Unsupervised Player Tracking in Sport Videos

Xiaoyi Liu[1] and Diego Klabjan[2]

[1] Northwestern University, Mechanical Engineering, Evanston IL 60208, USA
xiaoyiliu2021@u.northwestern.edu
[2] Northwestern University, Industrial Engineering and Management Sciences, Evanston IL 60208, USA d-klabjan@northwestern.edu

Abstract. Computer vision techniques have found widespread use in the realm of sports. With the growth of big data, it has become increasingly important for both coaches and broadcasters to analyze detailed player performance data in sports such as football and soccer. This includes player movements and poses, as well as team formations. A fully automated tracking and identification system would provide valuable insights towards these goals. However, this is a challenging task due to 1) the players' similar visual features caused by their jerseys, helmets, etc., if done unsupervised; and 2) the cost of annotating player tracking in videos for each sport if supervised.
To overcome the challenges of sport player localization and tracking with a moving camera, we present a novel unsupervised method that consists of three main steps. First, the method learns a transformation from a moving camera to a static camera to stabilize the frames. Second, it learns to localize sport players from frame subtraction by leveraging the difference between static background and moving objects. Finally, the method learns the object center displacement between consecutive frames to achieve continuous tracking.
The experimental results of our proposed method show that it performs as well as supervised models without fine-tuned detectors, with an improvement of more than 0.9% on the HOTA metric for soccer videos and 3.1% on the MOTA metric for NFL videos with respect to other unsupervised methods. These results demonstrate the effectiveness of our unsupervised approach and its potential to enhance sport player tracking in difficult scenarios.

Keywords: Object Detection, Object Tracking, Unsupervised learning, Sports Video Analysis

1 Introduction

The use of rapidly advancing scientific technologies have made modern sports more intriguing, have improved athletes' performance, and have provided an advantage to innovation. Despite this progress, there is still significant room for improvement in many aspects of sports. Sport analysts and coaches are constantly seeking an automated way to collect player tracking information, which can help them analyze and improve player performance. An automatic player

tracking system would also benefit the entertainment industry, offering audiences a more professional viewing experience.

Wearable technology has become the most common solution to this problem. Many hardware companies have developed GPS (Global Positioning System) and IMU (Inertial Measurement Unit) based systems that players must wear. Efforts have been made to reduce the weight and size of these wearable units, but they still present challenges such as battery life, comfort, and high cost.

Multiple Object Tracking (MOT), a case of sport player tracking, has gained significant attention due to the impact deep learning has had on computer vision. More and more models based on the detection-tracking logic have been proposed. Applying MOT methods to sports videos provides information on player positions and trajectories.

However, training MOT methods requires extensive and costly labeling efforts, leading to a lack of publicly available datasets for sport player tracking. Meanwhile, the amount of sports videos continues to grow rapidly. An unsupervised learning method is desired to take advantage of this abundance of data.

Our goal is to develop an automatic sport player tracking framework that does not require any manual annotations. There are several challenges associated with this task: 1) the camera view in most sports videos is often moving, 2) players have similar visual features due to their uniforms, helmets, etc., 3) occlusions are more common than in general videos, and 4) players often have extreme poses.

The primary focus of our work is to tackle these challenges in sport player tracking using an unsupervised system. The proposed system consists of four critical components, each addressing a specific aspect of the problem.

The first component of our system is designed to align consecutive video frames accurately. The goal of this component is to ensure that each frame is in the similar camera view relative to the previous frame. Proper frame alignment is essential for accurate player tracking and object segmentation, which are the subsequent steps in our system.

The second component of our system is responsible for segmenting all moving objects in the video frames. This component identifies all the objects that are moving within each frame, including players, ball, and other dynamic elements. This segmentation process is essential to ensure that the subsequent steps in our system only focus on relevant objects and not on any static or irrelevant elements.

Our third component is aimed at distinguishing players from all other moving objects identified in the previous step. The component leverages clustering methods to separate all objects and trains a contrastive learning model to differentiate players from other objects. The result is a set of players identified in each frame, which serves as input for the subsequent tracking step.

The fourth component of our system is an unsupervised tracking model that treats each object as a circle with height and width on the frame's heatmap. It predicts the displacement of centers using heads and is trained using continuous and conservative assumptions, without any human annotations.

Our contributions to the field of sport player tracking are:

1. The development of a novel unsupervised multiple object tracking framework that is specifically designed for sports videos. As such our work is the first fully unsupervised approach for player tracking in sports videos.

2. The introduction of an unsupervised learning component that aligns video frames, which enables us to overcome the challenge of camera movement in sports videos.

3. The introduction of a new segmentation method to detection, which has not been done before.

4. The evaluation of our proposed framework on a soccer video dataset and an NFL video dataset through experiments and comparison with state-of-the-art methods.

The paper is organized as follows. Section 2 reviews related work in the field, Section 3 presents the proposed detection and tracking method and its components, Section 4 discusses the training strategies, datasets, and experimental results, and finally, conclusions and future plans are outlined in Section 5.

2 Literature Review

In this section, we begin with a review of visual object localization techniques, followed by a comprehensive overview of the main methods and frameworks proposed for MOT, as well as contrastive learning, and sports video analysis. Our emphasis is on unsupervised approaches.

2.1 Visual Object Localization

Localizing sport players within video frames is a crucial step towards achieving a sport player tracking system. This task falls under the category of visual object localization, which is a critical issue in computer vision. Supervised versions of visual object localization have been extensively studied [13, 27, 28, 40, 42], but these approaches require large amounts of annotated training data, which can be expensive to obtain. Moreover, these models are vulnerable to domain shift, where the distribution of the training data differs significantly from that of the test data. Unsupervised approaches to visual object localization remain challenging but are gaining increasing attention due to their potential for more cost-effective and domain-robust solutions.

The study by Nair et al. in 2004 [26] presents an unsupervised framework for object detection that uses motion information, specifically background subtraction, to label the training examples automatically. Additionally, there are some other researches that use optical flow to detect moving objects as a source of motion information [1].

Self-supervised vision transformer (ViT) [10] has shown a great success in computer vision, and researchers [5] observe that self-supervised ViT features contain explicit information about the semantic segmentation of an image, which

does not emerge as clearly with supervised ViTs, nor with convolutional networks. Drawing from this observation, LOST [30] and TokenCut [33] utilize self-supervised ViT features and suggest the segmentation of a solitary salient object from each image by constructing a graph using DINO's patch features.

This work's primary distinguishing factor is its utilization of the temporal information present in videos. In contrast to prior research that has relied on pre-trained models, our approach employs the frame subtraction as a soft label to train the unsupervised object segmentation model from scratch, and has the capability to detect multiple objects at once.

2.2 Multiple Object Tracking

MOT refers to the computer vision problem that involves identifying and following multiple objects in a video sequence. The common approach to MOT is based on a two-stage strategy. First is a detection step and second is a tracking step that associates an identifier with each object. We focus on the tracking step in this section.

Recently, several deep learning-based MOT algorithms have been proposed. DeepSORT [35] is a deep learning-based extension of SORT [3] that tracks objects using a Kalman filter and the Hungarian matching algorithm. It includes appearance features generated by a deep neural network in its association metric. FairMOT [39] is similar to JDE [34] but aims to balance object detection and re-identification. ByteTrack [38] tracks objects by associating every detection box, using IoU scores to assign tracklets and recovering true objects. It uses the YOLOX [11] detector for the detection task.

The problem of tracking multiple objects in video sequences has been widely studied, but it remains a challenging task due to the need for large amounts of annotated data by supervised learning methods [7,22,25]. Unsupervised methods for MOT have received much less attention in the literature. UnOVOST [24] is built up to segment and track diverse objects by first grouping segments into short tracklets that are spatio-temporally consistent, then merging these tracklets into long-term consistent object tracks based on their visual similarity. SimpleReID [18] trains a re-identification model using labels predicted by SORT [3].

UnOVOST uses visual embeddings which capture more information than necessary. On the other hand, we use object heatmaps only of location information. Our tracking part is simpler than SimpleReID and it exhibits better results.

2.3 Contrastive Learning

Recently, contrastive learning has gained popularity and become widely adopted as a method for unsupervised visual representation learning [6]. It trains models by learning similarities between multiple inputs, through comparison of their representations. A framework for unsupervised object detection using multi-level supervision and contrastive learning between global images and local patches has been proposed in [36].

Contrastive learning can also be utilized to train unsupervised classification models. For example, [21] uses contrastive learning to classify sport players, in a fully unsupervised manner, by maximizing the distance between representations of players from different teams.

To filter out players from all moving objects detected in the video frames, we build a player classification task using contrastive learning. This technique allows us to distinguish players from other moving objects with high accuracy and without the need for human annotations. Our approach differs from [21] since the latter is distinguishing players by team, while we separate players from other entities such as balls and similar objects.

2.4 Sport Player Detection and Tracking

Player detection and tracking in sports videos is a crucial intermediate step in a sport video analysis process. It plays an important role in various other computer vision tasks such as player action recognition, automatic refereeing, goal prediction, highlight detection, ball tracking, and more. This task benefits both sport analyses and entertainment by providing insights and information that are essential for these applications. However, player detection and tracking in sports videos is challenging due to the fast movement of both players and the background, as well as similar visual features and non-rigid transformations of players. Despite these challenges, the player detection and tracking task has been widely researched, as seen in studies such as [12, 32].

Extensive research has been conducted on the topic of sport player tracking using conventional computer vision techniques, as reported in various surveys [25, 29]. Some studies utilized multi-camera systems with fixed cameras that work together to cover the entire playing field [14]. The commonly used player detection method in these fixed camera systems is background subtraction.

There has been a surge in research on sport player tracking using deep learning and computer vision methods [4, 31]. These approaches build on general object detection and tracking frameworks. However, due to the scarcity of training data, researchers either manually annotate images [20] or rely on automatically annotated data from previous studies [9].

The first annotated dataset for soccer games, SoccerNet [8], makes it feasible to train player detection and tracking models on a large scale. However, this is still a challenge for other sports without a similar dataset. A self-supervised detection and tracking model has been introduced in [16] when the players form a small portion of a frame, but it still relies on pretrained computer vision models trained on general annotated vision data. In contrast, our work does not require any annotations or pre-trained models, making it truly unsupervised. We are the first true unsupervised sports tracking work based on deep learning. In addition, our method works well regardless of the size of the players with respect to a frame.

3 Method

In this section, we first formulize the unsupervised sport player tracking task and then present our proposed framework. The framework consists of two stages: (1) training an unsupervised sport player detection network (as outlined in Section 3.2) and (2) training an unsupervised multiple object tracking network (as described in Section 3.3).

3.1 Task Description

The general form of multi-object tracking reads,

$$O = F(I), \tag{1}$$

where I is a sequence of frames $I = \{i_t\}_{t=1}^{T}$. The set of trajectories for all observed objects is denoted by $O = \{o_{i_t}\}_{t=1}^{T}$. For a given frame i, the trajectory slice o_i contains information about all the observed players, represented as $o_i = \{\mathrm{ID}^j, x_i^j, y_i^j, h_i^j, w_i^j\}_{j=1}^{N_i}$, where $(x_i^j, y_i^j, h_i^j, w_i^j)$ is the bounding box for player ID^j.

3.2 Unsupervised Sport Player Detection

The design of our deep neural network pipeline for player detection in sports videos consists of three steps: frame alignment, foreground segmentation, and player detection. The complete framework of our unsupervised player detection method is illustrated in Fig. 1.

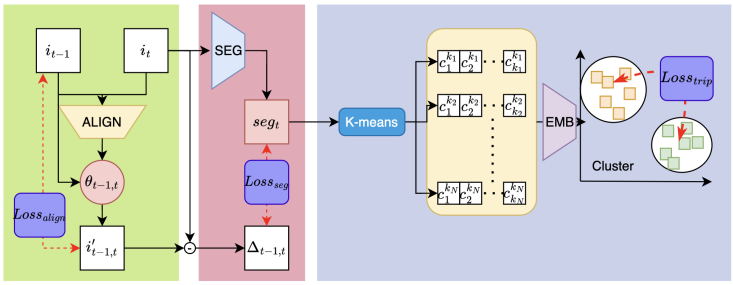

Fig. 1. Unsupervised Sport Player Detection left: frame alignment module, center: foreground segmentation module, right: player classification.

Frame Alignment The first challenge in sport player tracking is the fast-moving camera view. To overcome this challenge, we propose an unsupervised

frame alignment model $\mathbf{ALIGN}_{\tau_A}$ to learn the camera movements $\theta_{t-1,t}$ at timestamp t between consecutive frames $[i_{t-1}, i_t]$:

$$\theta_{t-1,t} = \mathbf{ALIGN}_{\tau_A}(i_{t-1}, i_t). \tag{2}$$

$\mathbf{ALIGN}_{\tau_A}$ is a neural network that takes two frames as input and has trainable parameters τ_A. This camera alignment model is designed under the assumption that the camera movements preserve lines and parallelism between different frames. Thus, it is suitable to assume that all movements are affine transformations and that $\theta_{t-1,t}$ can be represented as a vector of size 6 (rotation, translation, scaling). The details can be found in [17]. In theory, the transformation can be arbitrary, but in practice, we limit it to an affine transformation as it aligns rigidly with the changes in the camera view.

To enhance the training quality, we make use of reference frame i_{t-s} and current frame i_t, where s ranges between 1 and s_{max}, a hyperparameter.

This necessitates parameters $\theta_{p,q}$ that correspond to aligning frame i_p with frame i_q. Note that we allow $p \leq q$ or $q \leq p$. The neural network, $\mathbf{ALIGN}_{\tau_A}$, can be any type of an MSE network that outputs the transformation parameters and takes any two frames as input (not just consecutive). To ensure invertibility of the transformation, we employ the following loss function in the training process:

$$Loss_{align} = \Sigma_{t,s}(||i_{t-s} - i^{'}_{t,t-s}||_2^2 + ||i_t - i^{'}_{t-s,t}||_2^2), \tag{3}$$

where $i^{'}_{p,q} = Trans(i_p, \mathbf{ALIGN}_{\tau_A}(i_p, i_q))$. Here $Trans$ is based on [17]. The first term enforces the transformation to align the reference frame i_{t-s} and current frame i_t, while the second term ensures invertibility of the transformation.

Moving Objects Segmentation The process of aligning consecutive frames makes it possible to separate the moving objects by subtracting frames. However, as the camera is always rapidly moving and there is no background frame available for regular background subtraction, we use frame subtraction as soft labels to train a segmentation model $\mathbf{SEG}_{\tau_S}$. This model operates by taking a frame as input and generates a single logit for each pixel, with trainable parameters denoted as τ_S. The pixel's logit serves as the basis for calculating the probability of whether the pixel corresponds to the background or not.

The soft labels for the segmentation model $\mathbf{SEG}_{\tau_S}$ are obtained through frame subtraction between the input frame i_t and its aligned counterpart $i^{'}_{t-1,t} = Trans(i_{t-1}, \theta_{t-1,t})$ as shown in Fig. 1.

The difference between the frames is transformed into a binary feature map $\Delta_{t-1,t}$ using threshold δ based on

$$\Delta_{t-1,t} = \mathbf{1}(|i^{'}_{t-1,t} - i_t| > \delta). \tag{4}$$

The model $\mathbf{SEG}_{\tau_S}$ is trained using the cross-entropy loss function to minimize the difference between its prediction seg_t and the binary feature map $\Delta_{t-1,t}$ based on

$$Loss_{seg} = -\Sigma_j \left[\Delta^j_{t-1,t} \cdot \log seg^j_t + (1 - \Delta^j_{t-1,t}) \cdot \log(1 - seg^j_t)\right] \tag{5}$$

where j is a pixel. This optimization is with respect to τ_A and τ_S but we solve them sequentially by first optimizing τ_A, fixing it, and then τ_S.

Player Classification The moving foreground in sports videos is composed of all objects that are in motion against the playfield background, including players, balls, referees, and coaches. The final step of our player detection pipeline involves two key challenges: separating individual objects from the foreground and classifying them as either human or non-human.

We propose the following approach to address the aforementioned challenges. Our method leverages pixelwise clustering to identify potential individual players while dynamically expanding high-confidence positive and negative pools to encompass all pixels. Similar concepts have already been introduced in [21]. The main differences are in the following. 1) [21] starts contrastive learning with individual detections, while our method initializes a pool of patches generated through pixelwise K-Means for various values of K. Consequently, our model learns to distinguish not only 'parts of a player' but also the entire 'player' entity. 2) Our iterative strategy diverges from that of [21]. We incrementally augment our positive and negative pools with 'player' images of high confidence during each iteration. In [21], there is no utilization of such pools; instead, they update model weights to obtain new embeddings and seek convergence of cluster centers.

We first use K-Means to divide the foreground defined based on trained $\mathbf{SEG}_{\tau_S}$ to clusters. The number of clusters, i, impacts the final clustering results.

Let $\bar{p}_i^j$ be the binary encoding (mask) of pixels in cluster j when i clusters are specified. Let $\hat{p}_i^j$ be the bounding box of $\bar{p}_i^j$ as binary encoding. Furthermore, we introduce p_i^j as the corresponding bounding box of $\bar{p}_i^j$, represented as the pixel-wise product of $\hat{p}_i^j$ with the frame. We try a range of values for i from 1 to K, where K is a hyperparameter, and denote the union of all these bounding boxes as P, which can be expressed as $P = \bigcup_{i=1}^{K} \bigcup_{j=1}^{i} p_i^j$.

Our goal is to isolate individual players, so we train another convolutional neural network, $\mathbf{EMB}_{\tau_E}$, to generate embeddings for all object candidates, and differentiate players from other objects by their embeddings. Network $\mathbf{EMB}_{\tau_E}$ with trainable parameters τ_E takes elements in P as input and it outputs an embedding.

At the start of the training and inference process, two candidate pools are constructed from P. The first pool is the positive player candidate pool $P_{pos} \subseteq P$, where each candidate is chosen based on a strict criterion. To ensure that the bounding boxes correspond to complete single human bodies, we establish strict criteria for the boxes in P_{pos}. These criteria include 1) the center of $\hat{p}_j^j$ (bounding box) must be in $\bar{p}_i^j$ (cluster) and 2) a low proportion of pixels in $\bar{p}_i^j$ on the edges of $\hat{p}_j^j$. This eliminates boxes that encompass multiple individuals or only partial bodies. These rules are not specific to player detection in sports, but can be applied to other detection problems using rectangular bounding boxes.

The second pool is the negative player candidate pool $P_{neg} \subseteq P$, which is created by merging two clusters from the playfield with low Intersection over

Union (IoU). To this end, consider $P_{field} = \{$randomly cropped patches p with $\text{IoU}(p, P_{pos}) \leq \text{maxIoU}\}$. We start with $P_{neg} = \{Merge(p,q)|p,q$ randomly selected from $P_{field} \bigcup (P \setminus P_{pos})\}$ where $Merge(p,q)$ is the bounding box of $p \bigcup q$.

In each training iteration of $\mathbf{EMB}_{\tau_E}$, P_{pos} and P_{neg} are expanded based on the ranking of a bounding box confidence score calculated as

$$\text{score}(p) = \frac{||\mathbf{EMB}_{\tau_E}(p) - \mathbf{EMB}_{\tau_E}(P_{neg})||_2^2 - ||\mathbf{EMB}_{\tau_E}(p) - \mathbf{EMB}_{\tau_E}(P_{pos})||_2^2}{||\mathbf{EMB}_{\tau_E}(p) - \mathbf{EMB}_{\tau_E}(P_{neg})||_2^2 + ||\mathbf{EMB}_{\tau_E}(p) - \mathbf{EMB}_{\tau_E}(P_{pos})||_2^2} \tag{6}$$

where $\mathbf{EMB}_{\tau_E}(P_{pos})$ is the numerical average of all the embeddings of the candidates in P_{pos}, $\mathbf{EMB}_{\tau_E}(P_{neg})$ is the numerical average of all the embeddings of the candidates in P_{neg}, and p is any bounding box remaining in P. Set P_{pos} is expanded by all p with $score(p) > threshold_{pos}$ and P_{neg} by all those with $score(p) < threshold_{neg}$.

The model is trained using the triplet loss, with each triplet $(p_i, p_{i,+}, p_{i,-})_{i=1}^{M}$ being constructed by randomly selecting an anchor candidate $p_i \in P$, a positive candidate $p_{i,+} \in P_{pos}$ and a negative candidate $p_{i,-} \in P_{neg}$. The loss function of $\mathbf{EMB}_{\tau_E}$ is defined as

$$\begin{aligned} Loss_{emb}(P, \mathbf{EMB}_{\tau_E}) = \Sigma_i max(0, ||\mathbf{EMB}_{\tau_E}(p_i) - \mathbf{EMB}_{\tau_E}(p_+)||_2^2 - \\ ||\mathbf{EMB}_{\tau_E}(p_i) - \mathbf{EMB}_{\tau_E}(p_-)||_2^2 + \epsilon), \end{aligned} \tag{7}$$

where ϵ is the margin between positive and negative pairs.

3.3 Unsupervised Multiple Object Tracking

We introduce an unsupervised multiple object tracking model based on the CenterTrack approach [41] which is supervised. In this model, every object is treated as a circle on the frame's heatmap, and the width and height of the object are predicted by two regression heads as in [41]. The third center displacement head predicts the movement of each object center between frames. Although the model is trained in a fully supervised manner, we use self-supervision to train the center displacement head.

The center heatmap, denoted as $C(x, y, t) \in [0, 1]$, is defined for every pixel at each frame and is constructed using a set of Gaussian kernels centered at the object centers

$$C(x, y, t) = \max_{k=1,\ldots,K^t} G((x, y), (x_k^t, y_k^t); \sigma), \tag{8}$$

in which G is the Gaussian kernel with spread of σ, K^t is the number of objects on frame t, and pixel (x_k^t, y_k^t) is the center of each object.

Every pixel (x, y) at time t moving to $(x + \Delta x, y + \Delta y)$ after Δt is captured by

$$C(x + \Delta x, y + \Delta y, t + \Delta t) = C(x, y, t). \tag{9}$$

Assuming the time change to be very small, the above equation with Taylor series expansion can be reformulated as

$$C(x+\Delta x, y+\Delta y, t+\Delta t) = C(x,y,t) + \frac{\partial C}{\partial x}\Delta x + \frac{\partial C}{\partial y}\Delta y + \frac{\partial C}{\partial t}\Delta t + o(t^2), \quad (10)$$

which leads to

$$\frac{\partial C}{\partial x}\Delta x + \frac{\partial C}{\partial y}\Delta y + \frac{\partial C}{\partial t}\Delta t = 0. \quad (11)$$

Dividing by dt yields

$$\frac{\partial C}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial C}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial C}{\partial t} = 0. \quad (12)$$

The displacement heatmap $V(x,y,t)$ represents the speed of each pixel and is defined as

$$V(x,y,t) = \lim_{\Delta t \to 0}(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t}) = (\frac{dx}{dt}, \frac{dy}{dt}) = (V_x(x,y,t), V_y(x,y,t)). \quad (13)$$

The change in the value of C over time can be expressed as

$$C(x,y,t+\Delta t) - C(x,y,t) = \frac{\partial C}{\partial t}\Delta t = -(\frac{\partial C}{\partial x}V_x + \frac{\partial C}{\partial y}V_y)\Delta t, \quad (14)$$

where C is known and the unknown variables are V_x, V_y.

Taking a continuous sub-region A of the full image, and the summation for every pixel inside this region, leads to

$$Q(A,t) = \sum_{(p,q)\in A}[C(p,q,t+\Delta t) - C(p,q,t)] = -\sum_{(p,q)\in A}(\frac{\partial C}{\partial x}V_x + \frac{\partial C}{\partial y}V_y)|_{(p,q,t)}\Delta t. \quad (15)$$

Let us assume that the sub-region A is small enough so that V_x, V_y of each pixel inside A are the same: $\forall (p,q) \in A, V_x(A,t) = V_x(p,q,t), V_y(A,t) = V_y(p,q,t)$. We get

$$Q(A,t) = -V_x(A,t)\sum_{(p,q)\in A}\frac{\partial C}{\partial x}|_{(p,q,t)}\Delta t - V_y(A,t)\sum_{(p,q)\in A}\frac{\partial C}{\partial y}|_{(p,q,t)}\Delta t. \quad (16)$$

The trainable neural network $\mathbf{UMOT}_{\tau_U}$ outputs the displacement heatmap $V(x,y,t)$ with its loss function defined by

$$\begin{aligned} Loss_A(V_x^{\tau_U}, V_y^{\tau_U}) = ||Q^*(A,t) + V_x^{\tau_U}(A,t)\sum_{(p,q)\in A}\frac{\partial C^*}{\partial x}|_{(p,q,t)}\Delta t + \\ V_y^{\tau_U}(A,t)\sum_{(p,q)\in A}\frac{\partial C^*}{\partial y}|_{(p,q,t)}\Delta t||_2^2. \end{aligned} \quad (17)$$

Value Δt is 1/(frames per second). Ground truths C^* and Q^* are obtained based on (8) and (16) where the bounding boxes or objects are the outputs of $\mathbf{EMB}_{\tau_E}$. The tracking loss is defined as the sum of the loss function over all regions in the image I

$$Loss_{track} = \sum_{A \in I} Loss_A(V_x^{\tau_U}, V_y^{\tau_U}). \tag{18}$$

The overall loss is $L = Loss_{track} + Loss_{width} + Loss_{height}$ (see [41] about the latter two terms). With trained $\mathbf{UMOT}_{\tau_U}$, we consider a bounding box A in frame t. The model gives $V_x(A,t)$ and $V_y(A,t)$ which we use to offset A in frame $t+1$ to get $\bar{A}$. To $\bar{A}$ we find the closest bounding box in frame $t+1$ which is then set as the matching bounding box in frame $t+1$ to A.

4 Experiment Analysis

In this section, the training and testing data sets are introduced in Section 4.1. The evaluation metrics and baselines are then discussed in Section 4.2 and Section 4.3, respectively. The details of the proposed models' implementation can be found in Section 4.4, followed by the results on soccer and football game videos, which are presented in Section 4.5.

4.1 Training and Testing Data

In this study, we evaluate our proposed models on two popular sports: soccer and football. Our experiments are conducted in an unsupervised setting, where only raw video frames are used for training and no annotations are provided. The models are trained solely on the raw video frames without any external annotations.

SoccerNet-Tracking The SoccerNet-Tracking dataset [8] is a large public resource for tracking soccer players in video, Fig. 2. It consists of 200 sequences, each 30 seconds in length, and a 45-minute half-time. The annotations provided include bounding boxes, tracklet IDs, player jersey numbers, and team tags, with a frame rate of 25 frames per second. With over 3.6 million bounding boxes and over 5,000 unique tracklets, SoccerNet-Tracking is one of the largest multi-object tracking datasets available. We stress that annotations are used only to evaluate results (and not in training).

NFL Tracking The NFL Tracking dataset is an adapted version of the NFL 1st and Future dataset [15], which consists of 126 videos capturing NFL plays from both a sideline and end zone perspective. The original annotations include bounding boxes and ID tracklets of helmets. We expanded annotations by generating player bounding boxes based on the helmet information. We simply increased the size of the helmet bounding boxes by a defined ratio. Examples of the images and labels can be seen in Fig. 3.

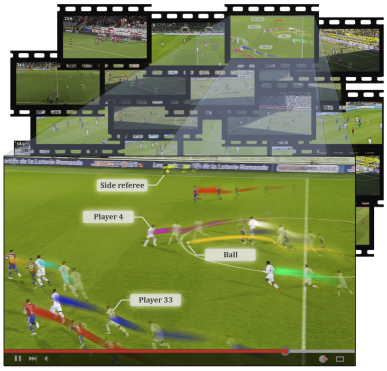

Fig. 2. SoccerNet-Tracking: This dataset includes 200 soccer videos of 30s each, representative of interesting moments from 12 soccer games, densely annotated with player tracklets, teams and jersey numbers [8].

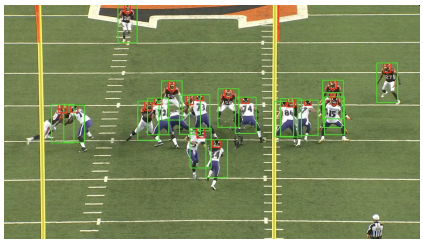

Fig. 3. NFL 1st and Future: This dataset includes 126 football play videos. The player bounding boxes (green) are generated by enlarging the helmet bounding boxes (red) in two directions.

4.2 Evaluation Metrics

We use the same evaluation metrics for the two datasets.

MOTA The Multiple Object Tracking Accuracy (MOTA) metric [2] is commonly used for MOT. It penalizes the ratio of missed boxes, false positive boxes, and identity switches computed over the number of ground truth boxes in the sequence.

HOTA HOTA (Higher Order Tracking Accuracy) [23] is a more recent metric proposed to equally weight detection and association. For more information, refer to the HOTA paper [23]. We use both MOTA and HOTA to evaluate the overall performance of our proposed method and the baseline models.

4.3 Baselines

To the best of our knowledge, no existing model has been designed specifically for the unsupervised sport player tracking task in moving camera settings. To evaluate the performance of our proposed model, we compare it to the results reported in [8] using three state-of-the-art general MOT models: DeepSORT [35], FairMOT [39], and ByteTrack [38]. These models were evaluated on the SoccerNet-Tracking dataset in [8]. Additionally, we trained the ByteTrack model using their published code on the NFL Tracking dataset as a baseline (DeepSORT and FairMOT have no open source code). In [8], experiments were conducted in two settings: "w/ GT" indicates that ground-truth detections were provided to the models, while "w/o GT" indicates that each model used its own unsupervised detector. We only consider "w/o GT." FairMOT-ft was fine-tuned in [8] on the training dataset for an additional 10 epochs.

4.4 Implementation Details

The setup for our experiments includes a server equipped with an NVIDIA GeForce RTX 3090, and PyTorch as the deep learning library. All neural networks are trained using the Adam optimizer [19] with an initial learning rate of 10^{-4}.

The neural network $\mathbf{ALIGN}_{\tau_A}$ consists of 5 convolutional layers with 3x3 kernels, interspersed with pooling layers, and capped by two fully connected layers. The output of the final layer is a vector of size 6. The training batch size is set as 64. The neural network $\mathbf{SEG}_{\tau_S}$ is a U-Net network that includes 4 encoding blocks and 4 decoding blocks, and returns a binary feature map. The training batch size is set as 16. $\mathbf{EMB}_{\tau_E}$ is a 3-layer convolutional neural network with 3x3 kernels, pooling layers, and two fully connected layers at the top. The final layer produces a vector of size 1024. The stopping criteria for training the $\mathbf{EMB}_{\tau_E}$ is when positive candidate boxes cover δ positive pixels. We set δ as 98% in this work. The training batch size is set as 64. The margin ϵ in (7) is 0.5. $\mathbf{UMOT}_{\tau_U}$ is based on [41] which uses DLA [37] as the network backbone, and the training batch size is set as 16.

Data augmentation To prevent overfitting due to the green playfield being the dominant feature in many frames of the sports videos, we randomly convert half of the training frames to grayscale.

Additionally, we employ data augmentation techniques such as scaling, rotation, and horizontal flipping to improve the quality of our model. The scaling factor ranges from 0.9 to 1, the rotation is limited to less than 10 degrees, and horizontal flipping is randomly applied with a probability of 50 percent.

Model selection The $\mathbf{SEG}_{\tau_S}$ model is trained to identify and extract all moving objects in the video frames during the training process. However, since the frame subtraction process involves pixels that change between frames, and $\mathbf{ALIGN}_{\tau_A}$ is not able to perfectly align frames, $\mathbf{SEG}_{\tau_S}$ learns to detect more noises as it is further trained. It is difficult to choose a model that can accurately segment human objects while avoiding distractions from other noises.

To prevent overfitting caused by the noises, we have created an auxiliary task to choose the best model $\mathbf{SEG}_{\tau_S}$ from the set of candidates $\{\mathbf{SEG}_{\tau_S^j}\}_{j=1}^{M}$ (M being the number of epochs). The goal is to select a model that can effectively ignore distracting elements, such as white lines and goalposts, which are known to contribute to these noises. The auxiliary task specifically focuses on filtering out artificially added white lines. The objective of the affiliate task is to choose the best model $\mathbf{SEG}_{\tau_S}$ from the set of candidate models $\{\mathbf{SEG}_{\tau_S^j}\}_{j=1}^{M}$ by minimizing the difference between the segmentation results on an original video frame f_i and an artificial frame f_i^a. The artificial frame f_i^a is created by adding long white lines or white curves to the original frame f_i in random directions. The model selection is performed by minimizing the squared L2 norm between the segmentation results of the two frames

$$\text{argmin}_j \Sigma_i ||\mathbf{SEG}_{\tau_S^j}(f_i^a) - \mathbf{SEG}_{\tau_S^j}(f_i)||_2^2. \tag{19}$$

Our proposed model has three modules in the detection framework and one module in the tracking framework. The modules are trained in a sequential order, starting with $\mathbf{ALIGN}_{\tau_A}$, then $\mathbf{SEG}_{\tau_S}$, followed by $\mathbf{EMB}_{\tau_E}$, and finally the tracking module $\mathbf{UMOT}_{\tau_U}$. After training $\mathbf{SEG}_{\tau_S}$ for 10 epochs, we choose the best model using the affiliate task specified by (19). We call the entire pipeline **USTP** (unsupervised sports tracking pipeline).

4.5 Experimental results

Our proposed model is composed of several neural networks, and to evaluate the effectiveness of each component, we also conduct an ablation study comparing the performance of the full model with models missing individual modules.

- "w/o $\mathbf{ALIGN}_{\tau_A}$": The $\theta_{t-1,t}$ for any timestamp t is set as vector of all 0 and the frame subtraction is conducted between original frames i_t and i_{t-1}.
- "w/o $\mathbf{SEG}_{\tau_S}$": The segmentation model is ignored, instead the $\Delta_{t-1,t}$ is considered as the moving object segmentation mask.

– "w/o $\mathbf{EMB}_{\tau_E}$": The initial rule filtered bounding boxes are considered as the detection results.

Table 1. Experimental results and ablation study for SoccerNet-Tracking: best result is formatted in bold, and second-best is underscored.

Methods	HOTA	DetA	AssA	MOTA
DeepSORT	36.663	40.022	33.759	33.913
FairMOT	43.911	<u>46.317</u>	41.778	**50.698**
ByteTrack	<u>47.225</u>	44.489	**50.257**	31.741
USTP	**48.127**	**52.632**	<u>44.091</u>	<u>46.845</u>
USTP w/o **ALIGN**	45.227	46.976	43.238	39.185
USTP w/o **SEG**	32.613	35.108	29.529	33.783
USTP w/o **EMB**	39.798	45.265	34.437	42.436

SoccerNet-Tracking The results are summarized in Table 1. **USTP** has the highest HOTA score and DetA score, while ByteTrack has the highest AssA score and FairMOT has the highest MOTA score. Among these models, **USTP** consistently ranks in the top two for all four metrics, with its best HOTA score resulting from its strong lead in DetA.

All ablated models show significant differences in performance compared to the full model. The model without the $\mathbf{SEG}_{\tau_S}$ module performs the worst among all metrics, demonstrating the importance of the segmentation model in learning high-level object features. Meanwhile, the model without the **ALIGN** module has performance that is closest to the full model, suggesting that the segmentation model can still learn even from unaligned frame subtractions.

As a reference point, supervised ByteTrack yields HOTA of 71.50 and MOTA of 94.57, which not surprisingly are much higher than the unsupervised counterparts. FairMOT-ft which is also supervised by using annotations yields HOTA of 57.88 and MOTA of 83.57.

In Fig. 4, we present instances of suboptimal inferences. These examples vividly illustrate the formidable challenge posed by player overlap to **USTP**. This challenge results in both detection errors and the loss of tracking IDs.

NFL Tracking We evaluated the model performance of **USTP** and its ablated versions on the NFL Tracking dataset, Table 2. **USTP** outperformed "ByteTrack" in terms of MOTA. This outcome can be attributed to several factors,

including the higher level of physical confrontation in football games compared to soccer games and the varying poses of football players during the game. **USTP** performs better than all of its ablated versions, demonstrating the effectiveness of each component in the framework. However, it is worth noting that the annotations in the NFL Tracking dataset are based on helmet bounding boxes, which could affect the validity of the performance metrics in validation.

In Fig. 5, we similarly present instances of suboptimal inferences. These examples underscore the complexity of player overlap in the context of detection. Notably, it is evident that football videos tend to exhibit a higher frequency of overlapping players, thereby resulting in lower performance metrics compared to soccer games.

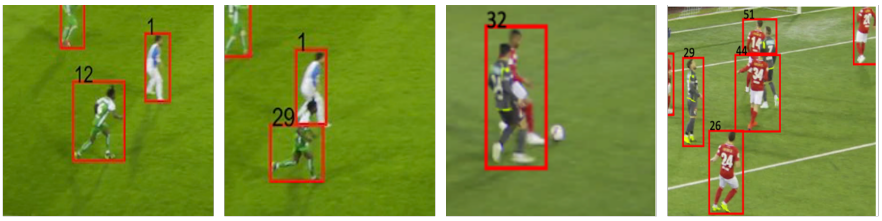

Fig. 4. Bad inferences examples in SoccerNet-Tracking of USTP: the left two frames show tracking ID lost; the right two show detection errors for overlapping players.

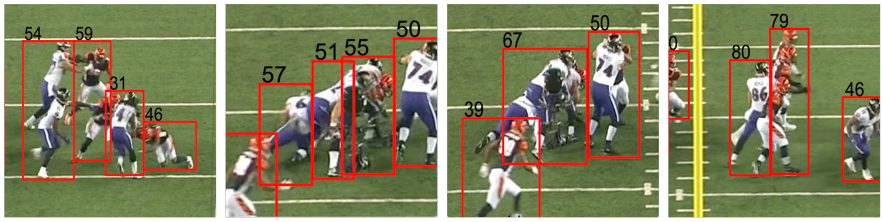

Fig. 5. Bad inferences examples in NFL Tracking of USTP: overlapping of players is challenging for detection.

4.6 Computational Time Analysis

We report training times for both **USTP** and ByteTrack for the NFL Tracking dataset. For **USTP**, the total training time amounts to 19.2 hours, consisting of 4 models: 0.3 hour for training $\mathbf{ALIGN}_{\tau_A}$ (with 689,000 parameters), 8.5 hours

Table 2. Experiment results and Ablation study for NFL Tracking: best result is formatted in bold, and second-best is underscored.

Methods	HOTA	DetA	AssA	MOTA
ByteTrack	**35.806**	**33.968**	**38.137**	<u>26.524</u>
USTP	<u>30.918</u>	<u>33.356</u>	<u>28.337</u>	**29.684**
USTP w/o **ALIGN**	25.947	27.605	25.387	22.822
USTP w/o **SEG**	21.470	23.487	19.074	21.554
USTP w/o **EMB**	23.802	27.901	21.494	24.725

for training $\mathbf{SEG}_{\tau_S}$ (with 7×10^6 parameters), 6.9 hours for training $\mathbf{EMB}_{\tau_E}$ (with 1.7×10^6 parameters), and 3.5 hours for training $\mathbf{UMOT}_{\tau_U}$ (with 15.7×10^6 parameters). On the other hand, ByteTrack requires a training time of 24.5 hours with 25.3×10^6 parameters.

We measure frames per second (FPS) in inference of **USTP** following the settings of [11] by using floating point 16-precision on a single GPU, and compare FPS with ByteTrack on the NFL Tracking data set. This result shows **USTP** is running 40% slower than ByteTrack with ByteTrack having FPS of 19.3 and **USTP** 11.8. The primary factor contributing to this disparity is that the **USTP** approach predicts the tracking problem incrementally, generating intermediate outputs such as camera movements and segmentation heatmaps at different stages.

5 Conclusion

In this study, we propose a novel deep learning-based unsupervised sport player tracking system. The system is comprised of several innovative modules that exploit the inherent characteristics of sport videos. It is important to note that this framework does not require any prior knowledge or pre-trained models.

Through our experiments, we demonstrate the generalizability of our model to different types of sport videos, including soccer and football games, which are known to be two of the most complex sports to track. This result highlights the versatility and robustness of our proposed system. The experimental results reveal that in the SoccerNet-Tracking dataset, **USTP** outperforms all existing unsupervised models. However, when comparing **USTP** to ByteTrack in NFL tracking, it exhibits superior results in terms of MOTA but lags behind in HOTA performance. We also observe that **USTP** has a big gap in association metrics compared to ByteTrack, which leads to the gap in HOTA. The divergence in performance between the two sports domains also offers valuable insights for the application of **USTP** . **USTP** demonstrates significantly superior performance

compared to other models in soccer, primarily due to the relatively low occurrence of player overlap in soccer games. This suggests that applying **USTP** to sports with similar characteristics, such as swimming and volleyball, is likely to yield major improvements over current state-of-the-art. However, it may not be as effective in sports characterized by frequent physical contact, such as football or basketball.

We believe that our model can serve as the new baseline for unsupervised sport player detection and tracking, and has the potential to be used for automatic label generation. In future work, we plan to enhance the performance of the model with regard to overlapping players and complicated poses.

References

1. Agarwal, A., Gupta, S., Singh, D.K.: Review of optical flow technique for moving object detection. In: Proceedings of 2016 2nd International Conference on Contemporary Computing and Informatics. pp. 409–413 (2016)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear MOT metrics. EURASIP Journal on Image and Video Processing pp. 1–10 (2008)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: Proceedings of 2016 IEEE International Conference on Image Processing. pp. 3464–3468 (2016)
4. Burić, M., Pobar, M., Ivašić-Kos, M.: Object detection in sports videos. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics. pp. 1034–1039 (2018)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294 (2021)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of International Conference on Machine Learning. pp. 1597–1607 (2020)
7. Ciaparrone, G., Sánchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., , Herrera., F.: Deep learning in video multi-object tracking: A survey. Neurocomputing (2020)
8. Cioppa, A., Giancola, S., Deliege, A., Kang, L., Zhou, X., Cheng, Z., Ghanem, B., Van Droogenbroeck, M.: Soccernet-tracking: Multiple object tracking dataset and benchmark in soccer videos. In: Proceedings of the 2022 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3491–3502 (2022)
9. D'Orazio, T., Leo, M., Mosca, N., Spagnolo, P., Mazzeo, P.L.: A semi-automatic system for ground truth generation of soccer video sequences. In: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. pp. 559–564 (2009)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2021)
11. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
12. Gerke, S., Muller, K., Schafer, R.: Soccer jersey number recognition using convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 17–24 (2015)
13. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
14. Hamid, R., Kumar, R.K., Grundmann, M., Kim, K., Essa, I., Hodgins, J.: Player localization using multiple static cameras for sports visualization. In: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 731–738 (2010)
15. Howard, A., J, A., Evans, B., Langdon, C., Huddleston, S., Cukierski, W.: NFL 1st and future - impact detection (2020), https://kaggle.com/competitions/nfl-impact-detection

16. Hurault, S., Ballester, C., Haro, G.: Self-supervised small soccer player detection and tracking. In: Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports. pp. 9–18 (2020)
17. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. arXiv preprint arXiv:1506.02025 (2016)
18. Karthik, S., Prabhu, A., Gandhi, V.: Simple unsupervised multi-object tracking. arXiv preprint arXiv:2006.02609 (2020)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2017)
20. Komorowski, J., Kurzejamski, G., Sarwas, G.: Footandball: Integrated player and ball detector. arXiv preprint arXiv:1912.05445 (2019)
21. Koshkina, M., Pidaparthy, H., Elder, J.H.: Contrastive learning for sports video: Unsupervised player classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4528–4536 (2021)
22. Leal-Taixé, L., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S.: Tracking the trackers: an analysis of the state of the art in multiple object tracking. arXiv preprint arXiv:1704.02781 (2017)
23. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking. International journal of computer vision (2021)
24. Luiten, J., Zulfikar, I.E., Leibe, B.: Unovost: Unsupervised offline video object segmentation and tracking. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2000–2009 (2020)
25. Manafifard, M., Ebadi, H., Moghaddam, H.A.: A survey on player tracking in soccer videos. Computer Vision and Image Understanding (2017)
26. Nair, V., Clark, J.J.: An unsupervised, online learning framework for moving object detection. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004)
27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems (2015)
29. Shih, H.C.: A survey of content-aware video analysis for sports. IEEE Transactions on circuits and systems for video technology (2017)
30. Siméoni, O., Puy, G., Vo, H.V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., Ponce, J.: Localizing objects with self-supervised transformers and no labels. arXiv preprint arXiv:2109.14279 (2021)
31. Theagarajan, R., Pala, F., Zhang, X., Bhanu, B.: Soccer: Who has the ball? generating visual analytics and player statistics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1749–1757 (2018)
32. Vats, K., McNally, W., Walters, P., Clausi, D.A., Zelek, J.S.: Ice hockey player identification via transformers. arXiv preprint arXiv:2111.11535 (2021)
33. Wang, Y., Shen, X., Yuan, Y., Du, Y., Li, M., Hu, S.X., Crowley, J.L., Vaufreydaz, D.: Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. arXiv preprint arXiv:2209.00383 (2022)
34. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: Proceedings of the European Conference on Computer Vision. pp. 107–122 (2020)

35. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: Proceedings of 2017 IEEE International Conference on Image Processing. pp. 3645–3649 (2017)
36. Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., Luo, P.: Detco: Unsupervised contrastive learning for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8392–8401 (2021)
37. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2403–2412 (2018)
38. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: Proceedings of the European Conference on Computer Vision. pp. 1–21 (2022)
39. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision (2021)
40. Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X.: Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems (2019)
41. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: Proceedings of the European Conference on Computer Vision. pp. 474–490 (2020)
42. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055 (2019)