Towards Automatic Sports Analytics: Team Affiliation, Jersey Number Recognition and Player Tracking.

Mariya Koshkina

A dissertation submitted to the Faculty of Graduate Studies

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Graduate Program in Electrical Enginnering and Computer Science

Lassonde School of Engineering

York University

Toronto, Ontario

September 2025

Abstract

Automatic sports video understanding can enhance athlete performance analysis, coaching strategies, and the viewing experience. A core challenge is the reliable identification and tracking of players in team sports, where athletes look visually similar, often occlude each other, and jersey numbers are visible only intermittently. This dissertation addresses these challenges through three interconnected tasks: team affiliation classification, jersey number recognition, and long-term multi-object tracking.

We first introduce a self-supervised method for team affiliation classification that uses contrastive learning to cluster players into teams without labeled data. This approach generalizes to unseen uniforms and games, reduces the burn-in time compared to color-based methods, and enables downstream applications such as team-conditioned heatmaps. Second, we propose a jersey number recognition pipeline that leverages advances in scene text recognition. By combining legibility filtering, pose-based torso localization, and sequence-level aggregation, the pipeline achieves strong results on a novel hockey dataset and the SoccerNet benchmark, and generalizes across different sports and camera geometries. Third, we present SportsSUSHI, a graph-based tracking framework that integrates domain-specific identity cues—team labels and jersey numbers—into the association process. This improves long-term tracking performance under frequent occlusions and moving cameras.

To support this work, we introduce a new university hockey dataset with annotations for team affiliation, jersey numbers, and tracking. Together with evaluations on SoccerNet and other benchmarks, our results demonstrate robustness within hockey and strong cross-domain generalization.

In summary, this dissertation contributes three key capabilities—unsupervised team classification, transferable jersey number recognition, and identity-aware tracking—that form a unified framework for sports video analysis. These methods provide accurate and generalizable tools

for understanding player behavior across diverse sports, laying a foundation for future advances in analytics, coaching, and media applications.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Sports video understanding and automatic statistics are valuable applications of computer vision. They can help coaches in training, enable efficient game analysis, and enrich the sports viewing experience. Identifying and keeping track of players for the duration of the game is one of the most fundamental tasks in automatic sports video understanding. It is vital for monitoring player movements and interactions throughout the game. It provides a dynamic view of the game, offering insights into player positioning, movement patterns, and physical engagements. Player tracking and player identification are crucial for athlete performance analysis. By understanding player movements, coaches can make informed decisions about player conditioning, substitutions, and tactical adjustments.

Traditionally, people tracking systems use appearance features to maintain the identity of the tracked person. In team sports, players on the same team have a very similar appearance. Players can be uniquely identified by their team affiliation and jersey number. Building systems that can accurately recognize these two attributes is valuable in itself but can also be used to facilitate player tracking.

In this work, we explore three main questions. Given a video of the game:

- How can a player's team affiliation be identified?
- How can a player's jersey number be recognized?
- How can a player be tracked for a long duration?

These tasks are difficult due to the dynamic nature of team sports. Motion blur, camera movement, change of illumination, and frequent occlusions make both recognition and tracking

challenging. Jersey numbers, in particular, are only visible on a fraction of the frames due to the player's position relative to the camera, motion blur, occlusions and fabric deformations. Player tracking is difficult not only because of the similarity in appearance of the players but also due to players' tendency to shift positions quickly and occlude each other for extended periods. This makes maintaining tracks for each player throughout the video clip extremely difficult.

To make team affiliation and jersey recognition methods applicable to real-life use cases, methods need to be able to generalize well to unseen games, with different teams and jersey numbers that were not present at training time. Ideally, the methods will also generalize well from one sport to another to avoid costly data annotation and network re-training. In our work, we propose an unsupervised team affiliation classification method that does not require labelled data to adapt to new previously unseen teams. We also propose a robust jersey recognition pipeline that generalizes well to unseen jersey numbers and performs well when applied to a different sport. Further, we aim to utilize both of these methods to improve the long-term tracking of players under challenging settings such as moving cameras and frequent occlusions.

Multiple-object tracking is a booming area of computer vision that concentrates mainly on pedestrian tracking due to the existence of large pedestrian-tracking datasets such as the MOT Benchmark [3]. Similarly, a large number of methods have been proposed for pedestrian re-identification due to the existence of datasets such as [105, 92]. Development of player tracking methods, jersey number recognition and team ID classification has lagged due to the lack of large public datasets for these tasks. To enable our work and to facilitate further advancements in the area, we propose a hockey dataset that is comprised of university hockey videos recorded with a static camera and capturing the whole rink. We annotate the dataset for 3 tasks: team affiliation, jersey number recognition, and player tracking. Detailed information on each part of the dataset is provided in the chapters corresponding to the tasks.

In the last couple of years, two new large-scale datasets for sports have been introduced: SoccerNet [33] and SportsMOT [25]. The introduction of these datasets enables comparison between proposed methods. In addition to staging experiments on our data, we compare the performance of our jersey number recognition and player tracking methods on publicly available SoccerNet [33] dataset.

Our contributions are:

- High-performing unsupervised method for team affiliation classification that generalizes

well to new previously unseen teams.

- Jersey number recognition pipeline based on scene text recognition that generalizes well to previously unseen jersey numbers as well as to new sports.
- Player tracking system that incorporates player ID (team affiliation and jersey number) for long-term player tracking.
- Novel hockey dataset with annotations for team affiliation, jersey number recognition and player tracking.

In the following three chapters, we address each of these three tasks: team affiliation, jersey number recognition and player tracking. We discuss previous work, our proposed methodology, datasets and experimental results.

Chapter 2

Team Affiliation

2.1 Introduction

Team membership classification (i.e. labelling each person on a playing surface as a member of team A, team B or a referee) is a critical task in sports video analytics: most inferences and statistics depend upon knowing which players are on each team, including attempts on goal, offsides, and player configurations. Accurate team affiliation labels can also improve player tracking. The problem can be challenging due to the extreme variations in player pose, occlusions, motion blur and uneven illumination.

Prior work (e.g., [61, 47]) has framed the problem as a supervised learning task in which labelled data (e.g., bounding boxes with team identifiers) are used to learn a classifier. Early supervised methods employed hand-crafted colour-based features [62, 60], while more recent approaches train convolutional neural networks (CNNs) on labelled datasets to perform player segmentation [47] and classification [61].

Unfortunately, the supervised player classification approach [61] has limited application, since it requires fine-tuning on every new game for optimal classifier performance. The team segmentation approach [47] has been found to generalize better but does not provide player instance segmentation and requires expensive pixel-wise annotation to train the system. For all of these reasons, an unsupervised approach is preferred.

To date, unsupervised approaches [71, 48, 26, 14, 87] rely solely on colour-based features such as colour histograms. While these are simple and lightweight, typically many frames are needed from each new game in order to learn the colour distributions, and these methods fail

when the two teams are wearing similar colours.

Our goal is to understand whether a more powerful representation, which may include both colour and configural information, can be learned in a fully unsupervised manner, and whether such a representation can reduce the number of frames needed for training and improve generalization to novel teams, jerseys, lighting and camera parameters.

To achieve this, we employ unsupervised contrastive learning to train a CNN to cluster players into two teams. We demonstrate our system's performance on a new hockey dataset and compare it to previously proposed unsupervised team affiliation learning approaches. Figure 2.1 demonstrates overall system design.

Our main contributions are:

1. We introduce what is, to our knowledge, the first unsupervised deep learning approach for team classification. This novel contrastive learning approach allows us to generalize to novel games, teams and jerseys without labelled data.

2. We introduce a new annotated hockey dataset that can be used to evaluate player detection and team classification algorithms.

3. We show that our novel unsupervised algorithm outperforms prior unsupervised approaches by a large margin, especially when only a small number of frames are available for unsupervised learning before team assignments must be made. This limits the burn-in time for real-time streaming applications and allows the system to adapt quickly to changes in lighting or camera parameters.

4. We show how our system for team classification can be used to produce accurate team-conditioned heat maps of player positioning, useful for coaching and strategic analysis.

2.2 Related Work

2.2.1 Player Classification

Automatic labelling of players according to team is critical for sport video understanding, including player tracking [62, 60, 87, 10], player configuration analysis, activity recognition [14] and detection of game events [26].

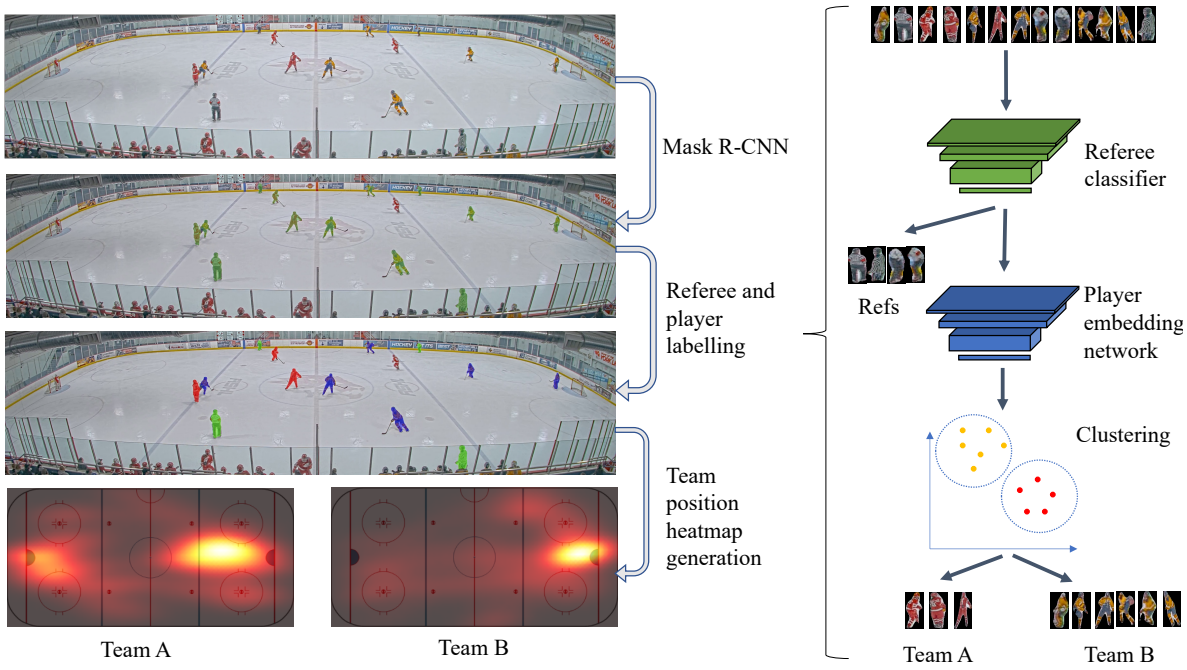

Figure 2.1: Overview of the proposed system. Mask R-CNN is first used to detect and segment each person on the playing surface. A pre-trained CNN is then used to classify referees, while the remaining players are passed to our embedding network for clustering into teams. This allows the production of heat maps showing the distribution of the two teams over the playing surface.

Early work relied on colour histograms [71, 10, 48, 26, 62, 60, 14] and 'bag of words' representations of colour features [87]. These approaches are lightweight, however, the exclusive reliance on colour features makes them more sensitive to illumination changes and could lead to lower performance when teams are wearing similar colours.

In recent years, supervised deep learning-based methods for player detection and player labelling have been proposed [61, 47]. These methods perform well but require labelled data for training. In [47], a CNN is trained to segment players and generate team pixel-wise descriptors, where pixels of players from the same team have descriptors that are close in embedding space. Pixels are then clustered to identify the players on the two teams. This method requires pixel-level team labelling to train the network and per-image pixel-level clustering at the inference stage. Moreover, it does not provide instance-level segmentation so would not be suitable for use in player location heatmap generation.

Lu et al. [61] also take a supervised approach, employing a cascaded CNN to learn team membership classification (team A, team B and others) from labelled data. This method has good results but does not generalize well and thus requires fine-tuning on labelled samples from a new game in order to be used for that game.

Clearly, both simple colour-based unsupervised approaches and more sophisticated CNN-based supervised approaches have limitations. Here we explore whether modern deep unsupervised learning methods can be used to overcome these limitations.

2.2.2 Contrastive Learning and Deep Clustering

Contrastive learning [36] is a self-supervised representation learning approach that aims to map similar objects to be close in embedding space and dissimilar objects further apart and has been shown to produce excellent results on a number of tasks [37, 21]. In our work we use a simple CNN trained with triplet loss [42] to learn a feature space that best separates players into two teams.

Recent work in contrastive learning [37, 21] shows excellent results in unsupervised representation learning on large datasets such as ImageNet[79] or COCO[57]. These methods are based on noise contrastive estimation and involve using an anchor (typically an augmented version of an original image), one positive (another augmented version of the same image) and a large number of negatives, randomly picked from the training set. This setup works well for a dataset

with a large number of categories, where randomly picked images are unlikely to contain many positives. In our setting, however, although we have a large number of images we have a relatively small number of categories (unique jersey designs). More precisely, ImageNet contains 1000 categories and our training dataset has 10. As a consequence, in our setting using random images as negatives results in a 10% of false negatives. This adversely affects training. For this reason, a simple triplet loss works much better in our setting.

Our work is inspired in part by deep clustering approaches [97, 99, 99] in which CNNs are used to jointly learn feature representations and cluster centres in an unsupervised fashion. In our approach, we use pseudo-labels from an initial k-means clustering as a supervision signal to train our contrastive learning CNN. The main divergence from prior methods is that we are only interested in learning feature space that will lead to good data separation - cluster centres can be quite different in each new game. Once trained, the network is only used to extract features from player images.

2.3 Method

2.3.1 Overview

Our general goal is to develop automatic sports analysis tools that provide valuable visualizations, statistics and analyses for coaches and players. Our current work is focused on hockey, but can easily be adapted to other team sports such as soccer, basketball and football. We design and evaluate a system that automatically detects players, classifies them into teams and returns a heatmap of the distribution of players for each of the two teams.

We employ video from a stationary 4K camera that captures the whole playing surface, and use the Mask R-CNN network [38]) to detect and segment all people on the ice, including the players from the two teams and the referees. Since the referee uniform is consistent across games, we first train a CNN to perform referee classification based on labelled data (referee, non-referee).

In order to classify players we employ an embedding CNN trained with triplet loss to extract a learned feature vector for each player image. We then use k-means to estimate cluster centres for the two teams from one or more initial frames of the video. On all subsequent frames, we assign each player to a team based on the closest cluster centre in the feature space. Using

a learned homography, we geo-locate each detected player on the ice surface and use kernel density estimation (KDE) to construct a heatmap representing the distribution of players across the playing surface for each team. Figure 2.1 shows the pipeline for our system.

2.3.2 Dataset

We introduce a new labelled hockey video dataset. Despite the variety of available sports video datasets [1, 33, 25], to the best of our knowledge our new hockey dataset will be the only publicly available sport video dataset that contains team affiliation labels.

The dataset is drawn from 15 different hockey games captured over two seasons. Seven of the games (season 1) are captured with a wide-field stationary 4K (3840 $\times$ 2160 pixel) 30 fps camera that captures nearly the whole rink. In order to better capture the whole rink, season 2 games are captured by two 4K cameras with 75-degree horizontal displacement, together capturing the whole rink with modest overlap. We defined a virtual camera with intrinsic parameters matching the two real cameras and extrinsic parameters equal to the mean of the two real cameras. Each of the two camera images was rectified to the virtual camera through a homography with the ice surface. The two virtual images were then smoothly blended. The resulting season 2 videos have a resolution of 5930 $\times$ 1080 pixels.

We manually close-cropped the videos to the 3840$\times$900 (season 1 games) and 5680$\times$904 (season 2 games) rectangle bounding the rink (Fig. 2.1). From each game, we randomly extracted a video clip of roughly 850 frames (28 sec). Each game contains a unique combination of player uniforms, and since play is active in each clip there is considerable variation in player pose, motion blur and occlusions between players. Players were automatically detected using Mask R-CNN (see below). To eliminate coaches and bench players, we applied a heuristic to exclude detections close to the bottom of the frame that had bounding box height less than twice the width.

For evaluation only, we manually annotated every 10th frame of each game clip, thus obtaining between 80-90 labelled frames per game. Annotations consist of:

1. Mask R-CNN detections
    - Class label (Team A, Team B, Referee)
2. Manual detections (including players not detected by Mask R-CNN)

- Class label (Team A, Team B)
- Estimated image projections of points of contact with the playing surface (skates on the ice)

To label the Mask R-CNN detections, extracted player images with segmentation masks applied were manually inspected one by one. Only the images that could be identified by visual inspection to belong to a player (Team A, Team B) or referee were labelled, the rest were marked as false positives. If there were multiple players within one extracted image the player with the most pixels in the mask was selected.

We use the Mask-RCNN labels of detected players to evaluate the accuracy of team classification algorithms and the manual detection annotations to evaluate the accuracy of our team positioning heatmaps.

The 15-game dataset was divided into training, validation and test sets with a 9-2-4 split. Both training and test set contains a mix of season 1 and season 2 games. One limitation of the dataset is that even though each game has a unique combination of teams playing, some teams appear multiple times through the dataset. We have ensured that the test set includes a game with previously unseen teams.

2.3.3 Player Detection and Segmentation

We employ Mask R-CNN [38] trained on MS COCO [57] to detect and segment all people on the playing surface. To adapt to the different resolution, aspect ratio and expected size of people in our video relative to MS COCO, we partitioned each frame into left and right images with a 40-pixel central overlap, running Mask R-CNN on each individually before merging results. Bounding boxes detected in the left image that overlap boxes detected in the right image by 45% or more are merged by selecting the larger of the two boxes. We define the estimated image location of each player as the mid-point of the lower boundary of the R-CNN bounding box.

2.3.4 Referee Classifier

Since referee uniforms are consistent across games, a supervised approach is appropriate. We use the referee/non-referee labels from our Mask R-CNN detections to train and evaluate a simple CNN classifier. This is the only way that labelled data is used in our system, aside

from evaluation. Our CNN classifier takes as input R-CNN detection images with segmentation masks applied and classifies them as referee or non-referee. We employ a small CNN with 3 convolutional layers (16, 32, and 64 output channels) and 3x3 kernels followed by 2 fully connected layers. We train the network with a binary cross-entropy loss function, employing the Adam optimizer.

2.3.5 Unsupervised Team Assignment: Feature Learning and Clustering

An ideal team labelling algorithm will be unsupervised, generalizing to new games without needing any labelled data, and will require minimal frames (burn-in time) from the beginning of the game to determine accurate labels for each player on the team.

Previous unsupervised approaches used colour features such as histograms and bag-of-colours. These approaches can be effective but since they do not consider spatial features, performance may suffer when teams are wearing jerseys with similar colour profiles, or when illumination variations render colour features unreliable. Here we explore whether an embedding CNN trained by contrastive learning can produce a more powerful representation that, by incorporating both colour and spatial features, can learn a reliable feature representation from fewer frames, and thus have a shorter burn-in time.

We employ a CNN with 3 convolutional layers (16, 32, and 64 channels) and 3x3 kernels, each followed by a pooling layer, and two fully connected layers. The last layer returns a feature vector of length 1024. We train our network using the Adam optimizer on a training set of games using a triplet loss [42]. Input is a triplet of extracted images with the R-CNN mask applied: an anchor image, a positive image and a negative image. The positive image is an image of a player believed to be from the same team as the anchor image, while the negative image is a player believed to be on the other team. The triplet loss function, when back-propagated, drives the network to decrease the distance in the embedding space between the anchor and positive images while increasing the distance between the anchor and negative images. In order to ensure that the learned representation does not exclusively rely on colour, we randomly convert 50% of training triplets to grayscale.

Unsupervised training of the embedding network requires a method for estimating whether two input images have the same or different labels. We seed this process with a simple colour-based distance measure, representing each image as a normalized RGB histogram with 8 bins

per colour channel and then using k-means to cluster players into two teams.

To form the triplets, we first rank the player images $\mathbf{x}_i$ in terms of their team assignment confidence scores p_{ij}, using a standard 'soft k-means' measure:

$$p_{i1} = \frac{\|\mathbf{x}_i - \mathbf{c}_2\|}{\|\mathbf{x}_i - \mathbf{c}_1\| + \|\mathbf{x}_i - \mathbf{c}_2\|} \tag{2.1}$$

$$p_{i2} = \frac{\|\mathbf{x}_i - \mathbf{c}_1\|}{\|\mathbf{x}_i - \mathbf{c}_1\| + \|\mathbf{x}_i - \mathbf{c}_2\|} \tag{2.2}$$

where $\mathbf{c}_j$ is the centre of cluster j and p_{ij} is the confidence with which image i is assigned to cluster j.

We consider only high-confidence samples ($p_{ij} > 0.9$)) for training to limit the label noise. We then randomly form triplets by sampling the anchor and positive images from one cluster (anchor and positive example) and the negative image from the other.

As the training proceeds we regenerate these pseudo-labels and training triplets, but replacing the histogram representation with the evolving embedded representation learned by the network. We train until convergence (no improvement on the validation data for 3 epochs) or a maximum of 30 epochs on the initial colour histogram pseudo-labels and then generate new pseudo-labels from the evolving embedded representation every 10 epochs (or until convergence). We find that the proportion of high-confidence samples grows over time, indicating that the network is learning a representation that improves data separation. Figure 2.2 illustrates this training process.

Once unsupervised training of the embedding network on the training set is complete, we apply the network to novel games with unseen teams and uniforms. We use the first n_{burn} frames of the unseen game as input to k-means to determine the two cluster centres for this new game in the pre-learned embedding space. Once the cluster centres are identified, we associate detected players in subsequent frames with the nearest cluster centre, and evaluate on annotated frames. Figure 2.3 illustrates our use of data for training and evaluation purposes.

2.3.6 Team Positioning Heatmaps

One of the many useful applications of player detection and labelling is the generation of team positioning heatmaps that can help coaches and players understand how their players and the players on the opposing team tend to be distributed throughout a game or portion of the game.

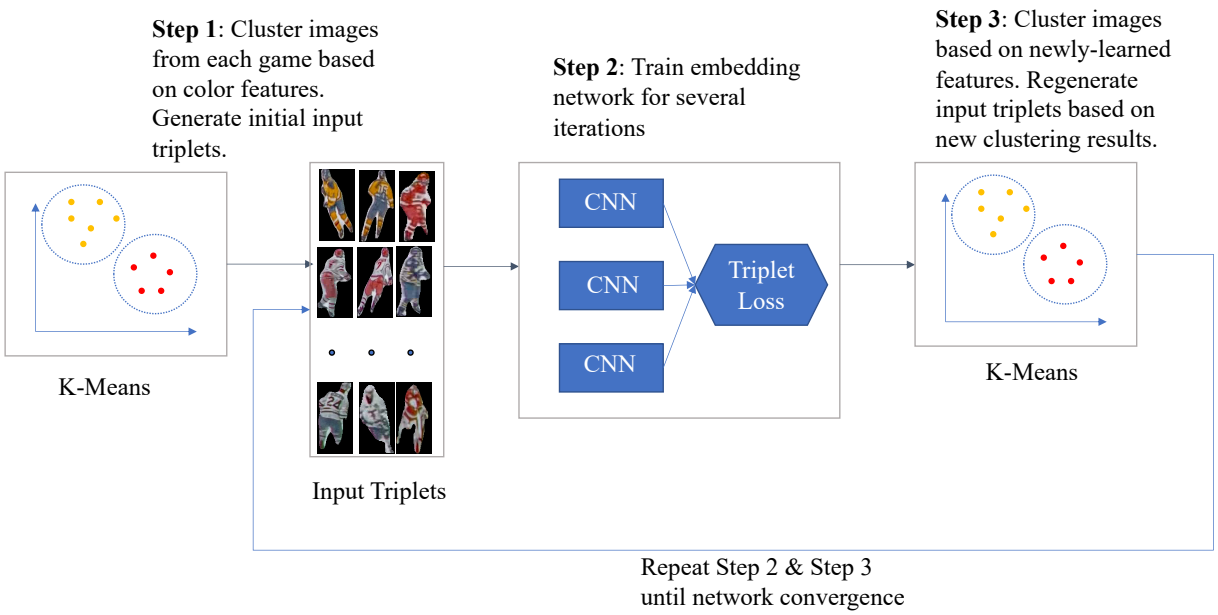

Figure 2.2: Self-supervised training of embedding network.

To generate these heatmaps, we first map player detections from image coordinates to the canonical playing surface. This is done using a homography transformation, which is a 2D projective mapping that relates points on a plane in the real world (e.g., the ice surface) to their projections in the image plane. In essence, a homography captures how the flat surface of the rink appears under perspective distortion in the video. Once the homography is known, any point detected in the image (such as the bottom midpoint of a player's bounding box) can be consistently transferred to its corresponding location on a top-down model of the rink.

The homography was computed by identifying 19 corresponding pairs of reference points between a video frame and a template model of the ice rink. These points included easily identifiable landmarks such as faceoff circles, goal creases, and rink corners. Given these correspondences, we estimated the transformation parameters using the standard least-squares reprojection method [65]. In this approach, the homography matrix H is chosen to minimize the reprojection error between the transformed image points and their known ground-truth positions on the rink template. This process ensures that, on average, points on the playing surface are mapped as accurately as possible to their true rink locations.

After calculating the homography matrix, we used it to transfer all detected player positions into rink coordinates. Combining these positions with the team affiliations inferred by our

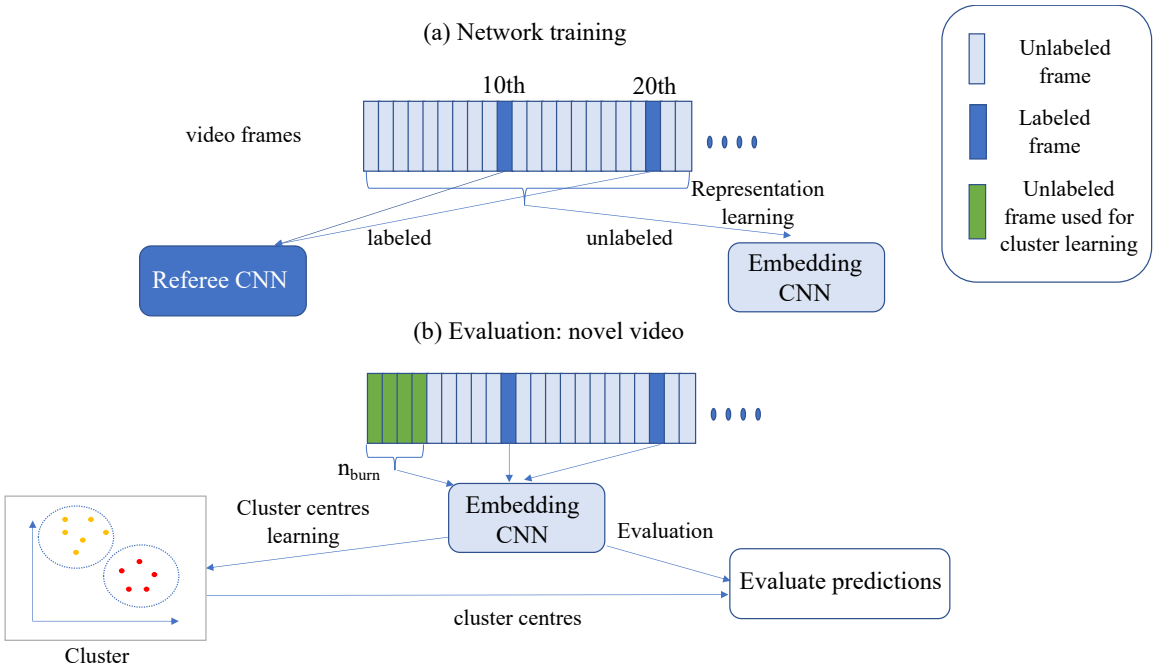

Figure 2.3: Data usage for training and evaluation. a) Labelled frames are used to train the referee classifier, but the embedding network representation is learned in an unsupervised fashion, without reference to labels. b) To perform inference on a novel video, we use the first n_{burn} frames to find cluster centres in the learned embedded representation. Labelled frames are used only for evaluation.

unsupervised contrastive learning algorithm over multiple frames, we then computed rectified player density maps (players per square metre per frame). These maps were generated by smoothing the projected positions using Gaussian kernel density estimation (KDE) [78, 73], which provides a continuous estimate of spatial occupancy. Figure 2.6 shows examples of these automatically generated maps, which highlight dominant player formations and spatial usage patterns.

2.4 Evaluation

2.4.1 Implementation Details

Our system is implemented in Python 3 with Pytorch and Sklearn. We use the publicly-available Mask R-CNN network and weights [7] with a confidence threshold of 0.6. Both referee and embedding networks take as input player images with segmentation masks applied, resized to 62×128 pixels, roughly the average size of a player image. To reduce the impact of illumination variations we applied an affine transform $I'_i(x, y) = aI(x, y) + b$ to the intensities I_i of all three channels $i \in \{R, G, B\}$ such that $\min_{x,y,i} I'_i = 0$ and $\max_{x,y,i} I'_i = 255$.

K-means computation of cluster centres entails 10 random initializations: The solution that minimizes the mean squared deviation from cluster centres is selected.

The code and data are available at `https://github.com/mkoshkina/teamId`.

2.4.2 Comparison with Other Unsupervised Approaches

We compare the performance of our unsupervised team affiliation algorithm against the two main previously proposed unsupervised team labelling approaches: colour histograms [71, 48, 26, 62, 60, 14] and bag-of-words representations of colour features [87]. Since the code and datasets for these previous approaches are not available, we performed a hyperparameter search using k-fold cross-validation to determine the optimal parameters and used k-means clustering to determine cluster centres. These optimal parameters were the number of bins per channel for the histogram algorithm and the number of words for the bag of colours algorithm. In addition, we evaluated whether to use the entire segmented player or just the upper half, since the lower half of the uniform is fairly consistent across teams, and also experimented with multiple colour spaces (see below).

We also experiment with replacing features learned by our contrastive learning network with features learned with convolutional autoencoder (see Section 2.4.2).

Comparison with previously used supervised approaches [61, 47] is not feasible as the code and datasets are not available.

Colour Histogram Algorithm

Our colour histogram method simply histograms the colours within the segmented player, normalizing them by the number of pixels. We experimented with RGB, LAB and HSV colour spaces, and also tried eliminating the luma or value channel (i.e., two-dimensional AB and HV spaces) to reduce sensitivity to illumination, but found optimal performance with RGB coding.

Cluster centres are then found using k-means, using Euclidean distance in the colour histogram space. The single hyperparameter is the number n of bins per channel: k-fold cross-validation revealed that $n = 8$ produces the best results for our dataset. We also found that performance was slightly better if only the upper half of the segmented player was considered as the player jerseys are most distinct between teams.

Bag-of-Colours Algorithm

In our bag-of-colours method, we employ the expectation maximization algorithm to fit a Gaussian mixture model (GMM) with n components to the normalized colours of the players in the initial training partition of the novel game. These components then form the words of a dictionary with which to encode players in subsequent frames. K-fold cross-validation reveals that $n = 35$ components yield optimal results. We use k-means clustering to find each team's cluster centres in this 35-dimensional space and assign players to the closest cluster. We again find that considering only the top half of the segmented player yields superior results. We also consider a variation of this approach, pretraining bag-of-colours, where the dictionary of colours is learned on the training set games.

Autoencoder

For additional comparison, we use a small convolutional autoencoder [70] trained on image reconstruction. The encoder network architecture is kept the same as our embedding network and the decoder mirrors the encoder. After training on the images in our training set, we

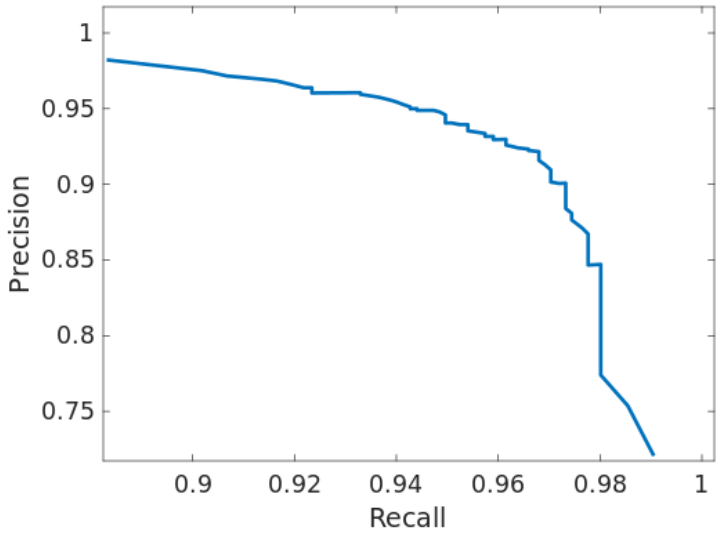

Figure 2.4: Precision-recall curve of the referee classifier.

used the encoder portion to extract a 1024-feature vector for each test image. We then use these features in the same setting as our embedding features to first learn cluster centres on the burn-in frames and then assign players to the closest centre for the rest of the frames.

2.4.3 Evaluation Methodology

We evaluate team affiliation labelling on players detected by mask R-CNN. These include false positives and imperfect segmentations. In addition, since the referee classifier is also imperfect, some referees will be incorrectly classified as players and will add noise to the contrastive learning process. We test both our supervised referee classifier and our unsupervised embedding network team classifier on the test set consisting of 4 games.

Accuracy is evaluated over the 30 annotated frames immediately following the burn-in interval. Since every tenth frame is annotated, this represents roughly 10 sec of video at 30 fps.

We assess the effects of noise in initial pseudo-labels on embedding network performance by considering different team assignment confidence scores p_{ij} thresholds. A higher confidence threshold leads to better clustering performance. We include this evaluation in supplementary materials.

Method	$\mathbf{n_{burn} = 1}$	$\mathbf{n_{burn} = 512}$
Colour Histogram	0.87 ± 0.031	$\mathbf{0.97 \pm 0.012}$
Bag-of-colours	0.76 ± 0.032	$\mathbf{0.97 \pm 0.018}$
Pretrained Bag-of-colours	0.86 ± 0.099	0.89 ± 0.189
Autoencoder	0.70 ± 0.076	0.92 ± 0.099
Embedding CNN	$\mathbf{0.94 \pm 0.009}$	$\mathbf{0.97 \pm 0.011}$

Table 2.1: Team classification accuracy as a function of the number n_{burn} of frames available for learning cluster centres prior to inference. We show the mean and standard error of the accuracy over 4 test games.

2.4.4 Referee Classification

For each game, we have 80-90 frames annotated frames and there are 3-4 referees on the rink, so for 9 training games we have 2000 referees in our training set, augmenting these by left/right reflections yields a total of 4000 training vectors. Employing a softmax threshold of 0.5, we achieve a mean accuracy of 98% with 93%, precision, 96% recall. Figure 2.4 shows precision recall curve for the referee classifier.

2.4.5 Team Classification

Table 2.1 shows the mean accuracy of team classification for all algorithms under evaluation. Results depend upon the number n_{burn} of frames available for learning cluster centres prior to inference. When n_{burn} is large (512 in this case), two colour-based and our methods perform fairly well, with colour-based methods rivalling our CNN approach. However, when n_{burn} is small (1 in this case), the performance of the colour-only methods drops dramatically, while our embedding CNN approach still performs very well.

This behaviour is shown in more detail in Fig. 2.5. We see that the simpler colour-based approaches and autoencoder approach improve continuously as the number of training frames increases, while our embedding CNN approach performs well even with only one training frame, improving only modestly thereafter. At least 512 burn-in frames are required before the pure colour approaches begin to rival our embedding CNN algorithm. The autoencoder method is

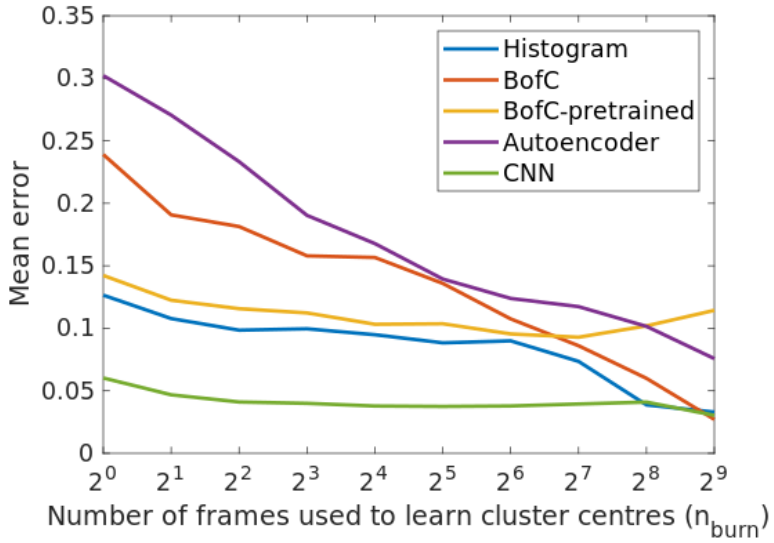

Figure 2.5: Error rate as a function of the number n_{burn} of initial frames used to learn cluster centres. We show the mean error of the accuracy over 4 test games

lagging behind even with 512 burn-in frames.

We believe that the advantage of our embedding CNN approach derives from the ability of our unsupervised contrastive learning network to learn from the training games an embedding space that is more effective for discriminating teams than colour histograms. This more discriminative space then allows well-separated cluster centres to be learned very quickly from the novel game.

2.4.6 Team Position Heatmaps Results

One useful application of player detection and team classification is to allow visualization of team positioning over the course of a game or a portion of a game. We demonstrate this by generating heatmaps for each game based on the 800-900 frames used for each game in our experiments. A learned homography is employed to back-project the image location (midpoint of the bottom boundary of the bounding box) to the playing surface. We also back-project our manual detections (Section 2.3.2) to form a ground-truth heatmap. Gaussian kernel density estimation [78, 73] is then used to estimate the player density (players per metre squared per frame) for both estimated and ground-truth heatmaps. The Gaussian bandwidth for KDE is calculated using Silverman's rule of thumb [80], and is roughly 30 pixels for all images (template

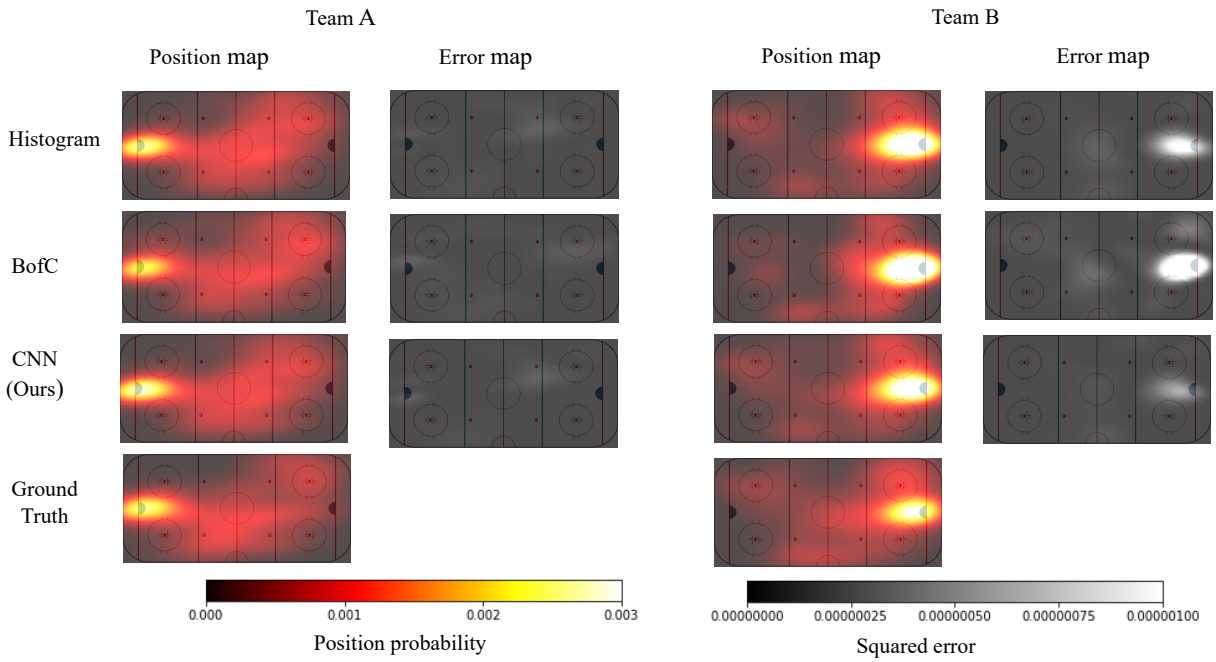

Figure 2.6: Team positioning heatmaps for a test game.

rink image size is 496x240 pixels).

Figure 2.6 shows example results from one test game for the three-team classification methods using $n_{burn} = 1$ frames to learn cluster centres. We see that our embedding CNN approach more consistently represents the true player densities than the pure-colour histogram or bag of colours approaches.

For quantitative evaluation, we scale the maps to integrate into one and then compute the KL-divergence between estimated and ground truth densities over our test set (Table 2.2). While the bag-of-colours algorithm outperforms a simple colour histogram, our embedding CNN approach substantially outperforms both pure-colour methods.

2.4.7 Runtime

Our experiments are conducted on a 3.6GHz Intel Core i9 CPU x 16 with 64 GB RAM and an Nvidia GeForce RTX 2080 GPU. Our method runs in real-time on segmented player images. It takes 21 milliseconds to learn team appearances for the game from a single frame and 11 milliseconds per frame for inference on subsequent frames. For convenience, we employed the widely available but non-real-time Mask R-CNN network [38] for player detection and segmen-

Method	Mean KL-divergence
Colour histogram	0.072
Bag-of-colours	0.069
Embedding CNN (Ours)	**0.047**

Table 2.2: KL-divergence of automatically-generated player positioning heatmaps from ground truth.

tation, which runs at roughly 5fps. If replaced with a real-time segmentation network, such as Yolact [15], our whole system will run in real-time. We leave this for future work.

Training of the embedding network takes 10-20 mins including finding pseudo labels for input images.

2.5 Conclusions

Our results demonstrate that a learned representation that can incorporate both colour and spatial features can produce superior results for team classification than a pure-colour approach. We also demonstrate that such a representation can be learned in an unsupervised fashion, using contrastive learning with a triplet loss. A major benefit is that unsupervised pre-learning of the representation allows for ultra-rapid learning of cluster centres from novel games, which limits the burn-in period, allowing online inference. We also show how this approach to team classification can be used to produce accurate team-conditional player positioning maps that can be useful for coaching and game analysis.

Chapter 3

Jersey Number Recognition

3.1 Introduction

Jersey number recognition is a critical task in sports video understanding and automated game analysis. One of the reasons it is so important is that sports video understanding depends fundamentally upon long-term tracking (over many minutes, i.e., many thousands of frames) of individual players. Since players on the same team are dressed to look almost identical, the jersey number is a very precious feature that can serve to disambiguate tracks, especially across frames in which players become tightly clustered, as is common in many team sports.

While important, jersey number recognition can be a very challenging task, as the jersey number is typically only clearly visible on a minority of frames, and motion blur, body pose variations, projective distortions, occlusions, and folds in the jersey material cause complex distortions all conspire to make reliable recognition difficult. Previous methods have approached the problem as a ground-up classification task, in which a network is trained from scratch on a large labelled dataset of detected players. A problem with this approach is that training the network from scratch requires a large labelled training dataset; Thus far, these have been proprietary and not released publicly. Here we study whether the problem can be made more accessible by making use of Scene Text Recognition (STR) systems, pre-trained on more general large-scale synthetic and text-in-the-wild datasets. We assess performance when using these systems out of the box and also when first fine-tuning on a modest jersey number dataset. We also assess how well such a system can generalize across sports, and very different camera geometries, with or without additional fine-tuning, and how best to aggregate image-level recognition to label

tracklets comprised of many frames.

To facilitate the research in the area we introduce a novel hockey jersey number dataset. It consists of hockey player images collected from university-level hockey games recorded from a stationary camera (setup explained in Section 2.3.2) as well as hockey player images from the McGill NHL public tracking dataset[2, 100]. The dataset has been manually annotated with the correct jersey number if it is legible by human eyes and a flag to indicate it is illegible otherwise. To our knowledge, this is the first image-level public dataset for hockey jersey number recognition.

In summary, our main contributions are:

- A novel image-level dataset for hockey jersey number recognition.
- A high-performance pipeline for detection, localization and frame-level recognition of jersey numbers.
- An analysis of how well this pipeline can generalize across sports, camera geometries and frame- vs tracklet-level jersey number recognition, with and without fine-tuning.

3.2 Related Work

The problem of jersey number recognition has been posed as image-level recognition [59, 88, 13, 58, 56] as well as tracklet-level recognition [90, 89, 20, 8]. Some methods detect and localize the jersey number region and then classify the numbers [59, 58, 56], while others assume that the image region containing the jersey number has already been cropped [88, 13, 32].

Progress on this problem has been slowed by the lack of public datasets that can be used to compare methods. This is now starting to be addressed with the 2023 release of the SoccerNet Jersey Number dataset [23, 4], although there are as yet no public datasets for other sports.

3.2.1 Image-level Jersey Number Recognition

Gerke et al. [32] and Li et al. [56] were among the first to apply CNN-based classification approaches to image-level jersey number recognition, and CNNs have been the dominant approach since this time. Liu et al. [58, 59] demonstrated the utility of body pose detection to improve classification with Faster R-CNN [75] and Mask R-CNN [38] architectures, respectively.

Vats et al. [88] demonstrated that multi-task training of a network on both holistic and digit-wise number classification results in better performance than a network trained on either task alone. They made use of a large, labelled dataset but unfortunately, it has not been made public. Also, despite its size, the training dataset does not include all possible jersey numbers and the system cannot generalize to other numbers. Bhargavi et al. [13] employed a similar approach but pre-trained using synthetic data and then fine-tuned on a small, labelled dataset of real images.

3.2.2 Tracklet-level Jersey Number Recognition

The visibility of a player's jersey number in video frames is often compromised due to motion blur and the player's position relative to the camera. In many instances, a player can be obscured by others, leading to multiple jersey numbers appearing in the same image. Identifying and pinpointing the jersey number of interest within a player's sequence of frames is a key step for recognizing jersey numbers at the tracklet level. Vats et al. [88] approach this by first classifying frames in each player's tracklet as legible or illegible. They then use only legible images to classify the number. On the other hand, Balaji et al. [8] propose a keyframe identification module that detects jersey numbers and filters out outliers (number detections that don't belong to the player in question or are too blurry for the recognition task). They use a jersey number detector and histogram-based features to detect and localize jersey numbers. However, the specifics regarding any additional data or annotations used to fine-tune their detector remain unclear. In our work, we take a similar approach to identify relevant images first. Instead of relying on hand-crafted histogram-based features, our method utilizes features derived from a person re-identification network to filter out distractions, such as other players blocking the main subject. Similarly to [88], we then apply a classifier to determine if the image contains legible numbers. Our approach is simple and shows superior performance on a challenging SoccerNet dataset.

For jersey number recognition at the tracklet level, there is an opportunity to integrate information across frames for better reliability. Chan et al. [20] and Balaji et al. [8] employed an LSTM while Vat et al. [90] used a temporal convolutional network to aggregate information over time. Vats et al. [89] have also explored the use of transformers for tracklet-level jersey number recognition within the multi-task approach introduced in [88]. They also make use of prior knowledge about the roster of players on the ice.

These prior approaches all treat jersey number recognition as a specialized classification problem requiring the design and training of a dedicated classification network. In contrast, we propose to explore a system based upon a more generally trained scene text recognition (STR) model, which will allow our approach to take advantage of progressive improvements in STR technology, adapt to different scenarios with little or no fine-tuning, and handle all possible jersey numbers, instead of being restricted to numbers that happen to be in the training dataset. As in [58, 59], we take advantage of body pose detection to localize the jersey number. As in [88] we use a weak-labelling strategy to generalize from image-level to tracklet-level annotation. But in contrast to prior tracklet-level approaches [20, 88, 90] we explore much simpler methods for integrating information across frames, demonstrating competitive results.

3.2.3 Scene Text Recognition

Scene Text Recognition (STR) is the task of recognizing text that occurs in the built environment (e.g., addresses, retail signs. traffic signs, license plates etc.). Several large datasets containing both synthetic and real data have been made available to train STR models. The current state-of-the-art model PARSeq [9] uses an encoder and decoder architecture in addition to a learned language model. It shows high performance on several challenging real-world datasets that include character occlusions, diverse orientations and varied illumination. Due to the lack of image-level jersey number datasets, STR has not previously been trained or evaluated on the jersey number recognition task. Here we explore how PARSeq can be integrated into a pipeline for jersey number recognition with and without fine-tuning.

3.3 Method

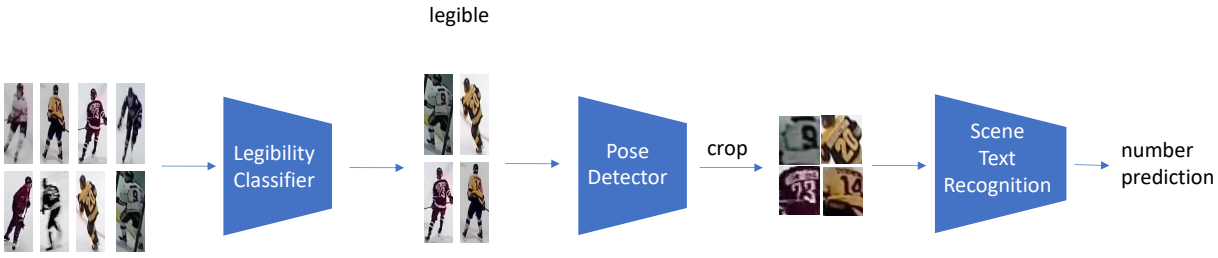

Figure 3.1: Pipeline of image-level jersey number detection and recognition.

3.3.1 Overview

To solve the jersey number recognition problem at the image level we introduce a simple yet very effective pipeline that detects, localizes and recognizes a jersey number of a player. We then extend this pipeline to tracklet-level jersey number recognition by addressing challenges specific to that task: filtering out distractors and combining image predictions into a single tracklet-level prediction. We describe all these components in detail in subsequent sections.

Image-level Task

Figure 3.1 shows an overview of our image-level jersey number recognition pipeline. In typical sports video, a jersey number is visible in only a minority of images. Thus, the first step in jersey number recognition is to identify in which frames the number is visible and legible. To perform this first task, we employ a binary CNN classifier based on an ImageNet[79] pre-trained ResNet34[39] model, fine-tuned on our new hockey dataset in which each player crop has been labelled as legible or illegible. To estimate a bounding box around the jersey number we employ a body pose detector and use the estimated pose keypoints to crop out the player's torso region. To classify a jersey number within this bounding box we employ the state-of-the-art STR system PARSeq [9] fine-tuned on a small number of hockey jersey number images.

Tracklet-level Task

To extend the above pipeline to the tracklet level (Fig. 3.6), we first use main subject filtering methods to identify frames that contain unoccluded players of interest. As in the image pipeline, we then employ our legibility classifier, followed by pose estimation to detect and localize jersey numbers. Finally, we use STR to recognize jersey numbers on each legible and unoccluded frame before aggregating image-level results over the entire tracklet.

3.3.2 Datasets

To explore generalization across different sports, camera geometries, and image- vs tracklet-level classification, we employ two datasets: Our own novel image-level hockey dataset (to be made public) and the recently-released tracklet-level SoccerNet soccer dataset [23]. For both datasets, reliable jersey number recognition is challenging due to diversity in illumination, occlusions, motion blur, pose variations and material deformations.

Hockey

To address the lack of publicly available image-level jersey number datasets, we introduce a new hockey jersey number dataset. We draw images from two sources:

- University Hockey - player images from 9 different games recorded with a stationary camera.
- McGill Hockey Player Tracking Dataset [2, 100] - player images from 8 different NHL games from broadcast videos.

Note, that the camera geometries are very different. While the University dataset is recorded with a fixed wide-field camera covering the whole rink, the McGill dataset is broadcast video, in which the camera zoom varies but is typically much more zoomed-in than for the university dataset. For both datasets, there is a lot of motion blur and partial occlusion. The university hockey images are especially challenging: A single camera device captures the whole rink and there is no pan and zoom, so player images are typically of lower resolution and jersey numbers are harder to decipher. To make a more diverse hockey dataset we combine images from both the university and NHL into a single labelled image-level jersey number dataset. We will make this dataset publicly available upon publication.

The hockey image-level dataset consists of cropped player images and has two types of annotation: legibility and jersey number. Player images were labelled as legible if the annotator could be certain of the jersey number. For jersey number recognition we used only a subset of these legible images to avoid excessive duplication of the same number. These images are labelled with a jersey number. We partitioned the data into training (10 games), validation (1 game) and test (6 games) - Table 3.1 details how this breaks down in terms of number of labelled images. Sample images from the dataset are shown in Figure 3.2.

Figure 3.3 shows the distribution of jersey numbers for training and test. There are 54 unique jersey numbers in the training set and 25 in the test set. Two numbers in the test set do not appear in the training set.

Soccer

In early 2023, SoccerNet [23] released the Jersey Number Recognition dataset and challenge [4], making it the first large public jersey number recognition dataset. This dataset consists of a

(a) Hockey Dataset (b) SoccerNet Dataset

Figure 3.2: Sample images from Hockey and SoccerNet datasets.

Part	Legibility		Jersey Number
	legible	total	
Train	4,706	94,036	3,531
Validation	923	14,138	233
Test	2,158	24,809	486
Total	7,787	132,983	4,250

Table 3.1: Hockey Dataset: number of labelled images by partition and annotation type.

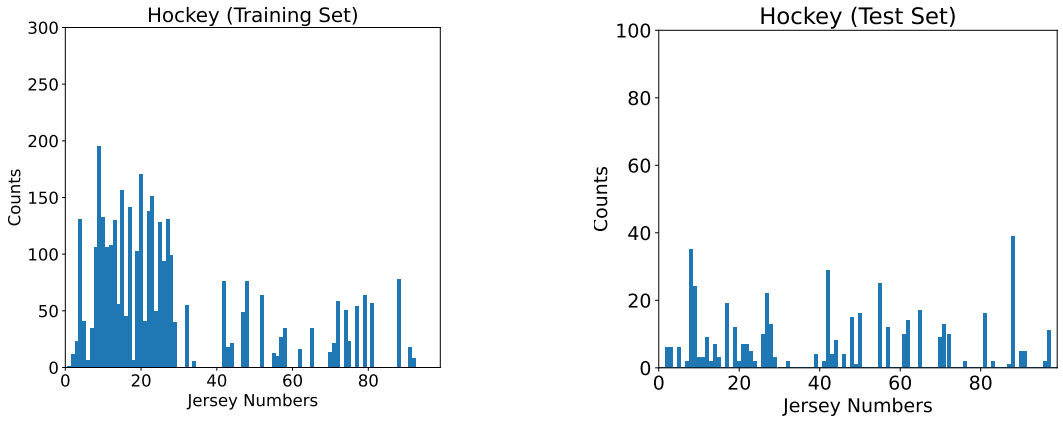

Figure 3.3: Hockey Dataset jersey number distribution.

	Train	Test	Challenge	Total
Tracklets	1,427	1,211	1,426	4,064
Images	733K	564.5K	748.6K	2,046K

Table 3.2: SoccerNet Jersey Number Dataset.

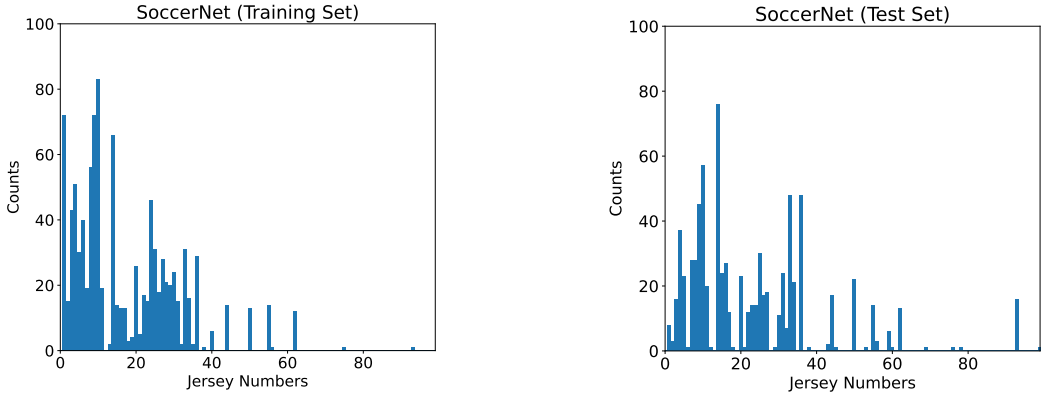

Figure 3.4: SoccerNet Dataset jersey number distribution.

collection of player tracklets and contains tracklet-level annotation. The dataset is partitioned into training, test and challenge, with a total of 4,064 tracklets. The average length of tracklets is 482 frames. Table 3.2 contains dataset statistics and Figure 3.2 shows sample images. During our experiments we discovered a flaw in the annotations: It includes tracklets for a soccer ball with the label of jersey number '1'. We added a component to our pipeline to identify soccer ball detections based on the average dimensions of the soccer ball images in the training set.

There are 55 unique jersey numbers in the training and test partitions and the test set contains 10 numbers that do not appear in the training partition. Figure 3.4 shows their distribution.

3.3.3 Image-level Task

Detection and Localization

A jersey number is typically only visible and legible on a fraction of the frames. To filter out images with illegible numbers we train a binary classifier to identify player images as either

Figure 3.5: Sample jersey number crops automatically extracted from player images.

legible (has a visible and decipherable jersey number) or illegible. Since jersey numbers are located on the torso of the player, we utilize a pose estimator to extract the torso region. This approach is simpler than training a dedicated jersey number detector because it does not require time-consuming jersey number bounding box annotation. Instead, it relies on a simple legible/illegible binary label and an off-the-shelf pose estimation network.

For our legibility classifier we employ a ResNet34 [39] model pre-trained on ImageNet [79] and fine-tune it on our binary hockey legibility dataset. Our hockey legibility dataset is highly imbalanced with only 5% of images labelled as legible. Although this reflects the true distribution, our experiments showed that using a balanced training dataset improved classifier performance. Therefore, we train with a balanced subset consisting of all legible images and a equal number of randomly selected illegible images. Test results are reported on the original imbalanced test data.

We train our binary legibility classifier for 20 epochs with a starting learning rate of 0.001 and momentum of 0.9. To improve the generalizability of our classifier we use Sharpness-Aware Minimization (SAM) [30] with SGD. We provide a careful ablation study of legibility model choice, as well as model generalizability analysis in Section 3.4.

We localize jersey number on the player image by extracting body pose keypoints using off-the-shelf body pose detector ViTPose[96] trained on MS COCO[57]. We then crop a rectangle defined by shoulder and hip joints padded by 5 pixels on the left, right, and bottom. A sample of the resulting crops from our hockey dataset is shown in Figure 3.5.

Recognition

Jersey number recognition is a specific case of Scene Text Recognition (STR). Recent STR models show very good performance recognizing text in the wild. We fine-tune leading STR model PARSeq [9] to recognize jersey numbers. PARSeq is trained on a collection of synthetic and real-world datasets including SynthText[34], COCO-Text[91], and TextOCR[81] (refer to [9] for a full list of training datasets.) There are several advantages to relying on the existing STR model for this task. It has been pre-trained on a vast number of images containing alphanumeric strings. It performs reasonably well on jersey number recognition tasks without any fine-tuning. Performance is further improved by fine-tuning on relatively small amount of jersey number data. Due to its token-processing nature, the model can predict jersey numbers that were not present in the training set, making it better suited to real-world applications. We fine-tune the model on legible jersey number crops from the hockey dataset for 25 epochs, limiting label length to 2 and using default PARSeq training settings.

Our proposed pipeline is simple, yet it outperforms previous methods. In the future, it can also benefit from advances in STR methods.

3.3.4 Tracklet-level Task

We adapt our image-level pipeline to the tracklet level by introducing two additional steps: main subject filtering and jersey number prediction consolidation.

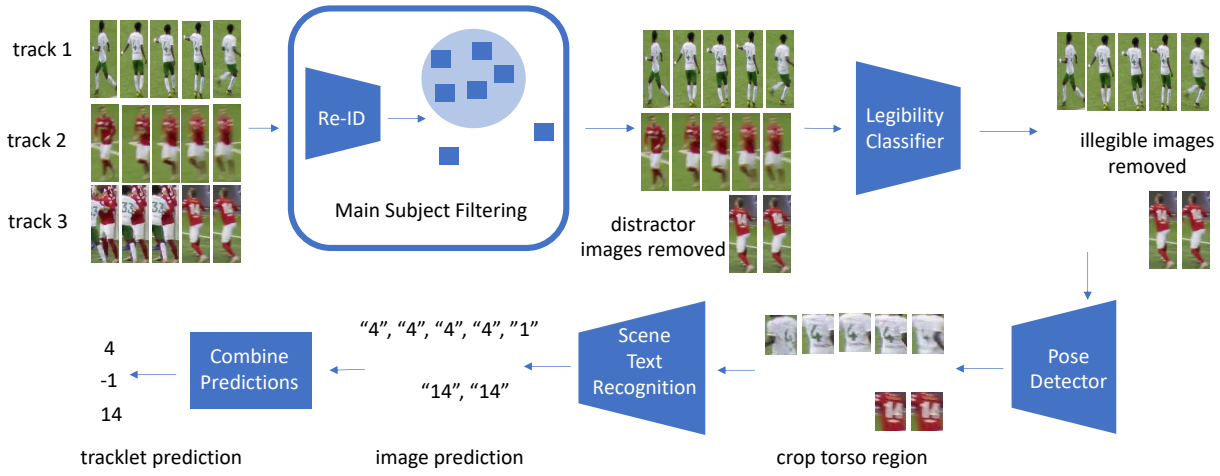

Figure 3.6: Pipeline of tracklet-level jersey number detection and recognition.

Main Subject Filtering

The SoccerNet dataset contains tracklets where the main subject is often occluded by other players. When the jersey number of the occluding player is visible it can affect both legibility and number predictions for the tracklet. This renders images where the main subject is occluded or multiple players are visible problematic. We study whether filtering out frames in which the main subject appears to be occluded can improve tracklet-level classification. To this end, we employ the Centroid-ReID [93] network trained on the Market1501 dataset [105] to extract a visual feature vector for each image in a tracklet. We fit an isotropic Gaussian to these vectors, and then exclude as outliers any images for which the feature vector lies more than N standard deviations from the mean. This process is repeated K times. In our experiments, this method leads to better overall results. Parameters for N and K were determined by grid search and cross-validation on a held out 30% subset of the training set tracklets. We found the optimal parameters to be $K = 3$, $N = 3.5$. Note, that the method is unsupervised; there are no labels for the main subject in the tracklet. We evaluate its performance based on its impact on the tracklet-level recognition task. Experiments show that this method leads to a boost in performance on the SoccerNet dataset (Section 3.4).

Detection and Localization

Extending our legibility classifier to the tracklet-level SoccerNet jersey recognition task is complicated by the lack of frame-by-frame labels. To overcome this barrier, we derive weak pseudo-labels from the tracklet-level annotations. We derive a set of positive (legible) pseudo-label instances by running our hockey-trained legibility classifier on the images within legible tracklets (tracklets with jersey number labels) and extracting instances deemed legible. Negatives are drawn from random images from illegible tracklets. We train the legibility classifier network using these pseudo-labels. At inference, a tracklet is deemed legible if it contains one or more images classified as legible.

Recognition

To fine-tune STR for the tracklet-level SoccerNet dataset we construct a weakly-labelled text recognition dataset based on tracklet-level data. In particular, we run our legibility classifier on

		Ground Truth	
		2 digit	1 digit
Prediction	2 digit	40%	7%
	1 digit	**48%**	5%

Table 3.3: Confusion matrix of STR predictions with regard to one- or two-digit jersey numbers.

all images in legible tracklets (tracklets with a jersey number label) and use these as pseudo-ground truth for fine-tuning.

At inference, we run the fine-tuned STR model on all images deemed legible in the tracklet. The result is a series of predicted jersey number labels: one for each legible image in the tracklet.

Prediction Consolidation

We propose a confidence-weighted majority vote approach to prediction consolidation. An important consideration when approaching this problem is the potential confusion between one- and two-digit jersey numbers. Two-digit jersey numbers are roughly twice as frequent as one-digit numbers in the SoccerNet dataset. Due to occlusions and variations in player pose, only one digit may be visible even when the jersey number consists of two digits (Figure 3.7). As a result, STR confusion regarding the number of digits in the jersey number is overwhelmingly due to mistaking a 2-digit number for a 1-digit number (Table 3.3).

Figure 3.7: Example of images where only one digit out of the two is visible. First row: true label 44, predicted 4. Second row: true label 34, predicted 3.

Method	Dataset Size	Accuracy
Li et al. [56]	12,746	86.7%
Liu et al. [58]	3,567	90.4%
Vats et al. [88]	54,251	89.6%
Bhargavi et al. [13]	3,000	89.3%
Ours	4,250	**91.4%**

Table 3.4: Previously reported results on image-level jersey number recognition task.

Model	Accuracy
Holistic Classifier (ResNet34)	48.1%
Multi-Task Classifier (ResNet34) [88]	65.2%
PARSeq (out-of-the-box) [9]	85.4%
Ours: PARSeq (fine-tuned on hockey)	**91.4%**

Table 3.5: Performance of jersey number recognition models on our hockey dataset.

We compute the tracklet-level prediction using a confidence-weighted majority vote of legible images. If the sum of confidences over frames is below a threshold, the tracklet is marked illegible. When some of the images in the tracklet are predicted to have two digits and others to have a single digit, we down-weight votes for one-digit numbers. Both of these measures provide a small boost to overall performance.

3.4 Results and Analysis

3.4.1 Image-Level Task

Our ResNet34[39] legibility classifier performs at 94.5% accuracy with F1-score of 71.7% on our hockey test set. We also evaluated a fine-tuned visual transformer model [27] (See Table 3.7) but found that, while it performs better when tested on the same dataset it was trained on, ResNet34[39] shows better results in generalizing to the new domain.

	Test: Hockey	Test: Soccer
Original	85.40%	80.51%
Fine-tune: Hockey	91.40%	83.90%
Fine-tune: Soccer	65.84%	87.45%

Table 3.6: Generalizability of the PARSeq STR model on hockey and soccer datasets with and without fine-tuning.

Model	$Hockey \rightarrow Hockey$		$Hockey \rightarrow Soccer$		$Soccer \rightarrow Soccer$		$Soccer \rightarrow Hockey$	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ResNet18 [39]	94.8%	71.4%	90.58%	93.0%	91.71%	94.15%	91.9%	65.3%
ResNet34 [39]	94.5%	71.7%	91.09%	93.7%	91.71%	94.17%	92.8%	63.2%
ViT [27]	94.8%	72.9%	86.9%	90.5%	90.75%	93.6%	92.6%	58.3%

Table 3.7: Performance comparison of three deep architectures for our legibility classifier. We examine how well different models generalize from $(TrainingDataset) \rightarrow (TestingDataset)$ and report both accuracy and F1 scores. Accuracy for Soccer is calculated at the tracklet level (a tracklet is deemed legible if it contains one or more legible images).

Experiment	Accuracy
Full	**87.45%**
No Bias	86.79%($\downarrow$ 0.66%)
No Bias, No Threshold	85.38%($\downarrow$ 2.07%)
No Main Subject Filtering	84.56%($\downarrow$ 2.89%)

Table 3.8: Ablation analysis of our soccer pipeline. We consider consolidation method with or without biasing toward two-digit jersey numbers. We also evaluate the effect of placing a threshold on the sum of confidences. The final row shows the performance with no main subject filtering. The results are on the SoccerNet test set.

Method	Test Acc	Challenge Acc
Gerke et al [32]	32.57%	35.79%
Vats et al [88]	46.73%	49.88%
Li et al [56]	47.85%	50.60%
Vats et al [89]	52.91%	58.45%
Balaji et al [8]	68.53%	73.77%
Ours	**87.45%**	**79.31%**

Table 3.9: Tracklet-level jersey number recognition performance on the SoccerNet Test and Challenge partitions. Results for other methods are cited from [8].

We evaluate jersey number recognition on image-level annotations for our hockey dataset considering only legible images and achieve an accuracy of 91.4%. Table 3.4 shows a comparison with methods previously reported in the literature. As a baseline, we evaluated both a ResNet34-based classifier trained on our data to predict a label 1-99, as well as the multi-task system described in [88] that uses a holistic classifier and digit-wise classifier heads. As with the results reported in [88], this multi-task training yields better results, but due to our small training set we see much lower performance than Vats et al. [88] reported. Without any fine-tuning PARSeq [9] trained on multiple synthetic and real scene text datasets achieves an accuracy of 85.4% on our hockey image-level dataset. The performance further improves with fine-tuning illustrating that the use of STR in the jersey number recognition pipeline is an appropriate choice.

3.4.2 Tracklet-Level Task

To evaluate recognition on the tracklet-level SoccerNet dataset we use the evaluation protocol followed in the SoccerNet Jersey Number Recognition Challenge. We evaluate the accuracy of tracklet-level labelling in which each tracklet may be comprised of both legible and illegible frames. Using the full pipeline with a legibility classifier and fine-tuned PARSeq model we achieve an accuracy of 87.45% on the SoccerNet test set and 79.31% on the challenge set. Table 3.9 shows the results of our method compared to previously reported on this dataset.

In Table 3.8 we present the results of several ablations. In particular, we demonstrate the effect of main subject filtering as well as different options for prediction consolidation. Our best

results are achieved using main subject filtering, as well as thresholding and 2-digit bias for prediction consolidation.

3.4.3 Implementation

The pipeline is implemented with PyTorch. The code and data are available at `https://github.com/mkoshkina/jersey-number-pipeline`.

3.5 Conclusion

We have introduced a robust pipeline designed for jersey number recognition at both image and tracklet levels. Our system outperforms previously reported results while requiring minimum fine-tuning. It generalizes exceptionally well to new jersey numbers as well as from one sport to another.

Furthermore, in an effort to foster continued research and development in this domain, we have introduced the novel publicly available dataset for image-level recognition of hockey jersey numbers.

Chapter 4

Player Tracking

4.1 Introduction

Player tracking is an essential task for game analysis. It involves detecting and following players for a period of time during a game. Player tracking is a specific case of a well-studied topic in computer vision - Multi-Object Tracking (MOT). Typically, MOT is studied in the context of pedestrian tracking (for example, for surveillance purposes) and vehicle tracking (for self-driving cars, and traffic analysis applications). The MOT task aims to track objects in a video sequence. Typically, the appearance or the number of objects is not known ahead of time. The output of an MOT algorithm is a set of lists of bounding box coordinates for each object and frame in which the object appears. Each such bounding box entry has an associated track ID for each unique object being tracked.

MOT is a challenging task because an object's appearance might vary depending on its position relative to the camera, objects might occlude one another or exit and then re-enter the field of view, objects can have a similar appearance, an object's motion can be unpredictable, etc.

Tracking players during a team sports event shares the same problem setup and challenges as tracking pedestrians. Generic MOT tracking algorithms can be applied to tracking players and will produce reasonable results. However, there are a number of important differences between generic MOT and team sport MOT:

- In sports, the objects to be tracked (the players) are typically fixed in number and remain the same over long periods. In contrast, in pedestrian and traffic scenarios a much larger

number of different objects will typically traverse the field of view, but each for much shorter periods.

- Pedestrian tracking focuses on short-term tracking, while player tracking in sports involves longer-term tracking, including recovery from long occlusions or departures from the field of view.

- Players on the same team often look similar, especially in winter sports like hockey, where gear obscures distinctive features, making traditional re-identification methods less effective.

- Players can be uniquely identified by their team id and jersey numbers.

Some of these differences make player tracking easier than general MOT and some more challenging.

In our work, we explore using domain knowledge to address the challenges of player tracking. In particular, we recognize that team affiliation and jersey number uniquely identify a player. Establishing both attributes for the player will enable us to track them for long periods. In our work, we follow the unified, hierarchical approach proposed by Cetintas et al.[19] that integrates short-term and long-term tracking, and innovate by incorporating domain-specific features specific to sports, including jersey number, team ID, and field position.

4.2 Overview

The MOT task has its roots in short-term single-object tracking. Earlier approaches include methods that identify and match keypoints between frames [83] such as the Kanade–Lucas–Tomasi feature tracker [86], contour tracking methods[84], template matching methods[17] and correlation filter approaches [41]. The advances in tracking-related research have been facilitated by the introduction of the annual VOT challenge [5] in 2013, and the MOT challenge in 2015 [3].

In recent years, due to the success of convolutional neural networks on object detection and classification, the prevailing approach for the MOT task is *tracking by detection*. Tracking by detection involves using an object detector to find all objects at each frame and then using a data association approach to determine which detection belongs to which track (See Figure 4.1). Since the object can be occluded or detections can be missed, it is common to also keep track of

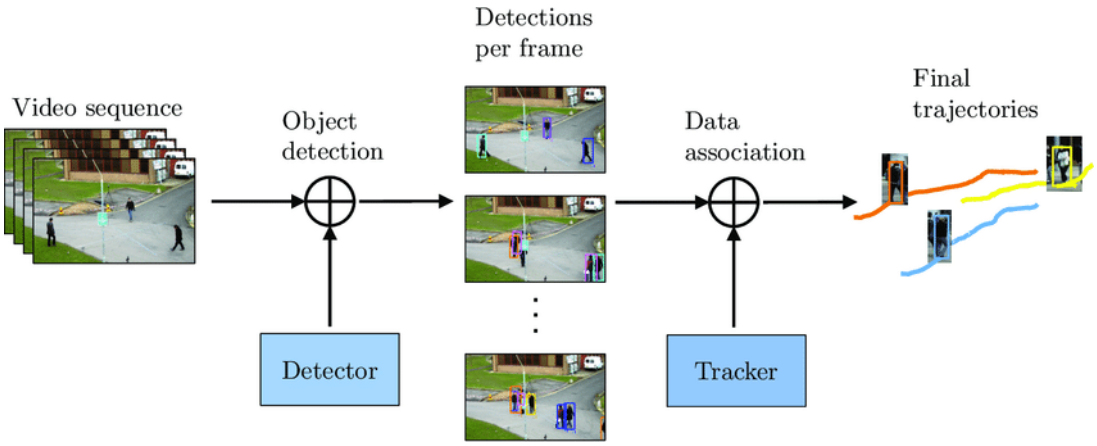

Figure 4.1: Tracking by detection paradigm. [Image from [54]]

'lost' tracks and to employ re-identification to match them to any new detections. The approach relies heavily on the detector and suffers whenever detections are poor. Data association poses many challenges as well due to the unknown number of targets, objects changing appearance, occlusions, similar object appearance, etc.

4.2.1 Problem Formulation

The problem of MOT tracking can be formulated as a multi-variable estimation problem. Given a sequence of N frames, we denote x_k^i to be a detection of the object k in frame i. Let $X = \{x_k^i\}$ be a set of all detections. A single trajectory hypothesis is an ordered list of detections $t_k = \{x_k^1, x_k^2, ..., x_k^N\}$, where some detections could be missing. An association hypothesis $T = \{t_k\}$ is a set of single trajectory hypotheses that attempts to account for all tracks. The objective of multiple object tracking is to find the optimal trajectories of all the objects, which can be modelled by performing MAP (Maximum a posteriori) estimation from the conditional distribution of trajectories given all the detections:

$$T^* = \underset{T}{\operatorname{argmax}}\, P(T|X) \tag{4.1}$$

4.2.2 Online vs Offline Tracking

MOT methods can be categorized as online and offline [64]. In online tracking, also known as causal tracking, the frames are processed sequentially and trajectories are computed up to the current frame. In offline tracking, both past and future frames are used to calculate the

optimal trajectories. Figure 4.2 illustrates the two approaches. Online tracking is suitable for applications that require real-time results. Due to its nature, online tracking cannot correct mistakes in past tracking results as new information becomes available. Offline (or non-causal) tracking benefits from processing the whole sequence which allows for methods to arrive at a globally optimal solution. Offline tracking is not suitable for real-time applications and may require significant computational resources.

In this proposal, we consider both online and offline methods.

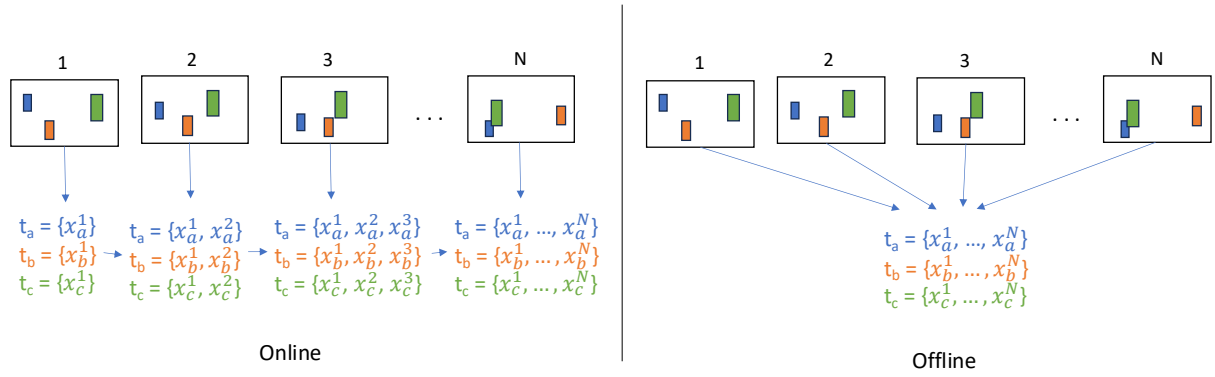

Figure 4.2: Online vs offline approach. Online methods take previous results and the current frame to produce tracks as they go. Offline methods consider all frames to produce the final tracking result.

4.2.3 Metrics and Evaluation Methodology

One of the driving forces of recent developments in object tracking is the existence of large MOT datasets such as MOT Benchmark[3] and PathTrack[68]. MOT Benchmark [55, 72] provides a dataset, evaluation framework and an annual challenge. Over the years, there have been many different metrics proposed to measure, analyze and compare the performance of different trackers. The current standard for MOT evaluation is comprised of two main sets of metrics: CLEAR (arising from the CLassification of Events, Activities and Relationships workshop) [82] and HOTA (Higher-Order Tracking Accuracy)[63]. In addition, track quality measures introduced in [94] and ID scores[76] are usually included in the evaluation.

Ground truth annotation for MOT tasks consists of a number of tracks. Each track is comprised of a sequence of bounding box coordinates indicating the location of the tracked object

in each frame. Before computing any metrics, matching between the hypothesis and ground truth tracks needs to be established. The current standard is based on a method proposed and detailed in CLEAR [82]. It involves matching based on proximity or more precisely Intersection-over-Union (IoU) of bounding boxes. An IoU threshold is typically set to 0.5 or 50%. Matching of hypothesis to ground truth needs to satisfy the constraint that one object cannot be accounted for by more than one hypothesis, and that one hypothesis cannot account for more than one object. The optimal matching is found using the Hungarian algorithm [53]. However, if a ground truth object is matched to the hypothesis at time $t-1$ and the IoU between them in frame t is above the threshold, then the correspondence carries over to frame t even if there exists another hypothesis that is closer to the actual target. In other words, matching is not done independently for each frame but promotes the continuity of tracks. The ground truth bounding boxes that cannot be associated with a hypothesis are counted as false negatives (FN), and the hypotheses that cannot be associated with a real bounding box are marked as false positives (FP). Every time a ground truth object tracking is interrupted and later resumed, it is counted as a fragmentation. Note that fragmentation is only counted if a ground truth box exists. Every time a tracked ground truth object ID is changed during the tracking duration, it is counted as an ID switch.

CLEAR Metrics

The most widely used metric to measure tracking performance is MOTA (Multi-Object Tracking Accuracy), which combines three sources of error to measure accuracy:

- FP - the number of false positives
- FN - the number of false negatives (missed detections)
- IDSW - the number of ID switches

$$MOTA = 1 - \frac{(FP + FN + IDSW)}{GT} \in (-\infty, 1] \tag{4.2}$$

where GT is the number of ground truth bounding boxes. MOTA is usually reported in %.

Detection Accuracy (DetA) is often reported as well. For two tracking algorithms executed on the same set of detections, detection accuracy can still differ, since it is based on the detections

that are included in the predicted tracks:

$$DetA = 1 - \frac{(FP + FN)}{GT} \tag{4.3}$$

Another commonly reported metric MOTP (Multi-Object Tracking Precision) measures localization precision. It is the average dissimilarity between all true positives and their corresponding ground truth targets. More precisely, it is an average overlap between all correctly matched hypotheses and their respective objects and ranges between 50% and100%.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{4.4}$$

where $d_{t,i}$ is IoU between the hypothesis i in frame t with its ground truth match, and c_t is the number of matches in frame t.

Track Quality

Another commonly reported set of metrics is track quality metrics. These ignore both the precision of detections and ID switches and instead measure how much of the track has been covered (possibly by multiple fragments).

- MT - Mostly Tracked. The number of ground truths tracks that have matching predicted tracks for at least 80% of the frames.
- ML - Mostly Lost. The number of ground truths tracks that are tracked for less than 20% of the frames.
- PT - Partially Tracked. The number of ground truths tracks that are tracked for more than 20% and less than 80% of the frames.
- FM – Fragmentations. The number of times ground truth object tracking is interrupted and later resumed.

ID scores

In some tracking applications, such as surveillance and sport, it is important to be able to maintain the correct identity of the track for longer periods through occlusions and missed

detections. MOTA tends to measure detection accuracy more than association accuracy. For that reason, the MOT Challenge has adopted another metric to better measure association accuracy. ID Precision, Recall and F1 score were first proposed in [76] in order to measure multi-camera tracking accuracy better. They were then adapted to single-camera tracking as well. Calculation of ID scores involves a different approach to matching proposed and ground truths detections. Instead of matching on the frame level as done in MOTA calculations, [76] performs matching on the trajectory level. This defines new types of detection matches. IDTPs (identity true positives) are matches on the overlapping part of trajectories that are matched together. IDFNs (identity false negatives) and IDFPs (identity false positives) are the remaining ground truth detections and predicted detections, respectively, from both non-overlapping sections of matched trajectories and from the remaining trajectories that are not matched. Based on these, ID scores are calculated as follows:

- IDP - Identification Precision:

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{4.5}$$

- IDR - Identification Recall:

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{4.6}$$

- IDF1 - Identification F1 Score:

$$IDF1 = \frac{2}{\frac{1}{IDP} + \frac{1}{IDR}} = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{4.7}$$

The Hungarian algorithm is used to select which trajectories to match so that the sum of the number of IDFPs and IDFNs is minimized. Proposals are generated by identifying matches with IoU values greater than the specified threshold between detection and ground truth boxes.

HOTA Metrics

HOTA (Higher-Order Tracking Accuracy) is the most recently introduced MOT metric. It was proposed only in 2021 [63] and was quickly adopted as a new standard metric by the MOT Challenge. The argument for introducing yet another metric is to provide a single metric that balances detection measurement and association accuracy. MOTA performs matching on

the local detection level and is biased towards measuring detection accuracy. The IDF1 score does trajectory matching and is biased towards measuring association. HOTA is introduced as a middle ground between them. [63] introduces a new measure called association accuracy or AssA. We formally define it by Equation 4.11, but intuitively it is the average alignment between matched trajectories. The HOTA metric is the geometric mean of detection and association accuracies averaged over different localization thresholds. Figure 4.3 illustrates the differences between the main MOT metrics in terms of detection vs association bias and demonstrates how metric HOTA balances the two.

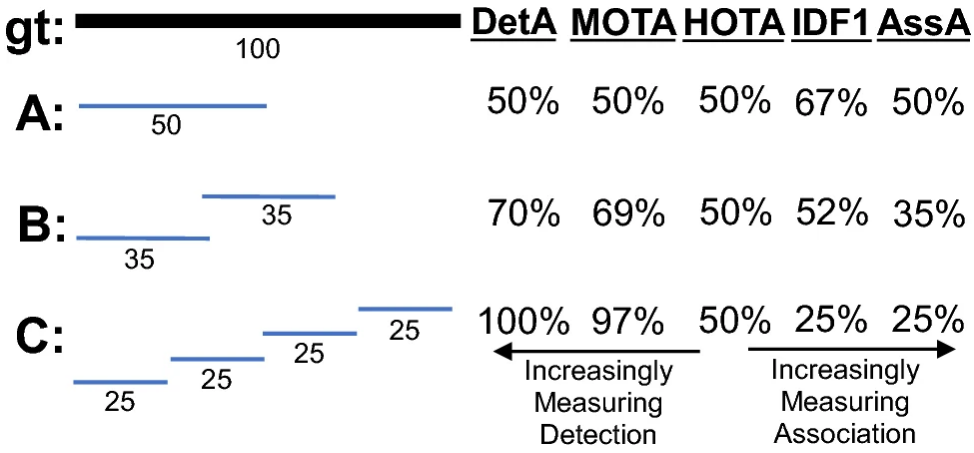

Figure 4.3: Comparison of main MOT metrics in terms of detection vs association measurement [Image from [63]]. Three different trackers are shown in order of increasing detection accuracy and decreasing association accuracy. While MOTA and IDF1 tend to focus on accurate detection and association, respectively, HOTA provides a balanced perspective. HOTA achieves this by explicitly combining a detection score (DetA) and an association score (AssA).

The matching of detections to ground truth is conducted on a frame-by-frame basis as in MOTA with one difference: matching is done to maximize the HOTA metric. [63] also introduces the concepts of True Positive Association (TPA), False Negative Association (FNA), and False Positive Associations (FPA) that are defined for each true positive (TP) match. Matching occurs at a detection level. As in MOTA, a true positive (TP) match is a pair consisting of a ground truth detection (gtDet) and a predicted detection (prDet), for which the IoU is greater than or equal to a set threshold. A false negative (FN) is a gtDet that does not have any prDet. A false positive (FP) is a prDet that is not matched to any gtDet. For a given ground truth object c,

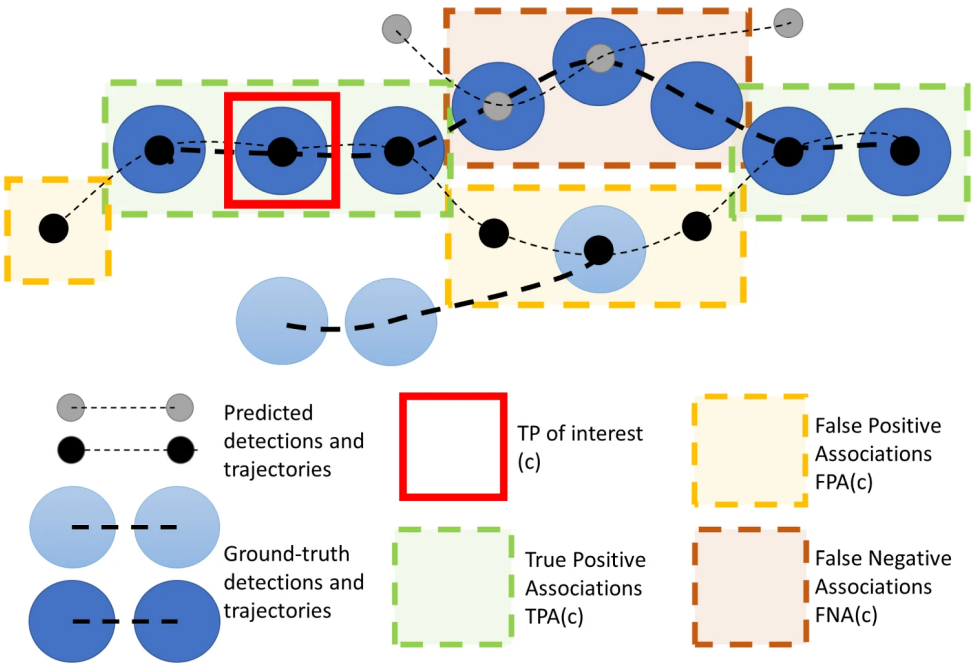

Figure 4.4: Association scores calculations for HOTA Association Accuracy [Image from [63]]

association metrics are illustrated in Figure 4.4 and are defined below. Note, that TPA, FNA and FPA are defined over all frames, and prID and gtID are predicted and ground truth ID of a single detection. TPA counts the number of correct one-to-one ground truth matches for a given object over all frames. For example, in Figure 4.4 for an object c (red) TPA(c) is 5 (green). Formally, TPA(c) is defined as: the set of TPs which have both the same gtID and the same prID as c:

$$\text{TPA}(c) = \{k\}, k \in \{\text{TP}|\text{prID}(k) = \text{prID}(c) \wedge \text{gtID}(k) = \text{gtID}(c)\} \tag{4.8}$$

For a given TP, c, the set of FNAs is the set of gtDets with the same gtID as c, but that have a different prID, or no prID (missed detections):

$$\text{FNA}(c) = \{k\}, k \in \{\text{TP}|\text{prID}(k) \neq \text{prID}(c) \wedge \text{gtID}(k) = \text{gtID}(c)\} \cup \{\text{FN}|\text{gtID}(k) = \text{gtID}(c)\} \tag{4.9}$$

For a given TP, c, the set of FPAs is the set of prDets with the same prID as c, but that were

either assigned a different gtID, or no gtID (false positive detections):

$$\text{FPA}(c) = \{k\}, k \in \{\text{TP}|\text{prID}(k) = \text{prID}(c) \wedge \text{gtID}(k) \neq \text{gtID}(c)\} \cup \{\text{FP}|\text{prID}(k) = \text{prID}(c)\} \quad (4.10)$$

Association accuracy for a given true positive c is then defined as follows:

$$\mathcal{A}(c) = \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)| + |\text{FPA}(c)|} \quad (4.11)$$

Notably, HOTA association accuracy is not influenced by fragmentation (See Figure 4.5). The interrupted and resumed track that covers 80% of the true track will have the same association accuracy as an uninterrupted one that covers 80% of the track, as long as the identity is preserved after interruption.

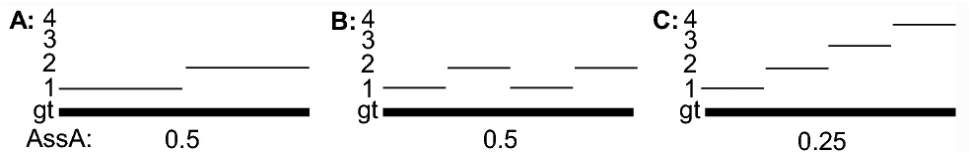

Figure 4.5: HOTA Association Accuracy for different fragmentation of the recovered track [Image from [63]]. Note that having multiple smaller fragments does not impact association accuracy as long as the total number of frames with association to the same ID remains the same (left and center diagrams). The accuracy is affected if the fragments are assigned to different IDs (right).

Finally, the HOTA metric combines both association and detection metrics:

$$\text{HOTA}_\alpha = \sqrt{\frac{\sum_{c \in \{\text{TP}\}} \mathcal{A}(c)}{|\text{TP}| + |\text{FN}| + |\text{FP}|}} \quad (4.12)$$

Note that α is the IoU threshold used for matching. In order to incorporate localization accuracy into the HOTA metric, the HOTA number reported is actually an area under the curve of HOTA scores across the valid range of α values between 0 and 1 (or more precisely an approximation of this area). For each separate value α, the matching is done separately before computing the metric.

$$\text{HOTA} = \int_0^1 \text{HOTA}_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\alpha \in \{ \begin{smallmatrix} 0.05,\ 0.1,\ \ldots \\ 0.9,\ 0.95 \end{smallmatrix} \}} \text{HOTA}_\alpha \quad (4.13)$$

4.3 Related Work

4.3.1 Generic MOT

Early Methods

Multi-object tracking began gaining the attention of computer vision researchers in the 1990s. Before the success of deep learning methods in the 2010s, proposed MOT approaches relied on traditional computer vision techniques and probabilistic methods. The Kalman filter [50] and its extensions were widely used in MOT solutions to predict the next location of the object [74, 77]. A bipartite assignment between tracklets and new detections was employed by many methods using both greedy [95] and Hungarian [44] algorithms. Linear programming [49] and k-shortest paths [11] have been applied in offline settings.

The offline MOT problem is often modelled as a graph where nodes represent detections or tracklets and edges between two nodes are potential associations. One trajectory corresponds to one flow path in the graph. A number of methods were proposed to use this model and solve for an optimal global solution using min-cost network flow algorithms [101, 22]. Zhang et al. [101] also introduced an occlusion model. After first finding an optimal solution on the graph containing all detections, potentially occluded objects are added to the graph and the optimization algorithm is executed again.

The min-cost flow formulation is guaranteed to find an optimal solution in polynomial time. Method [101] uses simple visual features such as color histograms, scale, and temporal distance to model association cost. The main limitation of Zhang et al. [101] is the complexity of the resulting graph. In the experiments conducted by the authors, there were only a few objects per frame being tracked (2 on average). Modern datasets tend to be much more crowded with the MOT17 dataset having on average 65 detections per frame. Finding globally optimal solutions on these graphs is prohibitively expensive. The other limitation is the simplicity of the visual features being used.

Modern Methods

Since the success of deep learning on object detection tasks, practically all current methods for MOT follow the tracking-by-detection approach in one form or another. Some methods approach the tracking problem as purely a data association task [16, 43]. However, some influential trackers

proposed in recent years extend existing object detectors to do tracking [12, 106].

Online Methods

The first method to adapt an object detector to the task of tracking is Tracktor++[12]. It adapts Faster R-CNN [75] to the tracking task by using its bounding box regressor to regress the bounding box of the object from a previous frame to the current one. It achieved state-of-the-art results at the time of publication. A Siamese re-identification network is used to re-identify objects that have been 'lost' due to occlusion, missed detection, or temporarily leaving the field of view. The proposed method is simple and solves most simple cases. Using a regressor to maintain the identity of the unoccluded objects from frame to frame reduces the number of identity switches and the number of false negatives and outperforms previous methods. In addition, since the regressor of the object detector is used, it is not trained to perform tracking. An object detector is trained on detection data alone. At inference time the regressor is re-purposed for the data association task. However, occlusions are not handled very well, which results in lower performance in crowded scenes. This is an effective short-term tracking approach that fails to address long-term tracking.

CenterTrack[106] is another recent tracking-by-detection method that dominated the leaderboards when it was published. It is based on the CenterNet[28] object detector. The authors proposed to do detection and tracking as a single step, tracking objects by their center points. Instead of predicting object detections as a list of bounding boxes, CenterTrack outputs a heatmap of object centers and object sizes. The proposed network takes two frames (current and previous) and a heatmap of previous point tracks. In addition to a center point and the object size maps, it outputs an offset map (See Figure 4.6). The offset map is a predicted displacement of the object center between the previous and current frames. This displacement is then used to associate detections between frames. A simple greedy algorithm uses the current frame detection and its displacement to associate it to the previous frame's detection. CenterTrack poses the tracking problem as tracking in consecutive frames. There is no re-ID when the object leaves the frame and reappears later. Its strength, however, is in local or short-term tracking. The network learns to predict object motion from one frame to another without relying on other approaches such as Kalman filters. CenterTrack improves further on the performance of Tracktor++ for simple cases. By incorporating displacement prediction, CenterTrack does a simple

motion prediction in the same network. Similar to Tracktor++, it can be trained on static images using augmentation, making it easier to apply to new datasets. Its major strength is in being real-time. On the downside, there is no re-identification or re-connection of broken tracks, which makes it unsuitable for long-term tracking.

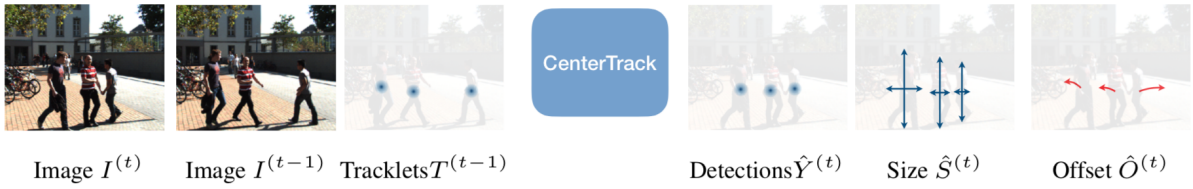

Figure 4.6: Inputs and outputs of CenterTrack[Image from [106]]

To address the lack of re-identification in the CenterTrack approach, an extension has been introduced in [104]. FairMOT[104] proposes a parallel re-ID branch that is trained together with the detection branch. The re-ID branch considers features from centers of objects and is trained as a classification task, where each tracked object is considered a separate class. They use cosine similarity between appearance feature vectors and bipartite matching to associate detections to tracks. They also use a Kalman filter to weed out detections that are too far out from predicted locations. The advantage of FairMOT is that it extends an already strong method to add a re-identification branch, hence improving its performance. It could potentially be improved by using a more sophisticated motion model than a Kalman filter.

The latest leader in high-performing online trackers is ByteTrack[103]. Most tracking-by-detection methods only consider detections with high confidence scores. Zhang et al. [103] argue that this results in a significant number of missed detections and lost tracks since partially occluded objects are often detected with lower confidence scores. They propose a method that considers all detections for tracking. It uses the Hungarian algorithm as an association method. Weights are computed as a weighted sum of bounding box IoU and re-ID feature similarity score. The main contribution of ByteTrack is its two-step online approach. Initially, it matches high-confidence detections to existing tracks, then it considers low-confidence detections, repeating the matching process with any unmatched tracks. This straightforward method set a new state-of-the-art performance record on the MOT17[3] benchmark. The authors also demonstrate how ByteTrack can be integrated with existing trackers, including CenterTrack and FairMOT,

enhancing their performance as well.

Offline Methods

Following the approach introduced by Zhang et al. [101], the authors of Lif_T[43] model the MOT problem as a network flow optimization problem. In the work of Zhang et al. [101] the graph is constructed with detections as nodes and potential associations between detections as edges. A trajectory is a path through the graph. An edge between detections represents that one detection follows the other in the trajectory. However, these detections don't have to be in consecutive frames since there could be missed detections and occlusions. There is a constraint that a detection can only belong to at most one trajectory. Hornakova et al. [43] argue that this formulation does not sufficiently model long-range dependencies and is not enough to produce high-quality long-term trajectories since edge costs only indicate whether the two detections follow each other. An example of long-range dependencies in the sports domain are frames that have a player's jersey number clearly visible. The frames don't have to follow each other but if the team and jersey number are the same, these detections should belong to the same track. The authors of Lif_T extend Zhang et al. [101] formulation by adding 'lifted edges' that connect detections that don't have to follow each other immediately in a track. These edges indicate similarity (or dissimilarity) between detections at a potentially large temporal distance that can encourage (or penalize) possible trajectories. The original optimization problem ([101]) minimizes the sum of the costs of edges that connect nodes that follow each other in a track and the cost of the connected nodes. The extended formulation ([43]) adds another term to the original optimization problem: the cost of lifted edges. In both [101] and [43] the flow variable that indicates whether two adjacent nodes are on the same trajectory is binary. Hornakova et al. [43] formulate a number of constraints and solve the problem using an integer programming solver.

The graph construction occurs in two steps. Initially, the graph includes detections within a short interval (2 seconds or less). The graph is made sparse by removing infeasible edges. After the initial graph is solved to form tracklets, they build on top of it a more extensive graph to link them. The weights of the edges are given by detection similarities. These similarities are calculated using various extracted features (such as re-ID network features, pixel correspondence features, etc.), which are then fused together with a separately trained network into a single

feature vector. The main downside of the method is that despite the multi-step approach it is computationally expensive.

MPNTrack [16] solves the data association problem in MOT (connecting detections into tracks) by formulating it as a graph flow problem based on the formulation proposed by Zhang et al. [101]. Re-ID features and geometric features are used to calculate edge weights. They use a graph neural network (GNN) and treat the problem as an edge classification problem. The GNN is trained to predict 'active' edges (those that connect nodes in a trajectory) using ground truth data. Building the graph based on all frames and all detections is of course too expensive. Therefore, they use a batch approach considering 15 frames at a time. That is a limited window to consider for re-id.

Cetintas et al. [18] use the MPNTrack [16] GNN approach to develop a unified model they call SUSHI that is scalable and generalizable, suitable for both short-term and long-term tracking. Instead of having a different approach and manual feature architecture for short-term and long-term tracking, they break up an image sequence into a hierarchy. They construct a graph with detections as nodes and employ a GNN to classify the edges. The resulting tracklets are then utilized to create a new graph, connecting them into longer tracklets, and so forth (see Figure 4.7). Unmatched tracklets persist and become the input nodes for the next hierarchical level, where they again have the opportunity to be associated with other tracklets over even longer temporal distances.

While the multi-stage approach has been employed in other works, SUSHI distinguishes itself by reusing the same architecture and weights across all levels. Additionally, SUSHI breaks down the image sequence into multiple levels, resulting in smaller graphs at each level, which enhances the system's speed and scalability. To avoid manual feature selection for each hierarchy level, each level has its own trained MLP layer to encode features (such as appearance and geometric features). This way the network is trained to select the most relevant features for each hierarchy level. Due to its speed and scalability, SUSHI is a step forward in offline long-term tracking. Weight-sharing and level-specific feature encoding allow for data-driven level-specific feature selection. The only potential downside of the hierarchical approach is that each level is solved separately, as opposed to a fully global approach. This means that mistakes made on connecting shorter tracklets will be propagated and still exist in the final tracking result.

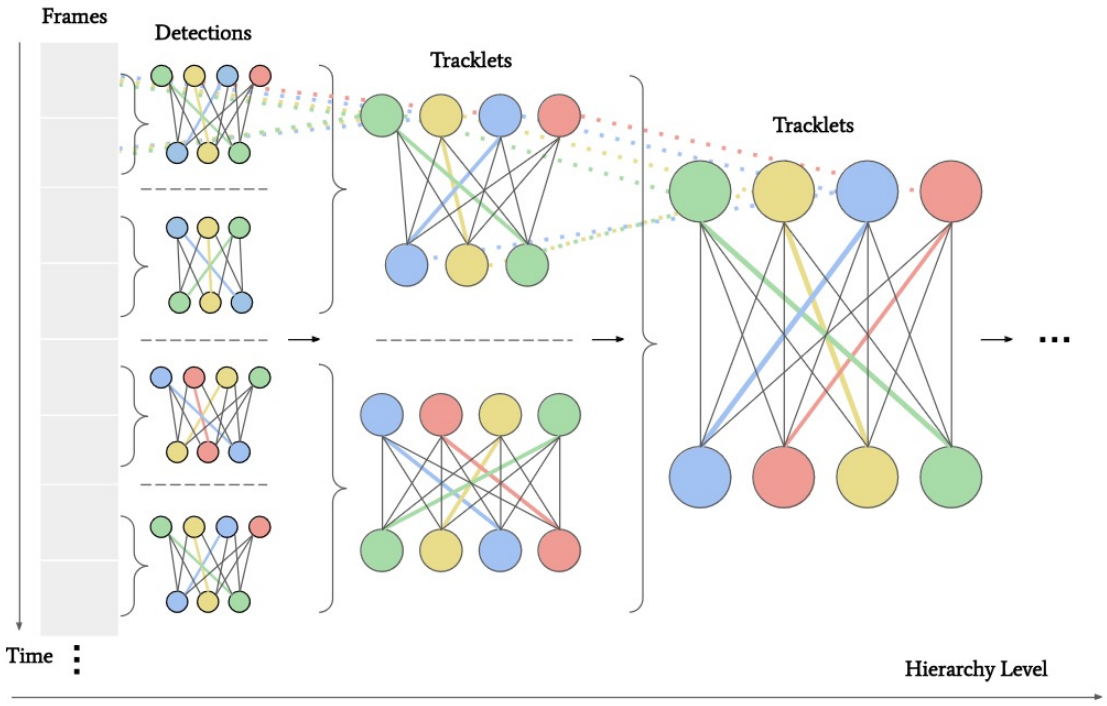

Figure 4.7: SUSHI: Hierarchy of tracking graphs [Image from [18]]

Performance and Discussion

In Table 4.1 and Table 4.2 we present the performance results of the MOT methods discussed in this section. Table 4.1 presents results on the MOT17 challenge with detections provided by the challenge. Some of the methods report tracking results with their own detections. These are listed in Table 4.2. As indicated in the tables, at the time of writing, SUSHI and ByteTrack outperform other trackers, with SUSHI showing superior performance in all metrics for public detections and the IDF1 metric for private detections.

The success of ByteTrack demonstrates that improved detections not only enhance association accuracy but also lead to better overall tracking performance. At the same time, SUSHI's high IDF1 score highlights the importance of a good long-term data association method. It should also be noted that while adapting object detectors to object trackers (such as Tracktor, CenterTrack, FairMOT) has yielded impressive results, this approach is limited to considering only a single frame or a pair of frames at a time. On the other hand, pure data association

Method	Type	Year	MOTA	IDF1	HOTA
Tracktor++[12]	online	2019	56.3	55.1	44.8
CenterTrack[106]	online	2020	61.5	59.6	48.2
MPNTrack[16]	offline	2020	58.8	61.7	49.0
Lif_T[43]	offline	2020	60.5	65.6	51.3
ByteTrack[103]	online	2022	67.4	70.0	56.1
SUSHI[18]	offline	2023	62.0	71.5	54.6

Table 4.1: Performance of popular MOT trackers on the MOT17 test set using public detections.

Method	Type	Year	MOTA	IDF1	HOTA
CenterTrack[106]	online	2020	67.8	64.7	56.2
FairMOT[104]	online	2020	73.7	72.3	59.3
ByteTrack[103]	online	2022	80.3	77.3	63.1
SUSHI[18]	offline	2023	81.1	83.1	66.5

Table 4.2: Performance of popular MOT trackers on MOT17 test set using private detections.

approaches, such as MPNTrack, Lif_T, and SUSHI, can establish longer-term associations but are susceptible to noisy detections. Effectively detecting and associating objects, particularly for long-term tracking, remains an open problem. Achieving optimal global tracking results efficiently remains a challenge, as a truly global association approach is still prohibitively computationally expensive for long-term tracking.

4.3.2 Player MOT

Most Multi-Object Tracking (MOT) methods focus on pedestrians or vehicles, while player tracking (or Multiple Athlete Tracking, MAT) remains less explored, largely due to its niche application and the historical lack of large public datasets. However, the recent introduction of SoccerNet [24] and SportsMOT [25] has spurred significant advancements in this field, as evidenced by a rise in research publications [67, 25, 45, 98, 69, 66, 46].

Player tracking leverages both appearance and kinematic features, as in MOT, but requires

additional sports-specific features like team IDs and jersey numbers due to the visual similarity among teammates [90, 102, 69]. Unlike pedestrians, players exhibit abrupt, non-linear movements [98, 45, 46, 66]. Pose extraction is also commonly used to enhance appearance features and manage occlusions [52, 51].

Zhang et al. [102] proposed a multi-camera tracking method incorporating team IDs, jersey numbers, and pose-guided feature extraction into a unified identity feature for track association. Their pipeline, trained separately for each game, demonstrated strong results on the APIDIS[1] dataset but required retraining for new games.

Yang et al. [98] introduced a Cascaded Buffered IoU (C-BIoU) tracker to handle rapid player movements. This method extends bounding box matching and improves short-term tracking, as seen in its performance on SoccerNet. Similarly, Huang et al. [46] combined ideas from ByteTrack and C-BIoU into Deep-EIoU, achieving superior short-term tracking results.

Lv et al. [66] addressed motion modeling by replacing the Kalman Filter with a Decoupled Diffusion Model, better suited for abrupt movements in sports. Their method, DiffMOT, achieved state-of-the-art results on SportsMOT but remains focused on short-term tracking.

Two of the most successful methods on SoccerNet dataset [24], Mansourian et al. [69] and Maglo et al. [67], both focus on improving player re-identification and using it to connect tracklets into longer tracks as a post-processing step. Maglo et al. [67] approach this by first running an existing short-term tracker, then using resulting tracklets to fine-tune the re-ID system in a self-supervised way to distinguish between players present in the game. They utilize contrastive learning, using detections that belong to the same tracklet as positives and detections that are captured in the same frame but belong to other tracklets as negatives. After fine-tuning the re-ID network, they use these new re-ID features to connect tracklets into longer tracks. Mansourian et al. [69], on the other hand, use an approach similar to that of Zhang et al.[102]. Their proposed tracker PRT-Track utilizes keypoint detection to build an appearance feature that is composed of appearance features from different body parts of the player. The resulting ID feature is less susceptible to occlusions. Maglo et al. [67] report best results on both SoccerNet[24] and SportsMOT[25]. This shows the importance of reliable re-ID feature and its role in long-term tracking. This approach, although effective, requires a lot of overhead as it includes training on every single test game separately at inference time.

Similarly to [69], our approach utilizes domain knowledge by extracting jersey number, team

id and re-ID feature. However, using hierarchical graphs provides a unified approach to both short-term and long-term tracking and does not require re-training on each new test game as does Maglo et al.[67].

4.4 Method

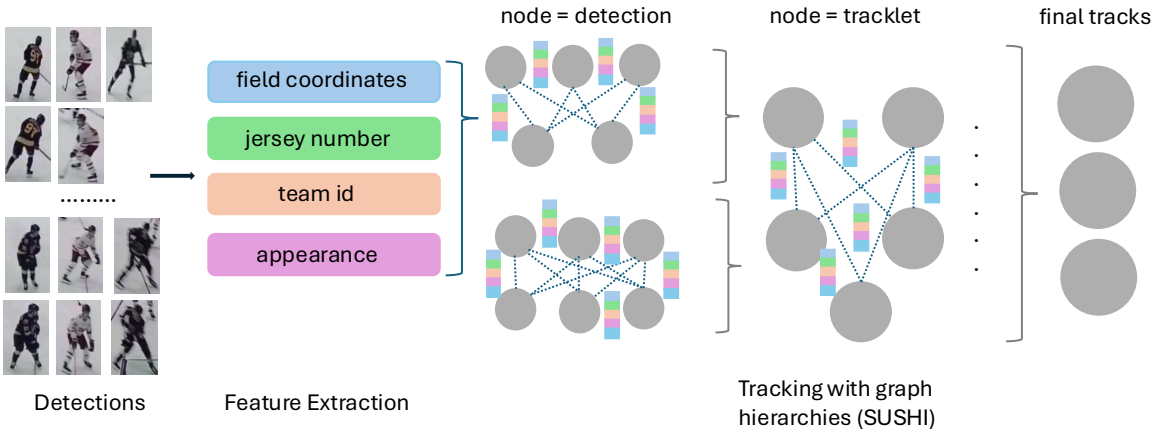

Figure 4.8: SportsSUSHI: we propose a player tracking system based on hierarchical tracker SUSHI[19]. Our feature extraction module, extracts features crucial for player tracking: jersey number, field coordinates, and team ID, in addition to classic re-ID features. The tracker then builds a hierarchy of graphs where each next level spans longer temporal durations. The initial graphs contain detections as nodes. Similarity measures between node features serve as the edge features. Each following layer uses the tracklets formed by solving the graph in the previous step as nodes.

Our tracking-by-detection approach takes detected bounding boxes of players and passes them through a feature-extraction module that predicts jersey numbers, field coordinates, team ID, and appearance features (Figure 4.8). The tracking is done offline by recursively building a hierarchy of graphs. The initial level contains detection as nodes. Edges between nodes are potential associations between detections on different frames. The initial step only considers consecutive frames. Edge features are based on the distance between detection features. The solution to the initial set of graphs yields short tracklets. These are then used to build a new set of graphs with tracklets as nodes and edges as associations between tracklets. These associations now span longer periods. The process continues for N levels. When the video clip is longer than

the longest temporal distance considered it is processed in a sliding window fashion with a simple stitching scheme as in the original method [18].

4.4.1 SportsSUSHI

We extend hierarchical graph tracker SUSHI[18] to support domain-specific features. The strength of SUSHI lies not only in considering long-term connections but in having an architecture that on one hand shares GNN weights between different levels but on the other learns separate feature encoders for each level. This allows the network to learn which features are more important at which temporal scale. At each level, once the edge features are encoded using an MLP the GNN is solved for the task of binary edge classification using neural message passing[16]. A linear program is then used to convert edge predictions into binary decisions and obtain final trajectories. For more information on SUSHI, refer to the original paper [18].

In the original SUSHI implementation, the authors use spatial features, such as the position and size of the bounding box, as well as the appearance (re-ID) feature. Unfortunately, these features are not sufficient for accurate player tracking. Based on our experiments we found that using sports-domain features improves tracking performance. Therefore, we introduce jersey number prediction and team ID to supplement appearance features. For broadcast video settings where the camera follows the play, comparing positions of bounding boxes between frames becomes unreliable, especially for longer timespans. To mitigate this, we use field registration as part of the pre-processing stage to estimate the homography matrix between the image and 2D playing field map. We then use projected player positions on the field as the spatial feature. In the following subsections, we discuss the feature extraction process in more detail.

4.4.2 Feature Extraction

Jersey Number Prediction

We make use of the jersey number recognition pipeline introduced in Chapter 3. The pipeline takes player detections as input and classifies them to identify player images with legible jersey numbers. For these player images, it uses pose estimation to select the torso area containing the number, before passing it to the scene text recognition network (STR) PARSeq[9]. Before decoding its prediction into a character string PARSeq produces a vector of confidences for each possible character value in each position. For a jersey number, $c_1 = [c_1(EOL), c_1(0), c_1(1)...c_1(9)]$

and $c_2 = [c_2(EOL), c_2(0), c_2(1)...c_2(9)]$ where EOL is an end-of-line character and $c_i(d)$ is a confidence of having digit d in position i. We can mark any images with predictions of EOL in the first position as illegible. For the rest, we calculate $c(d_i d_j) = c_1(d_i)c_2(d_j)$ for two-digit numbers and $c(d_i) = c_1(d_i)c_2(EOL)$ for one-digit numbers. The resulting 100-value vector encodes jersey number prediction.

Team ID Prediction

Our experiments show that the appearance feature is not sufficient for some datasets to distinguish between players on different teams. To address that we also include the team ID feature. The team ID feature in our system is a one-hot vector representing a prediction of whether the player belongs to team A, team B, or referee. We use the method proposed in Chapter 2 to first classify a person as a referee or a player, then extract an image embedding for each player using a previously trained embedding network. The first N=10,000 player images of each clip are then clustered to learn models for the appearance of Team A and Team B for the given game. All of the player images are then assigned a label based on which cluster center is closest. The team ID feature is a one-hot vector of length 3.

Field Coordinates

In typical tracking scenarios, the most powerful feature for association is object position, since object displacement is small for typical frame rates. However, this assumption is less reliable for broadcast videos, where sudden camera movements can cause the position of the player in the camera frame to jump. Sports field registration methods can potentially address this complication through online estimation of the homography relating camera pixels to field position. Several proposed methods use convolutional neural networks or transformers to find required key points and/or line segments in the frame [35, 67, 85]. Typically, RANSAC [29] is then used to estimate the homography matrix. In our work, we use the system proposed by Gutiérrez-Pérez et al. [35]. They use a convolutional network encoder/decoder architecture to detect key points and line segments on the field and predict each frame's frame-to-field homography matrix. After learning each frame's homography matrix we use it to project a middle point of the bottom of each detection bounding box in the frame (in pixels) to a position on a field (in meters). Field registration sometimes fails to detect key points in the frame. This typically happens when a

large zoom and camera angle don't capture enough field markings or when the camera motion results in the blurring of lines. To address this, we use linear interpolation of predicted camera parameters to approximate parameters for failed frames.

Person Re-ID

For sports such as soccer, visual appearance provides a lot of useful information despite the similarity of uniforms of players on the same team. As in the original SUSHI method, we extract the appearance feature vector using a person re-identification model FastReID [40] with ResNet50-IBN backbone [40] originally trained on the Market1501 dataset [105]. We fine-tune the model on the specific dataset being considered.

Edge Features

For each edge connecting two detections (or two tracklets) we calculate the edge feature vector. Each value of the vector corresponds to a measure of similarity between these two nodes in the given feature space (re-ID, jersey number, etc). We use cosine similarity for appearance features, jersey number, team ID, and absolute value for distance in meters of field coordinates. For nodes representing tracklets, appearance features are computed by taking a mean of individual detection values. For position, the distance between the last detection of the first tracklet and the first detection of the second tracklet is computed. An MLP (one for each hierarchy level) is then used to encode the features. To control the complexity of the graph, Cetintas et al. [18] employ edge-pruning to only consider the K closest nodes based on appearance and spatial features. We follow this approach as well with K=10.

4.5 Datasets

There are several player tracking datasets that became available in the last few years: SoccerNet[24], SportsMOT[25], and MHPTD[2]. All of these contain clips from broadcast videos. To explore longer-term tracking we introduce a new hockey tracking dataset. The dataset is unique in capturing the whole playing surface. We evaluate SportsSUSHI on our hockey dataset and the publicly-available SoccerNet[24] tracking dataset.

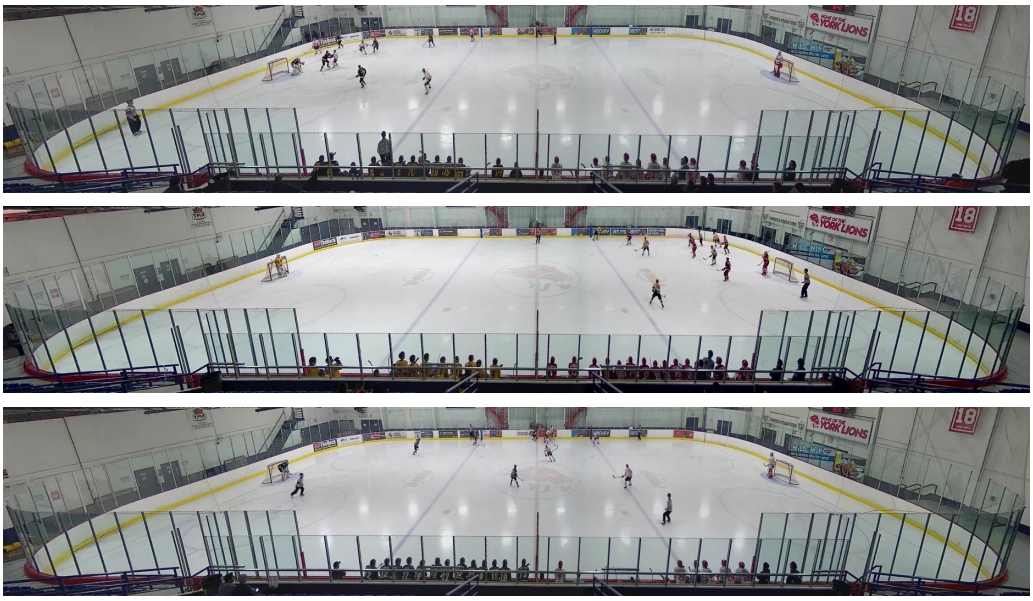

Figure 4.9: Sample frames from our hockey dataset.

4.5.1 Hockey

Recorded at university hockey games with a stationary camera at 30fps, the dataset contains 20 clips from 9 different games (Figure 4.9). We split it into a training set of 14 clips from 6 games and a test set of 6 clips from the remaining 3 games. The average clip length is 1311 frames and the longest is 1530 frames (Figure 4.10 shows the sequence length distribution).

The dataset contains ground truth in standard MOT annotation. In most MOT datasets when the person is occluded there are no bounding box annotations for them. T4o investigate tracking a person the whole time they are on the playing surface, we estimated their position through occlusion and included this in the annotation.

The annotation process was a two-step approach:

- Weak tracking ground truth was generated by annotators following each player in turn with a mouse pointer throughout the clip. Bounding boxes from an object detector were then associated with each such path, producing a set of imperfect tracks.

- These tracks were then manually corrected and refined to ensure accuracy. Any missing bounding boxes were added and any inaccurate bounding boxes were adjusted. We used the CVAT [6] annotation tool for this step. During this stage, jersey numbers (whenever visible) and team ID labels were added to the annotation.

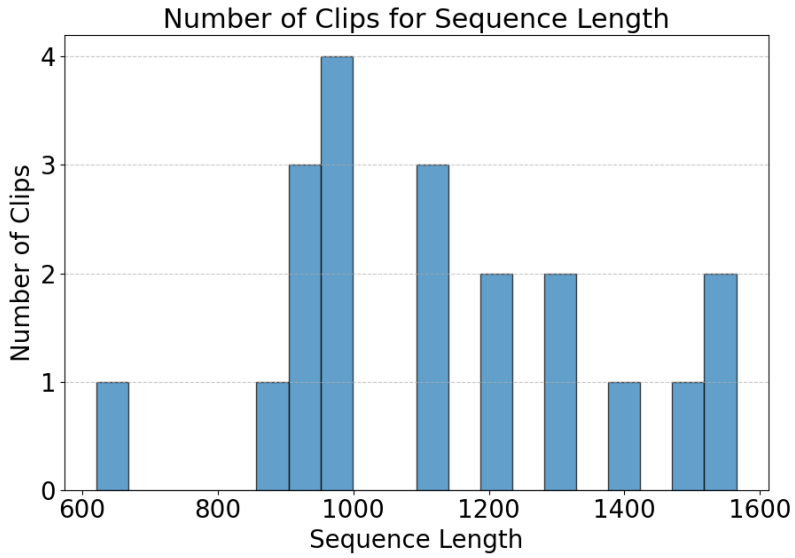

Figure 4.10: Hockey dataset sequence length.

Table 4.3 summarizes some statistics of the two datasets used.

4.5.2 SoccerNet

The SoccerNet[24] player tracking dataset introduced in 2022 was the first large-scale publicly available player tracking dataset. Its introduction and the challenge conducted spurred interest in the area. SoccerNet contains 200 clips, spread into train, test and challenge sets. The clips are from broadcast soccer games and contain a lot of challenging scenarios: camera motion, fast player movement, occlusions, and players leaving and re-entering a field of view. The latter challenges the ability of the tracker to re-identify the player and reconnect parts of the track after long intervals.

Dataset	Train	Test	Avg Length (frames)	Frame Rate (fps)	Dimensions (pixels)
Hockey (ours)	14	6	1,131	30	5930 x 1080
SoccerNet[24]	57	49	750	25	1920 x 1080

Table 4.3: Dataset information.

4.6 Experiments and Results

4.6.1 Re-ID Performance Analysis

We investigate how reliable the similarity of re-ID features is at different time gaps. To do that we take a subset of the test set and analyze the accuracy of matching the identity of the player in frame i to that same player in frame $i + N$ for different N. We can see from the results in Table 4.4 that re-identification does predictably worse at large time intervals. This is expected since pose, lighting, and position to the camera tend to be similar in the images closer together in time. The consecutive images (N=1) are almost identical. Once more time lapses, player location and pose change and personal appearance features become more important in distinguishing players. We can also observe that performance at longer intervals is higher for soccer than for hockey, because appearance cues such as facial features and hair color are more visible. These findings motivate the inclusion of domain-specific features such as jersey number and team ID.

	Soccer		Hockey	
Interval (frames)	Original	Soccer	Original	Hockey
1	98.0%	99.1%	99.1%	99.63%
50	51.2%	79.1%	32.5%	38.2%
100	35.7%	72.2%	25.5%	26.8%
300	33.0%	62.2%	29.9%	26.6%

Table 4.4: FastReID performance for matching player identities at varying frame intervals using cosine similarity of feature vectors. Accuracy is compared between the Market1501-trained model and its fine-tuned version for each dataset.

4.6.2 Implementation

For the jersey number recognition framework and team ID we use the methods described in Chapter 3 and 2 respectively. For the field registration we use the code and model weights from [35].

We train SportsSUSHI on the training partition of the player tracking dataset following the

training protocol outlined in Cetintas et al. [18], training earlier levels first for 500 iterations before unfreezing the next level, then continuing to train jointly. The model is trained with the Adam optimizer for 250 epochs. The code and data are available at `https://github.com/mkoshkina/sports-SUSHI`.

We evaluate tracking results using key standard HOTA[63] metrics: Detection Accuracy, Association Accuracy, and HOTA. Since our goal is reliable long-term tracking we are most interested in Association Accuracy and HOTA metrics. Note, that detection accuracy in tracking evaluation considers only detections included in predicted tracks. This means that even when evaluated on ground truth detections, tracking detection accuracy is usually less than 100%.

4.6.3 Soccer

The best results on the SoccerNet dataset were achieved by combining field coordinates, re-ID appearance, and jersey number features. Our SportsSUSHI tracker shows competitive performance compared to earlier published methods. Table 4.5 presents the results on both ground truth detections and YOLOX detections [31]. The current state-of-the-art approach by Maglo et al. [67] involves fine-tuning the reidentification network on each new game at inference time. Ours is the most performant approach that does not require inference-time tuning.

We show the contribution of various features to performance in Table 4.6. We show that using field coordinates instead of frame pixel coordinates introduces a significant performance improvement. The introduction of jersey numbers provides a further performance boost.

In this work, we do not consider team identification for the soccer dataset. Our current approach to team ID relies on the assumption that referees wear consistent uniforms across games, making them easily distinguishable using a binary classifier. This assumption does not hold for the soccer dataset, where referee appearance varies significantly. As such, we leave the development of a team ID system for soccer to future work. Additionally, we believe that the re-identification feature already provides a strong cue for both individual and team identity. Therefore, we do not expect substantial performance gains from incorporating team ID for soccer, though a more detailed analysis is also left for future work.

We explore the effect of the number of layers in the graph hierarchy. With 7 layers the furthest time gap in a track considered for reconnection is 256, for 9 layers it is 512 and for 10 layers it is 1024. The performance improves with additional layers since the model gets to

Method	HOTA	AssA	DetA
	Ground Truth Detections		
Maglo et al. [67]	**96.57**	**93.60**	99.65
Mansourian et al. [69]	90.77	82.53	**99.83**
Yang et al. [98]	89.2	80.00	99.4
Huang et al. [45]	85.44	73.57	99.24
SUSHI [18]	85.37	79.36	92.20
SportsSUSHI (ours)	<u>90.92</u>	<u>87.51</u>	94.80
	YOLOX Detections		
Maglo et al. [67]	**73.29**	**73.42**	**73.26**
Mansourian et al. [69]	59.77	58.55	61.09
ByteTrack [103]	60.56	52.45	70.10
SportsSUSHI (ours)	<u>71.36</u>	<u>69.99</u>	72.87

Table 4.5: Results on SoccerNet [18] test partition.

consider longer timespans. Note, that the total length of SoccerNet clips is 750 frames, so there is no need to consider more than 10 layers.

4.6.4 Hockey

In our hockey dataset, the camera is stationary and captures the whole rink. Therefore, using the original spatial feature (position of the player in the frame and bounding box size) is sufficient and we don't need to calculate the player's position on the rink. We use the jersey number, team ID, appearance, position in the image, and bounding box size. We show results on our test set in Table 4.7. All of the baselines we compare with have been fine-tuned on the hockey dataset. We use YOLOX[31] detections in our method. Since Maglo et al. [67] and Mansourian et al. [69] do not share their source code, we use several well-known MOT approaches for comparison. All of the baselines we compare with have been fine-tuned on the hockey dataset. SportsSUSHI shows superior results compared to previous methods.

Method	HOTA	AssA	DetA
	Ground Truth Detections		
SUSHI (9 layers)	85.37	79.36	92.20
SportsSUSHI (field position, 9 layers)	89.22	84.25	94.82
SportsSUSHI (field position, jersey numbers, 7 layers)	87.44	80.92	**94.87**
SportsSUSHI (field position, jersey numbers, 9 layers)	89.78	85.35	94.80
SportsSUSHI (field position, jersey numbers, 10 layers)	**90.92**	**87.51**	94.80

Table 4.6: Abblations on SoccerNet [18] test partition.

Method	HOTA	AssA	DetA
Tracktor++ [12]	44.33	34.09	58.08
CenterTrack [106]	59.70	48.96	**72.90**
FairMOT [104]	61.56	55.51	68.35
ByteTrack [103]	67.51	64.63	71.36
SUSHI [18]	69.51	67.47	72.51
SportsSUSHI (ours)	**71.24**	**72.82**	70.47

Table 4.7: Results on hockey dataset test partition.

4.6.5 Failure Cases

Considering longer timespans to re-connect tracklets and employing jersey IDs for re-identifying players yield promising results in player tracking. However, jersey numbers are not always visible or are often too blurry to decipher. Moreover, they offer no additional cues when tracking referees, who all wear the same uniform and are difficult to re-identify. The association errors observed in the hockey and soccer datasets primarily arise from failures to re-identify individuals after extended absences from the field of view or during occlusions (See Figure 4.11). A potential next step could involve incorporating a better motion model to predict player movements, which would enhance the utility of spatial features when appearance-based features are insufficient. We leave this to future work.

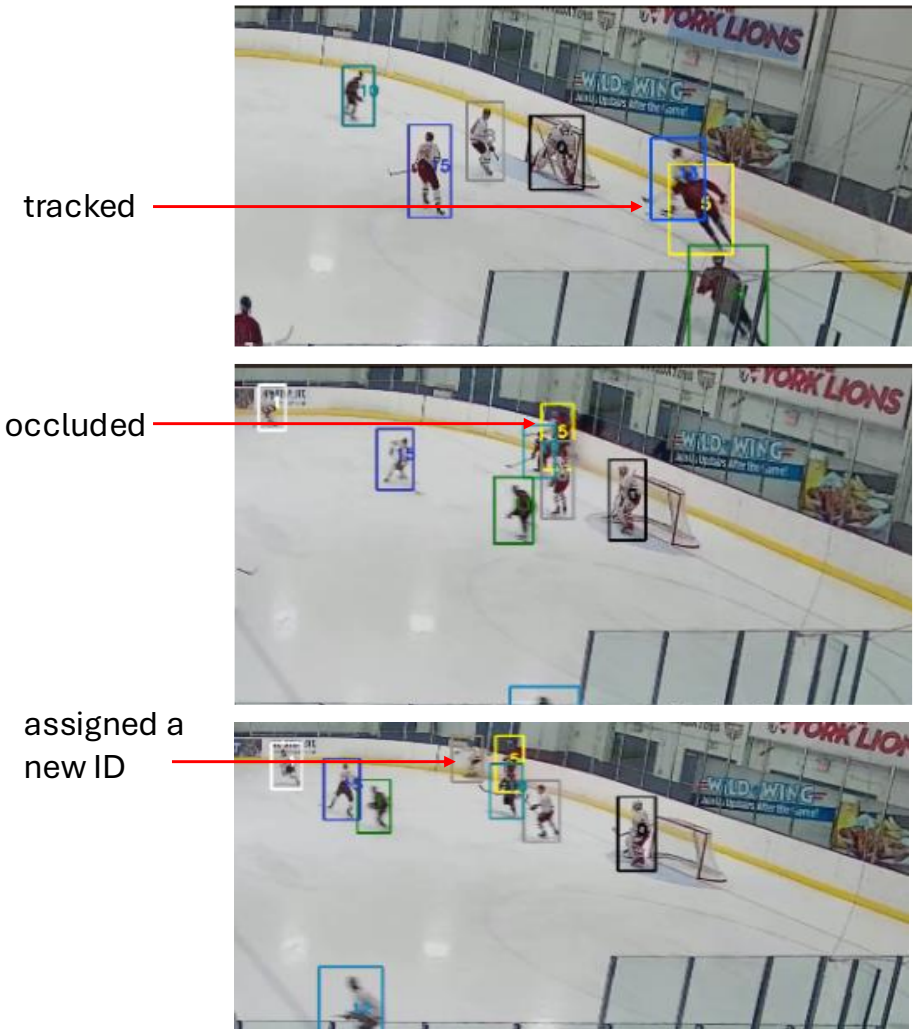

Figure 4.11: Example of tracking failure. Top image: before the occlusion. Middle image: the player is fully occluded and its track is interrupted. Bottom image: after the occlusion the player gets assigned a new ID (ID switch).

Chapter 5

Conclusion

In this dissertation, we presented a unified suite of algorithms that extract rich, identity-based information about players in team sports—information that is crucial for automatic game analysis. The core components of our framework include self-supervised team identification, robust jersey number recognition, and graph-based player tracking. A central theme of this work is the development of methods that generalize well to new players, new teams, and new, previously unseen games. In addition, we demonstrated cross-domain generalization using both a public soccer dataset and a new university hockey dataset introduced as part of this research.

Throughout the dissertation, we adopted a problem decomposition strategy that simplifies complex tasks by dividing them into smaller, more manageable subproblems. For example, in team identification, we first identify referees to simplify the player classification process; in jersey number recognition, we first filter for instances where jersey numbers are visible and readable. This design improves robustness and enables more effective handling of the diverse conditions present in real-world sports video.

To support further research in this area, we introduced a new dataset derived from university hockey games. This dataset contains annotations for player tracking, team affiliation, and jersey numbers, and serves as a valuable benchmark for evaluating identity-aware tracking systems.

In future work, we envision extending this framework to a broader range of sports, exploring more advanced motion models to better capture player dynamics, and incorporating domain-specific constraints. These extensions will further enhance the adaptability and performance of the system in diverse team sports environments.

At the same time, it is important to reflect on the ethical dimension of this research. Player

tracking technologies can, in general, raise concerns if used for surveillance or monitoring people without their consent. In this dissertation, however, the focus is strictly on the sports setting, where tracking is applied to public game recordings for the purpose of advancing analytics and performance understanding. All datasets used were either publicly available or collected with appropriate approval. By keeping the scope limited to team sports, this work aims to support fair and beneficial applications, while acknowledging that similar methods should be used with care in other domains.

In summary, this dissertation contributes three key capabilities essential to most sports analytics applications: team identification, jersey number recognition, and player tracking. Each component addresses a fundamental aspect of understanding and analyzing team sports from video. Together, they provide a foundation for scalable, accurate, and generalizable automatic sports video analysis across multiple domains.

Bibliography

[1] APIDIS basketball dataset. https://ispgroup.gitlab.io/code/apidis/. Accessed: September, 2025. 9, 55

[2] McGill Hockey Player Tracking Dataset (MHPTD). https://github.com/grant81/hockeyTrackingDataset. Accessed: September, 2025. 23, 27, 59

[3] MOT benchmark. https://motchallenge.net/. Accessed: September, 2025. 2, 39, 41, 50

[4] SoccerNet Jersey Number Recogntion. https://www.soccer-net.org/tasks/jersey-number-recognition. Accessed: September, 2025. 23, 27

[5] VOT challenge. https://www.votchallenge.net. Accessed: September, 2025. 39

[6] Computer Vision Annotation Tool (CVAT). https://github.com/cvat-ai/cvat, 2023. Accessed: September, 2025. 60

[7] Waleed Abdulla. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask_RCNN, 2017. Accessed: September, 2025. 15

[8] Bavesh Balaji, Jerrin Bright, Harish Prakash, Yuhao Chen, David A Clausi, and John Zelek. Jersey number recognition using keyframe identification from low-resolution broadcast videos. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, pages 123–130, 2023. viii, 23, 24, 36

[9] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196, Cham, 10 2022. Springer Nature Switzerland. 25, 26, 31, 34, 36, 57

[10] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *2011 International Conference on Computer Vision*, pages 137–144, 2011. 5, 7

[11] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011. 48

[12] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019. 49, 54, 65

[13] Divya Bhargavi, Sia Gholami, and Erika Pelaez Coyotl. Jersey number detection using synthetic data in a low-data regime. *Frontiers in Artificial Intelligence*, 5:988113, 2022. 23, 24, 34

[14] Alina Bialkowski, Patrick Lucey, Peter Carr, Sridha Sridharan, and Iain Matthews. Representing team behaviours from noisy data using player role. In *Computer Vision in Sports*, pages 247–269. Springer, 2014. 4, 5, 7, 15

[15] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9157–9166, 2019. 21

[16] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020. 48, 52, 54, 57

[17] Kai Briechle and Uwe D Hanebeck. Template matching using fast normalized cross correlation. In *Optical Pattern Recognition XII*, volume 4387, pages 95–102. International Society for Optics and Photonics, 2001. 39

[18] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22877–22887, 2023. viii, x, 52, 53, 54, 57, 59, 63, 64, 65

[19] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22877–22887, June 2023. x, 39, 56

[20] Alvin Chan, Martin D Levine, and Mehrsan Javan. Player identification in hockey broadcast videos. *Expert Systems with Applications*, 165:113891, 2021. 23, 24, 25

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning*, pages 1597–1607, 2020. 7

[22] Wongun Choi and Silvio Savarese. A unified framework for multi-target tracking and collective activity recognition. In *12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pages 215–230. Springer, 2012. 48

[23] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Scaling up SoccerNet with multi-view spatial localization and re-identification. *Scientific Data*, 9, June 2022. 23, 26, 27

[24] Anthony Cioppa, Silvio Giancola, Adrien Deliege, Le Kang, Xin Zhou, Zhiyu Cheng, Bernard Ghanem, and Marc Van Droogenbroeck. SoccerNet-Tracking: Multiple object tracking dataset and benchmark in soccer videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3491–3502, 2022. 54, 55, 59, 61

[25] Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. Sportsmot: A large multi-object tracking dataset in multiple sports scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9921–9931, 2023. 2, 9, 54, 55, 59

[26] Tiziana D'Orazio, Marco Leo, Paolo Spagnolo, Pier Luigi Mazzeo, Nicola Mosca, Massimiliano Nitti, and Arcangelo Distante. An investigation into the feasibility of real-time soccer offside detection from a multiple camera system. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(12):1804–1818, 2009. 4, 5, 7, 15

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 34, 35

[28] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. CenterNet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019. 49

[29] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 58

[30] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 30

[31] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint 2107.08430*, 2021. 63, 64

[32] Sebastian Gerke, Karsten Muller, and Ralf Schafer. Soccer jersey number recognition using convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 17–24, 2015. 23, 36

[33] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2, 9

[34] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016. 31

[35] Marc Gutiérrez-Pérez and Antonio Agudo. No bells just whistles: Sports field registration by leveraging geometric properties. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3325–3334, 2024. 58, 62

[36] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006. 7

[37] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 7

[38] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 8, 10, 20, 23

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 26, 30, 34, 35

[40] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. FastReID: A pytorch toolbox for general instance re-identification. *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9664–9667, 2023. 59

[41] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2014. 39

[42] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015. 7, 11

[43] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, pages 4364–4375. PMLR, 2020. 48, 51, 54

[44] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II 10*, pages 788–801. Springer, 2008. 48

[45] Hsiang-Wei Huang, Cheng-Yen Yang, Samartha Ramkumar, Chung-I Huang, Jenq-Neng Hwang, Pyong-Kun Kim, Kyoungoh Lee, and Kwangju Kim. Observation centric and central distance recovery for athlete tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 454–460, 2023. 54, 55, 64

[46] Hsiang-Wei Huang, Cheng-Yen Yang, Jiacheng Sun, Pyong-Kun Kim, Kwang-Ju Kim, Kyoungoh Lee, Chung-I Huang, and Jenq-Neng Hwang. Iterative scale-up expansioniou and deep features association for multi-object tracking in sports. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 163–172, 2024. 54, 55

[47] Maxime Istasse, Julien Moreau, and Christophe De Vleeschouwer. Associative embedding for team discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 4, 7, 16

[48] Zdravko Ivankovic, Milos Rackovic, and Miodrag Ivkovic. Automatic player position detection in basketball games. *Multimedia Tools and Applications*, 72(3):2741–2767, 2014. 4, 7, 15

[49] Hao Jiang, Sidney Fels, and James J Little. A linear programming approach for multiple object tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 48

[50] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 48

[51] Longteng Kong, Di Huang, and Yunhong Wang. Long-term action dependence-based hierarchical deep association for multi-athlete tracking in sports videos. *IEEE Transactions on Image Processing*, 29:7957–7969, 2020. 55

[52] Longteng Kong, Mengxiao Zhu, Nan Ran, Qingjie Liu, and Rui He. Online multiple athlete tracking with pose-based long-term temporal dependencies. *Sensors*, 21(1):197, 2020. 55

[53] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 42

[54] Laura Leal-Taixé. Multiple object tracking with context awareness. *arXiv preprint arXiv:1411.7935*, 2014. x, 40

[55] Laura Leal-Taixe, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, April 2015. arXiv: 1504.01942. 41

[56] Gen Li, Shikun Xu, Xiang Liu, Lei Li, and Changhu Wang. Jersey number recognition with semi-supervised spatial transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1783–1790, 2018. 23, 34, 36

[57] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 7, 10, 30

[58] Hengyue Liu and Bir Bhanu. Pose-guided R-CNN for jersey number recognition in sports. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 23, 25, 34

[59] Hengyue Liu and Bir Bhanu. JEDE: Universal jersey number detector for sports. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7894–7909, 2022. 23, 25

[60] Jingchen Liu and Peter Carr. Detecting and tracking sports players with random forests and context-conditioned motion models. In *Computer Vision in Sports*, pages 113–132. Springer, 2014. 4, 5, 7, 15

[61] Keyu Lu, Jianhui Chen, James J Little, and Hangen He. Lightweight convolutional neural networks for player detection and classification. *Computer Vision and Image Understanding*, 172:77–87, 2018. 4, 7, 16

[62] Wei-Lwun Lu, Jo-Anne Ting, James J Little, and Kevin P Murphy. Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1704–1716, 2013. 4, 5, 7, 15

[63] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, 2021. x, 41, 44, 45, 46, 47, 63

[64] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, page 103448, 2020. 40

[65] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996. 13

[66] Weiyi Lv, Yuhang Huang, Ning Zhang, Ruei-Sung Lin, Mei Han, and Dan Zeng. DiffMOT: A real-time diffusion-based multiple object tracker with non-linear prediction. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19321–19330, 2024. 54, 55

[67] Adrien Maglo, Astrid Orcesi, Julien Denize, and Quoc Cuong Pham. Individual locating of soccer players from a single moving view. *Sensors*, 23(18):7938, 2023. 54, 55, 56, 58, 63, 64

[68] Santiago Manen, Michael Gygli, Dengxin Dai, and Luc Van Gool. Pathtrack: Fast trajectory annotation with path supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 290–299, 2017. 41

[69] Amir M Mansourian, Vladimir Somers, Christophe De Vleeschouwer, and Shohreh Kasaei. Multi-task learning for joint re-identification, team affiliation, and role classification for sports visual tracking. In *Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports*, pages 103–112, 2023. 54, 55, 64

[70] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011. 16

[71] Pier Luigi Mazzeo, Paolo Spagnolo, Marco Leo, and Tiziana D'Orazio. Football players classification in a multi-camera environment. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 143–154. Springer, 2010. 4, 7, 15

[72] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016. arXiv: 1603.00831. 41

[73] Emanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 09 1962. 15, 19

[74] Donald Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. 48

[75] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 23, 49

[76] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016. 41, 44

[77] Mikel Rodriguez, Josef Sivic, Ivan Laptev, and Jean-Yves Audibert. Data-driven crowd analysis in videos. In *2011 International Conference on Computer Vision*, pages 1235–1242. IEEE, 2011. 48

[78] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27(3):832–837, 09 1956. 15, 19

[79] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 7, 26, 30

[80] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986. 19

[81] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8802–8812, 2021. 31

[82] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, John Garofolo, Djamel Mostefa, and Padmanabhan Soundararajan. The CLEAR 2006 evaluation. In *International Evaluation Workshop on Classification of Events, Activities and Relationships*, pages 1–44. Springer, 2006. 41, 42

[83] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010. 39

[84] Demetri Terzopoulos and Richard Szeliski. Tracking with kalman snakes. *Active Vision*, 20:3–20, 1992. 39

[85] Jonas Theiner and Ralph Ewerth. Camera calibration for sports field registration in soccer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1166–1175, 2023. 58

[86] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. *International Journal of Computer Vision*, 9:137–154, 1991. 39

[87] Xiaofeng Tong, Jia Liu, Tao Wang, and Yimin Zhang. Automatic player labeling, tracking and field registration and trajectory mapping in broadcast soccer video. *ACM Transactions on Intelligent Systems and Technology*, 2(2):1–32, 2011. 4, 5, 7, 15

[88] Kanav Vats, Mehrnaz Fani, David A Clausi, and John Zelek. Multi-task learning for jersey number recognition in ice hockey. In *Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports*, pages 11–15, 2021. 23, 24, 25, 34, 36

[89] Kanav Vats, William McNally, Pascale Walters, David A Clausi, and John S Zelek. Ice hockey player identification via transformers and weakly supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3451–3460, 2022. 23, 24, 36

[90] Kanav Vats, Pascale Walters, Mehrnaz Fani, David A Clausi, and John S Zelek. Player tracking and identification in ice hockey. *Expert Systems with Applications*, 213:119250, 2023. 23, 24, 25, 55

[91] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. COCO-Text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 31

[92] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer GAN to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018. 2

[93] Mikolaj Wieczorek, Barbara Rychalska, and Jacek Dabrowski. On the unreasonable effectiveness of centroids in image retrieval. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part IV 28*, pages 212–223. Springer, 2021. 32

[94] Bo Wu and Ram Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 951–958. IEEE, 2006. 41

[95] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75:247–266, 2007. 48

[96] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022. 30

[97] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning*, pages 3861–3870. PMLR, 2017. 8

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

space. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4799–4808, 2023. 54, 55, 64

[99] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016. 8

[100] Kua Chen Yingnan Zhao, Zihui Li. A method for tracking hockey players by exploiting multiple detections and omni-scale appearance features. *Project Report*, 2020. 23, 27

[101] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 48, 51, 52

[102] Ruiheng Zhang, Lingxiang Wu, Yukun Yang, Wanneng Wu, Yueqiang Chen, and Min Xu. Multi-camera multi-player tracking with deep player identification in sports video. *Pattern Recognition*, 102:107260, 2020. 55

[103] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, pages 1–21. Springer, 2022. 50, 54, 64, 65

[104] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021. 50, 54, 65

[105] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015. 2, 32, 59

[106] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. x, 49, 50, 54, 65