MV-Soccer: Motion-Vector Augmented Instance Segmentation for Soccer Player Tracking

Fahad Majeed[1,*], Nauman Ullah Gilal[1], Khaled Al-Thelaya[1], Yin Yang[1], Marco Agus[1], Jens Schneider[1]

[1] Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar.

{fama44316|jeschneider}@hbku.edu.qa

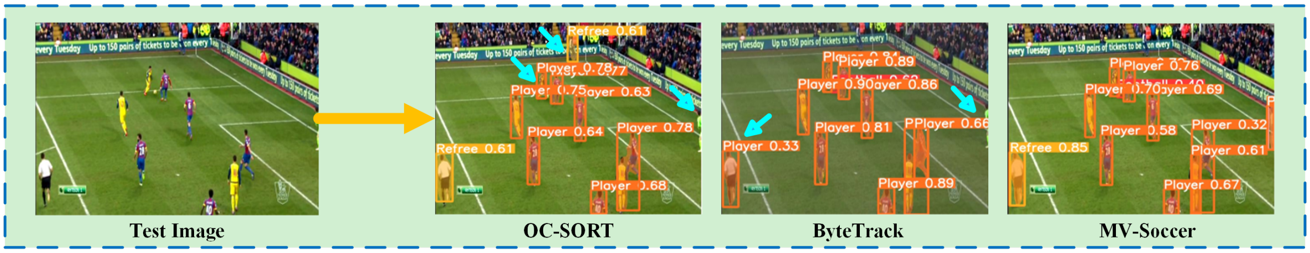


Figure 1. **Overview of *MV-Soccer* Framework:** Our motion vector augmented approach performs detection, instance segmentation and tracking in complex scenarios simultaneously. The test image is given on the left side, and the comparison of our method (MV-Soccer) with SOTA trackers (OC-SORT [9] and ByteTrack [51]) is shown on the right side. The cyan arrows indicate detection errors.

Abstract

*This work presents a novel real-time detection, instance segmentation, and tracking approach for soccer videos. Unlike conventional methods, we augment video frames by incorporating motion vectors, thus adding valuable shape cues that are not readily present in RGB frames. This facilitates improved foreground/background separation and enhances the ability to distinguish between players, especially in scenarios involving partial occlusion. The proposed framework leverages the Cross-Stage-Partial Network*53 *(CSPDarknet*53*) as a backbone, for instance segmentation and integrates motion vectors, coupled with frame differencing. The model is simultaneously trained on two publicly available datasets and a private dataset, SoccerPro, which we created. The reason for simultaneous training is to reduce biases and increase generalization ability. To validate the effectiveness of our approach, we conducted extensive experiments and attained 97% accuracy for the DFL - Bundesliga Data Shootout, 98% on the SoccerNet-Tracking dataset, and an impressive 99% on the SoccerPro (our) dataset.*

1. Introduction

Recent advances in artificial intelligence and deep learning have led to the development and pairing of numerous object detection, segmentation, and tracking algorithms with publicly available datasets [5, 12]. A crucial aspect of computer vision involves the simultaneous detection, segmentation, and tracking of multiple objects within a single frame, widely applicable across diverse domains [47]. Challenges in soccer video instance segmentation and tracking include the presence of diverse objects, edge ambiguity, and shape complexity [12]. Addressing these challenges requires a combination of custom algorithms and techniques. While modern segmentation and tracking methods perform well in situations for which they have been designed (e.g., detecting, counting, and tracking people in crowds or classifying their actions [4]), they fall short under challenging scenarios like soccer matches. This is typically caused by the players' diminutive relative dimensions on the screen, the similar appearance of members of the same team, the everyday use of pan-tilt-zoom cameras for televising games, etc.

A significant number of different industries, both commercial and scientific, use videos and images. Because of the enormous popularity of Artificial Intelligence (AI) in sports and the expansion into new marketplaces (like sport prediction markets), sports, and, specifically, soccer, have become one of the industry segments which has received the greatest attention from the field of video analytics [40]. The data gathered by detecting and tracking players in a soccer match using AI provides information that helps evaluate various tactical elements of the soccer game, including individual and team activities [27]. The extracted information certainly can be beneficial in devising strategies to evaluate and groom both professional and aspiring soccer players, leading to better sportsmen and teams by highlighting flaws and driving attention towards the weak points. When collecting individual player and team statistics from soccer footage, tracking objects is crucial for determining the overall distance run, ball possession time, or team configuration. Interpreting broadcast videos is a challenging problem since it demands effective decision-making despite cuts that break continuity [15, 24]. However, only a few datasets are available for training models and benchmark them in a standardized test environment [13].

Keeping in mind the above mentioned problems and challenges, this paper presents motion vector-based video detection, instance segmentation and tracking for soccer videos incorporating motion vectors. This work is inspired by the observation that motion vectors can provide discontinuities ("edges") between foreground and background and occlusion between soccer players during the game. The main contributions are as follows.

1. We propose a novel framework for real-time soccer player detection, instance segmentation and tracking pipeline using motion vectors generated by a DenseNet-based motion estimation from absolute frame differences called MV-Soccer.
2. We introduce a new dataset, SoccerPro, that comprises $1,495$ mp4 videos of soccer matches.
3. We provide annotations for four classes in consecutive as well as random frames

The proposed framework can perform well on soccer-related and other benchmark video datasets.

2. Related Work

This section reviews the related work in detection, instance segmentation, and tracking. We focus on deep learning, computer vision, and artificial intelligence techniques. These studies highlight open challenges that need to be addressed before successfully meeting the requirements of video instance segmentation and tracking for fully automated recommendation systems.

Detection. Real-time detection of objects enjoys significant attention in computer vision, targeting a wide range of applications such as autonomous driving, medical image analysis, object tracking, and robotics. Akan et al. [3] examine the difficulties associated with soccer video analysis and its application in various groups, such as player/ball detection and tracking, event detection, and game analysis. Several object detection and tracking methods have been proposed. Jiang et al. [30] find that the YOLO (You Only Look Once) family of deep architectures is one of the best detection techniques to date. Wang et al. [46] provide a solution that seamlessly integrates efficient training tools with the proposed architecture and the compound scaling method. The optimization of the training setup and object detection is achieved by proposing adaptable and efficient training tools, leading to the characterization of their optimized method as a "trainable bag-of-freebies".

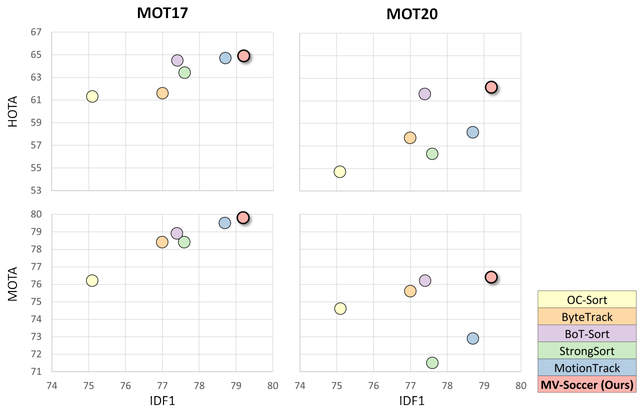


Figure 2. **MV-Soccer Performance:** We compare HOTA (top row) and MOTA (bottom row) metrics for both the MOT17 (left column) and MOT20 (right column) datasets, all plotted over IDF1 on the X-axis. Compared to state-of-the-art approaches (OC-Sort [9], ByteTrack [51], BoT-Sort [2], StrongSort [17], and MotionTrack [38], MV-Soccer (ours) achieves superior performance for all metrics. Note that all approaches perform better for MOT17 than the more challenging MOT20. Also, note that MOTA scores are generally higher (different Y-range between the two rows).

Hurault et al. [28] propose a self-supervised pipeline capable of detecting and tracking low-resolution soccer players under varying recording conditions, eliminating the necessity for ground-truth data. Naik et al. [36] present an approach to the soccer ball and player tracking based on YOLOv3 and Simple Online Real-Time (SORT), aiming to accurately classify detected objects in soccer videos and track them across diverse challenging scenarios. Nergård et al. [37] present an algorithm which detects and annotates the segments of the input dataset automatically. The method detects events using sliding windows and categorises them into a predetermined number of groups.

Segmentation. Video instance segmentation was introduced by Yang et al. [49], accompanied by a dataset named YouTubeVIS, which consists of $2,883$ high-resolution YouTube videos, a 40-category label set and 131k high-quality instance masks. Athar et al. [6] worked on object detection, segmentation, and tracking for rich and complex scenes. Mask-free video instance segmentation achieves high performance using only bounding boxes discussed by Lei et al. [32] to mark objects. The authors validate the proposed scheme on the YouTube-VIS 2019/2021, OVIS and BDD100K MOTS benchmark datasets. The proposed scheme greatly reduces the long-standing gap between fully and weakly supervised VIS on four large-scale benchmarks. Heo et al. [25, 26] develop a generalized framework for

VIS, called GenVIS, to improve memory efficiency. GenVIS achieves SOTA performance on challenging benchmarks without designing complicated architectures or requiring additional postprocessing. The key contribution of their work is the learning strategy, which is a query-based training pipeline for sequential learning with a novel target label assignment. Ghasemzadeh et al. [20] introduce a cohesive framework for automated sports analytics, production, and broadcast, which encompasses tasks such as locating the ball, predicting pose, and segmenting the instance mask of players in team sports scenes.

Tracking. It is always challenging to keep track of players in a soccer game visually due to nuisances such as occlusions, deformations, and changes in camera perspective. Heng et al. [19] introduce an innovative tracking approach termed Motion Features-based Simple Online and Realtime Tracking (MF-SORT). This method prioritizes the motion features of objects in data association, enabling a balanced trade-off between performance and efficiency. Cioppa et al. [14] provide a unique dataset, SoccerNet-Tracking designed for multiple object tracking that comprises 200 challenging soccer scenarios with sequences lasting 30 seconds each, along with a full 45-minute halftime segment for long-term tracking. Manafifard et al. [35] discuss a detailed survey on a player tracking in soccer videos. Yang et al. [48] provide a Cascaded Buffered IoU (C-BIoU) tracker to track multiple objects with irregular motions and indistinguishable appearances. Furthermore, the authors add buffers to expand the matching space of detections and track the irregular matching better than the prior techniques. Iwase et al. [29] propose a solution using a background subtraction method through multiple cameras to track soccer players. Scott et al. [41] propose SoccerTrack, a dataset that includes GNSS and bounding box tracking data annotated on videos recorded using an 8K-resolution fish-eye camera and a 4K-resolution drone camera. This dataset, however, also has some limitations, such as environmental conditions, being recorded on a single pitch, and with a limited number of sets of soccer jerseys.

Multi-object tracking, which combines camera motion compensation and Kalman filters, can track an object in real-time scenarios accurately [1]. The authors report better results in terms of MOT metrics on the MOT17 dataset: 80.5 MOTA, 80.2 IDF1, and 65.0 HOTA. OC-Sort [9] improves the tracking robustness using Kalman Filters in crowded scenes, and, when objects are in non-linear motion, by considering object observations to compute a virtual trajectory over the occlusion period to fix the error accumulation of filter parameters. MotionTrack [38] considers instead a transformer baseline for the MOT in an autonomous driving environment. Lu et al. [33] propose a learning approach to identify and track players in soccer videos. ByteTrack [51] introduces a generic association method that considers every detection box instead of only the high-score ones, utilizing similarities with tracklets to recover true objects and filter out the background detections. At the same time, StrongSORT [17] uses Gaussian-smoothed interpolation (GSI) to compensate for missing detections.

Datasets: The advancements in Multiple Object Tracking (MOT) owe much of their success to the multitude of publicly released datasets available to the community. Yu et al. [50] have undertaken the work closest to ours. The soccer player Multi-Object Tracking (MOT) domain lacks extensive benchmarks, and the available datasets are notably limited in size. Feng et al. [18] introduce the SSET dataset consisting of 282 hours of video, from which 80 tracking sequences are derived. SSET is primarily designed for single-object tracking rather than Multiple Object Tracking. But, our emphasis is on multiple classes, including their positions and trajectories. We thus annotated various timing sequences, encompassing the 90-minute game, the 45-minute half-game, and random sequences from any soccer game, skipping frames that were deemed static based on motion vectors. This marks the inaugural public release of tracking data spanning long, short, and random durations within the sports community.

3. Methodology

Our proposed architecture consists of three components: a backbone to perform feature extraction, a neck to perform video-frame resampling, and a head to perform motion compensation and tracking.

Architecture. The input data is a sequence of frames $(f_1, f_2, \dots, f_N)$, with $f_i \in \mathbb{R}^{H \times W \times M \times n}$, taken using a custom frame rate where M represent the depth of the frame. Fig. 3 shows the step-by-step pipeline of our methodology, while Algorithm. 1 summarizes our motion vector augmented instance segmentation and tracking flow.

Backbone. We feed images of size $608 \times 608 \times 3$ as input and pass them through a series of convolutional layers to extract features. We use the Cross-Stage-Partial Network53 (CSPDarknet53) in this work as a backbone for training, which involves downsampling operations through convolutional and pooling layers to capture features at different scales. The input frame undergoes a segmentation process divided into smaller slices to reduce the image's dimensions to 448×448 before being fed into the convolutional network. Initially, a 1×1 convolution is employed to reduce the channel count, followed by a 3×3 convolution to produce a cuboidal output (also see Fig. 3).

Path Aggregation Network (PAN). After extracting relevant features from the backbone layers, the neck stage concatenates feature maps from various layers of the backbone network and forwards them to the head. At the head stage, these feature maps undergo upsampling via deconvolution to reconstruct images, while the quality is enhanced

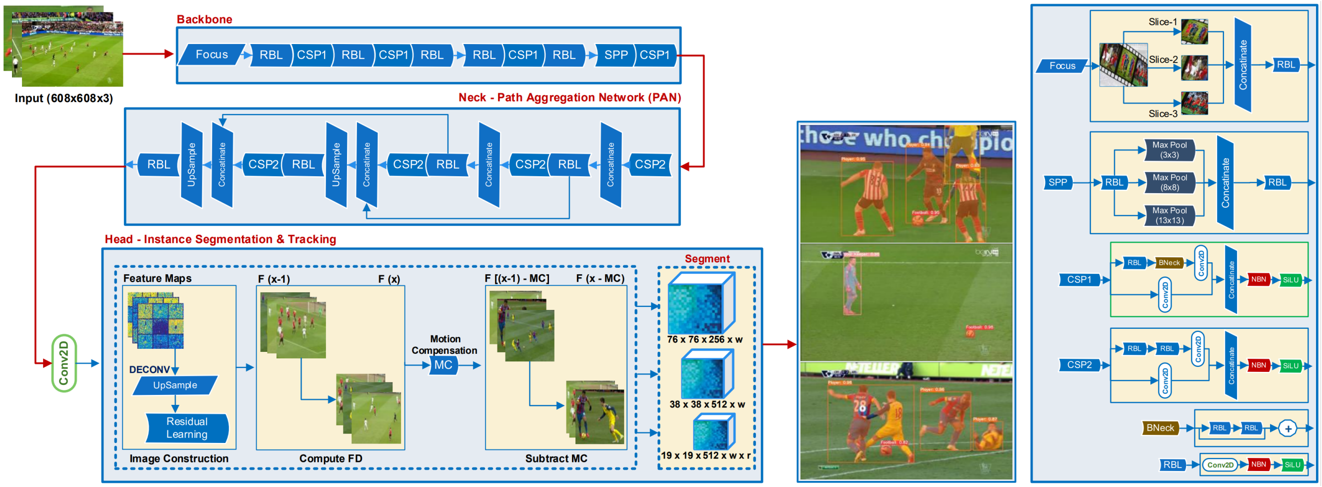


Figure 3. **Left:** view of our MV-Soccer pipeline. The architecture consists of three components. The **Backbone** extracts features from input frames. The **Neck** uses a Path Aggregation Network (PAN) to reduce the feature map's size and increase the features' resolution. The **Head** incorporates the motion vectors to perform motion compensation, instance segmentation and tracking. Three segmentation heads were utilized with their respective dimensions. Insets against a sky blue (right) backdrop provides a detailed view of the partial network modules. From top right to bottom right Focus, SPP, CSP1, CSP2, BottleNeck (BNeck) and Residual Block (RBL) modules.

by residual learning that uses residual blocks or skip connections to enable the network to learn residual mappings. This process facilitates accurate image reconstruction from the feature maps. The neck stage comprises RBL (Residual Block Layer), CSP2 (Cross Stage Partial Network2) and their working is shown in the bottom right corner in Fig. 3.

Motion Vector Extraction. The frames were extracted from videos for analysis using Bommes et al. [8] motion vector extraction technique, "mv-extractor" to determine the motion between them. Afterwards, the frames were decoded to BGR images, motion vectors, frame types and time stamps by utilizing H.264 and H.265 codecs[1] which offer high compression rates, excellent image quality, and widespread support (also see Algorithm 2). The accuracy of the predicted bounding boxes was increased by incorporating object motion between consecutive frames. However, motion vectors may also add shape cues, e.g., players partially occluding each other. Eqn. (1) formalizes the incorporation of motion vectors, for instance segmentation and tracking.

Let (b_i, b_j) be the center coordinates and w, h be the width and height of the predicted bounding box. Then, the motion vector $m = (m_0, m_1)$ (e.g., the polar vector mean of the vectors under the bounding box [5]) is incorporated by utilizing both per-object and per-pixel motion vectors.

$$b_i' = b_i + m_0,\ b_j' = b_j + m_1,\ w' = w,\ h' = h, \tag{1}$$

where b_i', b_j', w', h' are the updated bounding box coordinates with the motion vector incorporated.

Frame Differencing. A traditional, accurate, and robust optical flow estimation to capture the fast and complex motion patterns would be an option for soccer video instance segmentation and tracking but computing full-resolution optical flow is computationally demanding. Instead, we use a DenseNet-based method because of its ability to learn intricate motion patterns and handle complex scenes. To compute differences between two frames f_1 and f_2, we use absolute differences,

$$\Delta_{12}(j,k) = \left|\mathrm{f}_1(j,k) - \mathrm{f}_2(j,k)\right|, \tag{2}$$

where $\mathrm{f}_1(j,k), \mathrm{f}_2(j,k)$ represent the pixel value at coordinates (j,k) in the respective frame. We then convert the difference Δ_{12} to grayscale and threshold it to significant differences. The resulting thresholded image is returned as the output.

Motion Compensation. We predicted the displacement of players between frames by analyzing the motion vectors extracted from consecutive video frames using Lukas Bommes' mv-extractor [8]. This prediction helped us adjust the position of players in subsequent frames, reducing motion-induced distortions and improving the accuracy of segmentation and tracking algorithms.

Tracking Pipeline. The tracking pipeline utilizes a subportion of the BoT-SORT tracker [1] in MV-Soccer for a better tracker experience regarding motion enhancement, embedding and IoU. Algorithm 2 summarizes the working of drawing a motion vector by setting f as the current video frame and m_v as a NumPy array containing motion vectors. The function first checks if there are any motion vectors to draw (if $\mathrm{len}(m\_v) > 0$). Afterwards, it Keeps

[1] https://github.com/LukasBommes/mv-extractor/

Algorithm 1: MV-Soccer

```
Input: Video frame sequence $f_n$
Output: Instance segmentation & tracking results
Initialize the MV-Soccer model;
Load the pre-trained weights;
$f_{cur} \leftarrow$ null;
while Frames available do
    $f_{prev} \leftarrow f_{cur}$;
    $f_{cur} \leftarrow$ PREPROCESS($f_{next}$);
    /* Backbone */
    $x \leftarrow$ BACKBONE($f_{cur}$);
    $x \leftarrow$ DOWNSAMPLE(CONV(RBL($x$)));
    /* Neck (PAN) */
    $x \leftarrow$ CONCAT(UPSAMPLE($x$));
    /* Detection Head */
    $y \leftarrow$ OBJECT_DETECTION($x$);
    $z \leftarrow$ NON_MAX_SUPPRESS($x$);
    /* Mask Head */
    $y \leftarrow$ MASK_PREDICTION($y$);
    /* Post-processing */
    $y \leftarrow$ SCALE(UPSCALE($y$));
    if $f_{prev}$ =null then
        yield SEGMENT($y, z$);
    /* Frame Differencing */
    $\Delta \leftarrow |f_{cur} - f_{prev}|$;
    $m \leftarrow$ MOTION_VECTORS($\Delta$);
    /* Segmentation & Tracking */
    yield SEGMENT_AND_TRACK($y, z, \Delta, m$);
```

Algorithm 2: Draw Motion Vectors

```
Procedure Draw_MV($f$, $m\_v$)
if $l(m\_v) > 0$ then
    $num\_mvs \leftarrow$ shape$(m\_v)[0]$
    for $mv$ in split($m\_v$, $num\_mvs$) do
        $s\_pt \leftarrow (mv[0,3], mv[0,4])$
        $e\_pt \leftarrow (mv[0,5], mv[0,6])$
        cv2.arrowedLine($f, s\_pt, e\_pt, (0,0,255), 1,$
        cv2.LINE_AA$, 0, 0.1)$
    end for
end if
return $f$
```

iterating until it gets motion vectors, and for each motion vector, it extracts the starting point (s_pt) and ending point (e_pt) from the vector. It then uses cv2.arrowedLine to draw an arrow on the motion vector's frame. Frame differences are used to compute per-pixel motion estimation using a DenseNet. We then select motion vectors at the centroid of each moving object to perform motion compensation for the tracking stage. In addition, we also utilize per-pixel motion vectors. They are stacked onto the RGB image to serve as input for segmentation. In this fashion, we simultaneously apply detection, segmentation, and tracking (cf. Fig. 1). Fig. 4 presents a graphical overview.

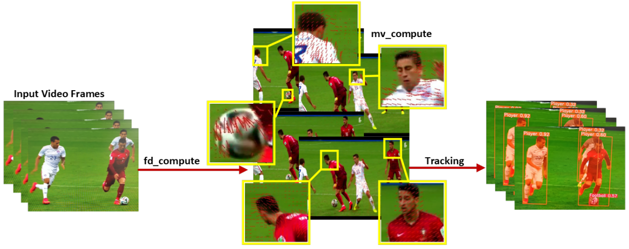


Figure 4. **MV-Soccer Tracking:** left-to-right, we give frames as input, followed by computing the frame differencing and passing them to the next stage to compute the motion vectors. Finally, per-pixel motion vectors are applied for better segmentation and tracking.

Post-Processing. For generating instance masks of each object, we leverage features from the extracted motion vectors, employing a combination of upsampling and convolutional layers to produce a binary mask for individual objects. Three segmentation heads were utilized with dimensions in height and width of 76 × 76, 38 × 38 and 19 × 19. Finally, the detected and segmented objects and their masks are post-processed to refine the bounding boxes and remove duplicates. This involves thresholding the confidence scores to 0.25, performing non-maximum suppression to remove overlapping boxes, and applying a mask to each remaining box to generate the final output.

Intuition for using Motion Vectors. The use of motion vectors in our pipeline is based on the assumption that soccer players moving on the field will be outlined by motion vector discontinuities. In particular, a static background (that is, in a full shot, static camera scenario) will have little to no motion vector magnitude. In this scenario, we believe motion vectors help in foreground/background separation, providing additional shape cues about the desired instance segmentation that are either not or only to a limited extent available in RGB images. Fig. 5, top and middle rows, shows an example of this scenario. Motion vectors are color mapped using the commonly used "HSV-to-RGB" approach: The magnitude is mapped to the V (value or luminance channel), whereas the direction of normalized motion vectors is mapped to the H (hue), a 360° color wheel. We set S (the saturation) to a constant of 0.5. In addition, we provide images of the edges in both the RGB and motion vector data, extracted using a 3 × 3 Sobel filter. Note that the Sobel edges of the motion vectors were computed using the motion vectors themselves and not the color-mapped image. Even in scenarios where the background is not static, e.g., due to the use of PTZ cameras,

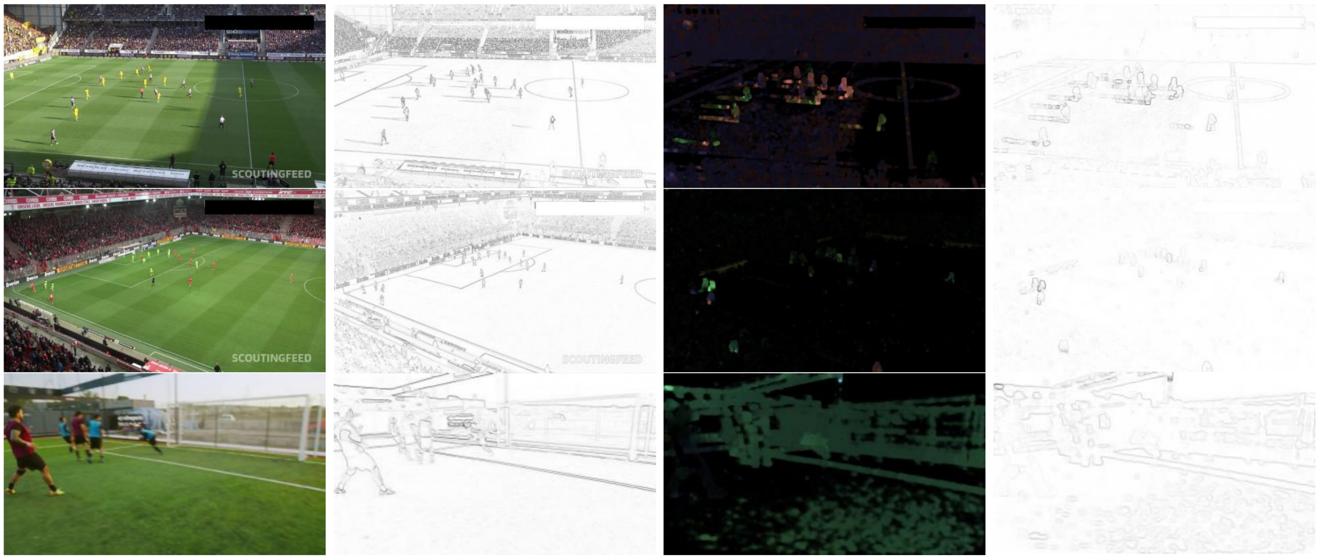


Figure 5. Visual motivation for including motion vectors. Top and middle rows: relatively static scenarios from the SoccerNet-Tracking repository dataset shot with a professional PTZ camera at 1080p (sequence fdf84965_0 & fdf84965_1). Bottom row: dynamic camera from the GitHub repository, likely to be recorded on a mobile phone at 720p. In each row, left to right: RGB image, Sobel edges of the RGB image, color-coded motion vectors, and edges of the motion vectors.

we observe that motion vectors tend to stand out against the background. Fig. 5, bottom, shows an example of this scenario (from GitHub repository). The video sequence appears to have been recorded using an un-stabilized mobile phone at 720p resolution. Note that the sequence depicted is very challenging for the following reasons. (1) The mobile phone is not stabilized using a gimbal and produces significant additional motion. (2) The video is blurry due to the mobile phone's lack of a proper optical lens array. (3) The phone's resolution is fairly low. As a result, the motion vectors have a significantly lower resolution and higher noise than in the previous scenario. Despite the challenges imposed by noisy and low-resolution motion vectors, the results show improved performance gains over existing frameworks that are using only RGB images.

4. Results

We analyzed the proposed methodology using the two benchmark datasets and one private dataset, SoccerPro, which we created. We used the pre-trained model provided by Torchvision, specifically trained on the COCO dataset. The model was trained simultaneously on all three datasets. All hyperparameters remained constant during the complete training procedure.

Training Setup. All experiments were performed on a machine running Ubuntu 22.04 with 512GB RAM, a Xeon(R) Gold 6226R CPU and an NVIDIA RTX 3090 GPU with 24GB RAM. The complete model was implemented in Python using Jupyter Notebook. The PyTorch framework was used due to its high stability. We used the AdamW optimizer with a learning rate scheduler $(10^{-5} \dots 0.01)$, momentum of 0.6, and a batch size of 16 and trained for 100 epochs.

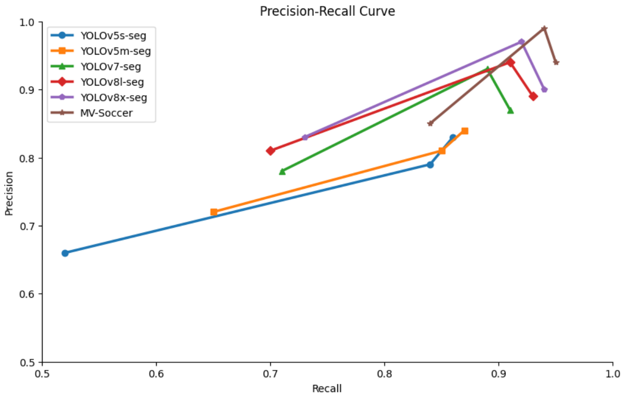


Figure 6. Precision-Recall curve for all five Instance Segmentation models and their comparison with MV-Soccer on all three datasets.

Datasets. This work uses the following three datasets. We used the complete DFL- Bundesliga Data Shootout dataset along with the subset of the SoccerNet-Tracking dataset and merged them with the SoccerPro dataset.

DFL - Bundesliga Data Shootout: The total size of this dataset is 37.55 GB consisting of videos acquired from DEUTSCHE FUSSBALL LIGA (DFL[2]), the German na-

[2] https://www.kaggle.com/competitions/dfl-bundesliga-data-shootout/data, Jan 2023.

tional football association. The videos in the dataset are encoded as mp4, with metadata provided in csv-format. 246 files in this dataset are partitioned into three parts: clips, test, and train [31].

SoccerNet-Tracking: The total size of this dataset is 187.8GB [10, 11, 13, 21–23] consisting of a 500 video dataset of soccer games from the six big European leagues, including Premier League (England), UEFA Champions League, Ligue-1 (France), Bundesliga (Germany), Serie-A (Italy), and LaLiga (Spain). The SoccerNet-Tracking dataset consists of 12 full soccer games captured from the primary camera with 1080p resolution at 25FPS. From this, a dataset comprising 200 clips lasting 30 seconds each was derived, accompanied by tracking data.

SoccerPro: The total size of this dataset is 4.7 TB, comprising $1,459$ videos of soccer games from English, EuroChamp, French, German, Italian, and Spanish leagues. Overall, our dataset consists of 47 full games: 13 full games were captured at a resolution of 720p and recorded at 50 FPS, 17 full games were captured at a resolution of 1080p and recorded at 30 FPS, and the remaining 17 full games were captured at a resolution of 1280p and recorded at 30 FPS.

Table 1. Comparative Analysis: Benchmarks and SoccerPro(ours)

Source	Length	Duration	Games	Field	Task
DFL	$2,00$	$3,0s$	9	Soccer	MOT
SN-Tracker	$2,01$	$5,0m$	$1,2$	Soccer	MOT
SoccerPro	$1,459$	$5,0m$	$4,7$	Soccer	MOT
Total	**1,860**	–	**68**	–	–

Dataset Creation and Annotation Pipeline. For creating a dataset, frames were extracted from videos with the help of a Python (videos-to-frames) conversion script. The extracted frames were categorized into consecutive and random (nearfield, midfield, and widefield) frames. Finally, the extracted frames were stored on disk.

Afterwards, we utilized Roboflow's[3] smart polygon (single click) feature to annotate and label our private SoccerPro dataset as well as the other two benchmark datasets. We annotated a subset of all three datasets and split them into three parts, i.e., 70% training, 20% validation, and 10% for testing (also see Fig. 7). We resized frames to 640 × 640 before subjecting them to augmentation, including adjustments to brightness and saturation within a range of ±25%, rotation ±15%, clockwise rotation of 90°, and horizontal flipping. In Tab. 2, we summarize the details of the annotated dataset along with their unique tracklets and number of bounding boxes.

Table 2. Annotated Dataset Details

Class	Unique Tracklets	Bounding Boxes
Player	$2,478$	$1,428,274$
Goalkeeper	378	$150,972$
Referee	$7,22$	$422,738$
Football	$5,94$	$309,824$
Total	**4,172**	**2,311,808**

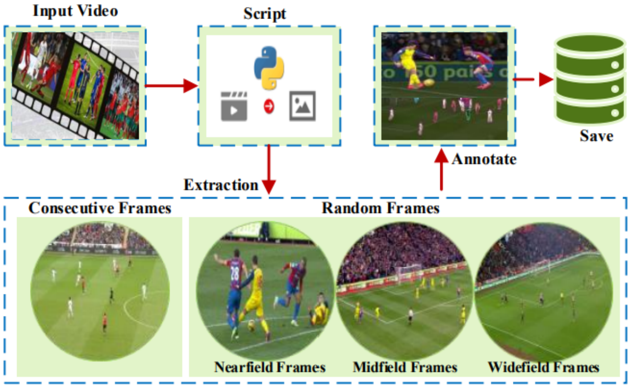


Figure 7. Dataset Creation and Annotation Pipeline.

Quantitative Comparisons.

Instance Segmentation: We use four metrics, i.e., Accuracy, Precision, Recall, and Mean Average Precision (mAP) to evaluate the instance segmentation model on benchmark and SoccerPro datasets see (supplementary materials Fig. S.3). Fig. 6 shows the precision-recall curve to evaluate the performance of our model. We computed the class-based score of all the classes and their model to analyse how well the method detects and segments each class separately see (supplementary materials Tab. 6). We also evaluated the combined classes' output along with their model's best results to analyse the overall performance of the method for detection and instance segmentation (cf. Tab. 3).

Table 3. **Quantitative Results:** Comparative Analysis of YOLO (v5, v7, and v8) and the proposed MV-Soccer, for instance segmentation on all three datasets.

Models	Precision	Recall	mAP$^{\text{box}}_{50-95}$	mAP$^{\text{mask}}_{50-95}$	Time$_{RTX3090}$
YOLOv5s-seg [43]	0.66	0.52	0.54	0.51	1.1ms
YOLOv5m-seg [43]	0.72	0.55	0.61	0.62	1.7ms
YOLOv7-seg [45]	0.78	0.71	0.70	0.67	11.4ms
YOLOv8l-seg [44]	0.72	0.70	0.72	0.70	14.2ms
YOLOv8x-seg [44]	0.79	0.73	0.76	0.74	18.3ms
MV-Soccer	**0.84**	**0.75**	**0.79**	**0.78**	**20.7ms**

Tracking: We evaluate the performance of our model for motion enhancement, embedding, and IoU against the most recent SOTA trackers available in literature: OC-SORT [9], MotionTrack [38], StrongSORT [17], ByteTrack [51] and BoT-SORT [1]. As summarized in Tab. 4, our approach consistently outperforms the other computer vision models regarding instance segmentation and tracking. To assess the tracking performance of our approach, we used three metrics: HOTA [34], MOTA [7], and IDF1 [39]. We also com-

[3] https://app.roboflow.com/fahad-majeed/

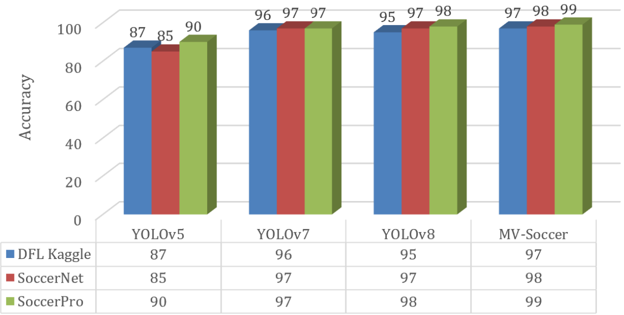
Table 4. Comparison of the tracking inference speed on the validation set of $MOT17$ and training set of $MOT20$ [16].

Trackers	MOT17				MOT20			
	HOTA↑	MOTA↑	IDF1↑	FPS↑	HOTA↑	MOTA↑	IDF1↑	FPS↑
Enhanced Motion:								
OC-SORT [9]	61.3	76.2	75.1	23.4	54.7	74.6	69.7	19.3
MotionTrack[38]	64.7	79.5	78.7	13.2	58.2	72.9	68.6	8.4
Embedding:								
StrongSORT [17]	63.4	78.4	77.6	9.3	56.3	71.5	70.2	6.7
IoU only:								
ByteTrack [51]	61.6	78.4	77.0	18.3	57.7	75.6	69.3	14.4
BoT-SORT [1]	64.5	78.9	77.4	8.5	61.6	76.2	74.7	7.6
MV-Soccer	**64.9**	**79.8**	**79.2**	**28.7**	**64.7**	**79.2**	**78.5**	**24.5**

Table 5. Overall Best Performance Comparison of Tracking Methods on combined MOT17 Validation and MOT20 Training Datasets.

Tracker	(w) Motion Vectors				(w/o) Motion Vectors			
	HOTA	MOTA	IDF1	FPS	HOTA	MOTA	IDF1	FPS
Enhanced Motion								
OC-SORT [9]	61.3	76.2	75.1	23.4	-	-	-	-
MotionTrack [38]	64.7	79.5	78.7	13.2	-	-	-	-
Embedding								
StrongSORT [17]	-	-	-	-	63.4	78.4	77.6	9.3
IoU only								
ByteTrack [51]	-	-	-	-	61.6	78.4	77.0	18.3
BoT-SORT [1]	-	-	-	-	64.5	78.9	77.4	8.5
MV-Soccer	**64.9**	**79.8**	**79.2**	**28.7**	**64.7**	**79.2**	**78.5**	**24.5**

pare the overall best performance of our model based on *with* (w) motion vectors and *without* (w/o) motion vectors on MOT17 validation and MOT20 training datasets simultaneously (cf. Tab. 5) and for their individual performances on both datasets see (supplementary materials Tab. 7). The overall accuracy of all the models compared with the proposed MV-Soccer for Instance segmentation and tracking is shown in Fig. 8.



	YOLOv5	YOLOv7	YOLOv8	MV-Soccer
🟦 DFL Kaggle	87	96	95	97
🟥 SoccerNet	85	97	97	98
🟩 SoccerPro	90	97	98	99

Figure 8. Comparison of instance segmentation and tracking accuracies between YOLO (v5, v7, and v8) and MV-Soccer.

Qualitative Comparisons. The proposed MV-Soccer framework performs better in instance segmentation and tracking soccer players than existing SOTA methods. Leveraging Bommes' MV-Extractor technique, MV-Soccer improves precision and recall across various classes, specifically the player class. This is evidenced by the substantially higher class-based scores attained by MV-Soccer, as demonstrated in the supplementary materials, Tab. 6.
The scheme shows robustness and accuracy in segmenting soccer players, excelling in challenging scenarios such as occlusion and diverse player poses. It also showcases remarkable generalisation capabilities by meticulously evaluating benchmark datasets (MOT17 and MOT20) and the SoccerPro dataset, ensuring consistent and reliable performance across diverse soccer tracking scenarios. Regarding computational efficiency, MV-Soccer maintains competitive real-time performance while achieving superior accuracy. The inference speed of MV-Soccer is commendable, especially considering its enhanced tracking accuracy compared to recent SOTA trackers, as shown in Tab. 4. Furthermore, MV-Soccer exhibits exceptional tracking accuracy, outperforming contemporary trackers in metrics such as HOTA, MOTA, and IDF1. This highlights the efficacy of MV-Soccer in accurately tracking soccer players' movements, which is crucial for applications in sports analytics and player performance analysis.

5. Conclusion

In this paper, we presented MV-Soccer, a real-time detection, instance segmentation and tracking approach using motion vectors. Our proposed framework leverages the Cross-Stage Partial Network53 (CSPDarknet53) as a backbone for instance segmentation coupled with motion vectors. We obtained motion vectors using a DenseNet motion estimator on absolute frame differences. To evaluate the models' confidence, extensive experiments were performed using current and previous versions of YOLO(v5, v7, and v8), along with a detailed comparative analysis with our MV-Soccer model. In addition, we also evaluated our model on the validation set of $MOT17$ and the training set of $MOT20$ dataset. Our method achieved 97% accuracy for the DFL - Bundesliga Data Shootout, 98% on the SoccerNet-Tracking dataset and 99% on our SoccerPro dataset. The proposed model achieved a tracking rate of 50 frames per second on an NVIDIA RTX3090.

In future, we plan to extend our work by using the complete SoccerNet-Tracking dataset for training, validation, and testing. We will extend our work with a focus on measuring the position and speed of the player, pose estimation, and action recognition.

6. Acknowledgements

This publication was made possible through the financial support of NPRP grants NPRP14S-0404-210137 from the Qatar National Research Fund, now known as the Qatar Research Development and Innovation Council (QRDI). The findings and opinions expressed herein are solely the responsibility of the authors.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 3, 4, 7, 8, 5

[2] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 2

[3] Sara Akan and Songül Varlı. Use of deep learning in soccer videos analysis: survey. *Multimedia Systems*, 29(3):897–915, 2023. 2

[4] A. Al-Shaery, A.S. Alshehri, N.S. Farooqi, and M.O. Khozium. In-depth survey to detect, monitor and manage crowd. *IEEE Access*, 8:209,008–209,019, 2022. 1

[5] S.R. Alvar and I.V. Bajić'. MV-YOLO: Motion vector-aided tracking by semantic object detection. arXiv:1805.00107v2, 2018. 1, 4

[6] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1674–1683, 2023. 2

[7] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008. 7

[8] L. Bommes, X. Lin, and J. Zhou. Mvmed: Fast multi-object tracking in the compressed domain. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1419–1424, 2020. 4

[9] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9686–9696, 2023. 1, 2, 3, 7, 8, 5

[10] Anthony Cioppa, Adrien Deliege, Floriane Magera, Silvio Giancola, Olivier Barnich, Bernard Ghanem, and Marc Van Droogenbroeck. Camera calibration and player localization in soccernet-v2 and investigation of their representations for action spotting. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pages 4532–4541, 2021. 7

[11] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Scaling up SoccerNet with multi-view spatial localization and re-identification. *Sci. Data*, 9(1):1–9, 2022. 7

[12] Anthony Cioppa, Silvio Giancola, Adrien Deliège, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Marc Van Droogenbroeck. Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3490–3501, 2022. 1

[13] Anthony Cioppa, Silvio Giancola, Adrien Deliege, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Marc Van Droogenbroeck. SoccerNet-Tracking: Multiple Object Tracking Dataset and Benchmark in Soccer Videos. arXiv:2204.06918, 2022. 2, 7

[14] Anthony Cioppa, Silvio Giancola, Adrien Deliège, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Marc Van Droogenbroeck. Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3491–3502, 2022. 3

[15] 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. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pages 4503–4514, 2021. 2

[16] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, 129:845–881, 2021. 8

[17] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023. 2, 3, 7, 8, 5

[18] Na Feng, Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, Yizhu Zhao, Yunfeng He, and Tao Guan. Sset: a dataset for shot segmentation, event detection, player tracking in soccer videos. *Multimedia Tools and Applications*, 79:28971–28992, 2020. 3

[19] Heng Fu, Lifang Wu, Meng Jian, Yuchen Yang, and Xiangdong Wang. Mf-sort: Simple online and realtime tracking with motion features. In *Image and Graphics: 10th International Conference, ICIG 2019, Beijing, China, August 23–25, 2019, Proceedings, Part I 10*, pages 157–168. Springer, 2019. 3

[20] Seyed Abolfazl Ghasemzadeh, Gabriel Van Zandycke, Maxime Istasse, Niels Sayez, Amirafshar Moshtaghpour, and Christophe De Vleeschouwer. Deepsportlab: a unified framework for ball detection, player instance segmentation and pose estimation in team sports scenes. *arXiv preprint arXiv:2112.00627*, 2021. 3

[21] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pages 4485–4494, 2021. 7

[22] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. SoccerNet: A scalable dataset for action spotting in soccer videos. *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, 2018-June:1792–1802, 2018.

[23] Silvio Giancola, Anthony Cioppa, Adrien Deliège, Floriane Magera, Vladimir Somers, Le Kang, Xinxing Zhou, Olivier Barnich, Christophe De Vleeschouwer, Alexandre Alahi, Bernard Ghanem, Marc Van Droogenbroeck, Abdulrahman Darwish, Adrien Maglo, Albert Clapés, Andreas Luyts, Andrei A. Boiarov, Artur Xarles, Astrid Orcesi, Avijit Shah, Baoyu Fan, Bharath Comandur, Chen Chen, Chenle Zhang, Chen Zhao, Che-Hsien Lin, Cheuk-Yiu Chan, Chun Chuen Hui, Dengjie Li, Fan Yang, Fan Liang, Fang Da, F. L. Yan, Fufu Yu, Guanshuo Wang, H. Anthony Chan, He Zhu,

Hongwei Kan, Jiaming Chu, Jianming Hu, Jianyang Gu, Jin Chen, João V. B. Soares, Jonas Theiner, Jorge De Corte, José Henrique Brito, Jun Zhang, Junjie Li, Junwei Liang, Leqi Shen, Lin Ma, Lin Chen, M L Santos Marqués, Mike Azatov, N. I. Kasatkin, Ning Wang, Qi Jia, Quoc Cuong Pham, Ralph Ewerth, Ran Song, Rengang Li, Rikke Gade, Ruben Debien, Runze Zhang, Sangrok Lee, Sergio Escalera, Shan Jiang, Shigeyuki Odashima, Shi-Jin Chen, Shoichi Masui, Shouhong Ding, Sin wai Chan, Siyu Chen, Tallal El-Shabrawy, Tao He, Thomas Baltzer Moeslund, W. C. Siu, Wei Zhang, W. Li, Xian Wang, Xiao Tan, Xiaochuan Li, Xiaolin Wei, Xiaoqing Ye, Xing Liu, Xinying Wang, Yan Guo, Yaqian Zhao, Yi Yu, Yingying Li, Yue He, Yujie Zhong, Zhenhua Guo, and Zhiheng Li. Soccernet 2022 challenges results. *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*, 2022. 7

[24] Ahmad Hammoudeh, Bastein Vanderplaetse, and Stéphane Dupont. Deep soccer captioning with transformer: dataset, semantics-related losses, and multi-level evaluation. arXiv:2202.05728, 2022. 2

[25] Miran Heo, Sukjun Hwang, Jeongseok Hyun, Hanjung Kim, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. A generalized framework for video instance segmentation. In *arXiv preprint arXiv:2211.08834*, 2022. 2

[26] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. Vita: Video instance segmentation via object token association. In *Advances in Neural Information Processing Systems*, 2022. 2

[27] Samuel Hurault, Coloma Ballester, and Gloria Haro. Self-Supervised Small Soccer Player Detection and Tracking. *MMSports 2020 - Proc. 3rd Int. Work. Multimed. Content Anal. Sport.*, pages 9–18, 2020. 1

[28] Samuel Hurault, Coloma Ballester, and Gloria Haro. Self-supervised small soccer player detection and tracking. In *Proceedings of the 3rd international workshop on multimedia content analysis in sports*, pages 9–18, 2020. 2

[29] Sachiko Iwase and Hideo Saito. Parallel tracking of all soccer players by integrating detected positions in multiple view images. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 751–754. IEEE, 2004. 3

[30] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022. 2

[31] Kaggle. DFL - Bundesliga Data Shootout — Kaggle. https://www.kaggle.com/competitions/dfl-bundesliga-data-shootout/data, 2022. 7

[32] Lei Ke, Martin Danelljan, Henghui Ding, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask-free video instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22857–22866, 2023. 2

[33] Wei-Lwun Lu, Jo-Anne Ting, James J Little, and Kevin P Murphy. Learning to track and identify players from broadcast sports videos. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1704–1716, 2013. 3

[34] Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129:1–31, 2021. 7

[35] Mehrtash Manafifard, Hamid Ebadi, and H Abrishami Moghaddam. A survey on player tracking in soccer videos. *Computer Vision and Image Understanding*, 159:19–46, 2017. 3

[36] B Thulasya Naik and Md Farukh Hashmi. Yolov3-sort: detection and tracking player/ball in soccer sport. *Journal of Electronic Imaging*, 32(1):011003–011003, 2023. 2

[37] Olav A. Nergård Rongved, Steven A. Hicks, Vajira Thambawita, Håkon K. Stensland, Evi Zouganeli, Dag Johansen, Michael A. Riegler, and Pal Halvorsen. Real-Time Detection of Events in Soccer Videos using 3D Convolutional Neural Networks. *Proc. - 2020 IEEE Int. Symp. Multimedia, ISM 2020*, pages 135–144, 2020. 2

[38] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17939–17948, 2023. 2, 3, 7, 8, 5

[39] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. arXiv:1609.01775v2, 2016. 7

[40] Melissa Sanabria, Frédéric Precioso, Pierre-Alexandre Mattei, and Thomas Menguy. A Multi-stage deep architecture for summary generation of soccer videos. arXiv:2205.00694, 2022. 1

[41] Atom Scott. SoccerTrack: A dataset and tracking algorithm for soccer with fish-eye and drone videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3569–3579, 2022. 3

[42] Tijeni. Upgraded YOLO with object augmentation. *Operations Research Forum*, 3:#60, 2022. 1

[43] ultralytics. YOLOv5 Instance Segmentation. https://colab.research.google.com/github/ultralytics/yolov5/blob/master/segment/tutorial.ipynb, accessed 30-Jan-2024, 2022. 7

[44] ultralytics. YOLOv8 Instance Segmentation. https://github.com/ultralytics/ultralytics, accessed 30-Jan-2024, 2023. 7

[45] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv:2207.02696*, 2022. 7

[46] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023. 2

[47] Feng Yang, Xingle Zhang, and Bo Liu. Video object tracking based on YOLOv7 and DeepSORT. arXiv:2207.12202, 2022. 1

[48] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the match-

ing space. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4799–4808, 2023. 3

[49] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[50] Junqing Yu, Aiping Lei, Zikai Song, Tingting Wang, Hengyou Cai, and Na Feng. Comprehensive dataset of broadcast soccer videos. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 418–423. IEEE, 2018. 3

[51] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 7, 8, 5