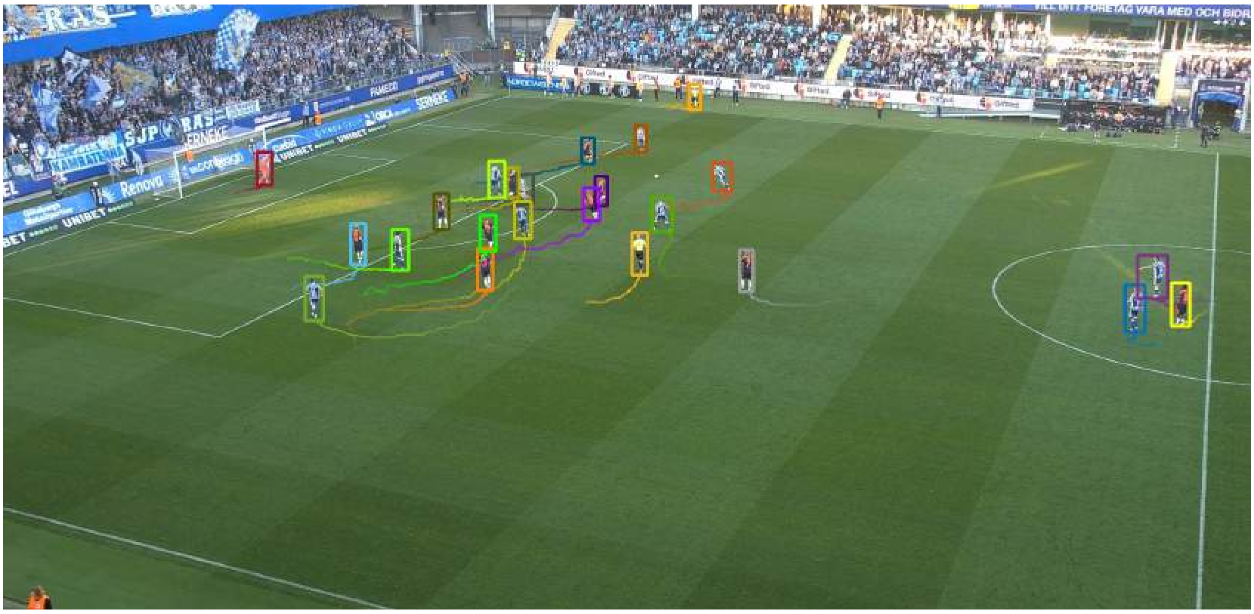
CHALMERS
UNIVERSITY OF TECHNOLOGY



Transformer-Based Multi-Object Tracking of Football Players Using Pseudo-Labeling

Adapting MOTRv2 with Pseudo-Labeling and Re-Identification for Football Tracking

Master's thesis in Complex Adaptive Sytems

Anton Hedén, Anthon Odengard

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Transformer-Based Multi-Object Tracking of Football Players Using Pseudo-Labeling

Adapting MOTRv2 with Pseudo-Labeling and Re-Identification for Football Tracking

Anton Hedén, Anthon Odengard

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Transformer-Based Multi-Object Tracking of Football Players Using Pseudo-Labeling
Adapting MOTRv2 with Pseudo-Labeling and Re-Identification for Football Tracking
Anton Hedén, Anthon Odengard

Supervisors: Anders Sjöberg, Fraunhofer Chalmers Center and
Prof. Lennart Svensson, Chalmers Department of Electrical Engineering
Examiner: Prof. Tomas McKelvey, Chalmers Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover image: An example tracking sequence demonstrating the performance of the MOTRv2 model with Re-ID, trained on the Mix dataset.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Transformer-Based Multi-Object Tracking of Football Players Using Pseudo-Labeling
Adapting MOTRv2 with Pseudo-Labeling and Re-Identification for Football Tracking

Anton Hedén, Anthon Odengard
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis investigates the problem of tracking football players in video sequences, with a focus on adapting modern multi-object tracking (MOT) methods to the specific challenges of football environments. The work is part of a broader effort to develop tools for analyzing football games using automated visual data. In this context, we utilize MOTRv2, a transformer-based tracking model originally designed for general-purpose MOT tasks, and apply it to the football domain, where challenges such as frequent occlusions, tight formations, and rapid movement are prevalent.

To address the lack of annotated football-specific tracking data, we implement a pseudo-labeling framework that allows the model to be trained on unlabelled domain data in a semi-supervised fashion. This approach enables progressive refinement of the model through multiple training cycles on domain-specific content. Our results show that MOTRv2 can be adapted to the football setting and performs well in many scenarios, particularly in open-play segments with clear player separation. However, limitations remain, including decreased tracking stability in crowded scenes and occasional ID-switches due to overlapping motion patterns.

Overall, this work demonstrates the potential of transformer-based trackers in sports applications and highlights the benefits of self-supervised training when domain-specific data is scarce. The findings offer insights for future improvements in automated sports tracking systems.

Keywords: Computer Vision, Multiple Object Tracking, Transformer, Football, Pseudo-Labeling, Re-ID

Acknowledgements

This project originated from an idea proposed by Professor Lennart Svensson, who also served as our academic supervisor. We are deeply grateful for his continuous support and engagement throughout the process. His guidance and early involvement were instrumental in shaping the direction of the work from start to finish.

We would also like to thank the Fraunhofer Chalmers Centre for providing a workspace and access to essential resources during the project. In particular, we appreciate Anders Sjöberg for his valuable feedback and for serving as a dependable point of contact. His insights and suggestions were especially helpful during key phases of the project.

Our sincere thanks go to IFK Göteborg for generously sharing video data and meeting with us. Their contributions gave us a clearer understanding of how a football club might apply the results of this kind of work, helping us stay grounded in practical relevance.

Lastly, we extend our special thanks to Tomas McKelvey for kindly agreeing to act as our examiner on very short notice. We are truly appreciative of his time and support.

Anthon Odengard, Anton Hedén, Gothenburg, 2025-05-28

List of Abbreviation

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BBox	Bounding box
IoU	Intersection over Union
MOT	Multi-Object Tracking
PL	Pseudo-Labeling
Re-ID	Re-Identification
SN	SoccerNet
TBD	Tracking by Detection
TBQ	Tracking by Query

Contents

List of Figures

List of Tables

1

Introduction

Multi-object tracking (MOT) is a task in computer vision that involves detecting and following multiple objects across video frames while preserving their identities [7]. It plays a crucial role in autonomous driving, surveillance, and sports analytics. MOT enables tracking players, referees, and the ball in football, offering insights into match dynamics and supporting performance analysis. Such data is also essential for applications like virtual reality simulations and tactical analysis in football [8–13].

Deep learning approaches integrating object detection with tracking [14, 15] have largely superseded earlier MOT techniques. Convolutional neural network (CNN)-based models are commonly employed for object detection, while separate tracking algorithms associate detections over time to maintain object identities. However, these approaches still face difficulties in dynamic scenes such as football matches, where frequent occlusions and motion complexity lead to ID switches and missed detections [15].

To improve tracking under such conditions, recent methods have started incorporating temporal information [15–17]. These approaches show progress. However, they often struggle with long-term identity preservation and fail to reliably handle persistent occlusions or complex movement.

Current Deep Learning techniques in MOT can generally be divided into two categories: *Tracking by Detection* (TBD) and *Tracking by Query* (TBQ). TBD methods detect objects in each frame and then associate them over time. Techniques like SORT [18], BoT-SORT [19], and DeepSORT [20] enhance this process by using a model-based approach like Kalman filtering and appearance-based features. Despite these improvements, TBD methods are sensitive to occlusions and temporary object disappearance.

In contrast, TBQ, particularly using transformer-based models, offers a more integrated approach by leveraging attention mechanisms to propagate object identities across frames. Models such as MOTR [21] and TrackFormer [22] have shown promise in addressing occlusions and reducing ID switches in complex scenes.

This project investigates the application of such attention-based models to the football domain, where challenges like crowded scenes, varying object scales, and long-term tracking are especially prominent. Specifically, we aim to determine whether transformer architectures can effectively combine spatial features with temporal context to improve tracking stability and identity consistency.

In addition, we explore the use of pseudo-labeling (PL) and re-identification (Re-ID) techniques. PL enables semi-supervised learning by generating and refining annotations on domain-specific, unlabeled data. Re-ID methods help maintain identity consistency during prolonged occlusions or complex movement.

Through this work, we aim to assess the effectiveness of these methods for football-specific MOT and contribute to building more accurate, stable, and identity-aware tracking systems.

1.1 Aim and Scope

This thesis aims to develop and evaluate an MOT system tailored to football matches that are recorded with stationary cameras. The primary goal is to achieve robust, long-term identity preservation of players across full-length games despite challenges such as occlusions, scale variation, and fast-paced movement.

Our work focuses on enhancing transformer-based tracking architectures by leveraging their strengths in modeling spatio-temporal dependencies. In particular, we explore how these models can be adapted to the football domain and how temporal context can improve tracking accuracy.

To address the scarcity of annotated football data, we investigate using PL as a semi-supervised learning strategy to generate additional training supervision. We also examine integrating Re-ID mechanisms to reduce identity switches further and improve tracking consistency.

While this research is grounded in football analytics, the approaches developed may apply to other domains that require long-term, identity-aware tracking in structured environments.

1.2 Limitations

This study is confined to an *offline* tracking framework. Real-time performance is not a constraint, and computationally intensive models are acceptable if they improve identity consistency and tracking robustness.

All experiments, tests, and evaluations were conducted using stationary-camera footage from IFK Göteborg’s internal recordings. This fixed-camera setup avoids complications arising from camera motion but introduces challenges related to the large field of view (FoV), causing distant objects to be represented with fewer pixels.

By clearly defining these boundaries, we narrow the project’s focus to identity-aware object tracking under static camera conditions within sports analytics.

2

Background

This section will introduce the fundamental concepts, challenges, and methodologies relevant to MOT in the football context. Starting by defining MOT and reviewing both traditional and modern tracking approaches, highlighting the limitations of TBD methods compared to the more recent TBQ paradigm. We then discuss the contribution of auxiliary techniques, such as PL and Re-ID. Additionally, we outline the standard evaluation metrics used in the field, including HOTA, MOTA, IDF1, and IDSW. Together, these foundations establish the necessary background for understanding how advanced deep learning methods can be applied and adapted to the specific demands of football analytics.

2.1 Datasets

Practical training and evaluation of MOT models depend heavily on the availability of diverse and well-annotated datasets. In the context of football analytics, datasets fulfill a dual purpose: they provide essential ground truth for supervised learning and model benchmarking. Crucially, they also must accurately represent the specific real-world conditions where the models are intended to operate to ensure robust performance on the task.

This section presents the datasets used in this work, highlighting their structure, content, and relevance to the tracking task. We include both large-scale public benchmark datasets, such as SoccerNet [23,24], and real-world data recorded by IFK Göteborg using Spiideo systems [25]. While benchmark datasets offer standardized evaluation protocols and rich annotations, proprietary datasets more accurately represent the operational settings in which football tracking systems are deployed, but often without labels.

Together, these datasets support the development of a reliable tracking algorithm and the study of domain-specific challenges, which will be addressed in subsequent sections.

2.1.1 SoccerNet: A Large-Scale Benchmark Dataset

SoccerNet [1] is a comprehensive platform providing benchmark datasets designed to support research in video understanding and analytics within the football domain. It includes hundreds of professional football matches, offering video footage, event annotations, player tracking, and camera calibration data. The dataset is structured to facilitate multiple tasks such as action spotting, pose estimation, temporal grounding, and multi-view tracking.

SoccerNet includes a tracking dataset of one hundred 30-second clips recorded at 25 frames per second, totaling 75,000 frames. Each frame contains bounding box annotations with object identities for players, referees, staff, and the ball. The clips are filmed from a broadcasting camera perspective, and the annotations maintain consistent identities across frames, enabling reliable identity-preserving tracking.

Its standardized splits and evaluation metrics make it popular for benchmarking MOT and action recognition models in football contexts. However, the SoccerNet tracking datasets consist of broadcast footage involving dynamic camera operations such as panning and zooming. Unlike stationary camera setups, these varying perspectives introduce unique challenges, such as scale changes, motion blur, and shifting backgrounds, which may affect the performance of tracking systems not explicitly designed for such dynamic conditions [26].

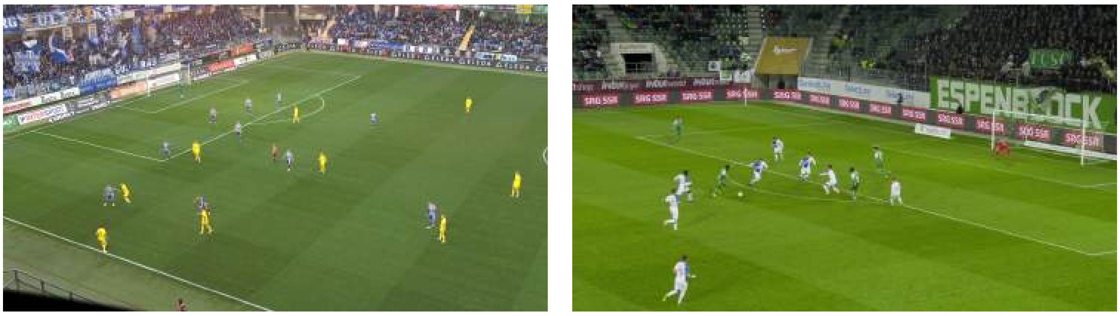
2.1.2 IFK Göteborg Dataset

In addition to standard benchmark datasets, our research incorporates a unique proprietary video dataset from the Swedish football club IFK Göteborg. This footage, captured using the professional sports video analysis system Spiideo [25], offers high-resolution panoramic views of an entire football pitch.

The recording setup utilizes two synchronized 4K cameras mounted on a gantry, with their feeds seamlessly stitched to create a complete field view. For this project, we focused on half of the pitch, similar to the view shown in Figure 2.1a, due to an image size limitation of 1920x1080 pixels during video access.

This dataset comprises actual footage from elite football matches and training sessions, accurately representing the real-world conditions in which professional football clubs use video analytics tools. Unlike typical broadcast footage, which often features dynamic camera movements and may not capture the whole field, Spiideo recordings provide a fixed, tactical view with consistent framing.

While this dataset offers valuable insights into real-world deployment, it lacks manual annotations for player positions or identities. This absence presents challenges for traditional supervised learning methods. However, its primary utility lies in unsupervised or weakly-supervised learning, domain adaptation, and rigorously testing tracking algorithms under authentic deployment conditions.



(a) Example from IFK Göteborg club data (b) Example from SoccerNet

Figure 2.1: Example frames from the two datasets utilized in this study. (a) A frame from the self-recorded match between IFK Göteborg and Kalmar FF was captured using the Spiideo system. (b) A representative frame from the publicly available SoccerNet [1] dataset.

2.2 Attention Mechanisms in Transformer Architectures

The transformer architecture has revolutionized many fields in machine learning, particularly in natural language processing [27], and has also shown promising results in computer vision [28–31]. Unlike traditional convolutional or recurrent operations, transformers use attention mechanisms to dynamically weigh different parts of the input, enabling flexible and global context modeling.

At the heart of this architecture lies the **attention mechanism**, which computes a weighted sum of value vectors based on the similarity between queries and keys. Given an input sequence $X = [x_1, x_2, \ldots, x_n]$, with $x_i \in \mathbb{R}^d$, we define:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \tag{2.1}$$

where W_Q, W_K, W_V are learned projections. Attention is then computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2.2}$$

2.2.1 Self-Attention

In **self-attention**, all three components- queries, keys, and values- are derived from the same input sequence X. That is show in equation 2.1.

This enables each element in the sequence to attend to all others, capturing contextual dependencies. In vision transformers, the input image is divided into patches and embedded into a sequence. Self-attention among these patch embeddings allows the model to reason globally about spatial relationships, such as object part interactions and large-scale structures. Since this operation is invariant to position, positional encodings are added to preserve spatial layout [32].

2.2.2 Cross-Attention

Cross-attention generalizes attention to operate between two different sequences. Queries come from one input X, while keys and values are derived from another input Y:

$$Q = XW_Q, \quad K = YW_K, \quad V = YW_V. \tag{2.3}$$

This mechanism enables one set of inputs to selectively attend to another, allowing for alignment and integration of heterogeneous information. In vision applications, cross-attention often enables object queries to attend to spatial feature maps extracted by a CNN backbone, as in DETR [28]. This interaction drives the detection process by associating queries with relevant visual evidence.

2.2.3 Multi-Head Attention

Multi-head attention enables the model to capture diverse types of relationships by computing multiple attention functions in parallel. For each head $i = 1, \ldots, h$, the queries, keys, and values are projected into different subspaces:

$$Q_i = XW_Q^{(i)}, \quad K_i = XW_K^{(i)}, \quad V_i = XW_V^{(i)}, \tag{2.4}$$

where each $W^{(i)}$ is unique to the i^{th} head and $d_k = d/h$. Each head performs its own attention computation

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i), \tag{2.5}$$

and the outputs are concatenated and linearly transformed:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O. \tag{2.6}$$

Multi-head attention improves the model's expressiveness by allowing each head to focus on different relationships, such as local vs. global patterns. In vision applications, this will enable models to capture texture, shape, and motion cues across image patches or video frames [22, 33].

2.3 Object Detection

Object detection identifies and localizes semantic entities within an image, such as people, balls, or vehicles. In contrast to image classification, which assigns a single label to an entire image, object detection predicts each instance's class and spatial coordinates, typically represented as bounding boxes [34, 35]. Figure 2.2 illustrates an example frame with detected objects, showing how bounding boxes and confidence scores are assigned to multiple entities.

At its core, object detection comprises two key components: classification (determining the object category) and localization (estimating spatial position). These tasks are commonly learned jointly through deep neural networks trained on annotated datasets [36].

Modern detection pipelines leverage convolutional or transformer-based architectures to extract high-level spatial features from images. These features are processed to generate object proposals, each associated with a class label and confidence score [34, 35]. Such detections serve as the foundation for downstream tasks like MOT [14, 22, 30].

The following subsections describe two dominant paradigms in contemporary object detection: convolution-based and transformer-based approaches.

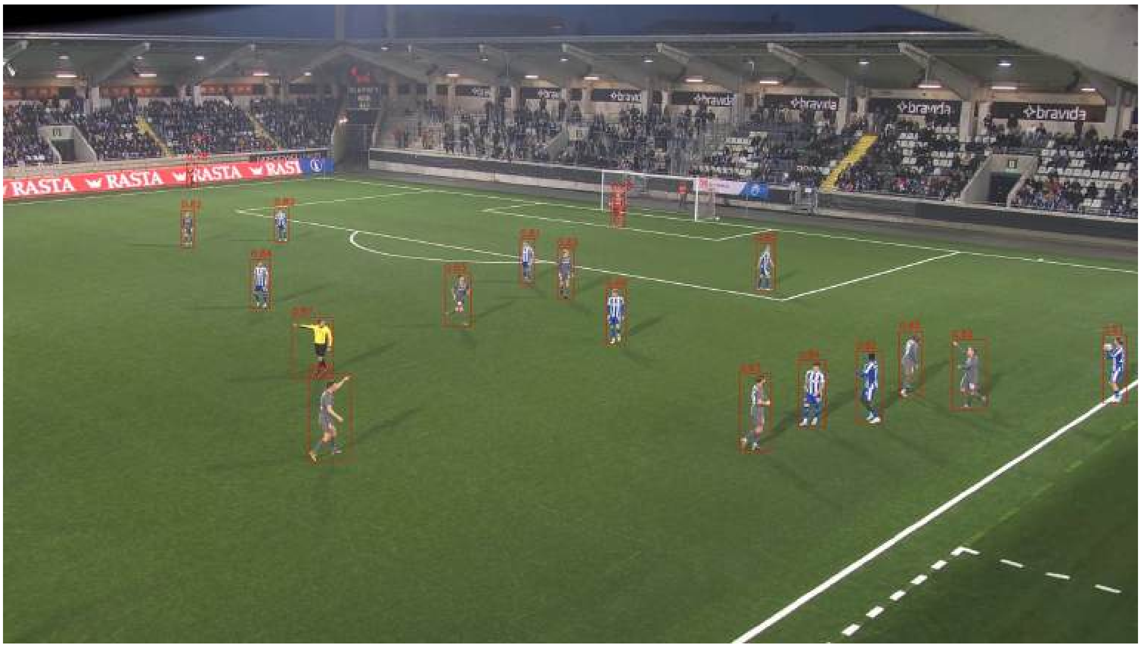


Figure 2.2: Visualization of object detections showing predicted bounding boxes and their associated confidence scores on a sample frame. The confidence score reflects the model's certainty in the presence and classification of each detected object.

2.3.1 Convolution-based Object Detection

Convolutional neural networks (CNNs) have long been central to object detection and play a key role in modern systems. CNN-based detectors typically fall into two main categories: one-stage detectors [34], which emphasize speed, and two-stage detectors [35], which often prioritize accuracy by decoupling proposal generation from classification. These models remain foundational for many real-time and high-accuracy detection pipelines.

This section describes two representative families of CNN-based detectors: YOLO (You Only Look Once), which exemplifies the one-stage paradigm, and Faster R-CNN, a two-stage detector.

2.3.1.1 YOLO: You Only Look Once

YOLO [34] is a family of real-time object detection models that reformulate detection as a single-stage regression problem. Unlike traditional two-stage approaches, which generate and classify region proposals, YOLO simultaneously performs object localization and classification in a single forward pass. This unified architecture enables high-speed inference while maintaining competitive accuracy, making it well-suited for time-sensitive applications like real-time tracking in video streams.

Rather than relying on computationally expensive region proposal networks, YOLO directly predicts bounding boxes and class probabilities from the entire image, leveraging a fully convolutional backbone network. This streamlined pipeline allows for highly efficient object detection with low latency.

A core advantage of YOLO is its ability to balance speed and accuracy [37], which is critical for downstream tasks such as MOT [38]. Its design ensures frame-by-frame detections can be generated in real time, even on modest hardware, enabling its widespread use in domains such as autonomous driving, surveillance, and sports analytics.

2.3.1.2 Faster R-CNN

Fast R-CNN [35] is a region-based object detection model that improves the speed and accuracy of its predecessor, R-CNN, by sharing computation across region proposals. Unlike R-CNN, which processes each region independently, Fast R-CNN computes a convolutional feature map over the entire image just once and then classifies regions of interest (RoIs) using features pooled from this shared map.

At its core, Fast R-CNN uses a convolutional backbone such as VGG16 [39] or ResNet [40] to extract spatial features from the input image. Region proposals, typically generated by an external method like Selective Search, are mapped onto the feature map. A RoI pooling layer extracts a fixed-size feature vector for each region, which is then passed through fully connected layers to produce both softmax class scores and bounding box refinements [35].

This design enables efficient end-to-end training via a multi-task loss that combines classification and localization. Since the feature extraction is shared, Fast R-CNN achieves significantly faster inference than R-CNN while improving accuracy.

2.3.2 Transformer-based Object Detection

Transformer-based approaches have recently gained traction in computer vision tasks, including object detection [28–30]. Unlike traditional CNN-based detectors that rely heavily on hand-crafted heuristics such as anchor boxes and non-maximum suppression [41], transformer-based models adopt a set-based prediction framework and leverage global self-attention mechanisms to reason about image-wide context.

This paradigm shift offers a unified and interpretable framework for object detection, with the benefit of being inherently extensible to tasks such as instance segmentation [31] and MOT [2,22]. In this section, we first describe DETR [28], which introduced the use of transformers in object detection, followed by Deformable DETR [29], a more efficient variant that addresses some of DETR's key limitations.

2.3.2.1 DETR: DEtection TRansformer

DETR [28] is a transformer-based object detection model that rethinks the traditional object detection pipeline by removing the need for hand-crafted components such as anchor boxes and non-maximum suppression. Instead, DETR formulates object detection as a set prediction problem, using a transformer architecture to directly output a fixed number of object predictions in one pass.

At the heart of DETR [28] is a CNN backbone that extracts a 2D high-level spatial feature map from the input image. This feature map is then flattened into a sequence of feature vectors, which serves as the input to the transformer encoder. The transformer encoder-decoder architecture then models relationships across the entire image and performs detection via attention mechanisms. Transformers, initially developed for natural language processing tasks [27], have shown great promise in vision tasks as well, where they enable models to capture long-range dependencies and contextual information across an entire image [42, 43]

The encoder processes the flattened feature map and applies self-attention [28], allowing the model to globally reason about all spatial regions in the image, which is especially important for capturing long-range dependencies and contextual cues.

The decoder introduces a set of learned object queries [28], each designed to attend to different regions of the encoded image features. These queries interact with the encoder output through cross-attention, where each query dynamically attends to all positions in the encoded feature map to decide where and what object to predict. This cross-attention mechanism is crucial: rather than scanning the image spatially or using sliding windows or region proposals, each query explores the image in a content-aware way, producing a single bounding box and class label prediction.

Because DETR predicts a fixed set of objects, the order of these predictions is not inherently tied to the objects themselves, making the output permutation-invariant. To handle this during training, DETR uses Hungarian matching [44] to find the optimal one-to-one assignment between the predicted boxes and the ground truth annotations. This set-based loss formulation avoids duplicate detections and eliminates the need for post-processing steps like non-maximum suppression.

While DETR is highly accurate and conceptually elegant, it is computationally intensive and relatively slow to converge during training. Nonetheless, it laid the foundation for a new class of detection models,transformer-based architectures, that can naturally extend to tasks like tracking by query [22], where temporal consistency and object identity preservation are critical.

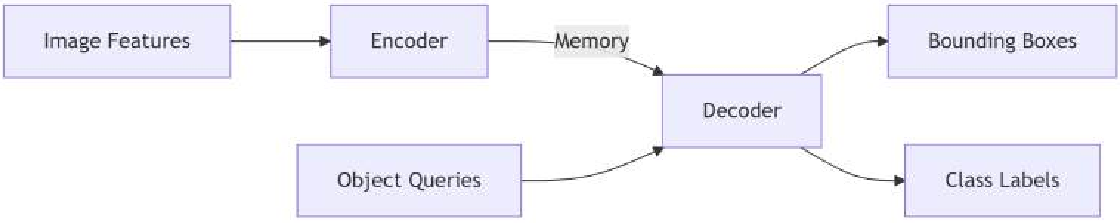


Figure 2.3: Schematic simplified view of DETR

2.3.2.2 Deformable DETR

Deformable DETR [29] builds upon the original DETR architecture by addressing its main limitations, namely, slow training convergence and difficulties in detecting small objects. While DETR relies on full self-attention over all spatial positions, this becomes computationally expensive and inefficient, especially for high-resolution images. To mitigate this, Deformable DETR introduces multi-scale deformable attention, which replaces the dense attention mechanism with a more efficient one. Instead of attending to all spatial locations uniformly, each query only attends to a small, learnable set of key sampling points across multiple feature scales. These sampling offsets are dynamically predicted and focus the attention on the most relevant regions of the image.

This sparsity reduces the computational cost and enhances the model's ability to detect small or occluded objects by aggregating contextual information from multiple resolutions [29]. As a result, Deformable DETR achieves significantly faster convergence and better detection accuracy, particularly for challenging object instances.

2.4 Tracking

Tracking refers to maintaining the identities and locations of objects as they move through a video over time. In MOT, this involves linking individual object instances across frames to form coherent trajectories, even as objects enter, exit, occlude each other, or change appearance. Prominent examples of successful MOT algorithms include SORT [14], DeepSORT [20], and ByteTrack [45].

The core challenge in MOT lies in accurately detecting objects in each frame and reliably associating them over time, despite various ambiguities such as visual occlusions or complex motion patterns [21, 22].

2.4.1 Tracking paradigms

Similar to object detection, where models are often divided between two-stage proposal-based methods and single-stage unified approaches, MOT has evolved around two dominant paradigms: Tracking-By-Detection (TBD) and Tracking-By-Query (TBQ). These paradigms reflect differing philosophies in how to model identity over time, whether by post-hoc association of per-frame detections [14, 45] or by learning persistent object representations that evolve across frames [2, 22, 46].

2.4.1.1 Tracking-by-Detection

TBD is a foundational paradigm in MOT that formulates the problem as a two-step process: detecting objects independently in each frame, then associating these detections across time to form coherent trajectories. Though rooted historically in computer vision, modern TBD approaches heavily rely on advances in object detection, as demonstrated by methods like SORT [14] and ByteTrack [45].

Instead of explicitly modeling complex temporal dynamics, TBD methods recover object identity retrospectively by evaluating spatial proximity, motion consistency, and appearance similarity between detections in consecutive frames. This paradigm emerged alongside deep learning detectors, allowing tracking systems to leverage powerful, pre-trained models for object identification. Such modularity enables tracking algorithms to benefit immediately from improvements in detection, largely independent of the specific detector employed.

TBD assumes per-frame detections are accurate and consistent enough to support reliable data association. To link detections, methods use cues like predicted displacement, appearance features, and spatial consistency, assuming objects maintain relatively stable visual characteristics and smooth motion over short intervals [20,45]. However, limiting associations primarily to adjacent frames hampers recovery from prolonged occlusions or long-term disappearance.

While effective in controlled environments, TBD performance degrades under challenging conditions such as high object density, frequent occlusions, or abrupt motion, typical in scenarios like professional football [15]. Here, independent frame-level detections can cause identity switches, fragmented trajectories, or missed tracks. Nonetheless, the TBD paradigm remains influential.

2.4.1.2 Tracking-by-Query

TBQ is a modern paradigm for MOT that redefines identity preservation across video frames. Unlike traditional approaches that rely on motion modeling or post-hoc data association, TBQ represents each tracked object as a learnable query vector that is propagated across time, enabling consistent tracking through motion, occlusion, and appearance changes.

TBQ builds on the Transformer architecture, originally developed for sequence modeling [27], and leverages its ability to encode rich temporal dependencies. Rather than processing frames independently, TBQ frameworks operate holistically over sequences, using persistent object queries and frame-level features to maintain identity continuity [2, 21, 22].

A core advantage of TBQ lies in its unification of detection and tracking within a single architecture. Inspired by set-based detection models like DETR [28] and Deformable DETR [29], TBQ interprets object tracking as a set prediction task. Object queries are dynamically updated using learned visual and contextual cues, eliminating the need for separate detection and association steps. This integration improves robustness in scenes with occlusions, motion blur, or visually similar objects.

Its end-to-end trainable structure allows object queries to learn temporal consistency and spatial attention jointly. These capabilities make TBQ particularly well-suited to dynamic environments like football, where reliable identity tracking and contextual reasoning are critical.

2.4.2 BoT-SORT

BoT-SORT [19] is a MOT-by-detection algorithm that builds upon the ByteTrack framework, introducing a series of enhancements designed to improve robustness and accuracy in complex tracking scenarios. One of its key contributions is the integration of camera motion compensation. By estimating and correcting for camera movement, BoT-SORT reduces errors in motion prediction that can lead to identity switches, particularly in dynamic scenes.

Another modification lies in the use of a refined Kalman filter. Unlike traditional approaches that estimate object size based on aspect ratio, BoT-SORT independently tracks bounding box width and height. This adjustment allows for more accurate object localization, mainly when scale variations occur due to perspective changes.

BoT-SORT combines spatial overlap using Intersection over Union (IoU) with appearance information derived from deep Re-ID features for data association. This fusion enables more reliable matching between detections and existing tracks, while a dual-thresholding strategy ensures that only strong associations are maintained, reducing false matches.

2.4.3 Transformer-Based Tracking with MOTRv2

MOTRv2 is a state-of-the-art end-to-end MOT framework that addresses challenges such as detection accuracy, identity consistency, and occlusion handling by building upon and extending the original MOTR architecture [2, 21]. The design is rooted in the Deformable DETR [29] transformer framework and introduces significant modifications that improve performance over traditional TBD methods and earlier end-to-end approaches.

In MOTR, object and track queries are used jointly to enable end-to-end MOT. Object queries are responsible for detecting new or previously untracked objects in each frame and are assigned to ground-truth instances using bipartite (Hungarian) matching. In contrast, track queries are persistent and maintain object identities across frames by propagating information from previous time steps, without being re-matched at every frame. These track queries are dynamically updated using their historical states and features extracted from the current frame. MOTR incorporates a temporal aggregation module to improve temporal consistency, fusing past and present information for each track query. During training, losses are computed at each time step and averaged over the sequence, encouraging stable learning and consistent identity tracking.

One of the main limitations of MOTR is its relatively weak detection performance compared to tracking-by-detection approaches that utilize powerful, standalone object detectors. To overcome this, MOTRv2 integrates the YOLOX detector [47], which provides high-quality detection proposals before the tracking module. This decoupling of detection and association tasks allows the model to benefit from task-specific detection priors while retaining the strengths of transformer-based tracking.

The overall architecture of MOTRv2 consists of two main stages: proposal generation and transformer-based tracking. In the first stage, YOLOX produces bounding box proposals for each video frame. These proposals include center coordinates, width, height, and confidence scores, and serve as object anchors. The tracking component then processes these proposals using a modified transformer decoder, which operates on a concatenation of track and proposal queries.

In MOTRv2, proposal queries are constructed from YOLOX detections by broadcasting a shared learnable embedding across all proposals and enriching them with sine-cosine positional encodings derived from the detection boxes and their confidence scores. Unlike the fixed learnable queries used in Deformable DETR and MOT, MOTRv2 combines these proposal-based queries with additional learnable queries to enhance tracking performance. These proposal queries are dynamically generated and work alongside the learnable queries to effectively model detection and motion cues.

Only new objects are present at the first frame ($t = 0$). The proposal queries are initialized from YOLOX outputs and processed using self-attention, followed by deformable attention to interact with image features. This results in track query predictions and bounding box offsets relative to the YOLOX proposals. The final detection boxes are obtained by adding these offsets to the original anchors.

In subsequent frames ($t > 0$), the transformer decoder inputs the concatenated set of track queries from the previous frame and new proposal queries generated from the current YOLOX proposals. The anchors consist of the earlier frames predicted boxes and the current YOLOX detections. Their sine-cosine encodings are used to create the positional embeddings. The decoder then updates both the object predictions and track queries. The updated track queries are carried forward to the next frame, allowing consistent identity propagation.

This architecture encourages a functional separation between the two query types: proposal queries detect new or missing instances, while track queries focus on maintaining object identities. Proposal and track queries can exchange information through self-attention mechanisms in the transformer decoder. This interaction helps avoid duplicate detections and improves the localization of tracked objects. Visual analysis of the self-attention maps confirms that the proposal and corresponding track queries for the same object share high similarity, supporting the effectiveness of this cooperative design.

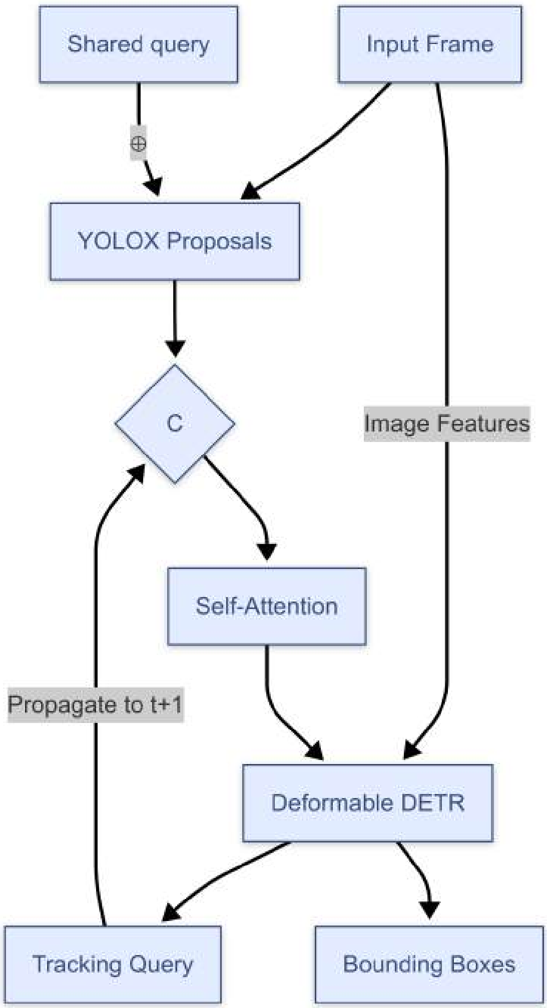


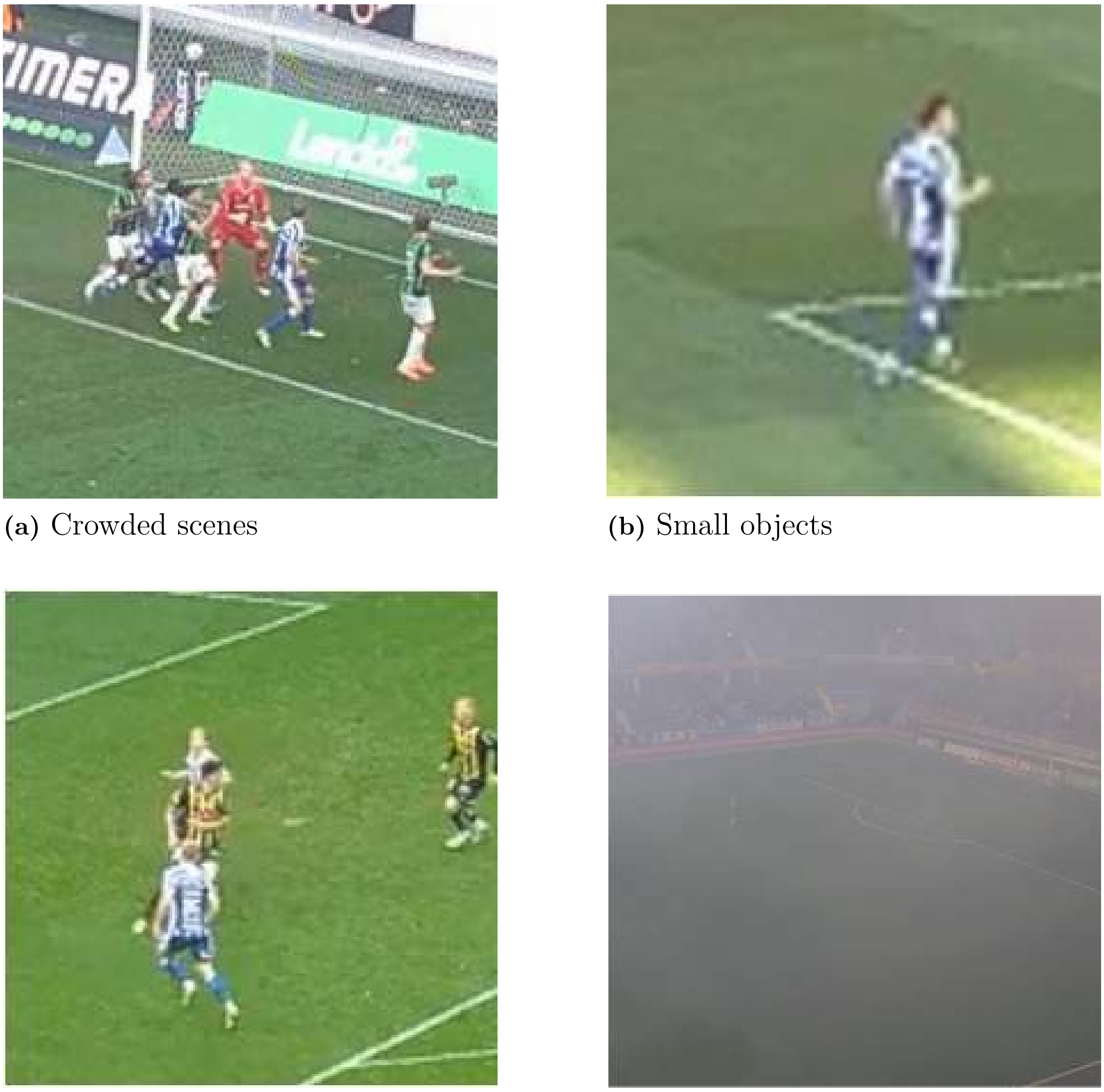
Figure 2.4: Simplified illustration of the MOTRv2 architecture, adapted from original source [2]. Symbols: ⊕ denotes addition; C denotes concatenation.

2.5 Challenges in Multi-Object Tracking

MOT in football and sports is a highly complex task involving detecting and associating multiple objects across frames while maintaining consistent identities over time [7,48]. Several inherent challenges arise in this domain due to the dynamic nature of football, environmental factors, limitations in available data, and the stringent performance requirements for real-world applications. As seen in Figure 2.5 (a–d), these challenges include small objects, crowded scenes, occlusions, and varying lighting conditions.

2.5.1 Occlusion and Crowded Scenes

One of the primary challenges in football MOT is the frequent occlusions, particularly in crowded scenes where players form dense groups [49, 50] as shown in Figure 2.5a and Figure 2.5c. Football involves intense physical contact and rapid transitions, leading to frequent occlusions during set pieces or fast breaks. Tracking algorithms struggle to maintain consistent identities across frames when multiple objects overlap or become hidden behind one another. The complex player interactions during such periods create highly ambiguous tracking conditions, further complicating long-term tracking and Re-ID tasks. In addition, low visibility, varying lighting conditions (Figure 2.5d), and the small number of pixels per player (Figure 2.5b) when they are far away further challenge the performance.



(a) Crowded scenes

(b) Small objects

(c) Occlusions

(d) Varying lighting conditions

Figure 2.5: Different hard scenarios from the IFK dataset

2.5.2 Appearance Variation

Football players' appearances can vary significantly across frames due to several factors, including lighting conditions, motion blur, and pose changes. These variations can degrade the tracking system's ability to maintain consistent identities across frames [7, 51]. For example, lighting changes caused by weather conditions or different times of day can alter the visual characteristics of players. Additionally, motion blur from high-speed movements can obscure key features, making visual matching more difficult. Even when conditions are stable, players often wear nearly identical uniforms and exhibit similar movements, making it especially challenging to distinguish between them based on appearance alone. This visual similarity significantly complicates Re-ID, increasing the risk of identity switches in crowded or dynamic scenes [4].

2.5.3 Lack of Annotations in Self-Filmed Data

In self-filmed datasets, such as those captured by Spiideo, the lack of annotated data presents a significant challenge [50]. These datasets may include high-resolution footage of the entire field but lack manual annotations such as player identities and bounding boxes. Training fully supervised models becomes difficult without such annotations, as the system lacks the necessary ground truth for training. This limitation necessitates using alternative methods, such as PL, to overcome the absence of labeled data.

2.5.4 Fast or Erratic Motion

The unpredictable nature of player movements in football, including rapid changes in direction and speed, presents a significant challenge for tracking systems [7, 51]. Traditional tracking methods often rely on assumptions of smooth motion and may struggle to maintain consistent tracking when players exhibit erratic or sudden movements. These unpredictable motions can lead to identity loss or incorrect track reinitialization, as the tracking system may fail to anticipate sudden changes in trajectory or velocity. Advanced tracking systems must incorporate motion models that account for these fast and erratic movements to ensure continuous tracking.

2.5.5 Domain Shift and Transfer Learning in MOT

MOT models trained on general datasets like MOT17 [52] or SoccerNet [24] often face performance degradation when applied to football due to domain shift—differences in data distributions between training and deployment environments. Football-specific challenges, such as unique camera angles, lighting variations, fast-paced motion, frequent occlusions, and visually similar players, are often not well-represented in generic datasets.

These domain-specific factors complicate appearance-based tracking, including motion blur from rapid movements and identical uniforms. To address this, transfer learning techniques adapt pre-trained models to the football domain, including:

- **Fine-tuning on target domain data**: The model, pre-trained on a general dataset, is further refined using a smaller, domain-specific dataset, allowing it to capture football-relevant features without re-training from scratch.

- **Domain adaptation**: Adaptation is performed using unlabeled or partially labeled target data, often through self-training, where high-confidence predictions are used as pseudo-labels to iteratively align the model with domain-specific characteristics.
- **Data augmentation**: Applying synthetic transformations, such as variations in lighting, occlusions, and viewing angles, enhances the model's robustness to real-world football scenarios.

Applying these transfer learning strategies may help mitigate the adverse effects of domain shift, potentially improving the robustness of MOT systems in real-world football scenarios. Given the variability in match conditions, player characteristics, and broadcast configurations, such adaptation is likely beneficial for developing more accurate and reliable tracking solutions in sports analytics. In the following section, we explore the practical impact of these techniques in more detail.

2.6 Pseudo-Labeling in Tracking

Pseudo-labeling (PL) is a core technique in semi-supervised learning (SSL) that enables the use of unlabeled data by generating artificial labels based on model predictions [53]. The core idea is to treat confident model predictions as supervisory signals, allowing models to improve iteratively even when data is limited. PL is especially valuable in domains where manual annotation is costly or scarce. The following section outlines the foundations of PL, its key variants, and its relevance to modern learning frameworks.

2.6.1 Semi-Supervised Learning

One of the most influential applications of PL is in the context of SSL. A foundational approach introduced by Lee [54] treats PL as a fine-tuning stage following supervised pretraining. In this framework, a model is initially trained on a small labeled dataset. Then, pseudo-labels, generated by the model's predictions, are assigned to unlabeled data and incorporated into continued training. These pseudo-labels reinforce confident predictions, and training proceeds by optimizing a joint loss: a supervised loss on labeled data and an unsupervised loss on pseudo-labeled samples, weighted by a time-dependent schedule.

Subsequent research has built on this core idea, addressing key challenges such as selecting unlabeled samples, managing label noise, and preventing confirmation bias. Sampling strategies vary from random selection to more structured approaches. For instance, FixMatch [55] applies a confidence threshold to filter pseudo-labels. Hybrid methods like MixMatch [56] combine augmentation, entropy minimization, and mixup regularization to improve generalization and robustness.

More recent innovations, such as Meta Pseudo Label frame PL as an optimization problem [3]. This method uses a teacher-student architecture where the teacher assigns pseudo-labels, and the teacher's parameters are updated based on the student's performance on labeled data. This dynamic feedback loop leads to more effective pseudo-label generation over time.

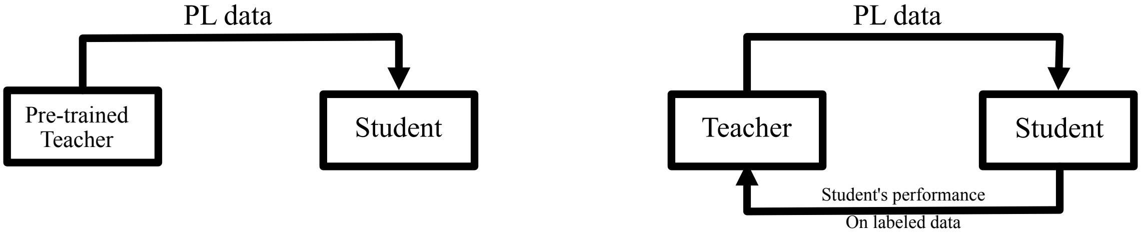


Figure 2.6: Left: In traditional PL, a pre-trained teacher model produces fixed pseudo labels, which are then used to train the student model. Right: The teacher and student are updated jointly in the meta pseudo-labeling framework. The student learns from the teacher's pseudo labels, while the teacher adapts its predictions based on the student's performance on labeled data. Illustration inspired by [3]

2.6.2 Self-Supervised and Unsupervised Learning

Although PL is most commonly associated with semi-supervised learning, its underlying idea also appears in many unsupervised and self-supervised learning methods. In particular, some self-supervised approaches use label-like signals, without relying on actual ground-truth labels, to guide the learning process. For example, frameworks like DINO [57] group similar data points together or enforce consistency across different views of the same input. These methods effectively assign temporary labels or categories to data and use them to train the model self-directedly.

This form of PL satisfies the key principle of assigning semantic structure to unlabeled data despite its lack of correspondence to predefined classes. Models are trained to pull together embeddings of augmented pairs and push apart representations of unrelated samples.

Virtual Adversarial Training (VAT) [58] is another self-supervised consistency technique, which perturbs samples within a small neighborhood to enforce local smoothness of model predictions. Like clustering-based methods, VAT produces pseudo-labels implicitly by assuming that samples close in input space should receive the same classification. These methods thus demonstrate how PL can extend beyond classification tasks and into general-purpose representation learning.

2.6.3 Challenges related to Pseudo Labeling

Despite its broad applicability, PL presents several critical challenges. A primary concern is confirmation bias, where incorrect predictions are reinforced over time, degrading model performance [56].

PL also suffers from distribution mismatch, an implicit assumption that labeled and unlabeled data share the same underlying distribution [59]. In practice, this often leads to poor generalization when a domain shift is present.

Selection strategies for pseudo-labels further influence model outcomes. Random selection may introduce noise, while metric-based selection, such as using prediction confidence or temporal consistency, aims to filter for higher-quality labels.

Finally, overfitting to noisy or biased pseudo-labels remains a persistent issue. Recent work explores remedies such as confidence-aware losses [59], label debiasing, and consistency regularization to increase robustness and generalization [60].

2.7 Re-Identification in Multi-Object Tracking

In the context of MOT, one of the significant challenges is maintaining consistent object identities across time, especially in scenarios involving long sequences, occlusions, and interactions between similar-looking objects [4, 7, 51]. Identity switches, or ID switches, occur when a tracked object's identity is mistakenly assigned to another object, often due to appearance ambiguities, detection failures, or occlusions. While modern trackers aim to minimize such occurrences, they remain persistent in complex environments such as sports footage [15].

2.7.1 Feature-Based Tracklet Splitting and Clustering

Feature extraction is central to identifying inconsistencies within tracklets. Global Tracklet Association (GTA) [4] employs OSNet [61] to compute discriminative appearance embeddings for each bounding box. GTA's tracklet splitter uses these embeddings to detect identity mixups within a single tracklet by clustering appearance features using a modified DBSCAN algorithm.

Unlike the standard DBSCAN, GTA's version reassigns initial outliers to the nearest cluster, assuming that every detection contributes useful identity information. Clustering is controlled via three hyperparameters: the minimum number of samples per cluster (s), the maximum neighbor distance (ϵ) using cosine similarity, and a maximum cluster count (k) to avoid over-fragmentation. This ensures that each tracklet fragment maintains visual consistency, improving identity purity prior to re-association.

2.7.2 Tracklet Re-Association via Connector Methods

Once tracklets are split, GTA applies a connector module to merge fragments likely belonging to the same object. This begins by constructing a symmetric distance matrix $D_{i,j}$ between all tracklet pairs:

$$D_{i,j} = \begin{cases} 1, & \text{if } i \neq j \wedge \Pi_i \cap \Pi_j \neq \emptyset \\ \frac{1}{N_i N_j} \sum\limits_{m \in \Pi_i} \sum\limits_{n \in \Pi_j} \left(1 - \frac{F_m^i \cdot F_n^j}{\|F_m^i\| \|F_n^j\|}\right), & \text{otherwise} \end{cases} \tag{2.7}$$

Here, Π_i and Π_j represent the temporal spans of tracklets T_i and T_j, while F_m^i and F_n^j denote the feature vectors at frame m and n respectively. N_i and N_j are the number of frames in each tracklet. This matrix quantifies the dissimilarity between two tracklets in feature space.

To exclude implausible pairings, spatial constraints are applied based on field geometry. The maximum allowed horizontal and vertical displacements are defined as:

$$\theta_{\text{hor}} = \beta \cdot \Delta_{\text{max,hor}}, \quad \theta_{\text{ver}} = \beta \cdot \Delta_{\text{max,ver}} \tag{2.8}$$

If the spatial offset between two tracklets exceeds these thresholds, their distance is set to 1:

$$D_{i,j} = 1, \quad \text{if } \Delta_{i,j,\text{hor}} > \theta_{\text{hor}} \vee \Delta_{i,j,\text{ver}} > \theta_{\text{ver}} \tag{2.9}$$

Finally, hierarchical clustering is applied to the filtered distance matrix. Tracklets are iteratively merged until no pair has a distance below a merging threshold α. This allows GTA to reconstruct long, consistent tracklets from fragmented sequences, ensuring continuity even across occlusions or earlier ID switches.

2.7.3 Graph-Based Tracklet Association

An alternative clustering approach is the construction of a tracklet similarity graph, where each node represents a tracklet and edges are weighted by feature similarity and spatio-temporal distance. Spectral clustering [62] can be applied to this graph to partition it into groups corresponding to individual identities. This method is particularly advantageous in dense or interactive scenes, where the relationships between tracklets are complex and non-linear. The graph structure also allows for global reasoning about identity assignments, increasing robustness to tracking noise.

2.7.4 Feature Representation Techniques

The success of Re-ID largely depends on the quality of extracted features. CNN-based models such as ResNet [36] are commonly used to encode visual appearance. Additionally, autoencoder architectures [63] offer an unsupervised approach to learn compact and informative embeddings, which can be helpful in settings with limited annotations. These representations serve as the foundation for splitting inconsistent tracklets and reconnecting visually and temporally similar ones.

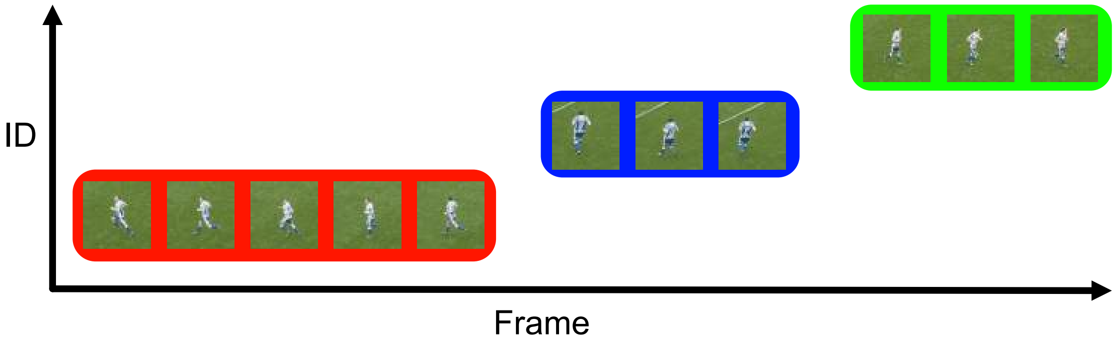


Figure 2.7: ID switching of the same player over time. Illustration inspired by [4].

2.8 Metrics

To properly evaluate the performance of MOT algorithms, it is essential first to understand the basic concepts of classification outcomes. These outcomes are fundamental to calculating various metrics such as precision, recall, and accuracy.

2.8.1 Intersection over Union

Intersection over Union (IoU) is a metric that evaluates the overlap between a predicted bounding box and the ground truth bounding box, also called the Jaccard index [64]. It is defined as the area of the intersection divided by the area of the union of the two bounding boxes:

$$\text{IoU} = \frac{|\text{Prediction} \cap \text{Ground Truth}|}{|\text{Prediction} \cup \text{Ground Truth}|} \tag{2.10}$$

And also illustrated in Figure 2.8.

IoU ranges from 0 to 1, where 1 indicates perfect alignment between the predicted and ground truth boxes, and 0 indicates no overlap.

Figure 2.8 illustrates this concept.

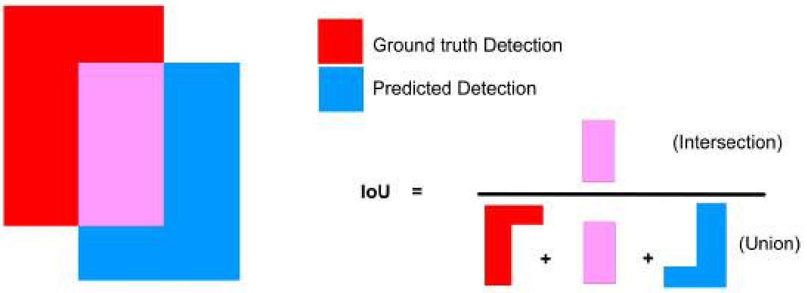


Figure 2.8: Illustration of Intersection over Union (IoU). The blue box is the prediction, the red box is the ground truth, and the shaded pink region represents their intersection.

2.8.2 Classification Metric

The performance of MOT systems can be evaluated based on four key detection/-tracking outcomes derived from comparing the predicted object detections and the ground truth objects. These outcomes include:

- **True Positives (TP)**: Correctly identified and detected or tracked an object shown in a green box in Figure 2.9.
- **False Positives (FP)** shown in the outlined red box in Figure 2.9: Incorrectly detected objects that do not correspond to any ground truth object.
- **False Negatives (FN)**: Ground truth objects that are not detected or tracked are shown in a red overlay in Figure 2.9
- **True Negatives (TN)**: Correctly identified non-objects (typically not included in tracking scenarios).

Note that these outcomes are often interdependent. For example, increasing the number of TP often leads to an increase in FP and a reduction in FN. Conversely, trying to reduce the number of FP may also reduce the number of TP and increase FN. Thus, it is a trade-off.

These detection/tracking outcomes form the basis for computing various performance metrics, such as precision, recall, and tracking-specific measures like MOTA and HOTA.

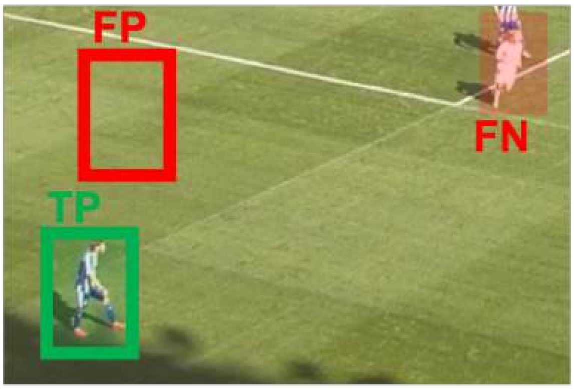


Figure 2.9: Visual example of True Positive (TP), False Positive (FP), and False Negative (FN).

2.8.3 Identity Switches (IDSW)

The Identity Switches (IDSW) metric measures the number of times a tracker incorrectly changes the identity assignment of a target over time [5]. An identity switch occurs when a ground truth object matched to a particular tracker hypothesis in frame $t-1$ is matched to a different tracker hypothesis in frame t.

This concept is illustrated in Figure 2.10. Here, the ground truth target is first matched to the red tracker trajectory, but starting from frame 4, the assignment changes to the blue trajectory—this is counted as one ID switch.

Formally, the IDSW metric is counted as follows:

- A ground-truth trajectory g is matched to tracker ID h_1 in frame $t-1$, and
- In frame t, the same trajectory g is matched to a different tracker ID $h_2 \neq h_1$.

These switches degrade the quality of long-term identity consistency, which is essential in applications such as sports analytics.

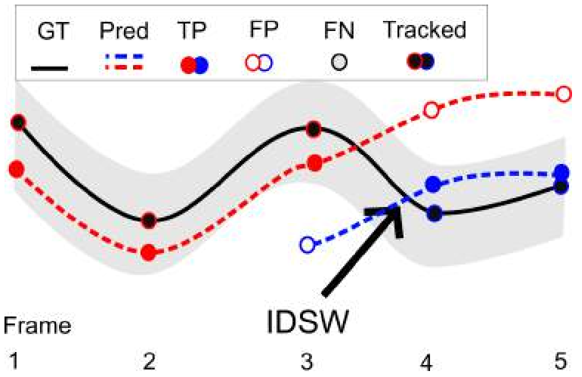


Figure 2.10: Illustration of identity switches (IDSW) inspired by [5].

2.8.4 Precision, Recall, and F1 Score

Tracking precision and recall evaluate the performance of object detection over time. Precision measures how accurately the predicted tracks match the ground truth, while recall measures how many ground truth tracks are correctly identified by the tracker. These metrics are critical for understanding the trade-off between missing objects (false negatives) and producing false detections (false positives).

The formulas are:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.12}$$

To capture both aspects in a single score, the F1 score is used, which is the harmonic mean of precision and recall:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.13}$$

A high F1 score indicates that the tracker correctly identifies objects and minimizes errors. These metrics provide a more nuanced evaluation than raw accuracy, especially in scenarios with class imbalance or variable object visibility.

2.8.5 Multiple Object Tracking Accuracy (MOTA)

MOTA (Multiple Object Tracking Accuracy) is a widely used metric to evaluate the performance of an MOT system. It combines multiple aspects of tracking performance, accounting for false positives, false negatives, and identity switches. The formula for MOTA is given by:

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FP} + \mathrm{FN} + \mathrm{IDSW}}{\mathrm{GT}} \tag{2.14}$$

where GT is the number of ground truth objects. A higher MOTA score indicates better tracking performance, reflecting fewer tracking errors.

While comprehensive, MOTA has a known limitation: it tends to be heavily influenced by the performance of the underlying object detector. Although all three error types are weighted equally on a per-occurrence basis in the formula, the cumulative number of false positives and false negatives across many frames often far exceeds the number of identity switches. This means a tracker can achieve a relatively high MOTA even with frequent identity switches if its object detection is very robust. Consequently, MOTA is often complemented with other metrics, such as IDF1 or HOTA, to gain a more complete understanding of tracking system performance.

2.8.6 Identification F1 Score (IDF1)

The Identification F1 Score (IDF1) evaluates how well a tracking system maintains objects' correct identities over time. It is the harmonic mean of identification precision and recall, where matches are based on identity consistency rather than just spatial overlap.

The formula for IDF1 is:

$$\mathrm{IDF1} = \frac{2 \times \mathrm{IDTP}}{2 \times \mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \tag{2.15}$$

where:

- **IDTP**: true positive identity matches (correct ID assignments),
- **IDFP**: false positive identity assignments (incorrect IDs),
- **IDFN**: missed identity matches (missed tracks).

IDF1 is especially useful for assessing identity preservation in MOT. A high IDF1 indicates that the system not only detects objects but also maintains their identities accurately over time.

2.8.7 Higher Order Tracking Accuracy (HOTA)

Higher Order Tracking Accuracy (HOTA) is a modern metric designed to provide a balanced evaluation of multiple object tracking performance [6]. Unlike traditional metrics focusing heavily on either detection or identity association, HOTA jointly measures both aspects. Specifically, it computes the geometric mean of detection accuracy (DetA) and association accuracy (AssA), reflecting how well a tracker detects objects and maintains their identities over time [6]. HOTA also uses an additional parameter α as a threshold for the amount of IoU needed for being classified as a TP, and calculates the mean of the results with different α.

Detection Accuracy (DetA) Detection Accuracy quantifies how well a tracker detects objects in each frame [6]. It is calculated as the Jaccard index over TP, FN, and FP at a given IoU threshold α:

$$\mathrm{DetA}_\alpha = \frac{|\mathrm{TP}_\alpha|}{|\mathrm{TP}_\alpha| + |\mathrm{FN}_\alpha| + |\mathrm{FP}_\alpha|} \tag{2.16}$$

This score reflects the tracker's ability to correctly detect objects without introducing extra or missing detections.

Association Accuracy (AssA) Association Accuracy evaluates how well the tracker maintains object identities over time [6]. For each matched detection c (a true positive), the alignment of its predicted and ground-truth trajectories is scored using a Jaccard index over TP associations (TPA), FN associations (FNA), and FP associations (FPA). This is visualized in Figure 2.11.

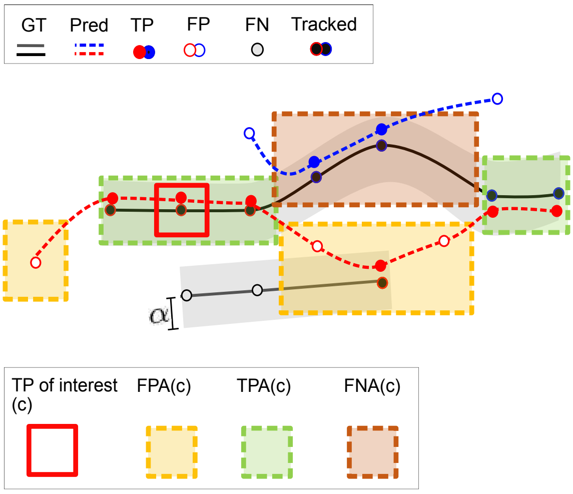


Figure 2.11: Illustration of TPA, FNA, and FPA for a matched detection c. Inspired by [6].

$$\mathrm{AssA}_\alpha = \frac{1}{|\mathrm{TP}_\alpha|} \sum_{c \in \mathrm{TP}_\alpha} \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)| + |\mathrm{FPA}(c)|} \tag{2.17}$$

The HOTA score is defined as:

$$\mathrm{HOTA} = \int_0^1 \mathrm{HOTA}_\alpha \, d\alpha \;\approx\; \frac{1}{19} \sum_{k=1}^{19} \mathrm{HOTA}_{\alpha_k}, \qquad \alpha_k = 0.05\,k,\;\; k = 1, \dots, 19. \tag{2.18}$$

Where

$$\mathrm{HOTA}_\alpha = \sqrt{\mathrm{DetA}_\alpha \cdot \mathrm{AssA}_\alpha} \tag{2.19}$$

where the alpha determines the classification scores of each

$$\text{Detection} = \begin{cases} \text{TP}, & \text{if IoU} > \alpha \\ \text{Not a TP}, & \text{otherwise} \end{cases} \tag{2.20}$$

The final HOTA score is computed by averaging over a range of IoU thresholds to ensure robustness to localization variations.

HOTA addresses the limitations of MOTA by explicitly disentangling detection and association performance. By computing the geometric mean of detection and association accuracies, HOTA ensures that high scores can only be achieved when both aspects perform well. This makes it a more balanced and interpretable metric for evaluating the actual effectiveness of tracking systems according to [6].

3

Methodology

This chapter presents our methodology for MOT in football, addressing challenges such as occlusions, crowded scenes, and limited labeled data from self-filmed footage. We build upon MOTRv2, a transformer-based tracking framework, and incorporate PL to leverage domain-specific unlabeled data effectively. Additionally, a post-process Re-ID module was utilized.

3.1 Datasets and Preprocessing

This section describes the datasets used for training, validation, and evaluation in our MOT pipeline. We combine the publicly available SoccerNet dataset with proprietary match footage from IFK Göteborg to balance data scale with domain relevance. Specific preprocessing steps were applied to ensure consistency across sources, including removing non-human objects such as the ball. We also manually annotated a subset of the IFK dataset to create custom validation and test sets tailored to our deployment context.

3.1.1 External Dataset: SoccerNet

We utilize the SoccerNet tracking dataset for training, as it is publicly available and closely resembles our evaluation domain regarding scene layout and object classes. Although the dataset includes ground truth annotations for players, referees, and the ball, we remove all ball-related labels to better align with the objectives of our tracking task. The ball exhibits distinct motion patterns and visual characteristics, which could divert model capacity away from learning discriminative features for player identification. Moreover, its inclusion may introduce noise during training and reduce the model's effectiveness in human tracking. By filtering out non-human objects, we ensure that the model focuses solely on players and referees, who constitute the relevant targets during training and evaluation.

3.1.2 Custom Validation and Test Sets

A portion of the proprietary IFK Göteborg dataset was manually annotated to enable supervised evaluation. This was necessary due to the absence of built-in player positions or identity labels. A dedicated validation set was created to monitor training performance and guide model selection. In addition, a supervisor selected a separate test set that remained unseen throughout development, ensuring an unbiased final evaluation.

Both the validation and test sets were annotated with bounding boxes and track IDs, allowing for standard MOT metrics such as HOTA, MOTA, and IDSW. The annotated data reflects the same tactical camera view and match conditions as the unlabelled footage, providing a realistic testbed for evaluating tracking models under real-world deployment scenarios. A comparison between the datasets is shown in Table 3.1.

Table 3.1: Comparison of SoccerNet (SN) and IFK Göteborg Football Club (IFK) datasets. "Camera": S (static), P+T+Z (pan, tilt, zoom). "GT": ground truth with bounding boxes and IDs. Image resolution is omitted, as all frames are 1920×1080.

Dataset	**FPS**	**GT**	**#Frames**	**#Games**	**#Seq**	**Camera**
SN Train	25	✓	42K	3	57	P+T+Z
SN Val	25	✓	4.5K	3	6	P+T+Z
IFK Train	25	X	>5M	>50	-	S
IFK Val	25	✓*	1.6K	3	3	S
IFK Test	25	✓*	2K	3	5	S

*Manually annotated during this project. - Noncountable

3.1.3 Data Preprocessing and Augmentation

For the tasks in this work, we perform preprocessing and augmentation on the datasets to increase model robustness and help with generalization. The data augmentation pipeline is crucial in handling variations in the data, such as scaling, flipping, and resizing. This augmentation is applied during the training phase, as detailed below.

Data Augmentation for Training: A combination of random horizontal flipping, random resizing, and random cropping for the training set. These transformations aim to introduce variability in the data, allowing the model to better generalize across different scenarios. The augmentation process is implemented as follows:

- **Random Horizontal Flip:** With a probability of 50%, images are flipped horizontally to simulate a mirrored view of the scene.
- **Random Resize:** A set of scales is used for resizing the image. The image height is resized to one of the given scales (e.g., 608, 640, ..., 992) or a random scale within the specified range.
- **Random Crop:** A fixed random crop of 800x1200 pixels is applied to ensure that the model sees different parts of the image and prevents it from overfitting to any one part of the frame.
- **HSV Augmentation:** The HSV (Hue, Saturation, Value) space is adjusted for color augmentation, making the model more robust to lighting and color variations in the data.

- **Normalization:** The images are normalized to match the typical distribution of pre-trained models. The standard normalization values used are the mean of [0.485, 0.456, 0.406] and the standard deviation of [0.229, 0.224, 0.225].

Data Augmentation for Validation: For the validation set, fewer augmentations are applied to retain data integrity. The primary augmentations include HSV augmentation and image normalization. Specifically, images are resized to a specific scale (800 pixels high) while preserving their aspect ratios.

This augmentation strategy aims to allow the model to handle various input conditions, contributing to more robust object detection and tracking performance in the real-world datasets used in this work.

3.2 Proposal Generation

As detailed in Section 2.3.1.1, YOLO-based detectors are widely used in real-time object detection for their efficient inference and prominent performance. In our tracking pipeline, we employ YOLOv11-l as the object proposal generator. This choice is motivated by YOLOv11-l's faster training and inference speeds compared to more complex detectors and state-of-the-art performance, enabling efficient processing of football footage. Additionally, YOLO's modular design and widespread adoption in the tracking community simplify implementation and integration, making it a practical and reliable solution for our domain-specific tracking needs.

To adapt the model to the specific domain of football analytics, we fine-tuned YOLOv11-l on the SoccerNet tracking dataset, with all ball annotations removed to focus solely on detecting players. This domain-specific fine-tuning improves the detector's robustness in player-heavy scenes and reduces false positives on irrelevant objects.

YOLOv11-l processes each frame independently and outputs bounding boxes with class confidences. These detections serve as the initial proposals for the subsequent tracking module, associating them temporally and assigning consistent identities.

3.3 Tracking with MOTRv2

MOTRv2, a transformer-based model known for its ability to handle long-term object dependencies, track re-initialization, and complex interactions in dynamic scenes, was chosen for the tracking architecture. MOTRv2 was selected due to its robustness in addressing key challenges in football tracking, including occlusions, crowding, and rapid player movement. Unlike traditional trackers that rely on explicit data association heuristics, MOTRv2 learns tracking end-to-end from data, leveraging a combination of self-attention and cross-attention mechanisms. This enables it to maintain object identities over extended periods, even when objects are temporarily lost or occluded, which is common in football scenarios.

3.3.1 Training and Fine-Tuning

MOTRv2 was initially trained on annotated football footage from the SoccerNet tracking dataset, which includes challenging scenarios such as crowding and occlusions. We incorporated pseudo-labeled data generated from IFK Göteborg footage to improve robustness and address domain gaps. Specifically, we experimented with three training configurations: training solely on SoccerNet, training on a combined dataset mixing SoccerNet and pseudo-labeled data, and training on SoccerNet followed by fine-tuning exclusively on pseudo-labeled data. These approaches aimed to enhance the model's adaptation and generalization across different football domains, bridging the visual and contextual differences between SoccerNet and Spiideo footage.

Table 3.2: Comparison of training configurations with different datasets.

Dataset(s)	**Epochs**	**Learning Rate**
SoccerNet only (SN)	61	2×10^{-4}
SoccerNet + PL (Mix)	25	2×10^{-4}
SoccerNet, fine-tuned on PL (SN→PL)	61 + 9	$2 \times 10^{-4} \rightarrow 2 \times 10^{-5}$

Following the configurations summarized in Table 3.2, the baseline MOTRv2 model was trained solely on the SoccerNet dataset for 61 epochs using a learning rate of 2×10^{-4}. This learning rate has previously been shown to work well for this model [2] and, in our testing, provided a good balance between fast convergence and stable training. For combined training, the model was trained on a mixture of SoccerNet and pseudo-labeled data for 25 epochs with the same learning rate. Since the combined dataset roughly contains twice the amount of data, the number of epochs was halved to achieve approximately the same number of gradient steps during training. This setup enabled the model to learn from both datasets simultaneously without increasing overall training time. Finally, to further specialize the model for pseudo-labeled data, fine-tuning was performed by continuing training on pseudo-labeled data alone for an additional nine epochs with a reduced learning rate of 2×10^{-5}, following the initial 61 epochs on SoccerNet. Throughout all training configurations, a lower learning rate was consistently applied to the pretrained backbone to preserve learned features, while a higher learning rate was used for the transformer components to allow more flexible adaptation. Gradual reduction of the learning rate during training aimed to promote stable convergence and improved generalization across different football footage domains.

Training was employed and distributed across four A100 GPUs to accelerate the process and ensure scalability. A custom data loader efficiently handled large datasets, with distributed samplers ensuring proper data partitioning in multi-GPU setups.

Learning rate drops were employed towards the end of training to find the optimum to ensure stable and efficient training.

The training employed the AdamW optimizer [65], chosen for its effectiveness in training transformer-based models.

3.3.2 Evaluation and Validation

Validation was conducted after each epoch to monitor the model's learning progress and detect signs of overfitting. Due to the extended evaluation time on the complete SoccerNet test set, we validated the model on both a subset of the SoccerNet test set and the custom validation set consisting of self-annotated IFK Göteborg data. During training, we tracked both training and validation loss on these datasets to ensure stable convergence. Once training was completed, the model was evaluated on a custom held-out test set using additional tracking-specific metrics. These included HOTA and IDSW, which are critical for assessing identity preservation and overall tracking performance. This post-training evaluation helped verify that the model minimized loss and maintained identity consistency across frames, even under challenging conditions such as occlusions and crowded player formations.

The model weights with the lowest loss on each validation set were saved. These saved weights were then used to evaluate different models to prevent the use of an overfitted model. Training curves can be seen in Figure 4.3.

3.3.3 Adaptations for Football Tracking

While MOTRv2's original architecture is designed for general-purpose tracking, we introduced several tailored modifications to align it with the specific demands of football analytics. Given the high number of simultaneously moving players in full-pitch views, we increased the number of object queries in the decoder to `40` using the `-num_queries` flag. This ensured sufficient capacity to maintain identity continuity across all players, even in dense scenarios such as corner kicks or transitional plays, where standard query budgets might underperform.

To further improve contextual modeling in dense environments, we enabled additional query interaction mechanisms by specifying `-query_interaction_layer QIMv2`, allowing richer query-to-query communication within the decoder. The `-extra_track_attn` flag was also activated to enhance temporal consistency by providing additional attention to historical object tracks. These choices emphasized long-range dependencies and inter-object relationships central to football, where formations and spatial coordination are often more informative than raw appearance.

Regularization techniques such as dropout were employed to improve training stability and reduce overfitting.

3.4 Pseudo-Labeling

The PL pipeline is composed of several steps to enable automatic labeling of player positions and movements. First, a tracker generates an initial estimate of the player tracks throughout a football match. From this initial output, confident tracks are identified and filtered. Each player in these confident tracks is then segmented. The segmentation allows each player and their movement pattern across the field to be extracted and composited onto a clean background. This makes it possible to place

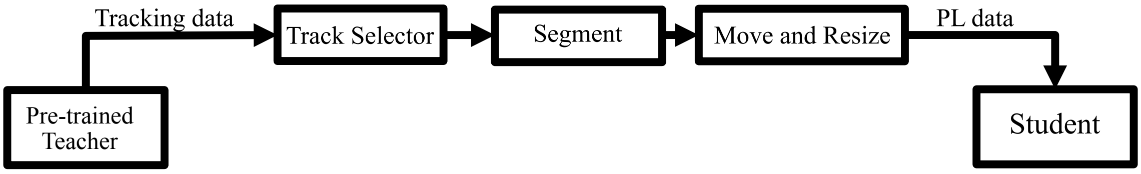


Figure 3.1: Visualization of the pipeline used to generate pseudo-labels from unannotated video data.

players at any position on the field with appropriate resizing. The full pipeline for this is illustrated in the Figure 3.1

3.4.1 Confident Track Selector

When selecting confident tracklets for PL, several criteria are applied to ensure the quality and reliability of the data. First, tracklets must be at least 4 seconds long (R1) to provide sufficient temporal information. To avoid ambiguous detections, any tracklet with overlapping bounding boxes with other players in the same frame is excluded by enforcing an IoU of zero (R2). Additionally, detections must have a confidence score of at least 0.6 (R3) to ensure reliability. The spatial movement of the tracked object is also essential; we measure the difference between the maximum and minimum pixel coordinates of the bounding box center in both x and y directions (R4), requiring a movement greater than 100 pixels to exclude static or nearly stationary tracklets that may indicate tracking errors. Tracklets with bounding boxes shorter than 40 pixels in height are filtered out to remove likely noise or false positives (R5). Finally, tracklets must be continuously tracked without missing frames to guarantee consistency (R6). These rules help select only high-quality tracklets suitable for generating accurate pseudo-labels. These rules are summarized in Table 3.3.

Table 3.3: Rules used to select confident tracklets for PL

Rule	Description	Value/Criterion
R1	Minimum Tracklet Length	$\geq$ 4 seconds
R2	BBox Overlap Tolerance	IoU = 0
R3	Detection Confidence Threshold	≥ 0.6
R4	Tracklet Motion Threshold	$\min(pos_{(x,y)}) - \max(pos_{(x,y)}) > 100$ px
R5	Minimum BBox Height	Box height > 40 px
R6	Consistent Tracking	No missing frames

3.4.2 Segmentation

The segmentation of players leverages the fact that the camera view is stationary. Thus, the players are typically the only rapidly changing elements in the scene. A static background image is first created by computing the median pixel value over 20 seconds. This method assumes that over a 20-second window, players will have moved enough not to dominate any single pixel location, resulting in a background image free of players.

Segmentation is then performed using the bounding boxes from the confident tracklets. Each bounding box's pixel values are subtracted from the background in the HSV color space, and then the pixels with a difference larger than the threshold (ϵ) are represented as a foreground/background in the binary mask.

This binary mask is further processed using morphological operations, dilation, and erosion [66]. These steps help fill in small holes, resulting in cleaner masks. This thresholding-based approach was chosen over machine learning models as it significantly speeds up the process by exploiting the stationary nature of the camera and field. It also provides a reasonably accurate segmentation without requiring additional training data.

3.4.3 Moving and resizing

Moving the tracklet is performed so that the desired frame aligns with the generated meeting point in both position and time. The entire tracklet is shifted by the same amount in time and space, preserving the natural motion direction relative to the player's movement within the frame.

A second-order polynomial was fitted to the box height in both the x- and y-directions to resize and position objects to appear appropriately scaled relative to their location in the image. This was done using linear regression, based on the observed heights of boxes at various positions. Specifically, for each game, the optimal constants (a, b, c) were found for both axes, resulting in:

$$A = \begin{bmatrix} a_x, & a_y, & b_x, & b_y, & c_x, & c_y, \end{bmatrix} \tag{3.1}$$

This defines a position-dependent height estimation function. Given an object with original height h_0 and position (x_0, y_0), resulting in a second-degree polynomial represented by

$$X_n = \begin{bmatrix} x_n^2, & y_n^2, & x_n, & y_n, & 1, & 1 \end{bmatrix} \tag{3.2}$$

The estimated height at that position is:

$$\hat{h}_0 = A \cdot X_0 \tag{3.3}$$

To compute the adjusted height h_1 at a new position (x_1, y_1), the same function is evaluated:

$$\hat{h}_1 = A \cdot X_1 \tag{3.4}$$

The new height is then calculated as:

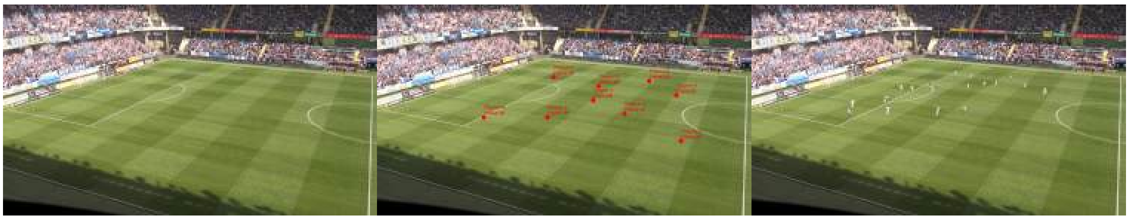
$$h_1 = \frac{\hat{h}_1}{\hat{h}_0} h_0 \tag{3.5}$$

This ensures that the object is resized proportionally to match its spatial context in the image.

3.4.4 Tracklet Generation

The segmented players, background image, and original tracking paths generate new training sequences. These synthetic sequences are created by randomly placing a set of coordinates shown in Figure 3.2 b, assigning a time, and determining which particular players should occlude each other at that location and time. The placement movements to and from these coordinates align with the original movement paths of the players, ensuring realistic motion patterns. At each point of interaction, 1–3 players are assigned to meet.

The player positions are randomly selected but constrained within the field's boundaries. Many such sequences are generated, each containing between 15 and 25 players on the field simultaneously, enabling the possibility of creating an infinite number of sequences.



(a) Background **(b)** Point generation **(c)** Pseudo-labeling

Figure 3.2: The pseudo label generation.

3.5 Re-Identification for ID Switch Correction

To address identity fragmentation and ID switches throughout a football match, we incorporate the Global Tracklet Association (GTA) framework [4] as a post-processing stage in our tracking pipeline. GTA tackles two common challenges in long-term MOT: internal ID inconsistencies within tracklets and the re-association of disjoint tracklet segments caused by occlusions or brief target disappearances.

3.5.1 Tracklet Splitting

The first step in GTA addresses the problem of tracklet fragmentation and identity switches, where detections from different players may have been mistakenly merged into a single tracklet. To correct this, we extract visual feature embeddings for each bounding box using OSNet [61], a re-identification backbone optimized for capturing fine-grained appearance differences. These embeddings are analyzed over time using a modified density-based clustering approach based on DBSCAN [67]. Unlike the original DBSCAN, our version ensures that all frames are assigned to a cluster, avoiding the loss of potentially valuable information due to outliers.

This step enables the system to split sequences where abrupt changes in appearance suggest an identity switch by clustering bounding boxes based on visual similarity within a tracklet, such as during occlusions, close player interactions, or tracking failures.

3.5.2 Tracklet Re-Association

After splitting, the second phase involves reconnecting tracklet fragments likely to correspond to the same player. A symmetric distance matrix $D_{i,j}$ is computed between all tracklet pairs as described in Equation 2.7. Each entry quantifies the average cosine dissimilarity between feature embeddings in tracklets T_i and T_j. However, if two tracklets overlap in time, meaning they contain frames from the same or intersecting time intervals, they are treated as mutually exclusive, since a single player cannot occupy two positions simultaneously. In such cases, the distance between them is set to the maximum value of 1.

To prevent implausible re-associations, we apply spatial filtering based on field geometry. Tracklet pairs whose horizontal or vertical displacements exceed predefined thresholds-scaled by a factor β-are also excluded by setting their distances to 1. These constraints eliminate tracklet pairs that could not feasibly correspond to the same player due to excessive movement.

The filtered distance matrix is then used for clustering. Tracklets are iteratively merged until no pairwise distance falls below a user-defined merging threshold α. This clustering step reconstructs longer, identity-consistent trajectories from shorter fragments, compensating for tracking failures during occlusion or heavy crowding periods.

Integrating this Re-Id correction step allows us to improve the continuity and accuracy of player tracks over extended sequences without requiring additional annotations or real-time supervision.

3.6 Summary of Design Choices for Football MOT

The methodological components described above were selected with the intent to address common challenges in football tracking, such as frequent occlusions, crowded player formations, lighting variability, limited annotations, and domain shifts. By integrating YOLOv11-l for detection and MOTRv2 for query-based tracking, the pipeline is designed to support identity continuity in dynamic scenes. Additionally, techniques like PL, domain-specific augmentation, and tailored data strategies are employed to improve adaptability to self-filmed and real-world football footage. These design choices are intended to promote robustness under varied conditions and facilitate scalable training without full reliance on extensive manual annotation.

3.6.1 Inference Pipeline

After training and fine-tuning, the model is deployed for inference, where it processes video frames to track players across a full football match. The inference pipeline follows these steps:

1. **Data Input and Preprocessing:**
   - Video frames are resized to a fixed resolution, normalized using predefined mean and standard deviation values, and augmented minimally to retain data integrity for evaluation.
2. **Object Proposal Generation:**
   - YOLOv11-l generates object proposals for each frame, detecting player bounding boxes with associated confidence scores.
3. **Tracking with MOTRv2:**
   - MOTRv2 processes the proposals through its decoder, applying attention mechanisms to track players across frames. Queries are maintained to preserve player identities, even when occlusions or rapid movements occur.
4. **Identity Consistency and Re-Identification:**
   - To reduce identity switches, **GTA** is used to refine fragmented tracklets, re-associating them based on appearance features and spatial constraints.

This end-to-end pipeline is designed to enable robust tracking of football players across a wide range of scenarios, aiming to handling the dynamic nature of the sport while maintaining consistent identity assignments. By combining YOLOv11-l's detections with MOTRv2's tracking capabilities and our adaptations, we hope to achieve high accuracy in player tracking in football footage.

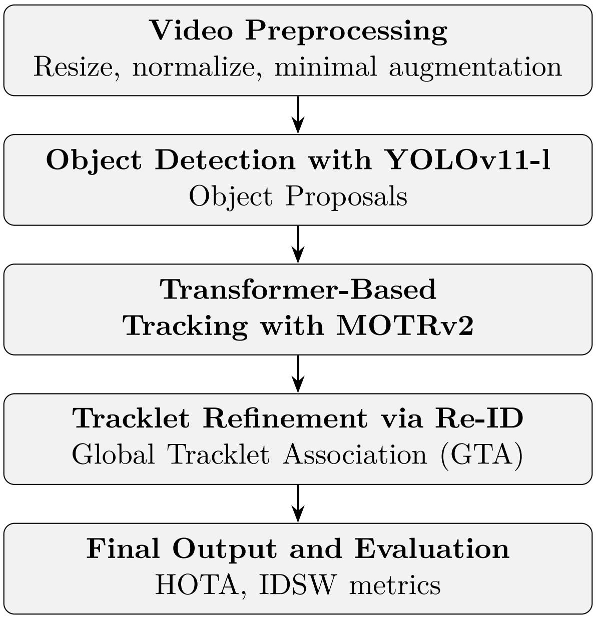

Video Preprocessing
Resize, normalize, minimal augmentation
Object Detection with YOLOv11-l
Object Proposals
Transformer-Based
Tracking with MOTRv2
Tracklet Refinement via Re-ID
Global Tracklet Association (GTA)
Final Output and Evaluation
HOTA, IDSW metrics

Figure 3.3: Overview of the football tracking inference pipeline.

4

Results

This chapter evaluates our MOT method on self-filmed football video data. We use both quantitative and qualitative metrics to assess tracking accuracy, identity preservation, and robustness under challenging conditions. The evaluation includes comparisons with baseline models, ablation studies on key components such as PL and Re-Id, and analysis of training strategies. All experiments are conducted on a custom-annotated dataset derived from real match footage, as described in the previous chapter.

4.1 Fine-tuning YOLO11l

The detector was trained on SoccerNet data, and its performance was evaluated on individual frames and detections within the SoccerNet test set. Training ceased when the validation box loss increased, as shown in Figure 4.1. To prevent overfitting, the weights from epoch 53 were chosen. As Figure 4.1 demonstrates, performance depends on the confidence threshold.

As part of the self-attention mechanism, the idea is to selectively focus on specific bounding boxes from the YOLO model for final tracking. Our primary goal is for the detector to accurately detect all players. Figure 4.2b, the recall plot, demonstrates that a low confidence threshold yields nearly perfect recall. However, as illustrated in the precision plot in Figure 4.2c), a lower threshold increases the number of false positives (FPs). Therefore, a low confidence threshold of 0.3 was chosen for the detections.

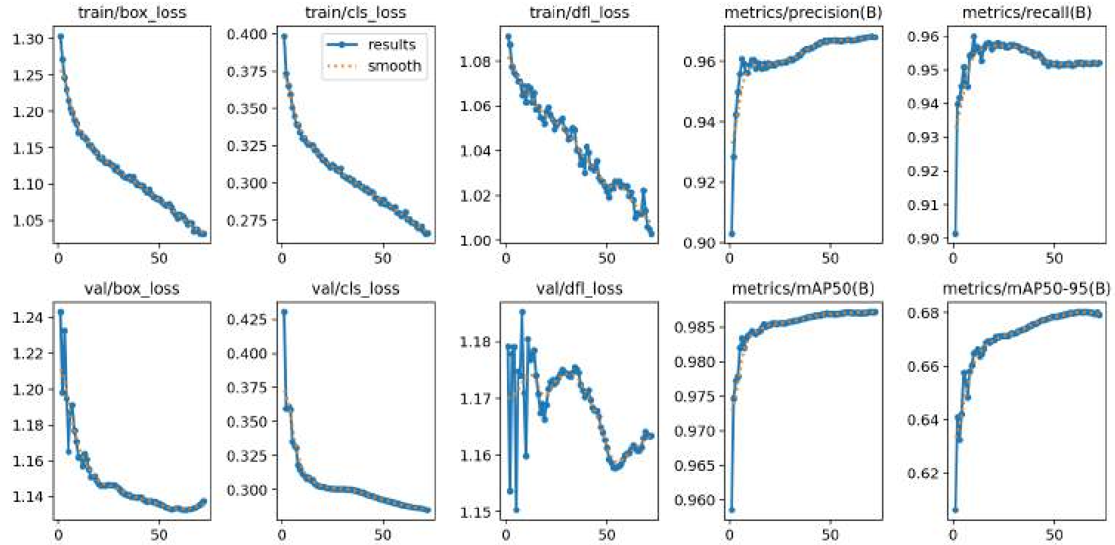
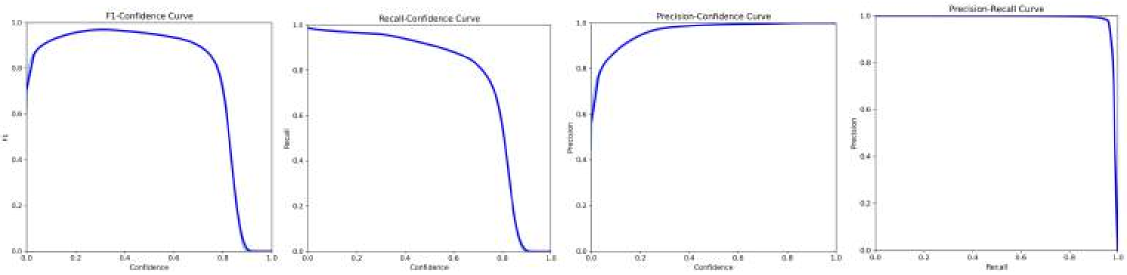


Figure 4.1: Training loss over number of epochs for the detector



(a) F1 **(b)** Recall **(c)** Precision **(d)** Precision-Recall

Figure 4.2: Performance metrics related to the confidence threshold of the detector model

Table 4.1: Performance Metrics of the YOLO11l Model

Model	F1 Score	Confidence Threshold
YOLO11l	0.93	0.34

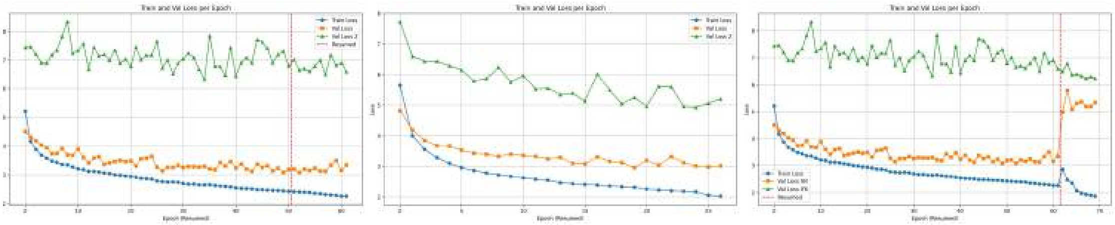
4.2 Effect of Pseudo-Labeling

The impact of incorporating pseudo-labeled data into the training process was assessed by evaluating the performance on four different dataset setups. One uses the Manually annotated SoccerNet data, one uses only the pseudo-labeled data, and then a Mix data set combining pseudo-labels and SoccerNet. Lastly, a model that was first trained on SoccerNet, then fine-tuned on pseudo-labels. These comparisons, detailed in Table 4.2, highlight the potential of PL for improving performance under limited supervision, resulting in an improved score for the Mix data.

Table 4.2: We compare the performance of MOTRv2 across different datasets. To ensure a fair comparison, all models were trained with identical parameters and a similar number of gradient steps, varying only in the training data. These results are obtained before parameter tuning and therefore may not be directly comparable to the final, optimized results presented elsewhere.

Dataset	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑	IDSW↓
SN	68.7	63.0	75.4	67.1	78.0	129
PL	61.4	58.7	64.4	62.6	70.6	301
Mix	**70.1**	**65.7**	**76.8**	**74.0**	**81.6**	**91**
SN->PL	68.2	62.5	74.8	68.1	81.3	378

Training loss per epoch for the different datasets could be spotted in Figure 4.1.



(a) SN

(b) Mix

(c) SN→PL

Figure 4.3: Training and validation loss per epoch when training MOTRv2 on different datasets

4.3 Baseline Comparison

Since the mixed dataset yielded the best performance, it was used for hyperparameter optimization. The optimized model with and without Re-ID was then compared to the state-of-the-art TBD tracker BoT-SORT, serving as a baseline to evaluate the impact of our architectural changes and added components. We also added our Re-ID module to the BoT-SORT tracker to assess the performance of the Re-ID module. Result of this can be seen in Table 4.3.

Table 4.3: Performance comparison between baseline model BoT-SORT and our MOTRv2 with our Re-ID and non-Re-ID.

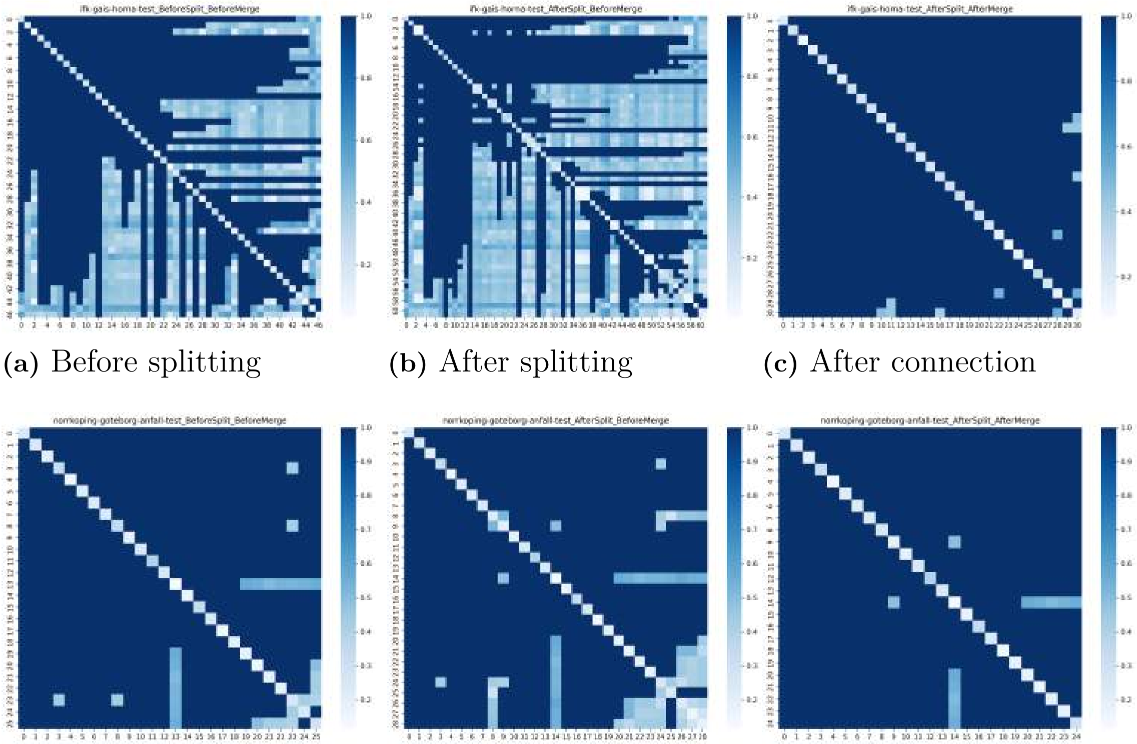
Model	HOTA↑	DetA↑	AssA↑	MOTA↑	IDF1↑	IDSW↓
MOTRv2	72.9	70.9	75.3	82.0	84.2	86
MOTRv2+Re-ID	**74.0**	**70.9**	77.6	**82.0**	**86.4**	77
BotSORT	72.8	68.7	77.4	78.4	83.8	54
BotSORT+Re-ID	73.3	68.7	**78.5**	78.4	85.1	**49**

4.4 Re-Identification Module

To evaluate the contribution of the Re-ID module, we compare model performance with and without this component. The Re-ID module is expected to improve the model's ability to maintain consistent identities over time, especially in occlusions and frequent player interactions.

The Table 4.3 shows that the Re-ID module improves the metric related to the association of the same ID to the same player over time, improving tracking consistency.

Figure 4.4 demonstrates the effectiveness of the Re-ID module in connecting similar tracklets and reducing their overall number. This is evidenced by the fact that after splitting and connecting (Figure 4.4c, Figure 4.4d), high cosine similarity is almost exclusively observed along the diagonal, indicating similarity with itself. This contrasts sharply with the state before splitting (Figure 4.4a, Figure 4.4d), where high cosine similarity exists between multiple distinct tracklets.



(a) Before splitting **(b)** After splitting **(c)** After connection

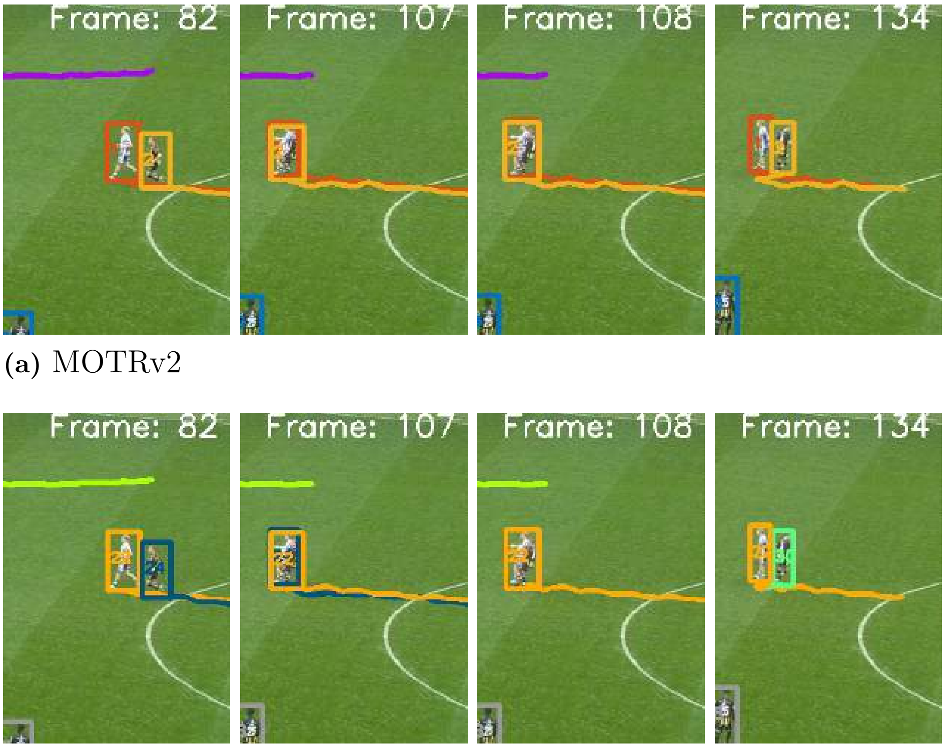
(d) Before splitting **(e)** After splitting **(f)** After connection

Figure 4.4: Cosine similarity matrices of visual features for different tracklets. Subfigures (a, d) show the matrices before splitting, (b, e) after splitting, and (c, f) after connecting. The plots represent two different sequences from the test set: (a-c) for the first sequence and (d-f) for the second.

4.5 Ocular Evaluation

This evaluation compares the BoT-SORT and MOTRv2 (Mix) models without any postprocessing step like Re-ID. We aim to identify instances where each model makes false predictions, misses detections, or incorrectly changes IDs, providing a visual understanding of the quantitative metrics.

Both models demonstrate satisfactory performance when visualizing the tracking results on the test set. However, MOTRv2 generally exhibits greater robustness to occlusions, more effectively preserving player identities when overlaps occur. This is illustrated in Figure 4.5, which highlights a sequence where MOTRv2 maintains correct identity tracking, while BoT-SORT fails to do so.



(a) MOTRv2

(b) BoT-SORT

Figure 4.5: BoT-SORT tracking error: An example sequence where the tracking of MOTRv2 and BoT-SORT differ

However, MOTRv2 introduces other issues that are not as frequently observed in BoT-SORT. One significant problem is that MOTRv2 sometimes assigns a double ID to the same player or rapidly switches between two IDs, as seen in Figure 4.6. This has the potential to create IDSW. Another problem is that if MOTRv2 loses track of a player, it immediately attempts to assign another player, often resulting in a double ID within a single frame or a limited number of frames for that new player before the lost tracklet disappears in the coming frame. This also potentially leads to IDSW, shown in red in Figure 4.7.

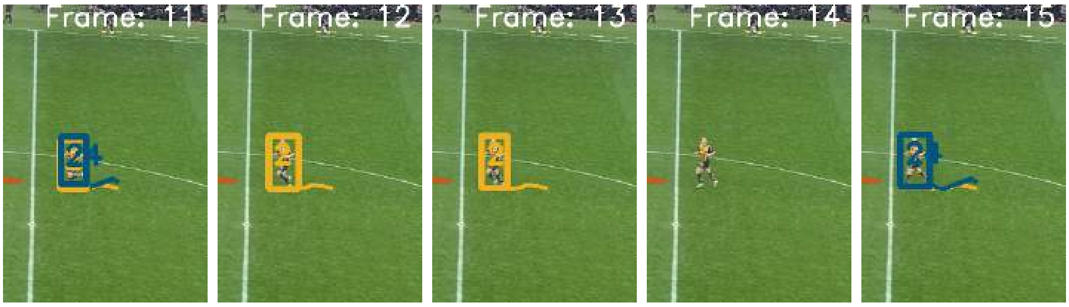


Figure 4.6: MOTRv2 tracking errors: instances of double ID assignments and missing tracklets.

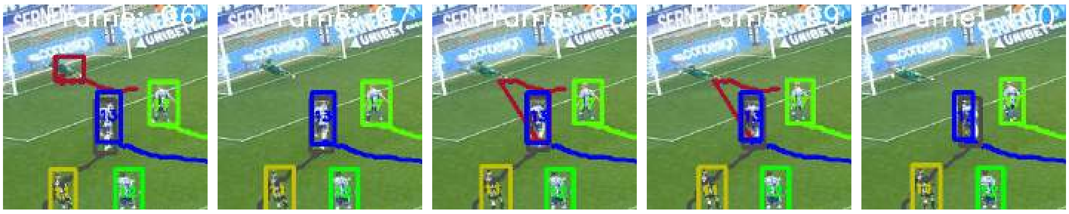


Figure 4.7: MOTRv2 tracking error: Assignment of a lost track ID to an already-tracked player.

5

Discussion

In this chapter, we reflect on our experiments' key findings and insights. We analyze the strengths and limitations of the tracking system, the effects of PL, and the role of Re-ID in enhancing performance. Potential directions for future work are also discussed.

5.1 Tracking

Tracking with TBQ methods like MOTRv2 appears promising and achieves results comparable to state-of-the-art, more model-based TBD approaches. However, due to the larger number of parameters that need to be trained, such methods require more data.

With a larger and more diverse pseudo-labeled dataset, TBQ approaches like MOTRv2 may outperform model-based trackers like Bot-SORT. Unlike MOTRv2, Bot-SORT relies more on fixed model components and offers limited support for end-to-end optimization. This can make it less flexible in adapting to the data, whereas TBQ methods benefit from the ability to optimize the entire tracking pipeline jointly.

5.2 Pseudo labeling

From Table 4.2, it is evident that the best-performing model utilizes a mixture of datasets, highlighting the benefits of combining diverse data sources to improve model performance. However, training solely on pseudo-labeled data is insufficient as it performs worse on almost all metrics in Table 4.2, indicating that the pseudo-labeled dataset still has limitations compared to real SoccerNet data. A potential reason for the performance drop could be that the edges of the segmentation masks used in the pseudo-label generation process are too sharp and not smoothly blended with the background. This may lead to unnatural transitions when a player is repositioned in parts of the field with different lighting conditions, making the synthetic data less realistic. Further research is needed to explore such limitations and improve pseudo-labeled samples' visual quality and consistency. The performance drop in using only the pseudo-label generation process suggests that, given more time, this process could be further refined to better approximate real data. Nevertheless, combining real SoccerNet data with more domain-specific pseudo-labeled data proves highly beneficial, enhancing the model's generalization capabilities.

5.3 Re-Identification

Re-ID has been shown to improve both TBQ and TBD models, though the performance boost diminishes as the models improve. For full game tracking, however, some form of Re-ID seems necessary for consistent long-term monitoring. Due to their focus on appearance features, Re-ID modules benefit from players being represented by more pixels, such as when they are closer to the camera or in higher-resolution images.

5.4 Future Improvements

This section outlines potential avenues for future work, focusing on enhancing tracking performance, improving pseudo-label generation, and optimizing training strategies in football player tracking.

5.4.1 Tracking System Enhancements

A significant challenge in tracking football players, particularly in a large field of view (FoV) with limited image resolution, stems from distant players represented by a very few pixels. This, combined with frequent player occlusion in sports, makes distinguishing individual players difficult even for human observers. To address this, several improvements related to the tracking setup and detector training can be explored:

- **Higher Resolution and Multi-Camera Setups:** Implementing higher-resolution cameras or a dual-camera setup (similar to broadcast styles) would increase the number of pixels representing each player. This enhanced pixel density would make it easier for the model to differentiate players, even when they are far away or occluded. A multi-camera system could further leverage camera calibration and triangulation to associate objects across different views, providing more robust and consistent tracklets.

- **Detector Training with Pseudo-Labeled Data:** Training the object detector directly on pseudo-labeled data, especially data that simulates the challenges of low pixel representation and occlusion, could significantly improve its ability to detect and distinguish players under these challenging conditions.

5.4.2 Pseudo-Label Generation Refinement

Improving the quality and realism of pseudo-labeled data is crucial for robust model training. Future work could focus on:

- **Dynamic Backgrounds in Pseudo-Labeling Videos:** Incorporating dynamic backgrounds into pseudo-label video generation would make the synthetic data more realistic, better mimicking real-world complexities and improving the model's generalization capabilities.

- **Camera Calibration for Pseudo-Labeling Accuracy:** Utilizing camera calibration during pseudo-label generation to ensure players are consistently within a defined grid (e.g., the field boundaries) at all times would lead to more accurate and reliable pseudo-labels, reducing noise and inconsistencies.
- **Class-Specific Pseudo-Labeling:** Integrating classification into the pseudo-label generation process would enable the model to distinguish between players from different teams and identify referees and linesmen. This would create more reliable pseudo-labels for match-like sequences, where specific roles (e.g., one referee, two linesmen) are consistent.

5.4.3 Optimized Training Strategies

Beyond data generation, the training methodology can be refined:

- **Iterative Pseudo-Labeling Training Loop:** Implementing an iterative training loop (e.g., train → generate pseudo-labels → train on new pseudo-labels → generate new pseudo-labels) could continuously improve model performance. For an offline setup, the model could fine-tune on pseudo-labels generated from the test data, then iterate this loop to achieve superior performance on that specific test set.
- **Pseudo-labeling on test set** During our project, we entirely excluded the test data from training to evaluate our model on unseen IFK games. However, as the PL technique can generate domain-specific data from a video stream without labels, as well as our offline approach, it would be permissible to use the test data video feed to create pseudo-labels for the games on which the algorithm is deployed. This approach allows the creation of challenging sequences from these games with the same players, same lightning conditions, producing data similar to the environment where the algorithm will be deployed. This could improve performance by intentionally overfitting to each match the model is applied to and fine-tuning it after the game, thereby enhancing the algorithm's performance on that specific match. However, this approach makes the model less generalizable and requires fine-tuning for each match, resulting in higher computational cost but likely increased performance on each game individually.
- **Runtime Pseudo-Label Generation:** Instead of relying on fixed epochs, exploring the generation of new pseudo-labeled data continuously during training could provide a more adaptive and dynamic training process, allowing the model to learn from evolving data distributions.

5.4.4 Re-Identification Module Optimization

While the Re-ID module has shown promise in connecting similar tracklets and reducing their overall number, further optimization is possible. Future work could focus on:

- **Enhanced Feature Learning for Low-Resolution Instances:** Developing Re-ID features designed to be robust even with low pixel representation per player could extend its benefits to more challenging tracking scenarios.
- **Tighter Integration with Tracking:** Exploring more seamless integration of the Re-ID module directly within the tracking pipeline, rather than as a separate postprocessing step, might lead to more real-time and coherent tracklet management.
- **Team and Number Classification:** A full-pitch field of view (FoV) is essential for robust long-term tracking over entire sequences. Furthermore, a dedicated classification module should be integrated into the Re-ID pipeline, combining OCR-based jersey number recognition with team affiliation classification. These semantic attributes can be embedded into the Re-ID representation to improve identity consistency. Incorporating such information enables more reliable reconnection of fragmented tracklets, particularly in cases where jersey numbers are occluded or temporarily invisible.

5.4.5 Incorporating Temporal Context

In an offline tracking setting, leveraging information from past and future frames to inform the association of objects in the current frame represents a promising direction. By conditioning the query representation on a broader temporal window including multiple frames into the future, the model could better estimate object trajectories, particularly in challenging scenarios such as complete occlusions or abrupt motion changes. Integrating temporal priors, such as where an object has previously been, and plausible predictions of its future positions in multiple future frames. It could improve the robustness and continuity of identity association. Nonetheless, effectively modeling and utilizing this bidirectional temporal context remains an open research challenge.

6

Conclusions

Tracking football players within a large field of view and with limited image resolution poses substantial difficulties. A few pixels represent distant players, and the frequent close proximity of athletes in sports often leads to occlusion. This combination of low pixel count and occlusion makes reliable differentiation between players challenging, even for human observers.

Tracking-by-query models, such as MOTRv2, necessitate significantly more training data to achieve performance levels comparable to tracking-by-detection models like BoT-SORT. As data-driven machine learning approaches, query-based trackers are inherently more sensitive to hyperparameter tuning than model-based trackers, which typically operate with fewer adjustable parameters. Nevertheless, our results underscore the immense potential of deep learning: given sufficiently large and domain-specific datasets, these modern approaches can rival, and even exceed, the capabilities of older, more model-based techniques. This suggests that the ultimate performance limit for deep learning methods in tracking may indeed be higher than for somewhat model-based techniques like BoT-SORT, provided optimal parameter configurations and extensive domain-specific data are available. However, as of now, a postprocessing step is still required to mitigate the increased number of IDSW generated by the data-driven MOTRv2 tracker, and we have not proven that MOTRv2 is on par or does improve such a metric compared to more model-based approaches such as BoT-SORT.

Furthermore, the utility of employing PL for generating domain-specific data has been demonstrated to effectively enhance model performance within that target domain.

Re-ID appears to yield modest performance improvements when applied as a postprocessing step. While its advantages are more pronounced on images featuring higher pixel representation per player, it consistently benefits the more model-based BoT-SORT and the more data-driven MOTRv2 architectures.

Bibliography

[1] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "Soccernet: A scalable dataset for action spotting in soccer videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 1792–179210, IEEE, June 2018.

[2] Y. Zhang, T. Wang, and X. Zhang, "Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22056–22065, 2023.

[3] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11557–11568, 2021.

[4] J. Sun, H.-W. Huang, C.-Y. Yang, Z. Jiang, and J.-N. Hwang, "Gta: Global tracklet association for multi-object tracking in sports," in *Proceedings of the Asian Conference on Computer Vision*, pp. 421–434, 2024.

[5] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[6] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "Hota: A higher order metric for evaluating multi-object tracking," *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[7] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[8] F. Anjou and A. Ekström, "Football analysis in vr-texture estimation with differentiable rendering and diffusion models," Master's thesis, Chalmers University of Technology, 2024.

[9] W. Spearman, A. Basye, G. Dick, R. Hotovy, and P. Pop, "Physics-based modeling of pass probabilities in soccer," in *Proceeding of the 11th MIT Sloan Sports Analytics Conference*, vol. 1, 2017.

[10] W. Spearman, "Beyond expected goals," in *Proceedings of the 12th MIT sloan sports analytics conference*, pp. 1–17, 2018.

[11] S. Llana, P. Madrero, J. Fernández, and F. Barcelona, "The right place at the right time: Advanced off-ball metrics for exploiting an opponent's spatial

weaknesses in soccer," in *Proceedings of the 14th MIT Sloan Sports Analytics Conference*, 2020.

[12] J. Fernandez and L. Bornn, "Wide open spaces: A statistical technique for measuring space creation in professional soccer," in *Sloan sports analytics conference*, vol. 2018, 2018.

[13] U. Brefeld, J. Lasek, and S. Mair, "Probabilistic movement models and zones of control," *Machine Learning*, vol. 108, no. 1, pp. 127–147, 2019.

[14] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, IEEE, 2016.

[15] H. Ganelius and J. Humayun, "Tracking players and ball in football videos," Master's thesis, Chalmers University of Technology, 2024.

[16] Z. Kalafatić, T. Hrkać, and K. Brkić, "Multiple object tracking for football game analysis," in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pp. 936–941, IEEE, 2022.

[17] P. Andrews, N. Borch, and M. Fjeld, "Footyvision: Multi-object tracking, localisation, and augmentation of players and ball in football video," in *Proceedings of the 2024 9th International Conference on Multimedia and Image Processing*, pp. 15–25, 2024.

[18] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, Ieee, 2016.

[19] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[20] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[21] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "Motr: End-to-end multiple-object tracking with transformer," in *European Conference on Computer Vision*, pp. 659–675, Springer, 2022.

[22] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8844–8854, 2022.

[23] A. Deliege, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, and M. Van Droogenbroeck, "Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4508–4519, 2021.

[24] A. Cioppa, S. Giancola, A. Deliege, L. Kang, X. Zhou, Z. Cheng, B. Ghanem, and M. 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*, pp. 3491–3502, 2022.

[25] Spiideo, "Sports video camera systems." `https://www.spiideo.com/sports-video-camera-systems/`. Accessed: 2025-05-26.

[26] A. Maglo, A. Orcesi, J. Denize, and Q. C. Pham, "Individual locating of soccer players from a single moving view," *Sensors*, vol. 23, p. 7938, 2023.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[29] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[30] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[33] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

[34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[37] S. Kang, Z. Hu, L. Liu, K. Zhang, and Z. Cao, “Object detection yolo algorithms and their industrial applications: Overview and comparative analysis,” *Electronics*, vol. 14, no. 6, p. 1104, 2025.

[38] N. M. Krishna, R. Y. Reddy, M. S. C. Reddy, K. P. Madhav, and G. Sudham, “Object detection and tracking using yolo,” in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1–7, IEEE, 2021.

[39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[41] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9759–9768, 2020.

[42] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[43] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, *et al.*, “A survey on vision transformer,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[44] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[45] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, “Bytetrack: Multi-object tracking by associating every detection box,” in *European conference on computer vision*, pp. 1–21, Springer, 2022.

[46] E. Yu, T. Wang, Z. Li, Y. Zhang, X. Zhang, and W. Tao, “Motrv3: Release-fetch supervision for end-to-end multi-object tracking,” *arXiv preprint arXiv:2305.14298*, 2023.

[47] Z. Ge, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.

[48] Y. Cui, C. Zeng, X. Zhao, Y. Yang, G. Wu, and L. Wang, “Sportsmot: A large multi-object tracking dataset in multiple sports scenes,” *arXiv preprint arXiv:2304.05170*, 2023.

[49] T. Zhang, B. Ghanem, and N. Ahuja, “Robust multi-object tracking via cross-domain contextual information for sports video analysis,” in *2012 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 985–988, IEEE, 2012.

[50] A. Scott, I. Uchida, N. Ding, R. Umemoto, R. Bunker, R. Kobayashi, T. Koyama, M. Onishi, Y. Kameda, and K. Fujii, “Teamtrack: A dataset for multi-sport multi-object tracking in full-pitch videos,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3357–3366, June 2024.

[51] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, “Deep learning in video multi-object tracking: A survey,” *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[52] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831.

[53] P. Kage, J. C. Rothenberger, P. Andreadis, and D. I. Diochnos, “A review of pseudo-labeling for computer vision,” *arXiv preprint arXiv:2408.07221*, 2024.

[54] D.-H. Lee *et al.*, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Workshop on challenges in representation learning, ICML*, vol. 3, p. 896, Atlanta, 2013.

[55] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[56] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *Advances in neural information processing systems*, vol. 32, 2019.

[57] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[58] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: a regularization method for supervised and semi-supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[59] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-labeling and confirmation bias in deep semi-supervised learning,” in *2020 International joint conference on neural networks (IJCNN)*, pp. 1–8, IEEE, 2020.

[60] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, PMLR, 13–18 Jul 2020.

[61] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3702–3712, 2019.

[62] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[63] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

[64] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[65] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[66] O. D. Team, "Eroding and dilating." `https://docs.opencv.org/4.x/db/df6/tutorial_erosion_dilatation.html`, 2025. Accessed: 2025-06-10.

[67] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, pp. 226–231, 1996.

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY