ReST: High-Precision Soccer Player Tracking via Motion Vector Segmentation

Fahad Majeed[1][a], Khaled Ahmed Lutf Al Thelaya[b], Nauman Ullah Gilal[c],
Kamilla Swart-Arries[d], Marco Agus[e] and Jens Schneider[f]
College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar
{fama44316, khalthelaya, ngilal, kswartarries, magus, jeschneider}@hbku.edu.qa

Keywords: Soccer Player Tracking, Motion Vectors, Computer Vision, Instance Segmentation, Sports Analytics.

Abstract: We present a novel real-time framework for the detection, instance segmentation, and tracking of soccer players in video footage. Our method, called ReST, is designed to overcome challenges posed by complex player interactions and occlusions. This is achieved by enhancing video frames by incorporating motion vectors obtained using the Scharr filter and frame differencing. This provides additional shape cues over RGB frames that are not considered in traditional approaches. We use the Generalized Efficient Layer Aggregation Network (GELAN), combining the best qualities of CSPDarknet53 and ELAN as a robust backbone for instance segmentation and tracking. We evaluate our method rigorously on both publicly available and our proprietary (SoccerPro) datasets to validate its performance across diverse soccer video contexts. We train our model concurrently on multiple datasets, thereby improving generalization and reducing dataset bias. Our results demonstrate an impressive 97% accuracy on the DFL Bundesliga Data Shootout, 98% on SoccerNet-Tracking, and 99% on the SoccerPro dataset. These findings underscore the framework's efficacy and practical relevance for advancing real-time soccer video analysis.

1 INTRODUCTION

Soccer player tracking is essential for coaches, analysts, and sports scientists. By analyzing player movements, teams gain insights into performance, strategy, and overall game dynamics. Advanced tracking systems provide detailed analysis of player positioning, speed, acceleration, and distance covered, which are crucial for optimizing training programs, enhancing strategic decision-making, and improving player performance (Bialkowski et al., 2014b). Recent technological advancements have enabled the collection of high-resolution spatial and temporal data. Systems like GPS trackers, optical tracking systems, and wearables have revolutionized player movement monitoring and analysis (Baysal and Duygulu, 2016; Csanalosi et al., 2020). Player tracking aids in performance analysis and plays a vital role in injury prevention and rehabilitation by identifying risky movement patterns and managing workloads (Ehrmann et al., 2016; Khaustov and Mozgovoy, 2020). Motion vectors, a crucial component of video compression technologies, represent the motion of pixel blocks between consecutive frames in a video sequence (Xu et al., 2016; Furht et al., 1997; ITU-T,

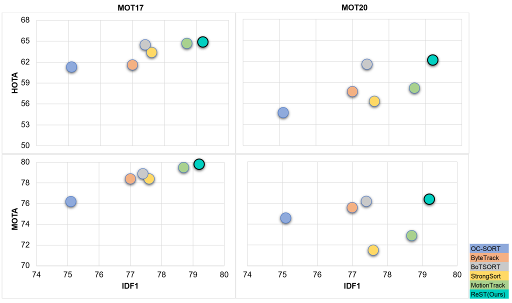

Figure 1: Comparison of HOTA (top row) and MOTA (bottom row) for both the MOT17 (left column) and MOT20 (right column) datasets (X-axis: IDF_1; see Sec. 4.3 for a description of these metrics) with the state-of-the-art (OC-Sort (Cao et al., 2023), ByteTrack (Zhang et al., 2022), BoT-Sort (Aharon et al., 2022), StrongSort (Du et al., 2023), and MotionTrack (Qin et al., 2023). Our method, ReST, achieves superior performance for all metrics.

[a] https://orcid.org/0000-0001-8447-2686
[b] https://orcid.org/0000-0003-2363-4586
[c] https://orcid.org/0000-0002-8197-4696
[d] https://orcid.org/0000-0002-2778-2802
[e] https://orcid.org/0000-0003-2752-3525
[f] https://orcid.org/0000-0002-0546-2816

Majeed, F., Al Thelaya, K. A. L., Gilal, N. U., Swart-Arries, K., Agus, M. and Schneider, J.
ReST: High-Precision Soccer Player Tracking via Motion Vector Segmentation.
DOI: 10.5220/0013168000003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 2: VISAPP*, pages 138-149
ISBN: 978-989-758-728-3; ISSN: 2184-4321

2021). By estimating object motion in a video, motion vectors enable efficient compression by predicting object movement from one frame to another.

In soccer player tracking, motion vectors can enhance tracking accuracy and efficiency. Traditional methods often rely on visual features or markers, which can be computationally intensive and prone to errors, especially in occluded or crowded scenes. Motion vectors offer a robust alternative by leveraging inherent motion information in video sequences, allowing for more accurate real-time player tracking (Majeed et al., 2024). Instance segmentation, a critical task in computer vision, involves identifying and delineating each object instance within an image or video frame (He et al., 2017). In sports analytics, instance segmentation plays a pivotal role by enabling precise localization and tracking of individual players amidst occlusion and complex interactions (Kirillov et al., 2019).

In this paper, we introduce a novel real-time Recognition, Segmentation, and Tracking (ReST) approach that uses enhanced motion vector instance segmentation to achieve high-precision tracking of soccer players. Our approach leverages advanced motion vector analysis to enhance player identification and tracking accuracy, particularly in challenging scenarios such as occlusions and crowded environments. By combining recognition, segmentation, and tracking in a unified framework, our method ensures that each player's position and identity are continuously monitored and updated with high precision. Furthermore, we address computational complexity by developing efficient algorithms for real-time analysis on standard hardware (Manafifard et al., 2017). Additionally, our system scales to large datasets and multiple players and integrates diverse data sources into a unified framework for comprehensive player movement analysis (Wehbe et al., 2014; Naik et al., 2022; Diop et al., 2022). Our unified framework, ReST, adds the following to the existing literature.

1. Motion vector instance segmentation—our novel approach provides accurate and dependable player tracking under challenging conditions (e.g., camera movement, zoom, partial occlusions).
2. SoccerPro—a new dataset featuring 1,495 soccer match mp4 videos with annotations with four classes for both consecutive and random frames.
3. ReST architecture—a new framework minimizing computational complexity and facilitating highly accurate real-time analysis on consumer hardware.
4. Scalability— Our system copes with large-scale datasets and multiple players effectively while delivering consistent and dependable performance.
5. Multimodality— Our unified framework uses multiple data sources, enabling a comprehensive understanding of player movement and enhancing tracking performance.

2 RELATED WORK

2.1 Segmentation

Video instance segmentation was pioneered with the introduction of the YouTube-VIS dataset (Yang et al., 2019). This dataset comprises 2,883 high-resolution YouTube videos annotated with a 40-category label set and 131k high-quality instance masks. Later benchmark data sets such as BURST (Athar et al., 2023) have contributed significantly to object detection, segmentation, and tracking in complex scenes. This has led to mask-free video instance segmentation (Ke et al., 2023) using only bounding boxes for object delineation. The latter method has been validated on benchmark datasets like YouTube-VIS 2019/2021, OVIS, and BDD100K MOTS, narrowing the gap between fully and weakly supervised video instance segmentation methods.

Aiming at improving memory efficiency, the generalized framework GenVIS (Heo et al., 2022; Heo et al., 2023) achieves state-of-the-art performance on challenging benchmarks without relying on complex architectures or additional post-processing. GenVIS uses an innovative learning strategy that involves a query-based training pipeline for sequential learning with novel target label assignment methods. DeepSportLab (Ghasemzadeh et al., 2021) is a comprehensive framework for automated sports analytics, production, and broadcast. The framework addresses tasks such as ball localization, pose prediction, and instance mask segmentation of players in team sports scenes, significantly advancing the field of sports video analysis.

2.2 Tracking

Traditional player tracking methods in soccer relied predominantly on visual features and markers. They employ cameras to capture player movements and use computer vision for analysis (Mazzeo et al., 2008). One of the earliest and most widely used approaches involves using multiple fixed cameras around the stadium. Systems like ProZone and TRACAB combine data from several cameras to create a 3D reconstruction of the playing field and to track players (Cintia et al., 2015; Linke et al., 2020). Despite their effectiveness, these methods often struggle with occlusion (players blocking each other from the cameras' view). Furthermore, the need for real-time image processing can make them computationally expensive (Zhang, 2012).

Computer vision-based techniques for object detection and tracking involve several key steps. Background subtraction analyzes differences between consecutive frames to separate moving objects (players) from the static background. Popular algorithms include Gaussian Mixture Models (GMMs) and Kernel Density Estimation (KDE) (Stauffer and Grimson, 1999; Zivkovic, 2004). Once identified, moving objects are segmented from the background to obtain clean player silhouettes using thresholding, morphological operations, graph-cut algorithms, etc. Features like color, texture, and shape are extracted from segmented player regions to distinguish individual players and track their motion across frames (Comaniciu et al., 2003). Various tracking algorithms, such as Kalman filters, particle filters, and mean-shift trackers, estimate player positions and trajectories based on the extracted features (Yilmaz et al., 2006).

Marker-based systems, on the other hand, rely on markers designed for easy detection that are worn by the players. These markers provide clear reference points, thus making the methods more accurate. However, they can be intrusive and less practical in professional settings where players may find the markers cumbersome (Rudovic et al., 2018). Despite advancements, these traditional methods face significant challenges, particularly with respect to scalability, computational complexity, and integration with other data sources like GPS and wearable sensors (Bialkowski et al., 2014b).

2.3 Motion Vector Analysis

Recent advances in motion vector extraction have shown significant potential for improving player tracking systems. Motion vectors, a crucial component of video compression technologies (e.g., H.264 and H.265), represent the motion of pixels or pixel blocks between consecutive frames. This enables efficient compression by predicting object movement (ITU-T, 2021; Furht et al., 1997). Motion vector frames add discontinuities or edges around moving objects in video sequences, data that is not necessarily available in traditional RGB frames. This information helps in resolving occlusions if objects do not move at the same speed. As a result, using a combination of motion vectors and RGB can enhance tracking accuracy (Alvar and Bajić, 2018; Kale et al., 2015).

Recent advances also include deep learning techniques for predicting and exploiting motion vectors. Convolutional Neural Networks (CNNs) have been applied to motion estimation tasks, achieving state-of-the-art performance by learning complex motion patterns from large video datasets (Ilg et al., 2017). Optical flow techniques, which estimate the apparent motion of pixels between consecutive frames, have improved with deep learning-based methods, achieving high accuracy and real-time performance (Ranjan and Black, 2017). Techniques for refining motion vectors to improve accuracy and reduce errors often utilize additional information like object segmentation or edge detection (Shah et al., 2021). (Liu et al., 2023; Naik and Hashmi, 2023) presented a deep learning-based real-time soccer player tracking framework that combines motion vectors with visual features, achieving state-of-the-art performance.

2.4 Sports Analytics

Advanced player tracking systems have broad applications in sports analytics, offering detailed insights into player performance, team strategies, and game dynamics. This data is crucial for optimizing training programs, making tactical decisions, and enhancing player performance (Bialkowski et al., 2014b). Player tracking data helps analyze movements, speed, acceleration, and other performance metrics, aiding coaches and analysts in evaluating player strengths and weaknesses and identifying areas for improvement (Kamble et al., 2019). Tracking data is also used to analyze team formations, player positioning, and passing patterns, helping coaches understand team play and develop counter-strategies (Bialkowski et al., 2014a). Moreover, tracking data monitors player workloads and identifies potential injury risks, assisting coaches and medical staff in managing training and preventing injuries (Christopher and Benjamin-Damon, 2021).

Extending beyond professional sports, monitoring is increasingly used in youth and amateur sports to improve performance. Integrating multi-source data, including video, GPS, and wearable sensors, allows for comprehensive analysis of the physical condition of players, contributing to a holistic approach to sports analytics (Murr et al., 2018).

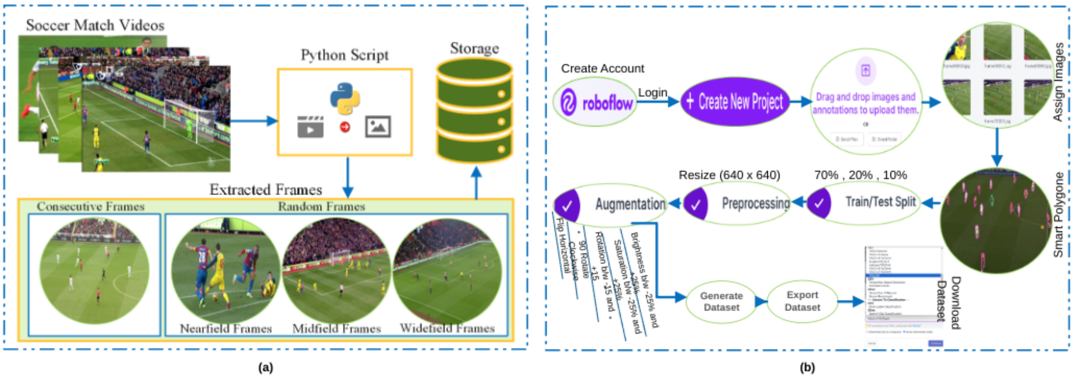

Figure 2: **(a)** Dataset creation pipeline from the given videos. **(b)** Annotating and labelling of the data set, step-by-step.

3 METHODOLOGY

3.1 Dataset Acquisition

For this study, we combine three datasets geared towards the multiple object tracking (MOT) task: the complete DFL-Bundesliga Data Shootout (Deutsche Fussball Liga (DFL), 2022), complete SoccerNet-Tracking (Deliège et al., 2021; Cioppa et al., 2022)), and our own, curated SoccerPro dataset. These datasets provide a diverse and comprehensive collection of soccer game videos, ensuring the robustness and generalizability of our method. The details of these datasets are given below and in Tab. 1.

DFL-Bundesliga Data Shootout. 37.55GB of mp4 videos acquired from the German National Football Association and the Deutsche Fußball Liga (DFL). It also includes csv metadata, totalling 246 files split into clips, test, and training sets (Deutsche Fussball Liga (DFL), 2022).

SoccerNet-Tracking. 187.8GB of videos from major European leagues, featuring 12 full soccer games captured at 1080p/25fps, from which 200 30-second clips were derived, along with tracking data (Deliège et al., 2021; Cioppa et al., 2022)). In total, it consists of 500 videos of the Premier League (England), UEFA Champions League, Ligue-1 (France), Bundesliga (Germany), Serie-A (Italy), and LaLiga (Spain).

Table 1: Benchmark datasets and SoccerPro (ours).

Source	Clips	Duration	Games
DFL	200	30s	9
SoccerNet-Tracking	201	50m	12
SoccerPro	1,459	50m	47
Total	**1,860**	**1,385h**	**68**

SoccerPro. Our dataset is the largest of the three at 4.7TB. It comprises 1,459 videos from various European leagues, captured at 720p to 1280p and 30 or 50fps. It includes 47 full games (13 captured at 720p/50fps, 17 at 1080p/30fps, and 17 at 720p/30fps).

3.2 Annotation and Labeling

We extracted frames from videos using a Python script. We then split frames into two batches, consecutive frames and non-consecutive frames, the latter with a randomized offset. In each batch, we further organize the data in three perspective categories: nearfield, midfield, and widefield (cf. Fig. 2a). Next, we utilized Roboflow's smart polygon (single click) feature to annotate and label our proprietary SoccerPro dataset and the two benchmark datasets. We also annotated a subset of each dataset by manually labeling the frames to create ground truth data for our supervised training pipeline (cf. Fig. 2b). Each player in a frame was annotated by a bounding box and an instance mask, providing the detailed spatial information necessary for instance segmentation. These preprocessing steps ensure that the input data is of high quality and suitable for our instance segmentation and tracking model.

3.3 Architecture

Our architecture consists of three components, dubbed Backbone, Neck, and Head. The Backbone is dedicated to image processing, data augmentation, and extracting feature maps from the input data. The Neck uses a path aggregation network (PAN) to reduce the feature maps at different scales. The Head incorporates the motion vectors and performs instance segmentation and tracking. Finally, our framework also uses both coarse- and fine-grained motion information extracted using a DenseNet operating on

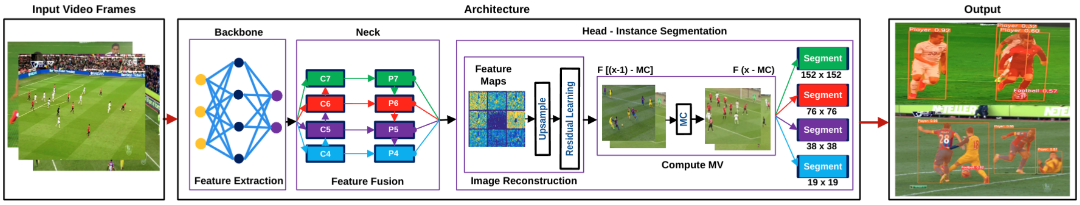

Figure 3: The ReST architecture consists of three components. The **Backbone** extracts features from input frames. The **Neck** uses a Path Aggregation Network (PAN) to reduce the feature maps at different backbone feature maps, labeled $\{C_4, C_5, C_6, C_7\}$ and pyramid feature maps, labeled $\{P_4, P_5, P_6, P_7\}$ with indices denoting the corresponding scale. The **Head** incorporates the motion vectors to perform motion compensation, instance segmentation and tracking. We use four segmentation heads with different dimensions, namely 152×152, 76×76, 38×38, and 19×19.

frame differences and a deep optical flow network, respectively.

3.3.1 Backbone and Preprocessing

Our preprocessing pipeline involves several key steps to prepare the data for training, including the integration of motion vectors for enhanced instance segmentation and tracking. We use the Generalized Efficient Layer Aggregation Network (GELAN), combining the best qualities of CSPDarknet53 and ELAN as a robust backbone for instance segmentation and tracking. We begin by extracting frames and motion vectors from the video data using the mv-extractor tool (Bommes et al., 2020). It decodes H.264 and H.265 video frames to RGB images, frame types, and time stamps. It also extracts the motion vectors. We then clean the extracted data by removing corrupt frames, checking for missing data, and correcting inconsistencies. At the same time, we synchronize the frames with their corresponding motion vectors. This is done by aligning the timestamps of the motion vectors to ensure that they correctly map to their respective objects. As we demonstrate in our results, the integration of motion vectors significantly enhances our model's ability to distinguish between players and track their movements accurately, even in challenging scenarios with partial occlusion.

Afterwards, we resize the processed images to 640×640 pixels with three color channels and store them on disk. Likewise, motion vectors are stored on disk but are reserved for the later head component. At this stage, we also compute normalization coefficients in order for the images and motion vectors to be on a consistent scale across different datasets.

3.3.2 Neck—Feature Pyramid Network

The extracted features are then forwarded to the Neck stage for feature fusion, specifically to the Path Aggregation Network (PANet). We use Feature Pyramid Networks (FPN) (Lin et al., 2017) to construct a multi-level feature pyramid that enables the accurate representation of objects at varying scales. For tasks where recognizing objects at different scales is crucial, the hierarchical nature of FPNs is particularly useful. In our scenario, this corresponds to players at various distances from the camera. By combining low-resolution, semantically strong features with high-resolution, semantically weak features, FPNs support the detection of small objects with (in our case) four different backbone feature maps, labeled $\{C_4, C_5, C_6, C_7\}$ and pyramid feature maps, labeled $\{P_4, P_5, P_6, P_7\}$ in Fig. 3, while maintaining accurate localization of larger objects. FPNs are typically built on top of a standard CNN, such as a ResNet, and consist of two main pathways, bottom-up and top-down.

Bottom-Up Pathway. The forward pass of the backbone network (e.g., ResNet) generates feature maps at multiple scales by applying a series of convolutional and downsampling layers. As we move deeper into the network, the spatial resolution of the feature maps decreases while the semantic richness increases.

Top-Down Pathway. Starting from the highest-level feature map (lowest resolution), this pathway progressively upsamples. At each level, the upsampled feature map is combined with the corresponding feature map from the bottom-up pathway using lateral connections, which are represented by 1×1 convolutions. The process to obtain the feature map P_l at level l of the pyramid can be formalized as follows.

$$P_l = \text{Upsample}(P_{l+1}) + \text{Conv}_{1\times1}(C_l). \qquad (1)$$

Here, the feature map at the higher pyramid level $l+1$ is upsampled and then combined with the corresponding feature map from the bottom-up pathway.

This fusion of high-level semantics with fine spatial details creates a set of feature maps with improved resolution and semantic content at each scale. The merged bottom-up and top-down feature maps are further processed by 3×3 convolutions to generate the final output feature maps. In our case, we repeat this process for four levels, resulting in feature maps P_4, P_5, P_6, P_7, with indices indicating scale.

Since these maps allow us to detect objects across a range of sizes in a single forward pass, they make our network more efficient and accurate.

3.3.3 Head—Instance Segmentation

The instance segmentation head is a crucial component designed to generate pixel-wise masks for each detected object in an image. Pixel-wise masks are particularly important in overlap scenarios which are not well described by simple bounding boxes. The architecture of the head can be broken into the following sequence.

Input Features. The instance segmentation head receives feature maps from the Feature Pyramid Network (FPN) that contain rich multi-scale information about objects of various sizes and appearances.

Mask Prediction. The core of the segmentation head consists of several convolutional layers that are applied to the feature maps. These layers progressively refine the spatial resolution while retaining the semantic information necessary for accurate mask prediction. Typically, a 4×4 deconvolution layer is employed to upsample the lower-resolution features, followed by a series of 3×3 convolutional layers. The output is a set of binary masks, where each mask corresponds to a single detected object.

Objectness and Class-Specific Masks. The segmentation head is designed to predict both the objectness score and the class-specific binary masks. During the training phase, the model learns to generate masks that are associated with specific object classes. This approach ensures that the masks not only delineate the object's boundaries but also correspond to the correct class label, enhancing the accuracy of instance segmentation.

RoIAlign. To maintain spatial alignment between the predicted masks and the input image, we utilize the Region of Interest Align (RoIAlign) operation. RoIAlign extracts fixed-size feature maps corresponding to each proposed region of interest (RoI) from the feature pyramid. This operation ensures that the mask predictions are accurately aligned with the original image, avoiding the misalignment issues that arise from quantization in RoIPooling.

The RoIalign in our framework leverages the rich multi-scale features provided by the FPN. This enables our framework to make precise mark predictions through convolutional layers and class-specific segmentations. It also ensures spatial accuracy. The instance segmentation head, therefore, provides a solid foundation for accurate object discrimination. However, since the hierarchy levels in the FPN are of different resolution, the following high-quality image reconstruction step is required to make the best use of the information stored in the FPN.

Image Reconstruction. The downsampled feature maps in the FPN require adequate upsampling in order to match the resolution of other levels in the same pyramid. For this task, we use deconvolution layers and residual learning to recover and enhance details lost during downsampling. First, the feature maps $\{\mathrm{FM}_l\}_{l=1}^{L}$ are upsampled through deconvolution layers, which restore spatial resolution by progressively enlarging the feature maps. Using weights W_l and biases b_l, this can be formalized as:

$$\widehat{\mathrm{FM}}_l = \mathrm{FM}_l \star^T W_l + b_l, \tag{2}$$

where $\star^T$ denotes the transposed convolution operation, acting as a pseudo-inverse of the convolution $\star$. This process increases spatial dimensions and enhances feature detail, contributing to finer reconstruction.

Residual learning is then applied to each level l to enhance fine details by adding prior feature information. The reconstruction R_l at level l is computed as:

$$\mathrm{R}_l = \mathrm{FM}_l + \mathcal{F}\left(\mathrm{FM}_{l-1}, W_r\right), \tag{3}$$

where $\mathcal{F}(\cdot)$ represents the residual function (e.g., a convolution operation) with weights W_r, and R_l is the output feature map with added detail from the previous layer FM_{l-1}. This approach allows for efficient gradient flow and enhances reconstruction by mitigating information loss through downsampling. This combination of upsampling and residual addition enables the model to accurately reconstruct images with preserved structural details and minimise information loss.

Frame Differencing. To capture motion across video frames and effectively isolate moving objects from static backgrounds, we employ a combination of frame differencing and optical flow techniques. This framework is applied across various field perspectives—nearfield, midfield, and widefield (cf. Fig. 2a)—to handle different spatial and temporal resolutions. We compute frame differences between consecutive and non-consecutive frames with different temporal offsets to detect motion over short and long time scales. The difference between frames highlights areas of movement, which typically correspond to players or other dynamic elements on the field. For consecutive frames, the frame difference $\Delta_{t,t-1}$ between frames I_t and I_{t-1} is calculated as the absolute difference in pixel intensities,

$$\Delta_{t,t-1} = |I_t - I_{t-1}|. \tag{4}$$

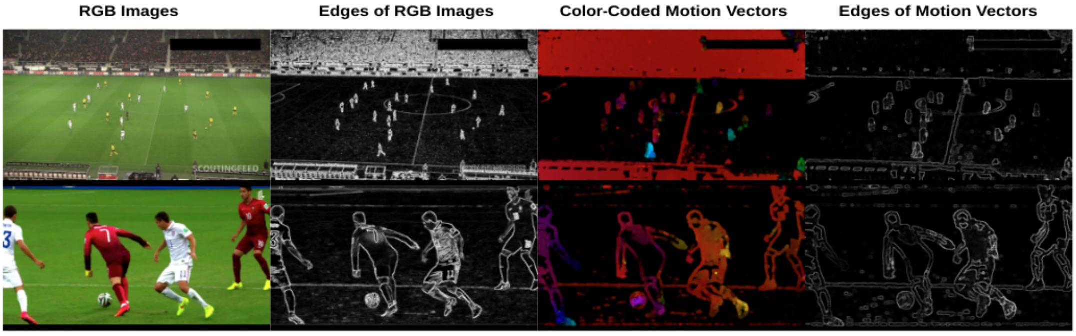

Figure 4: Motion vectors provide additional edge cues. Top row: relatively static scenarios from the SoccerNet-Tracking repository dataset shot with a professional PTZ camera at 1080p (frame ID 5dc4fe12_00083). Bottom row: dynamic camera from our SoccerPro dataset, 720p and likely to be recorded on a mobile phone. In each row, left to right: RGB image, Scharr edges of the RGB image, color-coded motion vectors, and edges of the motion vectors.

$\Delta_{t,t-1}$ highlights motion areas, allowing the identification of moving players.

To capture movement over longer intervals or various perspectives, we extend this method to compute differences between non-consecutive frames I_t and I_{t-k}, where $k \in \mathbb{N}$ is a variable temporal offset adjusted to the target perspective (e.g., nearfield for close action, widefield for more global play).

$$\Delta_{t,t-k} = |I_t - I_{t-k}|. \tag{5}$$

```
Procedure Draw_MV(F, MV, $M_{\text{inst}}$, $O_{\text{track}}$)
if $|MV| > 0$ then
    $N_{\text{mv}} \leftarrow \text{shape}(\text{MV})[0]$
    for mv in split(MV, $N_{\text{mv}}$) do
        start $\leftarrow (\text{mv}[0,3], \text{mv}[0,4])$
        end $\leftarrow (\text{mv}[0,5], \text{mv}[0,6])$
        cv2.arrowedLine(F, start, end, \
            (0,0,255), 1, cv2.LINE_AA, 0, 0.1)
    end for
end if
for each instance segment M $\in M_{\text{inst}}$ do
    Overlay M on F with a unique color
end for
for each object O $\in O_{\text{track}}$ do
    Track O across frames; update position
    Draw bounding box or contour \
        around O on F
end for
return $F$
```

Algorithm 1: Our approach to draw overlay motion vectors using OpenCV2 to analyze instance segmentation and tracking results. F: current frame, MV: motion vectors, M_{inst}: object instances, O_{track}: tracked objects.

This multi-scale differencing framework provides comprehensive motion information across different spatial and temporal resolutions, enhancing object tracking and distinguishing dynamic elements from static backgrounds.

Fine-grained Motion Estimation. To further enhance the accuracy of motion estimation, we incorporate a state-of-the-art optical flow algorithm based on deep learning, called RAFT (Recurrent All-Pairs Field Transforms) (Teed and Deng, 2020). Optical flow provides dense, pixel-wise motion estimation by analyzing frame-to-frame displacements, thereby offering precise motion tracking beyond the limitations of frame differencing. For each pair of frames I_t and I_{t-k}, the optical flow field $o_{t,t-k}$ is computed as

$$O_{t,t-k} = \text{OpticalFlow}(I_t, I_{t-k}), \tag{6}$$

where the optical flow algorithm estimates the displacement vector (direction and magnitude) for every pixel between the two frames. This provides a fine-grained understanding of player movements and ball trajectories. By combining frame differencing and optical flow, our method provides a robust framework for detecting and tracking motion across different temporal and spatial scales. This hybrid approach enables accurate motion estimation for both short-term dynamic actions and long-term strategic movements on the soccer field.

Motion Vectors. Building upon the motion detection established through frame differencing and optical flow, we employ Bommes' motion vector extraction technique (Bommes et al., 2020). This technique determines the motion vectors between consecutive and non-consecutive (nearfield, midfield, widefield) frames, providing essential data for subsequent analysis. We then decode the frames into RGB images, incorporating motion vectors, frame types, and timestamps using H.264 and H.265 codecs, known for their high compression rates, superior image quality, and broad compatibility. By leveraging object motion captured in these vectors, we enhance the accuracy of both predicted bounding boxes and instance seg-

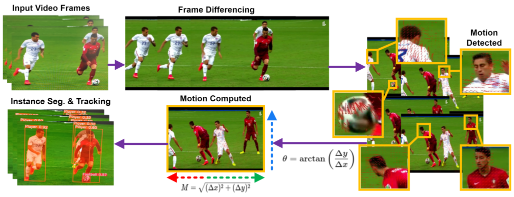

Figure 5: **Input** Frame shows the initial frame; **Frame Differencing** highlights motion areas; **Motion Detected** marks active regions with bounding boxes; **Motion Computed** illustrates direction and magnitude with vectors; **Tracking with Instance Segmentation** differentiates moving objects.

ments, particularly in scenarios involving partial occlusion among players.

To analyze the extracted motion vectors, we plot individual frames with object instance segmentation and motion vectors overlayed. Our approach to do so is outlined in Algorithm 1. To this end, we use OpenCV2 functionality and use either bounding boxes or contours to highlight the individual objects (players, ball, referees, etc.).

Rationale for using Motion Vectors. In our pipeline, we utilize the Scharr filter (Schar, 2000), a variant of the Sobel operator optimized for rotational invariance, to enhance the delineation of soccer players by detecting motion vector discontinuities under the assumption that player movement produces prominent motion edges against a static background. In particular, as can be seen in Fig. 4, motion edges do not necessarily coincide with RGB edges, providing additional information–that is not readily available from RGB-only–to help with object segmentation. We believe these additional shape cues to be essential for precise instance segmentation of players. Particularly, for a widefield, fixed-camera scenario, static regions yield negligible motion vector magnitudes, enabling clear foreground-background separation.

Tracking. Our tracking pipeline leverages a modified version of the BoT-SORT tracker (Aharon et al., 2022) to enhance motion tracking performance, embedding capabilities, and Intersection over Union (IoU) metrics. To process frame differences into motion estimates, we use a DenseNet with motion vectors estimated at object centroids for motion compensation during tracking, as shown in Fig. 5. Furthermore, per-pixel motion vectors based on an optical flow network (Teed and Deng, 2020) are combined with RGB images to facilitate our simultaneous detection, segmentation, and tracking processes.

4 EXPERIMENTS

We evaluated our proposed methodology using two benchmark datasets and a curated, proprietary dataset, SoccerPro. We use a model provided by Torchvision that was specifically pre-trained on the COCO dataset. We retrained the model concurrently on all three datasets, keeping hyperparameters consistent throughout the training process.

4.1 Training

For training, we divide the data into three parts: 70% for training, 20% for validation, and 10% for testing. We perform randomized image augmentation, including brightness and saturation adjustments within a ±25% range, rotation within ±15%, combined with the choice to perform 90° clockwise rotation and horizontal flipping (cf. Fig. 2b).

We conducted all experiments on an Ubuntu 24.04 LTS machine equipped with 512GB of RAM, an Xeon(R) Gold 6226R CPU, and an NVIDIA RTX 3090 GPU with 24GB of RAM. The entire model is implemented in Python using the PyTorch framework. We used the AdamW optimizer with a learning rate scheduler (10^{-5} to 0.01), momentum of 0.6, and a batch size of 8, and trained over 300 epochs.

4.2 Evaluation Metrics

We employ four primary metrics to evaluate the performance of our instance segmentation model on both benchmark and SoccerPro datasets: Accuracy, Precision, Recall, and Mean Average Precision (mAP). The precision-recall curve (Fig. 6) is used to comprehensively assess our model's performance. To thoroughly analyze our method, we computed the class-based scores for each class, evaluating how effectively the model detects and segments each class separately (see Tab. 2). Additionally, we assessed the combined output for all classes, determining the model's best performance across various classes to provide an overall evaluation of detection and instance segmentation capabilities.

4.3 Results

Our ReST framework significantly outperforms existing state-of-the-art methods in instance segmentation and tracking of soccer players. By leveraging Bommes' MV-Extractor technique, ReST enhances precision and recall across various classes, particularly the player class. This improvement is evidenced by the substantially higher class-based scores attained

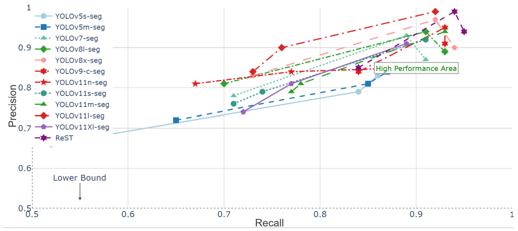

Figure 6: Precision-Recall curve for all (12) Instance Segmentation models and their comparison with ReST (Ours) on all three datasets.

by ReST, as demonstrated in Tab. 2. We also provide qualitative results in Fig. 7, showing segmentation and tracking results obtained on DFL-Bundesliga Data Shootout and the SoccerNet-Tracking and SoccerPro dataset using OC-SORT, ByteTrack and ReST (ours). The results clearly demonstrate that ReST (ours) performs better than the prior methods regarding instance segmentation and tracking.

Our framework exhibits robust and accurate segmentation of soccer players, excelling in challenging scenarios such as occlusion and diverse player poses. ReST's generalization capabilities are remarkable, as evidenced by its meticulous evaluation of benchmark datasets (MOT17 and MOT20) and the SoccerPro dataset. This ensures consistent and reliable performance across diverse soccer tracking scenarios. In terms of computational efficiency, ReST maintains competitive real-time performance while achieving superior accuracy. The inference speed of ReST is commendable, especially considering its enhanced tracking accuracy compared to recent SOTA trackers, as shown in (Tab. 3). Furthermore, ReST demonstrates exceptional tracking accuracy, outperforming contemporary trackers in the following key metrics.

IDF$_1$ (Identification F$_1$) is the regular F$_1$ metric applied to identity accuracy.

MOTA (Multiple Object Tracking Accuracy (MOTA). MOTA evaluates overall tracking performance by considering false positives (FP), false negatives (FN), and identity switches (IDSW). MOTA is defined as $1 - \frac{\text{FN+FP+IDSW}}{\text{GT}}$, where GT is the total number of ground truth objects.

HOTA (Higher Order Tracking Accuracy). This metric integrates various tracking facets into a unified metric, including localization and identity accuracy. It comprehensively evaluates tracker performance beyond traditional metrics like MOTA or IDF$_1$. HOTA is defined as $\text{A}_{\text{ass}} - \text{A}_{\text{loc}} - \text{FP} - \text{IDSW}$. In this context, A_{ass} denotes Assignment Accuracy, measuring the precision of bounding box assignments across different intersection over union (IoU) thresholds. A_{loc} represents localization accuracy, quantifying the average distance between predicted and ground truth bounding box centres. Finally, FP measures false positives, and IDSW is a measure of ID switches.

Table 2: Analysis of class-based scores on YOLO (v5, v7, v8, v9, and v11) and ReST (ours) on the combined datasets, for instance segmentation and tracking. All models in this table are variations of YOLO, except for ours. We report mAP both measured on box$_{50-95}$ and mask$_{50-95}$.

		Training				**Validation**			
				mAP				**mAP**	
Model	**Class**	**P**	**R**	**box**	**mask**	**P**	**R**	**box**	**mask**
v5s-seg	Player	0.61	0.84	0.75	0.33	0.57	0.81	0.69	0.30
v5m-seg		0.66	0.77	0.70	0.33	0.56	0.74	0.62	0.25
v7-seg		0.83	0.84	0.83	0.47	0.83	0.83	0.82	**0.41**
v8l-seg		0.77	0.82	0.74	0.38	0.76	0.80	0.74	0.29
v8x-seg		0.62	0.81	0.72	0.38	0.61	0.79	0.71	0.31
v9-c-seg		0.88	0.84	0.87	0.51	0.85	0.82	0.81	0.40
v11n-seg		0.70	0.80	0.72	0.37	0.77	0.77	0.76	0.32
v11s-seg		0.64	0.74	0.64	0.33	0.61	0.71	0.60	0.24
v11m-seg		0.68	0.80	0.70	0.35	0.67	0.77	0.69	0.29
v11l-seg		0.75	0.76	0.73	0.37	0.75	0.73	0.70	0.28
v11Xl-seg		0.69	0.69	0.67	0.32	0.65	0.66	0.65	0.25
ReST		**0.97**	**0.87**	**0.89**	**0.54**	**0.93**	**0.85**	**0.83**	**0.41**
v5s-seg	Goalkeeper	0.78	0.43	0.53	0.21	0.89	0.47	0.63	0.28
v5m-seg		**0.99**	0.44	0.75	0.32	**0.99**	0.50	0.74	0.37
v7-seg		**0.99**	0.79	0.88	0.40	**0.99**	0.79	0.88	0.39
v8l-seg		0.93	0.81	**0.96**	0.46	0.87	0.74	0.88	0.38
v8x-seg		0.93	0.77	0.87	0.46	0.93	0.77	0.83	0.42
v9-c-seg		0.93	0.79	0.84	0.47	0.92	0.78	0.90	0.44
v11n-seg		0.75	0.67	0.77	0.38	0.85	0.67	0.77	0.33
v11s-seg		0.72	0.72	0.87	0.42	0.79	0.81	0.91	0.41
v11m-seg		0.79	**0.83**	0.89	0.43	0.45	0.78	0.81	0.40
v11l-seg		**0.99**	0.82	0.86	0.44	0.90	0.72	0.80	0.41
v11Xl-seg		0.81	0.68	0.82	0.39	0.81	0.69	0.82	0.37
ReST		**0.99**	**0.83**	0.87	**0.49**	**0.99**	**0.81**	**0.92**	**0.45**
v5s-seg	Referee	0.85	0.46	0.56	0.31	0.76	0.42	0.49	0.22
v5m-seg		0.90	0.58	0.67	0.36	0.85	0.58	0.67	0.30
v7-seg		0.86	0.69	0.77	0.49	0.82	0.69	0.77	0.43
v8l-seg		0.69	0.69	0.72	0.34	0.61	0.58	0.64	0.28
v8x-seg		0.67	0.70	0.70	0.32	0.67	0.70	0.71	0.35
v9-c-seg		0.91	0.70	0.83	0.44	0.87	0.72	0.77	0.41
v11n-seg		0.81	0.66	0.77	0.42	0.84	0.65	0.73	0.36
v11s-seg		0.82	0.62	0.70	0.36	0.76	0.58	0.67	0.29
v11m-seg		0.84	**0.73**	0.72	0.41	0.81	0.69	0.66	0.34
v11l-seg		0.84	0.58	0.67	0.39	0.80	0.54	0.61	0.29
v11Xl-seg		0.74	0.65	0.70	0.43	0.74	0.66	0.73	0.35
ReST		**0.92**	0.72	**0.88**	**0.51**	**0.90**	**0.74**	**0.81**	**0.47**
v5s-seg	Football	0.36	0.35	0.30	0.10	0.31	0.30	0.24	0.12
v5m-seg		0.54	0.39	0.32	0.14	0.56	0.52	**0.46**	0.14
v7-seg		0.46	0.35	0.31	0.16	0.46	0.35	0.36	0.11
v8l-seg		0.47	0.35	0.30	0.10	0.36	0.26	0.22	0.09
v8x-seg		0.21	0.13	0.18	0.08	0.28	0.17	0.23	0.08
v9-c-seg		**0.63**	0.49	0.44	0.40	0.59	**0.53**	0.44	0.37
v11n-seg		0.53	0.26	0.30	0.13	0.73	0.26	0.34	0.14
v11s-seg		0.41	0.26	0.28	0.14	0.54	0.35	0.33	0.12
v11m-seg		0.36	0.22	0.26	0.11	0.44	0.26	0.30	0.11
v11l-seg		0.43	0.35	0.26	0.09	0.50	0.35	0.26	0.09
v11Xl-seg		**0.63**	0.18	0.25	0.13	**0.62**	0.18	0.28	0.13
ReST		**0.63**	**0.51**	**0.47**	**0.41**	0.61	0.48	0.44	**0.39**

Our results highlight the efficacy of ReST in accurately tracking soccer players' movements, which is crucial for applications in sports analytics and player performance analysis. The framework's ability to handle complex motion dynamics and partial occlusions and its real-time processing capabilities make it a valuable tool for enhancing the analysis and understanding of soccer games. Figure 8 shows the overall comparison of instance segmentation and tracking with respect to their accuracies on previous versions of You Only Look Once (YOLO) and ReST (ours).

Table 3: Tracking Results on MOT17 Validation and MOT20 Training Datasets. Bold: best performance per column.

	with Motion Vectors								without Motion Vectors							
	MOT17 Validation				MOT20 Training				MOT17 Validation				MOT20 Training			
Tracker	**HOTA**	**MOTA**	**IDF_1**	**fps**	**HOTA**	**MOTA**	**IDF1**	**fps**	**HOTA**	**MOTA**	**IDF_1**	**fps**	**HOTA**	**MOTA**	**IDF1**	**fps**
Enhanced Motion:																
OC-SORT[1]	61.3	76.2	75.1	23.4	60.2	74.5	76.3	19.7	54.7	74.6	69.7	19.3	52.4	73.1	69.3	17.6
MotionTrack[2]	64.7	79.5	78.7	13.2	63.4	77.4	78.2	9.7	58.2	72.9	68.6	8.4	57.4	72.2	67.8	8.2
Embedding:																
StrongSORT[3]	–	–	–	–	–	–	–	–	56.3	71.5	70.2	6.7	54.9	70.6	68.4	6.1
IoU only:																
ByteTrack[4]	–	–	–	–	–	–	–	–	57.7	75.6	69.3	14.4	57.3	74.5	68.7	12.7
BoT-SORT[5]	–	–	–	–	–	–	–	–	61.6	76.2	74.7	7.6	61.3	75.4	74.3	5.3
ReST (Ours)	**64.9**	**79.8**	**79.4**	**27.2**	**63.6**	**78.4**	**78.7**	**23.5**	**63.4**	**77.6**	**75.2**	**23.4**	**63.2**	**77.5**	**75.2**	**21.4**

[1] (Cao et al., 2023) [2] (Qin et al., 2023) [3] (Du et al., 2023) [4] (Zhang et al., 2022) [5] (Aharon et al., 2022)

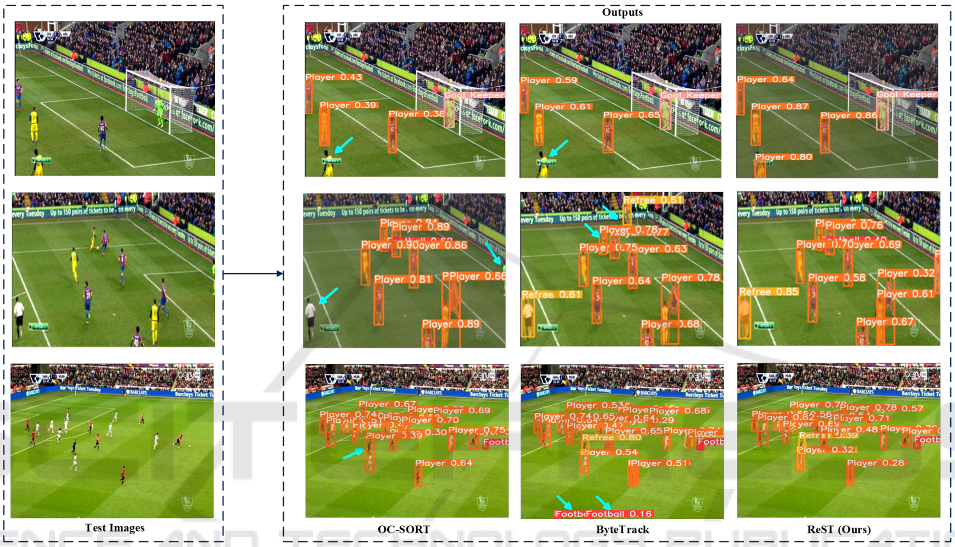

Figure 7: **Qualitative results:** Obtained, for instance segmentation and tracking outputs generated by ReST (Ours) on DFL-Bundesliga Data Shootout and the SoccerNet dataset using OC-SORT, ByteTrack and ReST (Ours). Up to down row: three different camera scenarios. (1) nearfield, (2) midfield, (3) widefield. Cyan arrows indicate the localization, segmentation and tracking errors. Our approach (last column) consistently provides better results in all three perspectives.

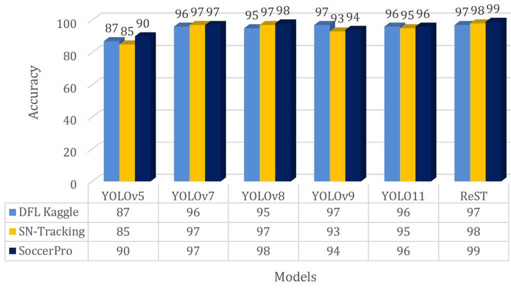

Figure 8: Comparison of instance segmentation and tracking accuracies between YOLO (v5, v7, v8, v9, and v11) and ReST.

5 CONCLUSION

This paper presents ReST, a novel real-time framework for detection, instance segmentation, and tracking using motion vectors. The proposed system leverages the Generalized Efficient Layer Aggregation Network (GELAN), combined with the strengths of the CSPDarknet53 architecture as the backbone for instance segmentation. Our framework is further enhanced by motion vectors obtained using a DenseNet motion estimator on absolute frame differences and fine-grained motion vectors based on a deep optical flow network. We demonstrated that integrating motion vectors provides ReST with additional shape cues that significantly improve the separation of foreground and background, particularly in scenarios involving partial occlusion of players. To rigorously evaluate our model's performance, we conducted extensive experiments comparing current and previous versions of YOLO (v5, v7, v8, v9, and v11) with our ReST model. Additionally, we assessed its performance on both the validation set of MOT17 and the training set of the MOT20 dataset. The results demonstrated remarkable accuracy, with ReST achieving

97% on the DFL-Bundesliga Data Shootout, 98% on the SoccerNet-Tracking dataset, and an impressive 99% on our custom SoccerPro dataset. Furthermore, the model operates in real-time, achieving a tracking rate of 50fps on an NVIDIA RTX 3090 GPU.

In the future, we will explore ReST's capabilities for additional applications, such as estimating players' 3D positions from the predicted motion vectors as well as estimates of speed and jump heights. We also see value in exploring the usefulness of our framework for other sports or even estimating events within the crowd of spectators.

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

Aharon, N., Orfaig, R., and Bobrovsky, B.-Z. (2022). Botsort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651.

Alvar, S. R. and Bajić, I. V. (2018). MV-YOLO: Motion vector-aided tracking by semantic object detection. In *IEEE 20th International Workshop on Multimedia Signal Processing*, pages 1–5.

Athar, A., Luiten, J., Voigtländer, Khurana, T., Dave, A., Leibe, B., and Ramanan, D. (2023). BURST: A Benchmark for unifying object recognition, segmentation and tracking in video. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1674–1683.

Baysal, S. and Duygulu, P. (2016). Sentioscope: A soccer player tracking system using model field particles. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1350–1362.

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. (2014a). Identifying team style in soccer using formations learned from spatiotemporal tracking data. In *IEEE International Conference on Data Mining Workshop*, pages 9–14.

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. (2014b). Large-scale analysis of soccer matches using spatiotemporal tracking data. In *IEEE International Conference on Data Mining*, pages 7255–730.

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

Cao, J., Pang, J., Weng, X., Khirodkar, R., and Kitani, K. (2023). Observation-centric sort: Rethinking sort for robust multi-object tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9686–9696.

Christopher, Raphael abd Brandt, C. and Benjamin-Damon, N. (2021). Systematic review of screening tools for common soccer injuries and their risk factors. *South African Journal of Physiotherapy*, 77(1):#a1496.

Cintia, P., Rinzivillo, S., Pappalardo, L., Pedreschi, D., and Giannotti, F. (2015). A network-based approach to evaluate the performance of football teams. In *Workshop on Machine Learning and Data Mining for Sports Analytics*, pages 46–54.

Cioppa, A., Giancola, S., Deliège, A., Kang, L., Zhou, X., Cheng, Z., Ghanem, B., and Van Droogenbroeck, M. (2022). Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3491–3502.

Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577.

Csanalosi, G., Dobreff, G., Pasic, A., Molnar, M., and Toka, L. (2020). Low-cost optical tracking of soccer players. In *Machine Learning and Data Mining for Sports Analytics: 7th International Workshop*, pages 28–39.

Deliège, A., Cioppa, A., Giancola, S., Seikavandi, M. J., Dueholm, J. V., Nasrollahi, K., Ghanem, B., Moeslund, T. B., and Van Droogenbroeck, M. (2021). Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 4508–4519.

Deutsche Fussball Liga (DFL) (2022). DFL-Bundesliga data shootout dataset. accessed Jul 2023.

Diop, C.-A., Pelloux, B., Yu, X., Yi, W.-J., and Saniie, J. (2022). Soccer player recognition using artificial intelligence and computer vision. In *IEEE International Conference on Electro Information Technology*, pages 477–481.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., and Meng, H. (2023). Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 25:8725–8737.

Ehrmann, F. E., Duncan, C. S., Sindhusake, D., Franzsen, W. N., and Greene, D. A. (2016). GPS and injury prevention in professional soccer. *The Journal of Strength & Conditioning Research*, 30(2):360–367.

Furht, B., Greenberg, J., and Westwater, R. (1997). *Motion Estimation Algorithms for Video Compression*. Springer.

Ghasemzadeh, S. A., Van Zandycke, G., Istasse, M., Syez, N., Moshtaghpour, A., and De Vleeschouwer, C. (2021). DeepSportLab: A framework for automated sports analytics, production, and broadcast. In *British Machine Vision Conference*, pages 1–14.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision*, pages 2980–2988.

Heo, M., Hwang, S., Hyun, J., Kim, H., Oh, S. W., Lee, J.-Y., and Kim, S. J. (2023). A generalized framework for video instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14623–14632.

Heo, M., Hwang, S., Oh, S. W., Lee, J.-Y., and Kim, S. J. (2022). VITA: Video instance segmentation with temporal attention. In *Advances in Neural Information Processing Systems*, volume 35, pages 23109–23120.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470.

ITU-T (2021). Recommendation h.264 (08/21). accessed Jul 2024.

Kale, K., Pawar, S., and Dhulekar, P. (2015). Moving object tracking using optical flow and motion vector estimation. In *4th International Conference on Reliability, Infocom Technologies and Optimization*, page #108.

Kamble, P. R., Keskar, A. G., and Bhurchandi, K. M. (2019). Ball tracking in sports: a survey. *Artificial Intelligence Review*, 52:1655–1705.

Ke, L., Danelljan, M., Ding, H., Tai, Y.-W., Tang, C.-K., and Yu, F. (2023). Mask-free video instance segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22857–22866.

Khaustov, V. and Mozgovoy, M. (2020). Recognizing events in spatiotemporal soccer data. *Applied Sciences*, 10(22):8046.

Kirillov, A., He, K., Girshick, R., and Dollár, P. (2019). Panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Linke, D., Link, D., and Lames, M. (2020). Football-specific validity of tracab's optical video tracking systems. *PloS one*, 15(3):#e0230179.

Liu, J., Huang, G., Hyyppä, J., Li, J., Gong, X., and Jiang, X. (2023). A survey on location and motion tracking technologies, methodologies and applications in precision sports. *Expert Systems with Applications*, 229:120492.

Majeed, F., Gilal, N. U., Al-Thelaya, K., Yang, Y., Agus, M., and Schneider, J. (2024). MV-Soccer: Motion-vector augmented instance segmentation for soccer player tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3245–3255.

Manafifard, M., Ebadi, H., and Moghaddam, H. A. (2017). A survey on player tracking in soccer videos. *Computer Vision and Image Understanding*, 159:19–46.

Mazzeo, P. L., Spagnolo, P., Leo, M., and D'Orazio, T. (2008). Visual players detection and tracking in soccer matches. In *IEEE 5th International Conference on Advanced Video and Signal Based Surveillance*, pages 326–333.

Murr, D., Raabe, J., and Höner, O. (2018). The prognostic value of physiological and physical characteristics in youth soccer: A systematic review. *European journal of sport science*, 18(1):62–74.

Naik, B. T. and Hashmi, M. F. (2023). Yolov3-sort: detection and tracking player/ball in soccer sport. *Journal of Electronic Imaging*, 32(1):011003–011003.

Naik, B. T., Hashmi, M. F., Geem, Z. W., and Bodke, N. D. (2022). DeepPlayer-Track: Player and referee tracking with jersey color recognition in soccer. *IEEE Access*, 10:32494–32509.

Qin, Z., Zhou, S., Wang, L., Duan, J., Hua, G., and Tang, W. (2023). MotionTrack: Learning robust short-term and long-term motions for multi-object tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17939–17948.

Ranjan, A. and Black, M. J. (2017). Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170.

Rudovic, O., Lee, J., Dai, M., Schuller, B., and Picard, R. W. (2018). Personalized machine learning for robot perception of affect and engagement in autism therapy. *Science Robotics*, 3(19):#eaao6760.

Schar, H. (2000). *Optimale Operatoren in der Digitalen Bildverarbeitung*. PhD thesis, University of Heidelberg. (in German).

Shah, S. T. H., Xuezhi, X., and Ahmed, W. (2021). Optical flow estimation with convolutional neural nets. *Pattern Recognition and Image Analysis*, 31:656–670.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252.

Teed, Z. and Deng, J. (2020). Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer.

Wehbe, G. M., Hartwig, T. B., and Duncan, C. S. (2014). Movement analysis of australian national league soccer players using global positioning system technology. *The Journal of Strength & Conditioning Research*, 28(3):834–842.

Xu, R., Tabman, D., and Naman, A. T. (2016). Motion estimation based on mutual information and adaptive multi-scale thresholding. *IEEE Transactions on Image Processing*, 25(33):1095–1108.

Yang, L., Fan, Y., and Xu, N. (2019). Video instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5187–5196.

Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys*, 38(4):#13–es.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2022). ByteTrack: Multi-object tracking by associating every detection box. In *European Conference on Computer*, pages 1–21.

Zhang, Z. (2012). Microsoft Kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition*, volume 4, pages 28–31.