streamlines the operating method and cuts expenditures. In this article, we will go over the algorithms that are utilized to follow

players on the field using only one video feed from a fisheye camera. It will review the current object identification and object tracking algorithms, both as standalone components and as part of a whole system, and provide a concise summary of their performance. In order to find the best combination for the final answer, we will test out several pretreatment and postprocessing strategies. The process of changing the player's location from the camera's coordinate system to the field's coordinate system will be detailed at the conclusion.

II. OBJECT DETECTION

Part A: Object detection theory Automatic object identification and localization in pictures or videos is made possible by a class of computer vision techniques known as object detection. There are a plethora of applications for object detection algorithms in fields as diverse as driverless cars, robotics, medical diagnostics, traffic monitoring, and many more. Despite deep neural networks' proven superiority over earlier algorithms, all object identification methods still face a number of obstacles. Variations in illumination, backdrops that are too busy or have too many textures, objects with varying proportions and sizes, different points of view, limited processing time, differences within classes, and an absence of data are all examples of these difficulties. B. Studies on object recognition Object detection algorithm research used to be quite structured, with several separate phases for different types of algorithms. The current trend in approaches is towards employing a single unit, such a deep neural network, to solve all the required processes; this is known as the end-to-end approach. In the last twenty years, object identification has gone through two major phases: the first, centered around more conventional techniques, and the second, centered around deep learning. Most algorithms used characteristics that were established by humans to identify edges, corners, colors, and textures when classical object identification was in use. Classification of the collected characteristics was then accomplished using statistical techniques. The competence of the algorithm developers and the accuracy of the specified characteristics were major factors in their performance. Among the most important algorithms developed during this time are the Deformable Part-based Model [10], the Viola-Jones detectors [8], and the HOG detectors [9]. The highest year for object identification using manually

set characteristics was around 2010. Even with the use of ensemble approaches and variants on effective techniques, progress remained sluggish and improvements were small [11]. Reintroduction of convolutional neural networks (CNNs) to computer vision occurred in 2012 [12]. It was wondered whether they may be used for object identification because of their capacity to learn high-quality and robust picture embeddings. Depending on whether they use a single neural network for object identification or split this process into two separate processes, object detection algorithms based on deep learning techniques may be classified as either single-stage or two-stage. Within the family of algorithms that includes R-CNN [13], Fast RCNN [14], and Faster R-CNN [15], there are some of the most well-known two-stage object detectors. Although the first method in this series provided a solid groundwork for object identification using convolutional neural networks, the succeeding algorithms swiftly outperformed it due to their incremental improvements. Notable achievements such as SPPNet [16] and FPN [17] have contributed to the advancement of model accuracy and had an impact on the most cutting-edge object identification algorithms used today. Object identification using convolutional neural networks was followed soon after by the emergence of single-stage techniques. For object localization and classification, these techniques combine the usage of a single neural network. As a result, they can achieve much greater frame rates than two-stage algorithms. You Only Look Once (YOLO) algorithms are a good example of a family that is both popular and effective. Significant enhancements have been made with each succeeding version [18], including versions 1, 2, 3, 4, 5, 6, v7, v8, YOLOX, YOLOR, and YOLO-NAS. Contemporary object identification techniques are surpassed by more recent versions, such as YOLOv8. One such one-stage technique that takes a somewhat different tack is SSD (Single Shot MultiBox Detector) [19]. It has been around for a while, but its great performance—particularly on devices with limited resources—keeps people using it. The attention mechanism and transformer structures, which have recently been introduced to computer vision, are noteworthy advancements that have their roots in natural language processing (NLP) [20]. When it comes to object detection, these designs really shine. Modern object identification algorithms use convolutional neural networks, however newer

methods like RT-DETR [21] show performance on par with or better than these.

III. OBJECT TRACKING

The idea of object tracking (A) Object tracking seeks to follow the movement of objects inside a video, in contrast to object detection, which primarily focuses on pictures. An algorithm does object tracking when it takes a starting set of object detections, gives each one a unique ID, then follows the objects as they move between frames in a movie. Put another way, this method treats the motion of each item as if it were a continuous trajectory. While object recognition and object tracking have certain commonalities, object tracking also has its fair share of unique obstacles, such as objects' rapid motion and their erratic trajectories. It is critical to construct an appropriate object model in order to do object tracking. Providing a distinct description of the item is required for this task. A crucial part of object modeling is the representation of the object's target, which may be in the shape of the object's center coordinates, a bounding box, a contour, a skeleton, a mask, or any combination of these. Moreover, numerical characteristics describing the object's visual attributes are required. These features may be acquired from the object's picture histogram, image intensity, or from image descriptors such as SIFT, SURF, ORB, HOG, or neural network embeddings. Finally, a suitable physical model should be used to depict the object's motion.

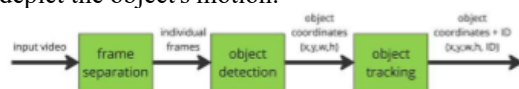


Figure 1. Object tracking pipeline

Part B: Algorithms for Object Tracking Object recognition on individual video frames is the first step in the present-day state-of-the-art algorithms' process of creating continuous tracks (Figure 1). A few of the most popular and powerful algorithms are going to be quickly described in this chapter. 1) Simplified Online and Real-Time Tracking, or SORT Back in 2016, we saw the introduction of the SORT tracking algorithm [22]. It outperformed state-of-the-art tracking algorithms because to its emphasis on simplicity and dependability. In the following stages, SORT used a combination of several popular strategies. an) Object recognition In order to identify

things in every single frame of the movie, an object detection method is used. The next stage of the method makes use of the item bounding boxes that have been generated. c) The Kullman filter The item's predicted position in the next frame is estimated using the Kalman filter using object movement modeling and the current object detections. Algorithm from Hungary A cost matrix is constructed to link the newly detected objects to their expected placements. Each potential combination of detections and predictions is represented by an intersection over union (IoU) value in this matrix. Next, we use the Hungarian method to determine which combination maximizes the sum of all crossings. d) Making new object IDs and removing old ones It is necessary to issue a new ID to each item that enters the video. It is necessary to delete an object's ID from the list of monitored objects if it vanishes from view. In place of IoU, DeepSORT [23] used neural network embeddings in the cost matrix and Mahalanobis distance to get even better results. Because neural network processing is required for each object picture, this increases processing overhead, but it takes location and item appearance into account. Second, ByteTrack When it came to enhancing the efficiency of object tracking algorithms, the ByteTrack algorithm's creators [24] nailed it. Problems with algorithm performance could not always stem from the tracking algorithm, despite common assumption to the contrary. It doesn't matter how sophisticated the tracking method is if the object detection algorithm doesn't provide high-quality object detections. Finding an optimal trade-off between false positives and false negatives is a significant performance hurdle for object identification algorithms. Each discovered item is given a confidence score using modern algorithms. Whether detections are accepted or dismissed is determined by this score and the confidence threshold that is established. Although there will be fewer false positives, there may be more missed detections if the threshold is set too high. On the other side, if the threshold is set too low, there will be few missed detections but many false positives. Giving the tracking algorithm access to all identified items, independent of confidence levels, and letting it prioritize its findings is the basic premise of the ByteTrack algorithm. While ByteTrack and the SORT algorithm have many similarities, ByteTrack splits the process of associating newly discovered items with their projected locations into two separate processes. A greater confidence score in a detection

is used to begin the matching process. The association is then executed using the remaining predictions and the lower-confidence detections after all feasible matches have been made. "Observation Centric Simple Online and Realtime Tracking" (OC-SORT) is the third component. Three issues with the SORT algorithm were pointed up by the OC-SORT [25] method's creators. When using the Kalman filter to anticipate an object's trajectory, they identified possible difficulties and proposed remedies. The core idea behind this method is that current object identification algorithms are quite good at what they do, and their results are often more trustworthy than expected object movement. The SORT method has certain drawbacks, such as being vulnerable to state noise, magnifying errors in time, and using the Kalman filter, which is estimation-centric, too. The OC-SORT team used many novel strategies to boost SORT's efficiency.

- Observation-Centric Momentum (OCM)
- Observation-Centric Recovery (OCR)
- Observation-centric Re-Update (ORU)

The optimization of the algorithm's Kalman filter and association components is the primary goal of these modules.

IV. DATASET

The open-source dataset SoccerTrack [26] is used to compare different methods and assess the efficacy of different object recognition and tracking systems. Annotated footage of football matches makes up this dataset. The dataset includes footage shot from both a drone mounted camera above the field and a high-resolution fisheye camera placed next to the field. Only footage captured by the camera next to the field will be used for this experiment. You can see a single frame from the camera that is placed adjacent to the field in Figure 2.



Figure 2. Image captured by fisheye camera next to the field

The dataset authors calibrated the cameras during video collection to eliminate fisheye lens distortion. Consequently, there is no fisheye distortion in this

dataset, and all future analyses will be conducted on such pictures. Twenty 30-second movies with annotations were included of the dataset when it was published. Afterwards, 66 more movies were added to the dataset. A related CSV file indicating the position of the player in each frame is included with each movie. We used five films to test our algorithm in this experiment.

V. PLAYER DETECTION AND TRACKING

A. Identifying players To find players on the pitch, the YOLOv8 [27] algorithm is used. A variety of sizes—from nano to small to medium to giant and extra-large—are available with this approach. The number of layers and convolutional filters vary throughout various variants, but the architecture stays mostly the same. The logic behind it is to have low-performance rapid versions and slower high-performance variants. Because it was freely accessible online and had the 'person' class already included, a model trained on the COCO dataset [28] was chosen to be used. Because the COCO dataset has 80 distinct object classes, the YOLOv8 model can identify all of them after training on it. The 'person' class was the only one that was kept for this experiment. Improving performance may be possible by tailoring the YOLOv8 model to the 'person' class. With so many different variants of the method examined in this work, however, such fine-tuning would impose a heavy strain on computer resources. The very high picture quality required to capture even the furthest individuals posed a substantial obstacle to player detection on the field. The camera's original resolution is 7680×4320 pixels, however after the calibration transform, the photos are shrunk to 6500×1000 pixels. It may be difficult to spot issues with really distant players, even with this high resolution.

Using such very high-resolution photos directly is beyond the capabilities of any state-of-the-art algorithm. Downscaling the picture to 640×640 , the input resolution needed by YOLOv8, would be the easiest option, but it would lead to an unacceptable loss of information and make the players almost vanish from the screen because of how tiny they are relative to the image. In order to overcome this obstacle, the sliding windows technique is used,

which involves cropping the picture into many partly overlapped regions.

The YOLOv8 algorithm is applied to each crop separately. Then, to get rid of any possible duplicate detections in the overlapping crop zones, a greedy non-maximum suppression approach is used. Crop width, crop height, and crop overlap ratio are the characteristics that define the sliding window method. We tried out two different sets of parameters: • Increased crop density: twenty-six crops per frame, 640×640 crop size, and twenty percent overlap in height and breadth. • Reduced crop density: 6 crops per frame, 1280×1280 crop size, 10% height / width overlap. Because the system tries to identify every person in the picture, it might produce a lot of false positives if there are other people in the area. All detections that are not located in the field are deleted in order to address this problem. A ray-casting technique is used to find out whether a detection is within the trapezoid created by the field corners when the coordinates of the field corners are known. You can see the players that the YOLOv8 algorithm found in Figure 3.

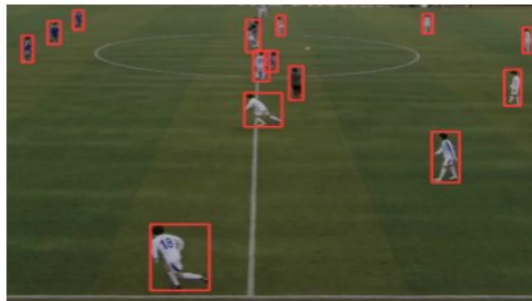


Figure 3. Players detections

An assortment of YOLOv8 variants were evaluated, each with its own set of preprocessing and postprocessing methods. The preprocessing methods included varying numbers of crops, while the postprocessing methods included rejection of individuals beyond the football field. Mean Average Precision (mAP), a commonly used measure, is used for this objective. Tables 1 and 2 display the results.

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT YOLOV8 VARIATIONS WITH HIGHER NUMBER OF CROPS

Higher number of crops: 640x640 crop size, 20% overlap			
YOLOv8 variation	Without detection rejection mAP [%]	With detection rejection mAP [%]	Frame processing time [s]
<i>nano</i>	74.60	80.30	1.01
<i>small</i>	77.01	82.68	1.09
<i>medium</i>	76.38	81.82	1.35
<i>large</i>	77.96	83.01	1.74
<i>extra-large</i>	78.61	83.97	2.27

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT YOLOV8 VARIATIONS WITH LOWER NUMBER OF CROPS

Lower number of crops: 1280x1280 crop size, 10% overlap			
YOLOv8 variation	Without detection rejection mAP [%]	With detection rejection mAP [%]	Frame processing time [s]
<i>nano</i>	61.01	65.73	0.68
<i>small</i>	68.46	73.55	0.70
<i>medium</i>	70.48	75.13	0.74
<i>large</i>	73.53	78.61	0.80
<i>extra-large</i>	74.55	79.42	0.88

Object detection performance was greatly improved by lowering the amount of false positives by discarding detections outside of the football field. As a result, there will be no comparisons made between tracking algorithms and detection rejection techniques.

B. Monitoring of players Following the recognition of players and the elimination of non-football-related items, the object tracking algorithm is informed of any remaining detections. The effectiveness of three object tracking algorithms—SORT, ByteTrack, and OC-SORT—is evaluated in this analysis. Figure 4 shows the most popular metrics used for object tracking.

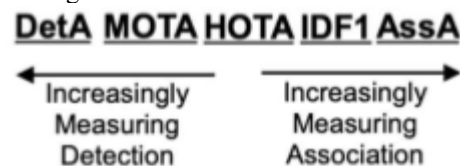


Figure 4. Object tracking metrics

The HOTA measure, which provides an ideal compromise between detecting and assessing association accuracy, will be used in this investigation. Tables 3 and 4 show the values of the HOTA measures for a larger and smaller number of crops, respectively.

TABLE III.
OBJECT TRACKING PERFORMANCE
(HOTA) WITH HIGHER NUMBER OF CROPS

Higher number of crops: 640x640 crop size, 20% overlap			
YOLOv8 variation	SORT HOTA [%]	ByteTrack HOTA [%]	OC-SORT HOTA [%]
<i>nano</i>	50.655	54.827	51.34
<i>small</i>	53.72	56.926	53.215
<i>medium</i>	52.563	56.645	52.821
<i>large</i>	54.420	56.603	53.949
<i>extra-large</i>	54.926	58.553	55.073

TABLE IV.
OBJECT TRACKING PERFORMANCE
(HOTA) WITH LOWER NUMBER OF CROPS

Lower number of crops: 1280x1280 crop size, 10% overlap			
YOLOv8 variation	SORT HOTA [%]	ByteTrack HOTA [%]	OC-SORT HOTA [%]
<i>nano</i>	40.224	39.373	40.459
<i>small</i>	47.197	47.254	47.838
<i>medium</i>	48.115	49.762	48.520
<i>large</i>	51.482	51.951	50.949
<i>extra-large</i>	51.347	54.106	52.064

The homographic transform (CRT) The camera view coordinate system is used to provide the player with their position once object recognition and tracking has been executed. Player position in field coordinate system in meters is necessary for a variety of calculations, including but not limited to: estimating various metrics (such as the player's distance traveled throughout the match), locating free kicks and fouls, calculating player distribution, and many more. Using a homographic transformation, we can convert the monitored players' positions from camera coordinates to field coordinates. At first, the field corners are manually annotated and the picture viewpoint is adjusted to a view above the field in order to estimate the transformation matrix. We just execute this technique for the one video frame since the camera was not moved throughout the whole match. Then, the monitored players' coordinates are transformed using the same matrix. Pixel values ranging from 0 to 4000 are now available for use as

the player's position in the field coordinate system. It is possible to linearly extrapolate the player's coordinates in meters for the whole video time from the known field dimensions (68x105m) (0-68 x 0-105 m). Figure 5 shows the results of changing the player's coordinates and the field view's viewpoint.



Figure 5. Perspective transformation with marked positions of tracked players

VI. CONCLUSION

This study contributes to the literature by shedding light on the problem of object recognition and tracking in video recordings and offering solutions, with a specific focus on the tracking of football players. Annotations of player whereabouts and football matches are part of the SoccerTrack dataset, which enables accurate assessment. Small identified objects and non-standard picture forms are two of the main obstacles in object detection. Methods including object tracking algorithms, YOLOv8 variants, and sliding window extraction are highlighted in the assessment along with their respective strengths and shortcomings. Beyond the scope of this research, future match analyses may build on the created and evaluated player tracking system's ability to provide precise player coordinates during a whole football match. This work may be used as a basis for future systems that employ a similar technique, despite the fact that player tracking is not flawless and sometimes makes errors. More research into improving tracking robustness, expanding the system to include more objects (such as the ball and referees), and optimizing a low-latency model for object recognition might be done in the future. This could lead to a decrease in processing needs for object detection.

REFERENCES

- [1]Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 1711–1721, 2018
- [2]Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J Seikavandi, Jacob V Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4508–4519, 2021.
- [3]Adrien Maglo, Astrid Orcesi, Quoc-Cuong Pham. Efficient tracking of team sport players with few game-specific annotations. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New orleans, United States. pp.3461-3471, IEEE, Jun 2022.
- [4]Liam Hennessy and Ian Jeffreys. The current use of gps, its potential, and limitations in soccer. Strength & Conditioning Journal, 40(3):83–94, 2018.
- [5]Alejandro Bastida Castillo, Carlos D Gómez Carmona, Ernesto De la Cruz Sánchez, and José Pino Ortega. Accuracy, intra-and inter-unit reliability, and comparison between gps and uwb-based positiontracking systems used for time– motion analyses in soccer. European journal of sport science, 18(4):450–457, 2018.
- [6]Filipe Trocado Ferreira. Video analysis in indoor soccer with a quadcopter. Master’s thesis, Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, 2014.
- [7]Stephen Karungaru, Kenji Matsuura, Hiroki Tanioka, Tomohito Wada, and Naka Gotoda. Ground sports strategy formulation and assistance technology developeent: player data acquisition from drone videos. In 2019 8th International Conference on Industrial Technology and Management (ICITM), pages 322–325. IEEE, 2019.
- [8]P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1. IEEE, 2001, pp. I–I.B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.
- [9]N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 886–893. J. Jones. (1991, May 10)
- [10]P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1–8.M.
- [11]R. B. Girshick, From rigid templates to grammars: Object detection with structured models. Citeseer, 2012.
- [12]A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105
- [13]R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 580-587
- [14]R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448
- [15]S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.