Progress in Artificial Intelligence (2022) 11:279–313
https://doi.org/10.1007/s13748-022-00290-6

REVIEW

Robust appearance modeling for object detection and tracking: a survey of deep learning approaches

Alhassan Mumuni[1] · **Fuseini Mumuni**[2]

Received: 25 May 2021 / Accepted: 16 August 2022 / Published online: 6 September 2022

Abstract
The task of object detection and tracking is one of the most complex and challenging problems in artificial intelligence (AI) systems that model perception. Object tracking has practical importance in AI applications like human–machine interaction, robotics, autonomous driving and extended reality. The fundamental task of object tracking is to detect objects in one video frame and maintain their identities or infer their trajectories across all subsequent frames. Real-world object tracking systems typically operate in highly complex and dynamic environments, with constantly changing object appearance and scene conditions, making it challenging to adequately characterize target objects with a single model. Traditional AI solutions rely on modeling handcrafted features based on rigorous mathematical formulations. This process is a highly non-trivial task and severely restricts end solutions to narrowly focused application settings. Today, deep learning techniques are the most preferred approaches due to their high generalization ability and ease of implementation. This paper surveys the most important deep learning-based appearance modeling techniques. We propose a unique taxonomy of approaches based on the architectural elements and auxiliary strategies that are employed in deep learning models for robust appearance modeling. The surveyed methodologies include data-centric techniques, compositional part modeling, similarity learning methods, memory and attention mechanisms, as well as approaches that integrate differentiable models within deep learning architectures to explicitly model spatial transformations. The fundamental principles, implementation details and application contexts, as well as the main strengths and potential limitations of the approaches are highlighted. We also present common datasets, evaluation metrics and performance results.

Keywords Visual tracking · Robust object detection · Generative modeling · Deformable part modeling (DPM) · Similarity learning · Attention mechanism

1 Introduction

Visual object tracking, or simply object tracking, is the process of maintaining an estimation of a specific object's (or set of objects') position(s) in a video sequence. This is closely related to the problem of video object detection [1], in which the task is to localize target object(s) in each image frame of a video sequence. However, with object tracking, an additional task is to predict a trajectory of the detected object(s). In many cases, object detection is a sub-task of visual tracking. The simplest case of visual tracking, *single object tracking* (SOT), considers the problem of tracking a single object in a video stream. The tracking task in most cases can be effectively accomplished by simply detecting the target object in each video frame [2]. *Multiple object tracking* (MOT) is a more complex problem involving the tracking of many objects simultaneously. Because of the complexity of MOT tasks, additional algorithms are often utilized to enhance robustness.

Important applications of object tracking include video surveillance [3], sports broadcasts [4], civil security applications [5], human–machine interaction [6], augmented reality [7], robotics and autonomous driving [8].

Visual appearance is the most important characteristic of physical objects that enables—in both biological and

✉ Alhassan Mumuni
alhassan.mumuni@cctu.edu.gh

Fuseini Mumuni
fmumuni@umat.edu.gh

[1] Cape Coast Technical University, P. O. Box DL 50, Cape Coast, Ghana

[2] University of Mines and Technology, P. O. Box 237, Tarkwa, Ghana

machine cognition—the effective recognition of different objects. Appearance modeling is aimed at encoding functional representations of visual features of objects that preserve their meaning under different viewing conditions. This is considered the most important task of the visual tracking problem [9,10]. The main task in robust appearance modeling is to extract useful visual information from training images that are invariant under different real-world phenomena (e.g., varying illumination, scale changes, occlusions and deformations). The learned visual representations are then used to aid detection and tracking, thus making it possible to accurately track objects regardless of variations in object or scene appearance.

Object tracking settings are usually highly dynamic in nature, with constantly changing object appearances and environmental conditions. The typical tracking setting is characterized by complicating factors such as object interactions, camera motion, cluttered backgrounds, non-uniform illumination, motion blur, changing object scales, occlusions, varying view angles, nonlinear object deformations, and changing scene conditions. Under these circumstances, a target object model captured under particular conditions may be incapable of representing the object in subsequent frames when the viewing conditions change.

1.1 Related works

Given the practical importance of visual tracking, a large number of surveys have been conducted on different aspects of tracking. Most of these surveys are dedicated to either classical machine learning approaches (e.g., [4,11–15]) or deep learning-based tracking techniques (e.g., [16–20]), while a few others (e.g., [21,22]) deal with both classical and deep learning approaches. Many surveys treat visual tracking techniques from the perspective of a given taxonomy defined according to various criteria [18–20,22]. For instance, Abbass et al. [16] classified tracking algorithms into methods that employ generative or discriminative models and techniques that utilize a combination of both approaches. They then presented an elaborate discussion of deep learning-based trackers under these broad methodological themes. Li et al. [20] introduced a taxonomy on the basis of network structure, function and training and presented a detailed description of deep learning-based trackers from the point of view of the proposed taxonomy. Similarly, in [19] Xu et al. categorized trackers into three groups, namely, deep network embedding-, description enhancement-, and end-to-end-based trackers. They further presented a detailed discussion on object tracking architectures and training methods for deep convolutional neural network (DCNN)- and recurrent neural network (RNN)-based trackers. Fiaz et al. [22] focused on techniques for tracking objects in noisy images. They classified visual tracking methods into correlation filter- and noncorrelation filter-based approaches and provided an extensive treatment of the common techniques in each of the categories based on the general architectures and tracking procedures.

Other works treat object tracking methods based on their constituent components (e.g., [15,21]) or the main sub-tasks [12,14,17] in the tracking pipeline. Notably, [15,21] presented deep learning-based visual trackers based on their key components and discussed extensively the application of deep learning methods in each component. In [15], Luo et al. classified MOT algorithms according to three different criteria: initialization method, image processing approach and output type. They then presented a generalized object tracking pipeline and the essential components of MOT models and, for each component, discussed the common issues and implementation details. Sugirtha and Sridevi [23] focus on the various stages of video object detection as well as tracking. [21] focuses exclusively on tracking-by-detection frameworks and the application of different deep learning techniques in the various sub-tasks of tracking.

Several surveys [4,21,24–27] deal with tracking issues in specific domains. These include animal tracking [25,28], human tracking in specific contexts (e.g., in football games [4,24]), football tracking [26], vehicle tracking [28,29], pedestrian tracking [21,24], or both vehicle and pedestrian tracking [27].

Datasets, evaluation metrics and extensive analysis of the performance of different trackers are presented in [16–18,20,22,24]. In addition to these surveys, the performance results of many state-of-the-art trackers are presented in the reports of annual object tracking competitions—notably, the Visual Object Tracking (VOT) for SOT trackers [30–32], and the Multiple Object Tracking (MOT) challenges [33].

Despite the importance of appearance modeling in visual tracking, only a few surveys [11,12] are dedicated solely to appearance modeling. However, even these surveys focus exclusively on classical approaches to appearance modeling. Till date, no single work has covered deep learning-based approaches to appearance modeling in sufficient detail. We propose this survey to address this gap.

1.2 Scope and outline of study

In view of the issues that have already been tackled by previous survey papers, we limit the scope of this review to studying deep learning-based robust appearance modeling techniques. We specifically focus on special deep neural network topologies and auxiliary strategies that are employed in conjunction with classical deep CNNs for invariant representation of visual appearance features. The techniques are aimed at improving the robustness of object tracking models in general settings. In addition, we discuss common evalua-

tion metrics and present quantitative performance results on several state-of-the-art visual trackers.

The paper is structured as follows. Section 1 provides a general background to the problem of object tracking, and highlights the importance of appearance modeling in visual tracking. It also explores related surveys of deep learning approaches to object tracking, and outlines the main differences with the current work. Section 2 presents a general framework of visual tracking and the various subtasks involved in the tracking process. In Sects. 3 to 7, we conduct a thorough survey of state-of-the-art deep learning approaches for encoding robust appearance features for object detection and tracking tasks. Section 8 presents common datasets, evaluation methods and performance results of the surveyed approaches. In Sect. 9, we summarize and discus the major issues of object detection and tracking algorithms. Section 10 explores potential developments and directions for future research. Finally, in Sect. 11, we conclude by recapping the main issues discussed in this work.

2 Appearance modeling in tracking

In this section, we present a generic structure of object tracker in the context of deep learning and summarize general approaches to appearance modeling based on deep learning techniques.

2.1 General framework of object tracking models

We present a generalized architecture of object tracking models and briefly describe its components. We utilize the conceptual framework for object tracking proposed by Wang et al in [10]. Per this framework, a tracker is essentially made up of a number of distinct components, each performing different functions: motion model, feature extractor, observation model, model updater, and ensemble post-processor [10]. With some modifications, we represent this generic architecture in the context of deep learning-based visual tracking in Fig. 1.

The appearance model encodes invariant representation of visual features, while the motion model estimates the location of the target object in subsequent frames. As shown in the diagram, the extracted features are used to build both appearance and motion models, which together form the basis for the observation model used to make predictions about target locations. In a deep learning setting, the observation model may be a neural network sub-model that aggregates the outputs of the appearance and motion models. An often critical component of most online trackers is the model updater. It performs periodic updates to allow temporal context of the video sequence to be incorporated in the tracking process.

There may also be an ensemble post-processor [10] (which we termed Auxiliary Module) for performing additional functions such as fusing the predictions of several tracklets in cases where multiple observations are made about the same object(s) (see Fig. 1). In particular, the data association and affinity computation [17] are common tasks that provide additional information that can be used to compensate for detection errors, and helps to localize target instances or to recover missing observations. Other post-processing tasks may include the removal of false detections or interpolating trajectories in case of discontinuities (e.g., due to occlusions) [34,35].

2.2 Overview of common Deep Learning approaches to appearance modeling

Invariably, the first step of object tracking involves learning an appearance model for the objects to be tracked. This requires extracting a compact set of invariant image features, based on which the tracking can be performed. We present the most common approaches to deep learning-based appearance modeling in the following sub-sections.

2.2.1 Classification-based deep CNN trackers

The simplest deep learning-based tracking approaches utilize deep convolutional neural networks as binary classifiers, where the main tracking task consists in distinguishing between the target object and background in each video frame. In general, feature extraction takes place in the initial CNN layers, while the classification process is performed in the last layers of the CNN model (e.g., [36–39]), but can also be performed in a separate machine learning model (e.g., in [40,41]). Support vector machines (SVMs) are particularly popular in this regard [40–43]. The described trackers are essentially end-to-end deep networks that directly predict the presence of target objects in the video frames under consideration. Some works [44] propose training CNN classifiers online to perform tracking. However, since the amount of training data that can be obtained online for training is naturally small, online training approaches are subject to severe overfitting. To overcome this limitation, approaches [36,41,45] have been proposed to train CNN models offline with external images or videos. Typically, to extract useful features, many approaches utilize off-the-shelf deep CNN models that have been pre-trained on large-scale datasets. Because of the domain shift problem [46], it is often necessary to fine-tune models using data from the target domain. In [45], Wang et al., for instance, performed offline training on large-scale image datasets and then fine-tuned online. [41] utilized pretrained CNN models and performed online learning using SVM.

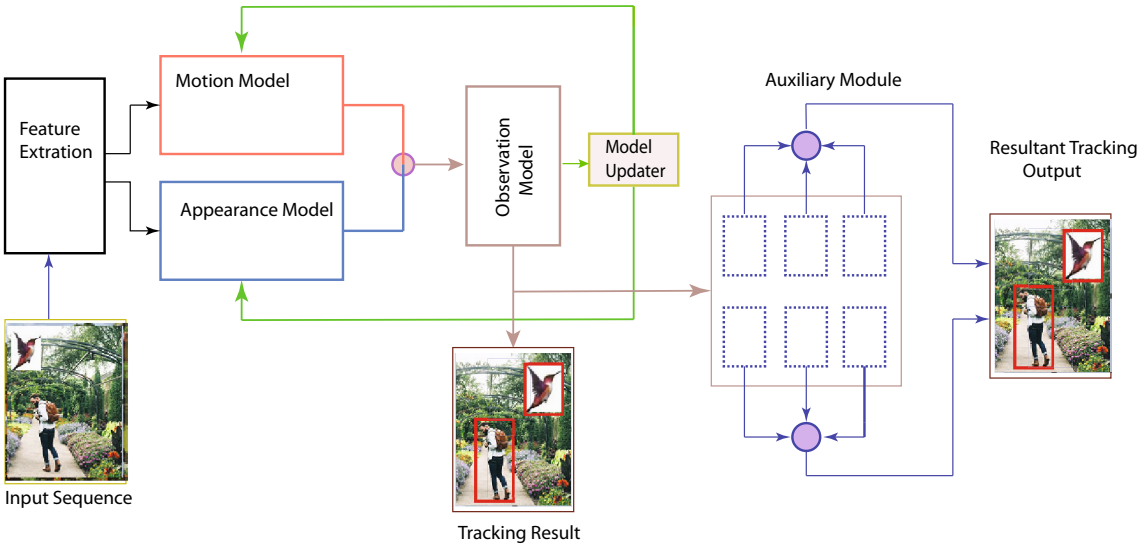

Fig. 1 General structure and workflow of object tracking algorithms

The main advantage of classification-based tracking approaches is the simplicity of the problem formulation and the ability to work seamlessly with large-scale datasets using pre-trained image classification models. However, because of this simplicity, it is often limited to SOT task or less challenging MOT scenarios.

2.2.2 Correlation filter-based trackers

Correlation filter (CF) [47] approaches have been widely used in deep learning-based tracking [48–53]. Correlation filter kernels utilize appearance features extracted by CNN models to perform cross-correlation to associate and locate target objects. The technique translates complex time-domain operations to simple, element-wise multiplications in the Fourier domain. Because of this simplicity, computational efficiency and high performance, correlation filters-based methods have becomeone of the most popular approaches for matching and locating target objects.

2.2.3 Tracking-by-detection approaches

Currently, the overwhelming majority of deep learning-based tracking algorithms are based on the so-called tracking-by-detection approaches. They perform tracking in two stages—detection and association. This involves first localizing target objects with object detectors in the initial frame and then finding correspondences among the initial detections and future detections in each subsequent frame. Such a decoupled formulation of the tracking problem allows to effectively tackle each of the two tasks—object detection and temporal association–separately through different robust appearance modeling techniques. A detailed scheme of this framework is shown in Fig. 2. We describe the important tasks below.

(a) Detection. The first step in tracking is usually to initialize the detector with a bounding box that describes the current location of the target. This can be accomplished manually or automatically [15]. For automatic initialization, bounding box proposals for probable target locations are generated by pre-trained object detectors. Many approaches utilize standard CNN-based object detectors such as Faster R-CNN (e.g., in [54]), SSD (e.g., in [55]) and YOLO (e.g., in [56]). Since two-stage detection frameworks such as [54] are generally more robust than their one-stage counterparts [57] like SSD [55] and YOLO [56], they are more commonly used in applications where robust performance is critical and computational efficiency is not a major concern. Two-stage detectors (shown in the diagram in Fig. 2) compute region proposals and align the encompassed features in the first stage and then predict their categories in the second stage. In contrast, one-stage detectors classify features in the first stage straightaway. While standard object detection pipelines are commonly used for the detection task, many recent approaches [56] have proposed to augment these detectors with additional robust appearance models or utilize custom detection models (e.g. [58–60] for robust object detection). Automatic target initialization requires that arbitrary targets in the initial frame be accurately detected and, in the case of MOT, appropriately assigned identifiers. However, owing to problems associated with complexity of real-world tracking settings, detections may be poor for arbitrary objects. To alleviate this problem, many approaches utilize advanced appearance modeling techniques to enhance the detection accuracy and robustness. This allows to more effectively detect the target objects at the initialization stage, as well as

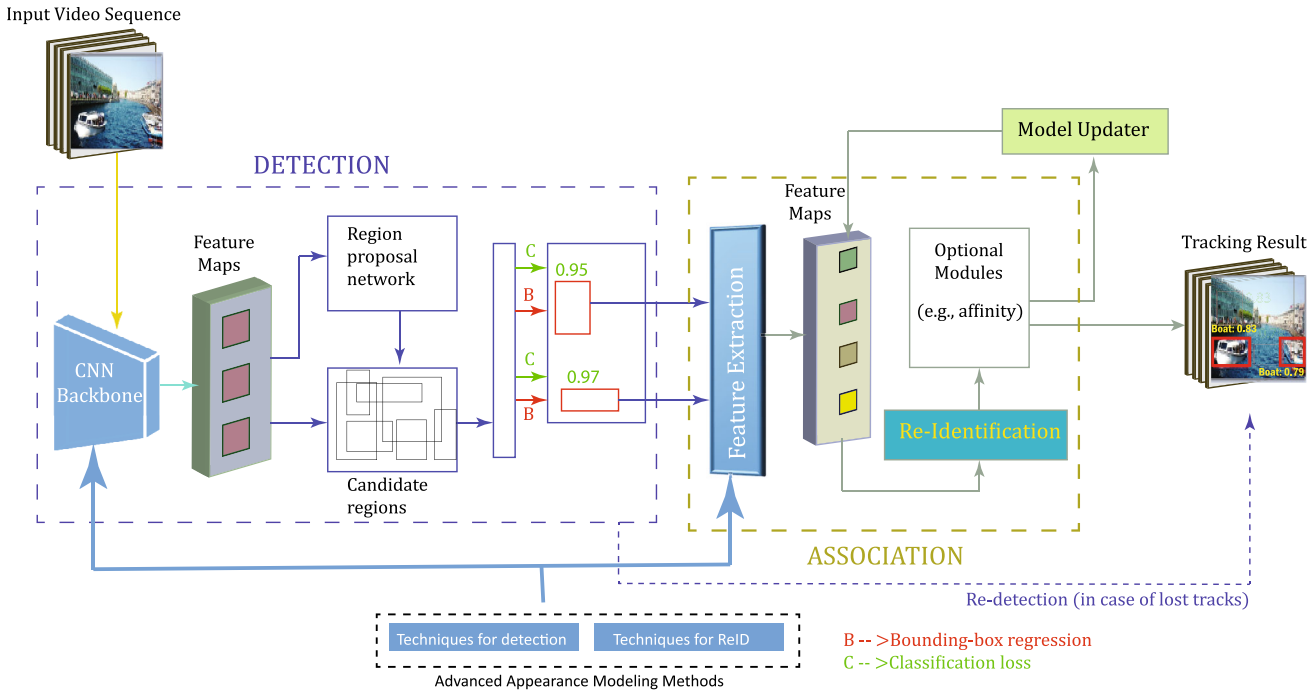

Fig. 2 A generalized tracking-by-detection-based appearance modeling framework for robust visual tracking. It incorporates a two-stage detection scheme and data association sub-models as the main components. Data association primarily involves re-identification and affinity matching. As depicted here, different techniques are utilized to encode robust features for detection and for extracting invariant features from the detected bounding boxes for re-identification

perform re-identification (Re-ID) and re-detections in subsequent frames regardless of appearance variations.

(b) Re-identification. For each of the generated bounding boxes, visual features are extracted for use by a re-identification sub-network. In general, the regions within the detector bounding boxes are taken as positive training samples, while regions outside the bounding boxes are considered as negative training data. Thus, for each object, there usually exists only one positive target sample and potentially infinite negative ones. To solve this sample imbalance problem, some authors [61] have proposed to sample several positive examples around the vicinity of each bounding box. However, this degrades the quality of positive samples and ultimately contributes to poor performance. State-of-the-art approaches tackle the data imbalance problem by utilizing advanced appearance modeling techniques that allow to encode invariant representation of visual features using one accurate positive sample generated by the detector. While both detection and re-identification need good features for robust performance, they typically utilize different kinds of features [62]. The detector performs inference at the object level (i.e., using high-level semantic features that are obtained from deeper layers), while re-identification operates on invariant, low-level features from shallower layers that allow to encode intra-class variations. Thus, it is common to adopt two different sets of robust feature representation schemes for detection and re-identification.

(c) Auxiliary tasks. In many state-of-the-art tracking algorithms, especially in MOT, additional subtasks such as affinity computation are frequently used to improve tracking performance in challenging situations. Several different techniques [63–66] have been proposed to enhance data association or compute affinity for matching candidate objects with target instances. In the literature, some of the most popular techniques include Bayesian methods (e.g., [63]), deep reinforcement learning (e.g., [64]), Hungarian algorithm (e.g., [66]), particle filter (e.g., [145] [67]) and linear programming (e.g., [65]). Most recently, a number of authors [68,69] have proposed replacing these data association techniques based on heuristics with differentiable neural network sub-models.

As a result of recent advances in robust visual feature embedding techniques, a number of authors [70,71] have proposed using detections alone to accomplish object tracking. These approaches formulate the tracking problem as a frame-to-frame re-identification task. For instance, in [70], Bergmann et al. proposed a detector-only tracking approach that outperformed more complex models in a range of multiple object tracking tasks on standard benchmarks. In this case, the re-identification model was trained offline and employed to perform detections in the tracking process.

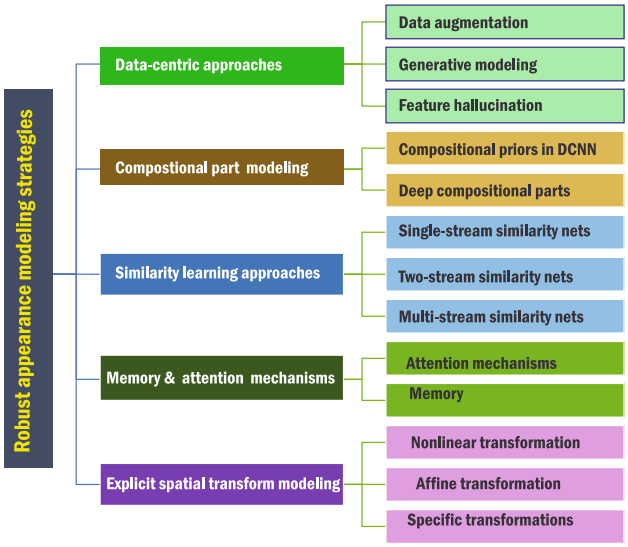

Fig. 3 Taxonomy of advanced deep learning-based appearance modeling methods discussed in this paper

However, Jia et al. [72] suggests these approaches may be weak against adversarial attacks. Other recent approaches [62,73–76] have suggested jointly performing detection and tracking as a one-step process so as to better leverage both processes. For instance, in [73] Feichtenhofer et al. applied both detection and tracking as complementary processes for better performance. That is, trajectory predictions are used to refine detections and vice versa.

2.2.4 Advanced deep learning-based appearance modeling techniques

As outlined above, classical deep learning techniques are inadequate for appearance modeling in complex domains. To overcome this limitation, several lines of work have been proposed. In the following sections, we explore these approaches in detail using the taxonomy depicted in Fig. 3. These advanced appearance modeling techniques facilitate invariant feature representation that enables accurate and robust detection and re-identification.

3 Data-centric approaches

One of the most important factors that accounts for the astounding success of deep learning approaches in machine vision tasks is the availability of large and rich annotated training data. However, visual tracking tasks usually involve dealing with arbitrary objects in an online manner, where the possibility of obtaining relevant training data in sufficient quantity is severely limited. This limitation often results in relatively poor generalization performance of deep learning methods in object tracking tasks as compared to other machine vision settings like object classification. Many authors have proposed to alleviate this problem by utilizing various techniques to generate large and diverse training data that cover all possible appearance conditions.

3.1 Manual data augmentation

An important problem in many practical machine vision applications is the class imbalance problem [77,78]—a situation where training data is excessively skewed towards some particular categories. More specifically, in object tracking settings, this is usually a relative scarcity of positive instances compared to negative ones [79,80]. This presents enormous difficulties to creating appearance models that are robust against different view conditions. One way to address the problem is by employing manual data augmentation techniques [81,82]. These approaches focus on manually generating more diverse positive samples that capture all possible appearance variations in the particular setting. In [81] Bhat et al. exploited different data augmentation strategies where positive samples are manually created to improve the robustness of the resulting model in object tracking tasks. Approaches utilizing synthetically generated data have also been suggested [83–85] to provide diverse positive samples for improved generalization performance. Augmenting training data with negative samples has also shown to be effective in visual tracking. For instance, in [79], Zhu et al. proposed to improve the discrimination of targets from semantic background (i.e., other objects in the scene) by introducing hard negative samples into the training data through data augmentation.

Despite the fact that manual augmentation techniques have successfully been used to improve robustness of deep learning models in many machine vision domains, they have limited scope of application in visual tracking domains. The main reason for this limitation is that in many visual tracking tasks, target objects are not usually known a-priori; the appearance details are determined online only upon initialization, making it challenging to apply manual augmentation in the tracking process. In addition, the process of creating new samples using manual data augmentation techniques such as [81,82] is notoriously time consuming and can only be achieved by an expert with an extensive knowledge of the end application domain. Moreover, in many cases, the manually created data may not be semantically rich and meaningful to capture complex appearance variations in real-world settings. This can lead to poor performance in practical applications. These issues are addressed by generative modeling techniques that perform automatic data augmentation.

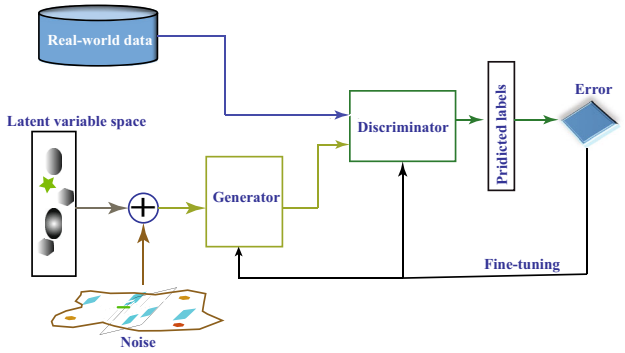

Fig. 4 Generalized architecture of Generative Adversarial Network (GAN) . The generator takes as input random noise and transforms that into an image samples. The discriminator computes the classification loss and propagates it through to the generator

3.2 Generative modeling

A recent trend is to employ deep learning algorithms to automatically generate relevant training data to extend and diversify the original data. The main idea of generative modeling is to automatically create "artificial" data that contain predictive features as the tracked instance. The use of generative methods is desirable both from the point of view of their ease of implementation and from the point of view of their scope of application; models based on them are generally invariant under more diverse transformations of the target appearance, including complex nonlinear transformations which cannot be generated manually.

3.2.1 Automatic data generation based on Generative Adversarial Networks

The most popular class of approaches [80,86,87] for generating training data in object tracking domains is based on generative adversarial network (GAN) [88] architectures. The GAN approximates the distribution of the input data by sampling from that distribution. This, thus, overcomes problems of sample scarcity and data imbalance. A GAN is a composite neural network made up of a generator and a discriminator that are designed to compete with each other (Fig. 4). Usually, the discriminator is simply a standard CNN classifier whose task is to distinguish generated images from real ones. The generator's goal, on the other hand, is to generate as realistic as possible data that makes it difficult for the discriminator to discriminate.

A repeated process of generation and discrimination is carried out until convergence, when the generator learns to synthesize data that is so close to the input sample that the discriminator is unable to distinguish between the real and generated data.

In many machine vision settings, the goal of generative modeling is often to generate artificial samples that look as realistic as possible. In contrast, common implementations [80,90–92] of GANs in object tracking domains are designed to accomplish feature-level generation. This typically consists in first generating an output mask from convolution features and then using it to alter output features from training images in a way that produces artificial variations which are subsequently learned through adversarial training. In [90] Yin et al. proposed a GAN-based tracker which generates random masks adversarily with the help of cropped images placed around input image samples. The masks are then used to produce richer appearance variations that are learned by the model. [91] employs a CNN classifier that leverages attention mechanism to enhance the robustness of the network in [90] against appearance drifts.

Most of the recent GAN-based approaches (e.g., [80,92]) additionally exploit strategies to select a subset of features—the most robust with respect to the given context—out of the generated samples. The goal is to improve performance by retaining only the most robust features of the tracked instance which can then be used to train a final classifier. In [92], Javanmardi et al. argued that randomly masking out features to produce appearance variations, as implemented in [90], for example, may lead to potential loss of useful information which may be disadvantageous. To address this problem, they proposed to generate an adaptive mask that aligns the most informative features of local image regions of the most recent scenes with that of earlier target images. In [80], the authors proposed a tracker that augments positive samples through adversarial learning. They incorporated a generator-discriminator pair into a conventional CNN architecture, specifically a VGG-M model [93]. They utilize the generator to generate masks which are subsequently used to adaptively mask out input convolutional features from positive samples. This procedure produces multiple output features corresponding to different appearance changes. Further, they trained a discriminator to be robust to these visual appearance variations.

There are a number of GAN-based approaches (e.g., [89,94–96]) that formulate the tracking problem as a similarity learning problem. To provide robustness to more diverse tracking problems, Han et al. [89] utilized two separate GAN modules to handle sample- and feature-level generation (Fig. 5). First, a sample GAN (SGAN) model generates diverse training samples which are then fed into a feature GAN (FGAN) that learns to generate diverse features for different appearance conditions such as deformations, occlusions and motion blur.

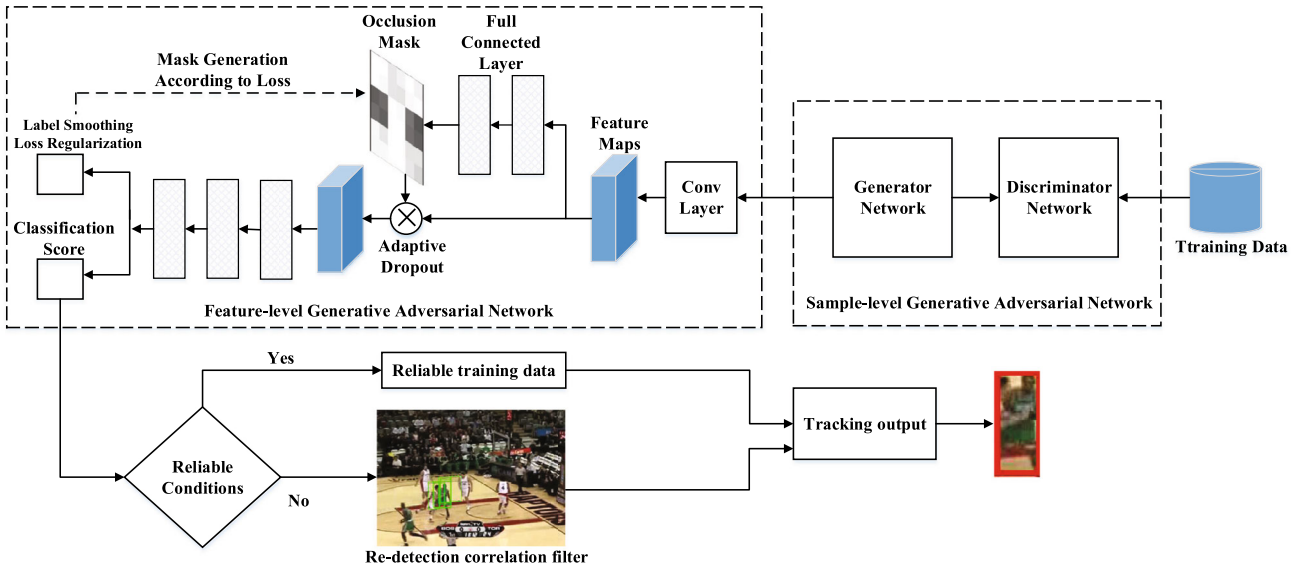

Fig. 5 Generative adversarial network (GAN)-based appearance modeling approach proposed in [89]. It utilizes sample-level data generation sub-model based on the conventional GAN architecture and feature-level generation sub-model to diversify features by occlusion masking

3.2.2 Other generative modeling methods for automatic data augmentation

Although GANs remain the predominant approaches for generative modeling, the use of other generative modeling techniques in robust image feature generation has been growing over the years. Researchers have explored a number of related techniques to improve the quality of feature representation and generalizability. Most notably, approaches based on autoencoders [100–102] and variational autoencoders (VAEs) [95,98,99,103] have demonstrated good performance. To address overfitting problems arising from small training data, Liu et al. [102] employed an auto-encoder subnetwork to impose constrain on the loss function. In [98], Kim et al. used a conventional variational autoencoder (VAE) to implement a deep learning model for learning rich spatial information about objects. They demonstrated the use of conventional variational autoencoders (VAEs) in generating rich appearance features for tracking. In [99], Lin et al. used a custom variational autoencoder consisting of three encoder branches to extract visual features at different semantic levels for video object segmentation and tracking. The extracted visual features are used to enhance Mask R-CNN segmentation robustness in tracking. The branches provide different semantic levels of generalization: the input layer is sensitive to simple image features such as lines and their orientation in certain areas of the visual area, while the response of other layers is more complex, abstract, and position-independent of the image. Similar functions are realized in the cognitron by modeling the organization of the visual cortex. Methods have also been developed that combine different generative schemes to produce better appearance features. For example, Wang et al. [95] proposed a generative modeling technique using the earlier developed Siamese Instance Search Tracker (SINT) [104] as a backbone model. Their generative modeling approach uses two different subnetworks—Positive Sample Generation Network (PSGN) based on VAE architecture to generate and augment positive samples, and a so-called Hard Positive Transformation Network (HPTN) based on deep Q-network to create occlusion and deformation patterns that can be learned by the discriminator. The final component, the Siamese network, is used to infer the similarity between the target sample that is initialized in the initial frame and candidate samples in subsequent frames. Common generative modeling-based trackers and their constituent components are summarized in Table 1.

3.2.3 Feature hallucination techniques

In contrast to the aforementioned methods such as [80,90–92,94–96] which aim to improve robustness by generating feature masks to increase the diversity of training data, some of the more recent generative modeling approaches, known as hallucination methods (e.g., [97,105,107,108]), are aimed at directly transferring different visual phenomena from training data to unseen data, thereby generating novel views. The concept of hallucination has been motivated by the ability of humans to imagine new visual contexts from observations [97,105,106,108]. The main idea is to learn image transformations from exemplar images and then apply this knowledge to unseen object classes in novel contexts. These techniques, therefore, allows to learn robust visual feature representations that can be applied across multiple domains and tasks. These approaches generally utilize an encoder-

Table 1 Representative generative modeling-based trackers and their construction

References	Constituent generative modeling sub-models	Function of generative modeling components	Base model
[94]	Custom GAN (generator and discriminator)	Generator generates "similar" and "dissimilar" samples for discrimination by the discriminator	Siamese network
[92]	Standard GAN (generator and discriminator)	Generates and discriminates positive samples	MDNet [36]
[97]	Encoder-decoder network (HAT)	Hallucinates novel views	MDNet [36]
	Selective deformation transfer (SDT)	Selects right transformations for transfer	
[80]	Custom GAN (fully connected CNN as generator and a CNN classifier as discriminator)	Generates and discriminates positive samples	VGG [93]
[90]	Standard GAN (generator and discriminator)	Generates and discriminates positive samples	VGG [93]
[95]	VAE (Positive Sample Generation Network)	Generate positive samples	
	Deep Q-network (Hard Positive Transformation Network)	Create occlusions	
[98]	Standard Variational Auto-Encoder	Generates robust features for training a base model	Siamese network
[99]	Encoder	Constructs compressed features	Mask R-CNN backbone
	Proposal decoder	Extracts high-level features	
	Auxiliary decoder	Extracts low-level features	
	Augment decoder	Aggregates multi-level cues	

decoder scheme where the encoder learns transferable image transformations from pairs of exemplar images (e.g., different poses, scales, illumination conditions) of the same class, and the decoder's task is to learn to apply these learned transformations to new categories. For instance, in [90] Wu et al. proposed to generate new image samples using an encoder-decoder network based on what they termed Adversarial Hallucinator or AH. The hallucinator generates transformed images which are then used to train CNN classifiers. In addition, they incorporated a so-called selective deformation transfer (SDT) sub-model to select and transfer the most relevant transformations to unseen contexts. In [106], Wei et al. proposed a re-identification framework, PTGAN, that uses a GAN to transfer persons in labeled datasets to novel styles (i.e., appearance conditions such as different backgrounds, illuminations and view angles), while preserving useful features that define the identity of the persons. Amirkhani et al. [109] employ visual style transfer technique to compose new training dataset from an existing dataset and combined them to achieve a larger and more diverse data for training object trackers. The various data augmentation methods described in this section are summarized in Table 2.

4 Compositional part modeling

A part model of an object is understood as the set of simple geometric primitives that provides a meaningful representation of that object. The rationale for this approach is based on the fact that appearance variations of object parts are generally much less drastic than the possible variations of the object as a whole. Hence, simpler models and smaller datasets can be used to effectively obtain robust models. Many different approaches are used to encode compositional parts as information priors in deep learning pipelines (Fig. 6). In general, object classes are represented as mixture of parts, with each part representing specific appearance instances such as different viewpoints [110,111], size variations [112], pose instances [113] or occlusion extend [114]. In many tracking applications (e.g., [115,116]) compositional part models serve to enhance robustness of object detectors. The main strength of compositional parts is their ability to handle complex transformations such as nonlinear deformations and significantly occluded objects, even if trained without including transformed examples [114,117]. Two broad strategies of part-based approaches can be identified: approaches that explicitly formulate part models as representation priors and those based on deeply learned parts. In the first family of approaches, object parts are manually modeled independently before using some algorithm, usually

Table 2 Summary of the major data augmentation approaches

Method	Design[a]	Main Purpose	Major shortcomings	Works
Basic data manipulation	Manual	Increase diversity of existing data by applying transformations to produce more positive (target) or negative (background) samples	Limited to situations where desired categories already exist; laborious process	[79,81,82]
Data synthesis using computer graphics tools	Manual	Generate data (from scratch) in situations where no training data exists	Domain shift between synthetic and real data; tedious process	[83–85]
Generative modeling	Automatic	Expand training samples using examples from similar categories	Requires large amounts of training data; no reliable metrics to determine quality of generated samples	[80,87,89–92,100]
Hallucination	Automatic	Transfer the visual style of data to new domain or context	Typically requires examples from the target domain which may not be accessible in some situations	[97,105,106]

[a]Design denotes the method of composing the augmentations

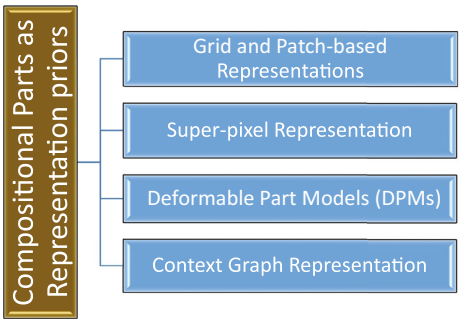

Fig. 6 Taxonomy of part modeling approaches based on representing compositional parts as information priors in deep learning pipelines

a machine learning model, for feature classification. In the second case, part-level representations are directly learned end-to-end from deep CNN feature maps.

4.1 Part models as representation priors in deep CNNs

A large number of approaches [118–123] propose to explicitly model compositional parts as representation priors in object detection and tracking pipelines. These approaches usually approach feature learning as a two-step process; building informative, invariant mid-level features as vectors of compositional parts and using deep CNN models to learn robust representations for these parts. The simplest approaches to compositional part modeling utilize natural images that are artificially divided into grids or smaller patches [119,121,122,124,125]. In [122] for example, Tian et al. proposed a part-based pedestrian detection technique utilizing a pool of human body parts defined as a rectangular human body grid and then trained a CNN classifier to learn relevant features for each of these parts by sliding filters over the entire grid. Another common method for compositional part modeling is to segment training images on the basis of low-level pixel properties—superpixels [126,127]. This approach is based on the intuition that pixels sharing common visual characteristics in a given region may represent a unique semantic context. Superpixels are commonly defined by clustering algorithms [128]. However, newer approaches [129–131] have proposed learning superpixels end-to-end with deep neural networks.

More sophisticated compositional part modeling techniques such as [110,111,117,132–134] encode additional information such as spatial dependencies among constituent parts. To handle object deformations, for example, deformable part models (DPMs) [135,136], encode deformations from part displacements. DPMs are often used to help with the object detection sub-task, where they help to encode robust features in region-based CNN detection models [115,116]. For instance, in [137] Ouyang et al. used deformable part models to generate region proposals containing deformable object parts. After this, a dense subgraph discovery (DSD)-based filter is used to select the most useful region proposals.

Richer part-based methods model the structural features of an object based on its constituent parts and their spatial

relationships [133]. In this regard, structural information of objects in images is represented using simple sub-entities that are themselves described by even simpler entities. The most advanced part models (e.g., [138–141]) are typically described by hierarchical graph structures in the form of nodes and links which encode more detailed information about the spatial properties of the constituent parts, including local interactions. In [139], for instance, Wang et al. proposed an appearance model for object tracking using a graph-based architecture consisting of multiple CNNs to encode visual features of local parts. The learned features are then fused using a regularization framework. Similarly, in [138] , Nam et al. employed separate CNN sub-networks in a hierarchical, tree-like arrangement to model the appearance of different parts. In their implementation, the edges of the structure characterize the structural relationships that exist among the different parts (represented by the different CNN sub-networks). To simplify the representation, some graph-based approaches (e.g., [142,143]) utilize superpixel information to segment images into parts which are then defined as graph elements.

Despite the aforementioned advantages of using information priors in the form of compositional parts, the approach has a number of significant drawbacks. First, object tracking based on parts results in the loss of high-level information, thereby reducing performance in some cases. Second, building rich part models is usually a labor-intensive and time consuming process. Another area of difficulty when using explicit part models as representation priors relates to the inability of human experts to manually identify good parts that are optimal for visual recognition tasks. In view of these limitations, several authors propose to learn part representations automatically in an end-to-end manner.

4.2 Deeply learned quasi-compositional part representations from mid-level CNNs features

In [144–146] it was shown that in deep convolutional neural networks, part-level information is present in the mid layers and that extracting features from these layers could provide contextual hierarchy in object representations. This concept has two main advantages. First, it does not generally require additional model parameters since these mid-level features are mined from existing layers of the network. Also, the requirements for adapting filters or for exploiting complex network structures for learning invariance is eliminated, thus providing a more simple approach to appearance modeling. Inspired by this finding, a large number of recent approaches [50,147–154] exploit this idea to design end-to-end deep CNN models to learn quasi-part representations directly from image-level data. These methods unify the processes of part modeling and feature representation by jointly extracting part-level features from deep CNN layers and learning suitable representations from the extracted parts. In [50], Ma et al. used features from early CNN layers to encode more nuanced spatial details while employing the last activation layer to capture object semantics. Many approaches employ special strategies such as dedicated compositional part filters [153,154], unsupervised clustering [155,156], special activations [157–159] or pooling techniques [153] in selected CNN layers to learn high level compositional parts. For instance, to overcome the limitations of conventional pooling techniques like average pooling and max pooling in encoding part-level information, Ouyang and Wang [160] proposed a part-based CNN model that incorporates a deformation layer between the fully connected layer and the last convolutional layer to capture part deformations. Ouyang et al. [153] extended this concept by introducing deformation- or def-pooling which is designed to replace conventional pooling layers at multiple locations within a deep CNN network.

More recently, advanced compositional-part-modeling approaches (e.g., [151,153,154,161,162]) that utilize complex network architectures consisting of several independent sub-networks have emerged. For instance, Wu et al. [162] propose an approach for robust visual tracking using multiple deep learning sub-networks to separately observe different sub-regions of the input frames. Each sub-model is designed to learn specific local features from a target sub-region. Qi, et al. [148] employ several independent CNN trackers to learn mid-level spatial features from different convolutional layers. The predictions of these trackers are then adaptively fused by means of an online decision-theoretic learning approach using Hedge algorithm. An overall high-performance tracker is obtained based on the weighted sum of the predictions of all trackers. Yang et al. [154] proposed to integrate multiple CNN-based compositional part extraction modules, called P-CNN, into different layers of pre-trained CNN models—AlexNet [163] and VGG19 [93]. The P-CNN utilizes part filters which are optimized to select part-level descriptors from feature maps of designated convolution layers (i.e., layers to which P-CNN modules have been attached). In [151] Mordan et al. introduced "Deformable Part-based Fully Convolutional Network (DP-FCN)", which utilizes a (FCN) network [152] together with a number of custom extensions for part-level feature learning. The fully convolutional network is responsible for extracting task-specific features of each image class into feature maps. In addition, a deformable part-based region-of-interest (RoI) pooling layer encodes part-level representations of the resulting feature maps. The deformable RoI pooling layer partitions the image-level feature maps into $n \times n$ region proposals (i.e. square grids) and performs alignment of parts. The final extension, at the end of the whole structure, consists of two separate network branches that perform semantic classification and deformation-aware localization by exploiting the effects of part displacements. [153] proposed a deep CNN architecture

that jointly learns object deformation and part-level feature representations, as well as incorporating context information. The approach was implemented using the ZFNet architecture (proposed in [164]) as a CNN base model with additional branches consisting of part-level kernels and classification sub-networks. By changing the configuration of this CNN, different detectors are obtained, leading to variability, and hence better generalization performance in specific situations. In addition, the approach further enhances generalization by allowing the sharing of deformable parts among different object categories.

While deeply learning compositional parts from CNN layers can provide better generalization in unseen domains [147], they are typically less transparent compared to their explicit model counterparts, and ultimately suffer from the black-box syndrome [165] commonly encountered in deep neural networks. Another limitation pertaining to compositional part modeling in general is that the approach is not suitable for objects without distinct parts. Also, non-rigid object parts can often exhibit many different shape and form variations that completely diverge from the learned representations and thereby making it difficult for the approach to work well. Because of these limitations, in some scenarios, they may be more prone to catastrophic failures than traditional part-based models designed explicitly to account for anticipated conditions. The main approaches to modeling compositional parts in the context of object detection and tracking are captured in Table 3.

5 Similarity learning approaches

When tracking objects using deep learning methods, the network is required to learn very reliable visual features that remain stable under many different conditions. In this case, the deep learning model relies on learning invariant visual features from large datasets and then performing predictions based on matching corresponding features in candidate images to the previously learned representations. Since in most tracking applications the target appearance is captured only in the initial frame, it is often not possible to obtain sufficiently rich features for tracking. Many traditional deep learning approaches tackle this problem by training offline utilizing large-scale datasets before fine-tuning online on the specific visual tracking task. But this often requires performing parameter updates online using gradient decent, which is computationally expensive and generally too slow for most practical applications. The second option is to combine classical algorithms such as particle filters [166] and HoG-like features [167] with CNNs or to utilize specialized deep learning architectures (e.g., [95]) to encode robust object appearance. These techniques are often more complex, highly specific and require more prior knowledge about the target domain. All these considerations led to the widespread use of similarity learning algorithms [168,169]. Similarity learning trackers are typically offline trackers in that they learn similarity embedding completely offline using available datasets that are similar to the target domain.

5.1 General principles of similarity learning

Similarity learning approaches to appearance modeling differ from conventional deep learning methods in that they do not directly learn visual features for each object instance or category. Instead, they learn a function that predicts the similarity of input images. The decision boundary is defined by a similarity measure [170] which can be independently computed as a distance metric [171,172] or learned directly from input images [66,104,173] using a neural network. In place of the usual prediction error-based loss functions employed in traditional CNNs, similarity learning methods use special loss functions such as contrastive loss [174] to force semantically similar image samples to be embedded in close proximity while forcing dissimilar images apart. Another important task in similarity learning is to minimize the intra-class differences between objects while, at the same time, maximizing the interclass differences. One major challenge with distance metrics is in defining the right size of the distance, which must be large enough to include all intra-class appearance variations but small enough to exclude interclass appearance differences. Deeply learned similarity metrics solve this problem but they are often not transparent and may be subject to higher error rates when trained using insufficiently large data. To further enhance robustness, some approaches impose temporal constraints (e.g., [115]) or additional spatial constrains (e.g., [175,176]) on the definition of similarity metrics. The main idea in [175] and [176] consist in dividing images into sub-regions and then learning similarity measures for corresponding regions independently before combining the individual metrics to obtain a global similarity metric. Once a similarity is learned, the tracking process involves initializing the target object in the first frame and then performing exhaustive search in subsequent frames to locate the most probable region within the search area that might contain the target. Thus, re-identification in the context of similarity learning consists in finding a candidate region with the minimum distance within the threshold specified by the metric. The rest of this section explores common similarity learning approaches categorized into different network topologies and similarity embedding mechanisms.

5.2 Single-stream similarity networks

The simplest similarity learning approaches are based on single-stream networks [9,177–179]. They typically consist of deep convolutional neural network architectures that

Table 3 A summary of compositional part modeling methods and their major characteristics

Method	Description	Auto[a]	Comp.[b]	References
Grid or patch representation	Partitions training images into equally sized rectangular parts	×	Low	[119,123,124]
Deformable part modeling	Represents objects with their constituent parts as well as possible deformations and part displacements	×	High	[136,137]
Hierarchical graph representation	Employs more granular parts to represent objects in a scene and while encoding contextual relationships among the parts.	×	Very high	[138–141]
Super-pixel representation	Utilizes low-level pixel characteristics of the images to define parts	×	Medium	[126,127]
Composing parts from CNN feature maps	Mines compositional parts from intermediate CNN layers	✓	Low	[147,149,153]
Learning parts using a dedicated network per part	Employs multiple dedicated sub-networks to independently learn and aggregate different object parts	✓	Low	[161,162]

[a]Auto denotes approach where the construction of compositional parts is usually exclusively automated.
[b]Comp. denotes the relative complexity of the model design

employ contrastive loss at the end of the deeper layers to learn similarity embedding. In [9], Moujahid et al. proposed a single-stream similarity embedding network that uses soft cosine similarity metric to compute similarity. During tracking, the approach samples candidate locations around the initialized target and computes similarity for each candidate region. The region with the highest score is taken as the new target location. A major limitation of the method is that the model needs to make an assumption about the probable location of the target. For this purpose, a motion model is employed. In [179] Ning et al. proposed a single-stream similarity network which employs contrastive loss layer to implicitly learn the similarity from sample targets and background images selected by RoI layers. Despite its simplicity and closeness in structure to traditional deep CNN architectures, current literature emphasizes the use of more complex topologies such as two-stream and multi-stream networks for enhanced similarity encoding.

5.3 Two-stream Siamese networks

In recent years, visual tracking approaches using pairwise, deep similarity learning architectures based on two-stream and multi-stream networks have become very popular in many machine vision domains [180]. In particular, the Siamese network [181,182]—a two-stream network architecture—is currently the most popular visual tracking approach for solving most SOT problems. Their success in SOT is evidenced by the results of the annual Visual Object Tracking (VOT) Challenge, where the top-performing short-term trackers in recent years [30–32] have mostly been Siamese-based architectures.

A generalized architecture of the Siamese network is shown in Fig. 7. It consists of two identical CNN branches with shared parameters. The network is trained by feeding into the two branches a pair of similar (i.e., objects of the same class) and dissimilar (objects belonging to different classes) images. The features extracted by the two branches are compared and fused by means of a contrastive loss mechanism whose goal is to learn a similarity function to correctly predict object similarity given any pair of images. During tracking, one of the branches is fed with the initialized target (i.e., an image patch containing the object), while the other branch takes as input a search area encompassing the whole scene or part of it. Essentially, the search of candidate objects consists in shifting the exemplar patch over the entire search

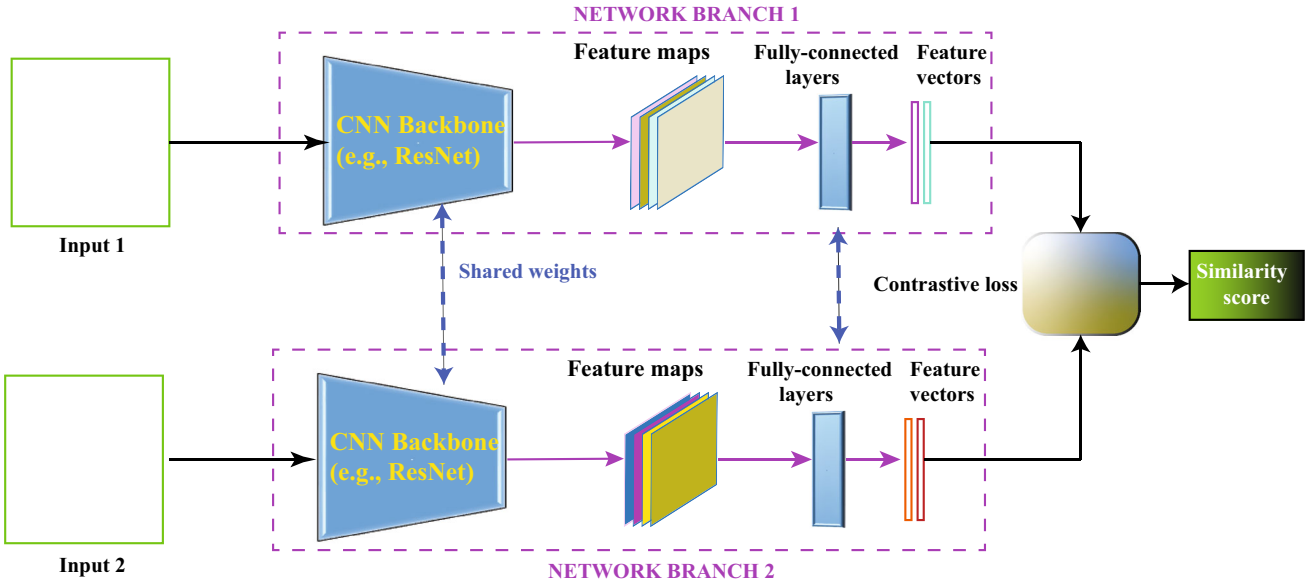

Fig. 7 General structure of Siamese network

area while computing similarity for each location. An extensive review of Siamese architecture is presented by Chicco in [180]. The author detailed several applications of Siamese networks.

In one of the pioneer works, Tao et al. in [104] proposed Siamese Instance search Tracker (SINT) based on conventional two-stream Siamese framework that employed Radius Sampling method proposed in [183] to sample candidate objects for tracking. In [184], Bertinetto et al. introduced SiamFC which employs a dedicated cross-correlation layer on top of the Siamese branches. In this case, the search for candidate targets during tracking is reduced to computing cross-correlation between the target patch and the search patch. Similar to [184], CFNet [185] utilizes cross-correlation layer to estimate similarity; but in contrast to SiamFC, CFNet additionally employs a correlation filter unit as a differentiable CNN module in the template image branch of the Siamese framework to help learn varying appearance cues. GORUN [186], on the other hand, employs a Siamese framework to learn target appearance features while applying fully connected CNN layers to fuse the extracted features. [187] proposed to use region proposal network (RPN) on top of a traditional Siamese architecture to perform object detection. Zhu et al. [79] extended the SiamRPN model by proposing DA-SiamRPN, which incorporates a so-called distractor-aware sub-module to transfer learned representations of semantic negative object interactions in complex scenes to the online tracking process. To handle out-of-view and full occlusion problems in long-term tracking, they also proposed a strategy to incrementally expand the search area to provide a global view in order to recover the lost object (through re-detection) once it reappears.

Some Siamese-based approaches propose to fuse features of different abstraction levels from multiple CNN layers [188] or learn low- and high-level features in separate Siamese networks [189,190] before combining the results for inference. In [189], He et al. proposed a special Siamese framework consisting of a double two-stream network structure. The network is made up of an appearance branch that extracts invariant visual features from shallower layers and a semantic branch that exploits deeper features to encode high-level semantic representation. The similarity scores for the two branches are computed separately in the training phase before being combined to obtain a final similarity result during tracking. The appearance and semantic branches are aimed at enhancing the network's discriminative and generalization abilities, respectively.

Fundamentally radical modifications of the standard Siamese architecture have also been proposed. Notably, Zagoruyko and Komodakis in [191] investigated a number of new Siamese network architectures, including a so-called pseudo-Siamese network. While Siamese architectures employ two identical CNN streams with shared weights, the Pseudo-Siamese architecture proposed in [191] employs two stream networks with unshared weights. According to the authors, the technique allows more parameters to be adjusted easily during training. The authors further extended this concept with the introduction of a so-called 2-channel network, which operates based on completely uncoupled two-stream networks. From the results of their studies, the performance of these different models seem to depend strongly on the specific application scenario. Despite their promise, these approaches have not yet been fully exploited in object tracking domains.

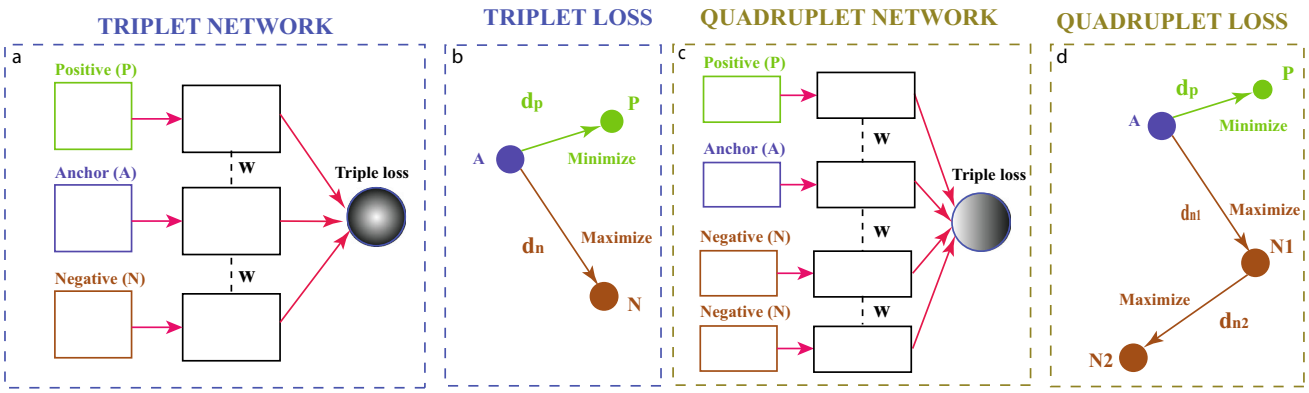

Fig. 8 Structure of triplet and quadruplet networks with their respective losses

5.4 Multi-stream similarity networks

Multi-stream networks are a special type of Siamese architectures that employ, typically, three (triplet networks) or four (quadruplet networks) CNN branches to learn image similarity. Multi-stream models provide more advanced feature embedding mechanisms than two-stream Siamese networks.

(a) Triplet trackers. Triplet networks [192–195] (Fig. 8a) are made up of three identical neural networks with shared parameters and are trained by using three groups of input samples at a time: a target instance *P*, a positive sample from the target class *A*, known as anchor or reference, and a negative sample *N* (i.e. a sample from a different class). Generally, a triplet network uses triplet loss functions [192] to learn similarity (Fig. 8b). The idea is to minimize the distance *dp* between the target *P*and the reference *A* and maximize the distance *dn* between the negative *N* and target *P*. During inference, the objective is to determine whether the input image at anchor channel is closer to the reference or negative sample. Thus, training with triplet loss allows to compare similarity in relative terms rather than simply determining absolute correspondence of two input images. This way, more expressive visual features are extracted compared to two-stream architectures [194].

(b) Quadruplet network trackers. Most quadruplet network trackers [196–198] employ quadruplet loss for similarity learning. For instance, Chen et al. [199], and Dike and Zhou [200] propose to use quadruplet networks with quadruplet loss that jointly learns similarity using the entire scene (search area) in addition to the three patches used in triplet network architectures. The quadruplet network (Fig. 8c) samples from four images consisting of a positive image *P* representing the target object; an anchor or reference image *A*, which is also a positive sample (i.e., an instance of the target object); and a pair of dissimilar images *N1* and *N2* that are different from *A* and *P* samples representing two negative instances. The quadruplet loss (see Fig. 7d) involves minimizing the distance *dp* between the positive sample *P* and the reference image *A*, maximizing the distance *dn1* between the negative instance *N1* and the reference *A*, and maximizing the distance *dn2* between the two negative samples *N2* and *N1*. Although conventional quadruplet networks use quadruplet loss, some new approaches have proposed using different loss combinations [196,198]. In [198] Zhang proposed a quadruplet network with shared weights using multi-task loss - a combination of pairwise (i.e., contrastive) loss and a triplet loss. The pairwise loss learns the similarity between an exemplar patch (reference image) and a search area (candidate image), while the triplet loss compares positive and negative instances against the reference image. By using these losses in combination, the relationship among the input samples is better exploited for robust representation. Similarly, Dong et al. [196] proposed a four-stream network and introduced a special loss function with both pairwise loss and triplet loss within the same quadruplet network architecture.

5.5 Approaches to online-learning with similarity models

A significant limitation of conventional similarity learning approaches is that the similarity embedding is learned completely offline and is generally fixed—further updates are often not applicable once the model is deployed online. The visual appearance changes inherent in most tracking scenarios, especially in long-term tracking tasks, make it challenging to achieve robust performance with these models. Consequently, to enhance robustness in complex scenarios, some approaches resort to incorporating robust motion models to complement predictions [201]. Another common solution is to embed Correlation Filters (CF) into the Siamese network (e.g., in [185]) to handle appearance variations online. Recently, several online learning mechanisms [185,193,202,203] have been proposed that allow Siamese networks to update learned appearance embeddings during the tracking process. [203] uses an LSTM-based neu-

Table 4 A summary of compositional part modeling methods and their characteristics

Architecture	Main principle	Mode[a]	Typical loss function	References
Single stream	Extracts and compares target with non-target (background) regions of the same image	Online	Contrastive loss	[177,178]
Two stream	Computes similarity of target and template images by performing cross-correlation of search and template input streams	Offline	Contrastive loss	[183,184,186]
Three stream	Compares the similarities between a target and a different instance from the target category on one hand, and between the target and background on the other	Offline	Triplet loss	[193–195]
Four stream	Compares the similarities between a target and three different samples: two dissimilar background samples and a positive instance from the target category	Offline	Quadruplet loss	[196,199,200]
	Combines a two-stream and a three-stream sub-networks into a composite, four-stream architecture	Offline	Contrastive and triplet loss	[196,198]

[a] Mode denotes the mode of training (i.e., either offline or online) that is natural for the particular method

ral network to determine when updates are required and then performs updates by modifying the appearance features stored in external memory. In [193], Liu, et al. extended the SiamFC model proposed in [184] from two-stream network to a three-stream network in which the third stream is used for online model update, while the other two streams are used in the usual way to learn similarity embeddings. In addition, the network includes a Faster R-CNN-based detector known as localization network that allows it to re-establish a lost target. Similarly, Shi et al. [204] uses a triplet net extension to improve both SiamFC [184] and SiamCAR [205] through online model updates. Siamese networks are also increasingly being used in MOT as part of a more complex architecture to perform specific tasks in the tracking pipeline—for example, feature extraction [65,206,207], data association [208] or affinity computation [209,210]. The important properties, topologies and operating principles of similarity learning models are presented in Table 4.

6 Memory and attention mechanisms

An emerging trend in visual appearance modeling for object tracking tasks is the increasing use of memory and attention to improve performance. The concept of attention [211] is based on selective processing of input signals to enhance robustness and efficiency. Since different features have different discrimination and generalization abilities [212], utilizing all visual features with equal priority for visual tasks such as tracking is inefficient and may produce sub-optimal results. Visual attention [213–215] provides a mechanism to adaptively select and process the most semantically useful features for a given task while at the same time ensuring compactness and efficiency of representation. On the other hand, memory [203,216,217] endows the model with the ability to preserve learned representations over time. Memory (e.g., [218]) and attention mechanisms (e.g., [219,220]) have also been proposed as a means of incorporating context to enrich visual representation in object detection and tracking tasks. Chen and Gupta in [218] proposed Spatial Mem-

ory Network (SMN) to characterize contextual relationships among objects in images. Li et al. [219] proposed to model global and scene-level contexts using Attention to Context Convolutional Neural Network (ACCNN). Most attentional networks are implemented using feedback architectures such as RNNs. By virtue of their feedback arrangements, RNNs are also naturally endowed with memory. Beyond this natural occurrence, memory and attention do often perform complementary roles in machine vision tasks. In particular, since memory capacity is often limited, attention can enable selective storage of relevant information. Conversely, recall of stored information can also leverage attention to enable fast and efficient retrieval of information.

6.1 Attention in visual tracking

The attention mechanism works by adaptively re-weighting network parameters so as to prioritize more relevant features or relevant areas of interest for subsequent processing. The original work on visual attention—proposed by [211]—use attention to enhance the computational efficiency and at the same time increase the robustness of deep learning models in classification tasks. Attending to specific objects locations in large scenes can also be used to enhance visual search in challenging object detection tasks. This has been demonstrated with impressive results in [221]. Attention mechanisms [222–224] are recently being widely used to develop robust models for online trackers. They are able to adapt trackers to visual appearance changes of target objects over long time periods. Kahoú et al. [222], for example, implemented attention mechanism using RNN-based framework that performs spatial "glimpses" on relevant and informative regions of a scene. For target localization, the model uses a binary classification module to classify image features at the various locations. In [222], Kosiorek et al. utilized both spatial and feature attention mechanisms to allow a deep learning network to search in the right regions of a scene as well as select relevant features that are important for the tracking task at hand.

Recently, approaches based on modified RNN architectures like Long Short-Term Memory networks (LSTMs) [225,226] and Gated Recurrent Units (GRUs) [207,227], have been introduced. They allow deeper models to proces longer video sequences without the effects of vanishing gradients. In [227] two GRUs were used within a Recurrent Autoregressive Network to separately learn visual appearance and motion models. Instead of conventional recurrent networks based on RNNs, an increasingly large number works [121,213,228,229] propose to use special CNN configurations to learn different types of attentions. For instance, Stollenga et al. [230] implemented an attention mechanism by using special feedback arrangements constructed on the basis of Maxout networks [231]. In their approach, the synaptic weights of the feedback connections are learned using reinforcement learning techniques. This is done so as to enable the tracking model adapt its convolutional filters to important features present in the input images. In [59], Chu et al. proposed to use spatial graph transformer for learning attention.

More recent works (e.g., [59,232–235,237]) have explored the use of deep neural networks based on transformer [236] architectures as an alternative method of encoding attention in visual tracking models. In contrast to RNN-based attention models which utilize feedback in recurrent network topology to process information sequentially, the transformer employs feedforward attention blocks within an encoder-decoder structure. They can process larger amounts of data in parallel and model relatively longer-range dependencies. This allows them to learn inherent interdependencies between different entitiesin different parts of an image to help model the global context of the underlying scene. TrTr [232], for instance, incorporates transformer units within an encoder-decoder network that utilizes self- and cross-attention mechanisms to model contextual relationships between template and search image features in a single object tracking framework. TransTrack [235] proposes a transformer-based query-key method for multiple object tracking that is capable of effectively detecting and tracking new objects that appear in the scene during the tracking process. It employs two decoders—one for object detection and the other for propagating object features to the following frame—and a single encoder for learning robust feature maps through attention. The feature maps serve as input queries (object and track queries) for the decoders. That is, one decoder predicts bounding box detections using object query, while the other one aims to estimate the current locations of features from previous frames with the help of the track query. This allows the model to identify new objects that were not previously present in the scene. Trackformer [237] uses single encoder to learn both object and track queries and matches tracks entirely using self- and cross-attention operations. Approaches based on transformer architectures are presently one of the most impressive visual tracking models.

6.2 Long-term memory in visual tracking

The memory in RNN-based approaches (e.g., [203,217] does not provide long-term storage, as these models do not contain actual memory (i.e., storage). To address long-term storage needs, some authors (e.g., [215,238,239]) have proposed various techniques to enhance the information storage capacity of deep learning models. Chanho et al. [215], for example, proposed to increase the information storage capacity of conventional LSTM methods using Bilinear LSTM. Chen et al. [238] proposed a dedicated memory mechanism, referred to as Long Range Memory (LRM), to cache pre-

viously extracted local and global features as intermediate features for re-use by later frames. However, the ability to retain information over long term periods requires actual storage resources which are absent in approaches based on neural networks. A number of works [203,240,241] have proposed using explicit memory that provides reading and writing capabilities to deal with visual appearance variations over long periods of time. With this approach, the storage capacity of deep neural networks can easily be enlarged by increasing the size of external memory. In [203], Yang and Chan proposed Dynamic Memory Networks to overcome the problem of low capacity of LSTM-based approaches. Instead of keeping object appearance information as weight parameters in deep neural networks, the proposed approach stores visual feature information in external memory and retrieves relevant appearance details as needed. Appearance changes are handled by updating the stored information in memory. Because the method uses external memory, long-term appearance variations can be stored. The approach employs LSTM to control the writing and reading of information into and from memory. In addition, a spatial attention mechanism is used to direct the LSTM input to the probable locations of the relevant target. In [240], Deng et al. proposed an external memory to store features extracted from detections (i.e., features located within the bounding boxes) in a video sequence to be subsequently combined with features from later video frames.

7 Approaches for learning spatial transformations

A prevalent problem in object tracking settings is the apparent variation in objects' visual appearances emanating from phenomena such as non-rigid deformations, changes in object proximity and camera view angles, rotations and pose variations. These changes, in turn, result in geometrically transformed objects in the captured images, thus making it difficult to adequately encode the object's appearance in all possible contexts using a single appearance model. To address this problem, one promising class of approaches [242–256] seek to embed additional convolutional or pooling layers as independent, dedicated differentiable units in deep CNNs to explicitly learn geometric transformations. The most well-known methods in this class are those proposed in [257] and [258]. As shown in Table 5, these approaches can broadly be categorized into three groups [259]: methods that address (1) affine transformations, (2) general (including arbitrary and nonlinear) transformations, and (3) specific (or single) transformations.

7.1 Approaches to modeling affine transformations

Many spatial transformation modeling approaches [242,243, 243–248,248–253] specifically target affine transformations. In [257] Jaderberg et al. proposed spatial transformer network (STN), which embeds a differentiable model, called spatial transformer, to learn the parameters of affine transformations of a target object. The learned transformation parameters are then used to generate new sampling kernels which are applied to extract features from input data. Approaches based on spatial transformers have already become very popular in many machine vision tasks—including detection and tracking [245–248]. In most of the implementations, the spatial transformers are embedded in base CNN classification models or placed on top of detection heads to align input images to canonical views. For instance, Qian et al. [251] proposed a method to allow the detection of heavily deformed pedestrians in fish-eye camera views. Because of the lack of wide field of view (FoV) pedestrian detection datasets, they first transformed canonical images into fish-eye views by means of a so-called Projective Model Transformer (PMT) and then utilized a so-called Oriented Spatial Transformer Network (OSTN) consisting of a pair of STNs to learn fish-eye image transformations. Spatial transformers have also been employed to help generate positive samples in different poses for adversarial training [260,261]. In [253], Li et al. used an STN to learn localization information for latent compositional parts in a pedestrian re-identification framework. Luo et al. [252] (Fig. 9) combined STN and re-identification modules in a similarity learning framework for robust person re-identification. The STN learns affine transformation parameters and is able to accurately sample the most similar holistic image patches that match target (partial) persons in distorted and cropped images. Similar to the STN-based models, Xie et al. [243] proposed to incorporate a custom affine transformation manifold in a Faster R-CNN object detection model in order to learn geometric transformations of target objects, and to adapt and align detection bounding boxes to object shape. The bounding box alignment allows to better capture spatial features in the effective area of the tracked object. To encode possible deformations, three different kernel sizes are used for RoI pooling. Additionally, a multi-task loss simultaneously optimizes the robustness and accuracy of detections.

7.2 Approaches to modeling nonlinear transformations

Approaches such as [243] and the STN-based methods [242, 243,245–248,250–252] employ explicit geometric transformation operations to learn spatial appearance variations. As a result, they cannot effectively handle complex, non-analytical transformations. To overcome this shortcoming,

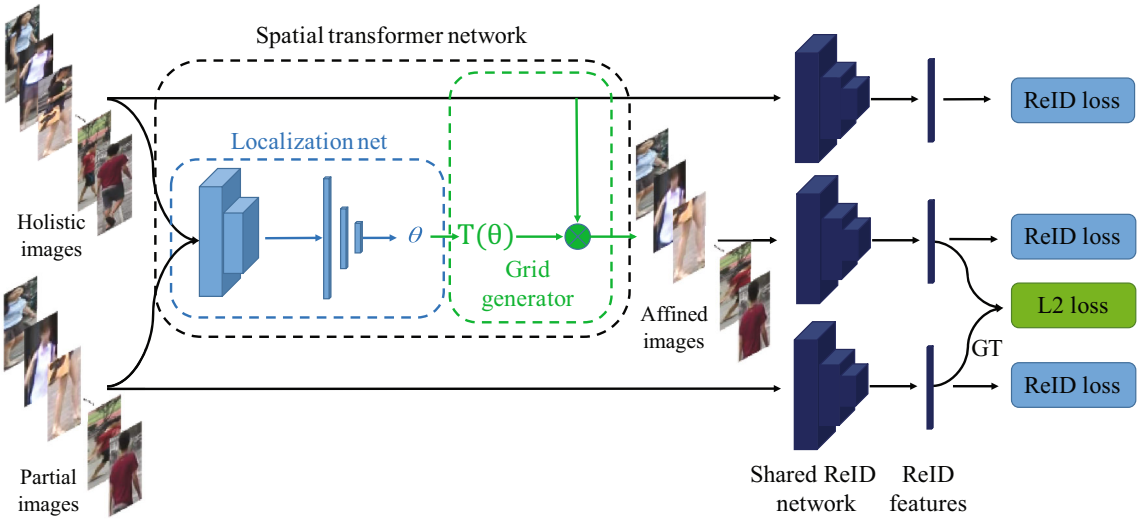

Fig. 9 Spatial Transformer Network (STN)-based person re-identification framework—STNReID [252]. The approach employs an STN in a Siamese network configuration to perform re-identification of persons in cropped and severely warped images

Dai, et al. in [258] introduced the (DCN), a technique that allows arbitrary nonlinear geometry transformations to be learned. The approach embeds a module that allows arbitrary deformations to be applied to the sampling kernels of its convolutional and RoI pooling layers. When incorporated into a standard CNN network, these deformable kernels can be applied on input features to learn geometric transformations. Following the original work in [258], several works utilizing the method for better visual feature encoding in object detection and tracking tasks have been proposed [62,133,248,254–256,262]. For instance, Cao and Chen [256] proposed Deformable Convolution Network Tracker (DCT) which consists of using deformable convolution modules in multiple CNN branches dedicated to different domains. In [62], it was shown that deformable convolutions can help to align re-identification features with detections, thereby significantly improving the accuracy and robustness of tracking. In contrast to the above approach to learning spatial deformations by adaptively changing the shape of convolutional kernels, Johnander et al. [263] proposed to encode target transformations by composing filters as linear combination of smaller filters.

Based on the knowledge [264] that expanding the receptive field improves generalization to spatial transformations, some approaches [253,265–267] proposed to expand the receptive field by replacing the CNN's conventional dense convolutions with dilated or atrous convolutions [268,269]. For instance, in [265] Chen et al. composed a visual tracker which uses a ResNet-50 backbone with dilated convolutions within a Siamese network structure to learn robust appearance features for tracking. Similar to [265], Jiang et al. [266] employed dilated convolutions based on a Hybrid Dilated Convolution (HDC) [270] organization to learn rich feature hierarchies. Zhang et al. [271] proposed irregular atrous convolutional scheme to further enhance feature representation in object tracking tasks.

7.3 Approaches to modeling single transformations

In contrast to the techniques considered in Sects. 7.1 and 7.2 which model general (affine and nonlinear) transformations, a common line of work aims to encode specific geometric transformations by applying predefined transformations in a pre-processing step (e.g., [272–276]) or by using multi-scale features [277,278] before using layers CNNs to learn these transformations. These techniques are mostly incorporated in standard backbone feature extraction and object detection models such as VGG [93]. They are commonly designed to encode rotations [279], scale variations [277,280], and perspective distortions [281,282]. Multi-scale methods are arguably the commonest of these techniques. In [280], Szegedy et al. proposed to use of differently sized convolutional filters to extract multi-scale features from input images. Fang et al. [277] employ a spatial arrangement of filters to encode features of varying sizes. Other approaches, for example, [283,284] adopt special pooling mechanisms to dynamically adjust the scales of visual features. Even though methods in this category are less general as compared to other geometric transformation techniques, they still find widespread use in object detection and tracking applications due to their low computational overheads.

Table 5 Approaches to tackling geometric transformations in visual tracking settings

Types of transformations	General functional mechanism	Representative trackers
Nonlinear transformations	Adaptation of receptive fields	[133,248,255,256]
Affine transformations	Analytical transformation operations	[243,245–247]
Single transformations	Predefined variable filters or specific image warping	[272–275]

8 Datasets, evaluation metrics and performance results of state-of-the-art object trackers

This section presents the common datasets, evaluation metrics and performance results of state-of-the-art visual trackers surveyed in this work. We focus on datasets for which quantitative performance results are available for many of the approaches surveyed. Conversely, in the presentation of performance results, we pay less attention to approaches that have not been evaluated on popular datasets. Also, we focus on a subset of metrics for which we have several results on the selected datasets. Nonetheless, for each dataset, the selected subset of performance metrics is the most important, and is broad and can adequately characterize the performance of visual trackers.

8.1 Datasets

To allow the training and evaluation of object tracking models, a large number of video datasets [30–32,286–295,298, 299] have been composed. The videos in these datasets are typically captured under challenging conditions like varying illumination and scale, occlusion, blur, background clutter, deformation, as well as in-plane and out-of-plane rotations. This allows researchers to train robust trackers and evaluate their ability to handle different real-world situations. The major features of common object tracking datasets are summarized in Table 6. In addition to these dedicated object tracking datasets, visual trackingmodels that rely on tracking-by-detection methods may utilize large-scale video object detection datasets such as ImageNet VID dataset [301] and the YouTube-BoundingBoxes dataset [302] . In the following paragraphs, we present a brief description of some of the most important visual tracking datasets.

(a) SOT datasets: The large-scale datasets used for training single object trackers include the Visual Object Tracking (VOT) family of datasets—VOT15 through to VOT20 [30–32,287–289]; the Object Tracking Benchmark (OTB) line of datasets—OTB-50 [303] and OTB-100 [286]; Need for Speed (NfS) [294]; UAV123 [295]; GOT-10k [297]; LaSOT [298], and TrackingNet [299].The Visual Object Tracking (VOT), LaSOT, GOT-10k and Object Tracking Benchmark (OTB) lines of datasets are the most popular datasets for training and evaluating SOT algorithms. Some of the SOT datasets focus on narrow application domains such as people tracking in video surveillance scenarios (e.g., [32]) and vehicle tracking (e.g., [295]). There are also many SOT datasets (e.g., LaSOT [298],GOT-10k [297] and TC-128 [296]) that aim to capture generic objects and scenes. The OTB-100 dataset, for example, contains one hundred (100) challenging labeled video snippets with a general focus. The TC-128 has 128 labeled video clips with a large diversity of object categories captured under different conditions. It particularly focuses on object and scene color variations. The VOT family of datasets and the OTB-100 [286] focus on human tracking.

(b) MOT datasets: Existing multiple object tracking datasets are typically domain-specific datasets, with many dealing with pedestrian or vehicle tracking. The most popular datasets are the MOT series [290–292]. Through several iterations starting from MOT15 to MOT20, a large number of these benchmark datasets have been collected through several MOT Challenges. To date, a total of 44 video snippets totaling about 36, 000 seconds of streaming content [292] are available through the MOTChallenge. The latest MOT dataset, MOT20 [292], contains 8 new (4 training and 4 test sets) video sequences. The MOT datasets are domain-specific, all dealing with pedestrian detection and tracking. The KITTI object detection dataset [293] is another popular dataset used for training and evaluating multiple object tracking models. The dataset is intended for vehicle and pedestrian detection and tracking. It contains a total of 50 short videos, 21 of which are for training and the remaining 29 for testing. Wen et al. recently introduced a new dataset, the UA-DETRAC dataset [285], for vehicle tracking.

8.2 Evaluation metrics

Many performance benchmarks and evaluation metrics have been proposed to quantitatively assess the quality of object tracking algorithms and validate their use in different situations. They also allow researchers to compare the performance of different models. Typically, different datasets or families of datasets provide different evaluation protocols and metrics. We briefly introduce the metrics used to compare visual trackers explored in this paper, and refer the reader to appropriate sources for more detailed information on the specific metrics.

Table 6 Common object tracking datasets

Dataset	Type	FPS	Domain	No. videos	No. frames
UA-DETRAC [285]	MOT	25	Vehicles	100	140,000
OTB-100 [286]	SOT	30	Humans	100	59,040
VOT series [30–32,287–289]	SOT	30	Humans	60	10,390
MOT15 [290]	MOT	Varied (7–30)	Pedestrians	22	11,283
MOT16/17 [291]	MOT	Varied (14–30)	Pedestrians	14	11,235
MOT20 [292]	MOT	25	Pedestrians	8	13,410
KITTI [293]	MOT	10	Vehicles and pedestrians	50	19,000
NfS [294]	SOT	30 and 240	Diverse (23 classes)	100	383,000
UAV123 [295]	SOT	30	Vehicle tracking from air	123	11,2578
TC-128 [296]	SOT	30	Diverse (color information)	129	55,346
GOT-10k [297]	SOT	10	Diverse (563 classes)	10,000	56,000
LaSOT [298]	SOT	30	Diverse (70 classes)	1400	3,520,000
TrackingNet [299]	SOT	Varied	Diverse (27 classes)	30,643	14,431,266
UAVDT [300]	MOT	30	Vehicle tracking from air	100	80,000

Table 7 Results of surveyed state-of-the-art trackers on the Visual Object Tracking (VOT) datasets—VOT15, VOT16 and VOT17 datasets

Model	VOT2015			VOT2016			VOT2017		
	EAO↑	A↑	R↓	EAO↑	A↑	R↓	EAO↑	A↑	R↓
SiamFC [184]	0.289	0.534	0.88	0.235	0.53	0.46	0.188	0.495	2.049
SiamFC+ [304]	0.31	0.57	–	0.30	0.54	0.38	0.23	0.50	0.49
SA-Siam [189]	0.310	0.590	1.260	0.290	0.540	1.080	0.236	0.500	0.459
ECO [305]	–	–	–	*0.375*	0.55	**0.20**	0.280	0.48	**0.27**
ECO-HC [305]	–	–	–	0.322	0.54	0.30	0.238	0.49	0.44
CCOT [53]	0.303	0.54	0.82	0.331	0.536	0.895	0.267	0.49	*0.32*
AFSL [90]	*0.366*	*0.62*	0.98	0.342	0.58	1.08	–	–	–
MDNet [36]	**0.378**	0.603	*0.693*	0.257	0.54	0.34	–	–	–
Staple [51]	0.300	0.56	0.86	0.295	0.544	0.378	0.169	0.519	2.507
MemTrack [306]	0.275	0.558	1.729	0.272	0.527	1.438	0.243	0.494	1.774
SiamRPN [187]	0.349	0.58	1.13	0.344	0.56	0.26	0.244	0.49	0.46
SiamRPN+ [304]	0.38	0.59	–	0.37	0.58	*0.24*	*0.30*	0.52	0.41
VITAL [80]	–	–	–	0.322	0.56	0.27	–	–	–
DaSiamRPN [79]	–	**0.630**	**0.660**	**0.411**	**0.610**	0.220	**0.326**	**0.560**	0.340
VTAAN [91]	–	–	–	0.327	1.41	1.98	–	–	–
AVA [92]	–	–	–	0.366	0.53	0.68	–	–	–
MDSLT [195]	0.296	0.692	1.052	0.258	0.542	0.396	–	–	–
GDT [133]	–	–	–	0.353	0.585	0.774	0.258	*0.558*	0.645
C-RPN [188]	–	–	–	0.363	*0.594*	0.95	0.289	–	–

For each metric, the best result is in bold font, while the second best is in italic

(a) SOT metrics: In this work, we present performance results for VOT15 through to VOT20, as well as for TrackingNet, LaSOT and GOT-10K datasets. We briefly describe the important metrics used on these datasets for performance evaluation. Details about these metrics are presented in the original works [30–32,287–289,297–299]. The most important performance evaluation metrics provided by the VOT family of datasets are *accuracy* (A), *robustness* (R), and the *expected average overlap* (EAO). *Accuracy* describes the preciseness of localization of the target, that is, how well the estimated bounding box for a tracked object matches the ground-truth bounding box. The metric is given as a fractional number which is computed as the ratio of successfully tracked frames to the total number of frames in the given video sequence. A successful track is considered to be a track whose region overlap exceeds a certain pre-

Table 8 Results of surveyed state-of-the-art visual trackers on the Visual Object Tracking (VOT) datasets—VOT18, VOT19 and VOT20

Model	VOT2018			VOT2019			VOT2020		
	EAO↑	A↑	R↓	EAO↑	A↑	R↓	EAO↑	A↑	R↓
TrTr [232]	*0.493*	0.606	**0.110**	*0.384*	0.601	**0.228**	–	–	–
Siam R-CNN [2]	0.408	0.609	0.220	–	–	–	–	–	–
UPDT [81]	0.378	0.536	0.184	–	–	–	0.278	0.465	0.755
SiamRPN++ [307]	0.414	0.600	0.234	0.292	0.580	0.446	–	–	–
ATOM [308]	0.401	0.590	0.204	0.292	0.603	0.411	–	–	–
DiMP-50 [309]	0.440	0.597	0.153	–	–	–	0.274	0.457	**0.740***
D3S [310]	0.489	0.597	0.178	–	–	–	*0.439*	*0.699*	0.769
SAMN [311]	**0.521**	**0.652**	*0.145*	**0.408**	**0.639**	*0.231*	**0.461**	**0.720**	0.794
DiMP [309]	0.441	0.597	0.152	0.321	0.582	0.371	–	–	–
SiamBAN [265]	0.452	0.597	0.178	0.327	*0.602*	0.396	–	–	–
Ocean [271]	0.467	*0.640*	0.150	0.327	0.590	0.376	0.430	0.693	*0.754*
DaSiamRPN [79]	0.383	0.586	0.276	–	–	–	–	–	–

For each metric, the best result is in bold font, while the second best is in italic

Table 9 Results of surveyed state-of-the-art trackers on other popular SOT datasets—TrackingNet, GOT-10K and LaSOT

Model	TrackingNet			GOT-10k			LaSOT		
	Prec.↑	Pnorm.↑	Success↑	AO↑	$SR_{0.5}$ ↑	$SR_{0.75}$ ↑	AUC↑	P↑	Pnorm↑
CFNet [185]	–	–	–	0.293	0.265	0.087	0.275	0.259	0.312
SiamFC [184]	53.3	66.3	57.1	0.348	0.353	0.098	0.336	**0.339**	0.420
ECO [305]	49.2	61.8	55.4	0.316	0.309	0.111	0.324	*0.301*	0.338
CCOT [53]	–	–	–	0.325	0.328	0.107	–	–	–
MDNet [36]	56.5	70.5	60.6	–	–	–	0.397	–	0.460
Staple [51]	–	–	–	0.246	0.239	0.089	0.243	0.278	0.278
SiamRPN [187]	–	–	–	0.483	0.581	0.270	–	–	–
AD-LSTM [217]	60.6	70.7	64.3	0.401	0.433	0.186			
SiamRPN++ [307]	*69.4*	50.0	73.3	0.517	0.616	0.325	*0.496*	–	*0.569*
DaSiamRPN [79]	59.1	73.3	63.8	–	–	–	0.415	–	0.496
Siam R-CNN [2]	**80.0**	**85.4**	**81.2**	*0.549*	**0.728**	**0.587**	–	–	–
DiMP-50 [309]	68.7	*80.1*	*74.0*	**0.611**	*0.717*	*0.492*	**0.569**	–	**0.643**

For each metric, the best result is in bold font, while the second best is in italic

determined threshold value. *Robustness*, also called failure score, is the number of times a tracker loses its target and needs re-initialization. *Expected average overlap* is a composite metric that characterizes the combined effect of the robustness and accuracy measures. For GOT-10k, we report results for average overlap (AO) and success rate (SR) scores. The success rates are measured using overlap thresholds of 0.75 (SR0.75) and 0.5 (SR0.5). TrackingNet uses *precision* (P), *normalized precision* (Pnorm) and *success* (S) to quantitatively measure the performance of trackers. *Precision* measures the distance error or deviation, in pixel units, between the center positions of the ground-truth and the predicted bounding box of the target object for each frame. Precision is usually measured as the percentage of frames in which this deviation is within a given limit. With *normalized precision*, the raw precision values are normalized to account for the influence of different image sizes or resolutions. In this case, the distance error values are measured relative to image sizes. Success is computed as the region overlap ratio (i.e., the *Intersection over Union* or IoU) between the predicted and ground-truth bounding boxes. Again, a threshold value is set, above which a track is considered to be successful. The default value for this threshold is usually 0.5, and the percentage of frames whose region overlap ratios are greater than 0.5 gives the success score for the particular model. LaSOT, similar to TrackingNet dataset, provides *precision* and *normalized precision* for evaluation. Another important metric is the *area under curve* (AUC). This metric is obtained by first varying the overlap threshold between 0 and 1 and computing the success score at each threshold for the entire sequence. The average value of the success scores at each (sampled) overlap threshold value gives the AUC score.

Table 10 Results of surveyed state-of-the-art trackers on MOT17 dataset

Model	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
RelationTrack [60]	75.6	*80.9*	*75.8*	43.1	21.5	*9,786*	**34,214**	–
FairMOT [62]	67.5	–	69.8	37.7	20.8	–	–	2,868
FairMOTv2 [312]	73.7	**81.3**	72.3	43.2	17.3	27,507	117,477	3,303
Tractor [70]	56.3	–	55.1	21.1	35.3	**8,866**	235,449	1,987
SOMOT* [313]	71.0	–	71.9	42.7	15.3	39,537	118,983	5,184
CTTrack [71]	61.5	–	59.6	26.4	21.9	14,076	200,672	2,583
TransMOT* [59]	*76.7*	–	75.1	*51.0*	16.4	36,231	93,150	2,346
TraDeS [75]	69.1	–	63.9	36.4	21.5	20,892	150,060	3,555
DMAN* [314]	48.2	75.7	55.7	19.3	38.3	26,218	263,608	2,194
FPSN-MOT [208]	44.5	–	–	23.4	31.2	25,639	156,422	4,775
Ref. [215]	47.5	–	51.9	18.2	41.7	25,981	268,042	2,069
Ref. [58]	51.3	77.0	47.6	21.4	35.2	24,101	247,921	2,648
MPNTrack [315]	58.8	–	61.7	28.8	33.5	17,413	213,594	**1,185**
ArTIST-T [316]	56.7	–	57.5	22.7	37.2	12,353	230,437	*1,756*
ByteTrack* [317]	**80.3**	–	**77.3**	**53.2**	14.5	25,491	*83,721*	2,196
TransCenter [233]	68.8	79.9	61.4	36.8	23.9	22,860	149,188	4,653
CorrTracker* [318]	76.5	–	73.6	47.6	*12.7*	29,808	99,510	3,369
TransTrack* [235]	74.5	80.6	63.9	46.8	**11.3**	28,323	112,137	3,663

For each metric, the best result is in bold font, while the second best is in italic. The marker "*" denotes instances where private detectors are used

Table 11 Results of surveyed state-of-the-art trackers on MOT20 dataset

Model	MOTA↑	MOTP↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDS↓
RelationTrack [60]	67.2	79.2	70.5	62.2	8.9	61,134	104,597	4,243
FairMOTv2 [312]	61.8	78.6	67.3	68.8	**7.6**	103,440	88,901	5,243
ByteTrack* [317]	**77.8**	–	**75.2**	*69.2*	9.5	26,249	87,594	*1,223*
Tractor* [70]	52.6	–	52.7	29.4	26.7	6,930	236,680	1,648
TransMOT* [59]	*77.5*	–	**75.2**	**70.7**	9.1	34,201	**80,788**	1,615
FairMOT* [62]	58.7	–	63.7	66.8	*8.5*	103,440	*88,901*	6,013
TransCenter* [233]	61.0	*79.5*	49.8	48.4	15.5	49,189	147,890	4,493
CorrTracker* [318]	65.2	–	69.1	66.4	8.9	79,429	95,855	5,183
ArTIST-T [316]	53.6	–	51.0	31.6	28.1	**7,765**	230,576	1,531
SiamMOT* [319]	67.1	–	69.1	49.0	16.3	–	–	–
SOMOT* [313]	68.6	–	*71.4*	64.9	9.7	57,064	101,154	4209
Tractor++ [70]	51.3	–	47.1	24.9	26.0	*16,263*	253,680	2,584
deepTAMA [69]	47.6	–	48.7	27.2	23.6	38,194	252,934	2,437
MPNTrack [315]	57.6	–	59.1	38.2	22.5	16,953	201,384	**1,210**
TransTrack* [235]	64.5	**80.0**	59.2	49.1	13.6	28,566	151,377	3,565

For each metric, the best result is in bold font, while the second best is in italic. The marker "*" denotes instances where private detectors are used

(b) MOT metrics: For evaluating the performance of multiple object tracking algorithms, the most commonly used metrics are the *multiple object target accuracy* (MOTA) and its newer extension—the *multiple object tracking precision* (MOTP). MOTA is computed using 3 main parameters: missed tracks (*false negatives* or FN), *false positives* (FP), and identifier assignment errors (i.e., identity switches). The *Identity* or *ID Switch* metric (IDS) measures the total number of times the IDs of correctly tracked objects are erroneously changed. Since the proportion of missed tracks is usually several orders of magnitude higher than false positives, FN scores greatly influences the overall MOTA scores. BYTE, recently proposed by Zhang et al. in [317] aims to mitigate this challenge by grouping detections into high- and low-confidence predictions. The high confidence bounding box detections are first matched with tracklets.

All tracklets that remain unmatched are then associated with detections from the low-confidence group. This differs from the common approach where low confidence detections below a given threshold are rejected. The new method significantly reduces false negatives and enhances the overall tracking performance. Per the MOTA metrics, tracking success can be categorized as *mostly tracked* (MT)—i.e., for tracks with tracking success of 80% and above; *mostly lost* (ML)—success not exceeding 20%, and *partially tracked* (PT)—success between 20% and 80%. The MOTP metric measures the localization accuracy of tracked objects. Other notable evaluation metrics commonly used in visual tracking models include *false alarm per frame* (FAF) and *fragmentation* (Frag). FAF is calculated as the number of false positive instances detected in each frame. Frag is determined by the number of times a tracker loses a tracked instance in an earlier frame and re-establishes (i.e., re-detects) it in a later frame. Most MOT datasets come with specific object detectors that can be used in the detection stage. This is to ensure a fair comparison of different approaches. That notwithstanding, researchers are still able to use private or custom detectors on these datasets. For a detailed overview of the various evaluation protocols and metrics, readers can refer to [290] and [292].

8.3 Quantitative performance results of visual trackers

In Tables 7, 8, 9, 10 and 11, we present quantitative performance results of the surveyed trackers on selected large-scale visual tracking datasets. For the metrics marked with up arrow (↑), higher numerical values are better, while those shown with the down arrow (↓) indicate metrics for which lower numerical values are better. As already mentioned, we selected the particular datasets that have been widely used to evaluate many of the surveyed approaches. Tables 7, 8 and 9 present results for SOT methods, while Tables 10 and 11 capture results on MOT datasets.

Tables 7 and 8 present results on the popular VOT family of datasets. The evaluation metrics used are the expected average overlap (EAO), accuracy (A) and robustness (R). These metrics are briefly described in Sect. 8.2. The reader may refer to [287] and [288] for further details on the computation procedures. Results on other popular SOT datasets - specifically, TrackingNet, GOT-10k and LaSOT - are presented in Table 9. For the TrackingNet dataset, results are presented in terms of success, precision (prec.) and normalized precision (Pnorm.). The GOT-10k dataset results are based on average overlap (AO), success rate at 0.5 and 0.75 overlap thresholds ($SR_{0.5}$ and $SR_{0.75}$). For LaSOT, precision (P), normalized precision (Pnorm) area under curve (AUC) are used. Details on the calculation of these metrics are available in [299], [297] and [298].

In Tables 10 and 11, we present results for multiple object tracking methods using MOT17 and MOT20, respectively. The metrics used here are the multiple object target accuracy (MOTA), multiple object Tracking Precision (MOTP), identification-F1 (IDF1), mostly tracked (MT), mostly lost (ML), false positives (FP), false negatives (FN) and ID Switch (IDS). We refer interested readers to [291] and [320] for a detailed discussion on these metrics.In some cases where the same model has been tested using public and private detections, we provide results for both detections.

9 Summary and discussion

In Sects. 3 to 7, we have reviewed the main deep learning approaches for enhancing robustness of appearance models in object detection and tracking tasks. The reviewed techniques address different issues: sample efficiency, geometric transformations, object deformations, occlusions, complex backgrounds, and object interactions. Each technique approaches the problem of robust feature extraction and representation differently, offering advantages in terms of a combination of generalization performance with respect to general or specific appearance changes, computational efficiency, model adaptability and sample efficiency. Section 8 presents the common datasets and evaluation metrics, as well results of the surveyed object tracking models on some of the popular datasets. A broad summary of the common features, main rationale, architectures and limitations of the most important approaches covered in this work is given in Table 12.

Currently, methods based on similarity learning approaches, especially two-stream Siamese architectures, are the most common techniques due to their simplicity, computational efficiency and the possibility for few-shot learning. However, disadvantages associated with phenomena such as occlusions, background clutter and object interactions that are common in many complex MOT environments and long-term tracking scenarios limit their scope of application. In these scenarios, similarity learning approaches are often used in conjunction with other techniques in more complex pipelines. Solving problems such as occlusions and complex background clutter is most effective using compositional part modeling techniques, which treat the appearance model as a composition of spatially related entities. However, the process of creating models by this means is very time-consuming. A new trend is to automatically learn compositional parts from input samples. However, this is often challenging in many practical tracking applications since it requires training with large corpus of relevant data. GAN-based approaches have been proposed to address the problem of data scarcity and severe data imbalance by generating appropriate samples in the training process. Unlike in gen-

eral machine vision tasks that mostly deal with sample-level generation, adversarial learning in object tracking contexts typically involve feature-level generation.

While extending training datasets with GANs has proven to be an effective way to learn invariant features robust to different appearance conditions, these models are generally harder to train and, in some situations, achieving convergence may be unattainable. There is also a lack of reliable empirical performance metrics to assess the quality of GAN-generated data. Moreover, they also introduce additional computational overhead, thereby hampering their suitability for real-time applications. Attention-based models provide a good balance of efficiency and robust performance. Unlike conventional approaches to visual recognition where entire input images are processed with equal "attention"—and as a result learn both useful and irrelevant features of the object and scene – in models using attention mechanisms, only the most informative image segments necessary for the particular task are processed. This greatly reduces computational costs and increases detection efficiency while maintaining invariance to image transformations. In addition, the inclusion of memory in attention models allows long-term appearance characteristics to be preserved for future use. Another way to improve the robustness of deep learning-based appearance modeling is to integrate specialized CNN modules to explicitly model spatial transformations. The modules are differentiable and can seamlessly be incorporated into standard CNN models like the faster R-CNN framework (e.g., as in [247,321]) and trained end-to-end without modifying the structure of the base model. These techniques provide a fast and reliable means of encoding robust appearance models that can generalize well under various conditions. However, when applying them in general settings, difficulties arise due to their narrowly-defined formulation—they focus mainly on spatial transformations. For this reason, photometric effects (e.g., random noise, shadows, reflections and illumination variability) can greatly reduce their effectiveness.

10 Future research directions

A recent trend in object tracking is the development of [object detection and tracking] techniques [95,98,102,148,250,253,267,322] that combine different approaches dedicated to specific tasks into complex models in order to overcome the limitations of the individual approaches. Indeed, many of the approaches surveyed utilize two or more fundamental methodologies so as to ensure more accurate and robust detection and tracking performance. The resulting hybrid architectures consist of a set of dedicated sub-systems for feature representation using a combination of various mechanisms such as GANs, part models, visual attention and similarity learning approaches. For example, [253] employed a complex tracker that utilize a wide range of techniques. These include multi-scale kernels to encode scale variations; dilated convolutions to increase the receptive field; deeply mined quasi-compositional parts from multiple convolution layers; a spatial transformer network (STN) to learn affine transformations of latent compositional parts, as well as also modeling additional spatial constraints to better encode visual features. Similarly, [267] utilized a deep CNN configuration that involves an STN, a GRU and atrous convolutional layers. Zhang et al. [250] proposed a Siamese framework, within which an STN is employed to learn affine transformations of compositional parts for robust tracking. Lee et al. [323] introduced a memory model in a Siamese model to enable long-term tracking. In [322], attention mechanism is used to extract robust features from compositional part models.

Some of these hybrid models require the use of sophisticated fusion algorithms, as well as refinement methods. In [267] a GRU is used to fuse different features produced by the model components. In [324], a soft-max-based fusion mechanism is proposed for aggregating low-level features. In addition, a high-level spatial feature fusion is used to combine features from different components, including the soft-max fusion output and channel and spatial attention sub-modules. The techniques for fusing hybrid models are still at an early stage of development, hence, there is still a lot of room for the development of better fusion strategies to harness the strengths of individual approaches in a unified framework. The most promising application of future hybrid trackers would be to enable generic object tracking algorithms that generalize across multiple domains.

The main directions envisaged for future work include the following.

- *Robust feature transfer:* More effective techniques for transferring useful features from existing large-scale datasets to novel visual contexts and challenging application settings would be highly beneficial and compensate for the difficulty in creating large-scale tracking datasets.
- *Generic appearance models for tracking in open domains:* Many practical tracking application scenarios are characterized by openness, where arbitrary objects can appear on and disappear from the scene. Most of the current appearance models, however, work in specific, closed environments, in which the number of object categories are known and fixed. A relatively unexplored approach is learning robust generic appearance models in open environments.
- *More advanced hybrid fusion methods:* More sophisticated "hybridization" techniques that rely on both low- and high-level context information as well as advanced decision making capabilities to aggregate visual features will significantly improve the robustness and reliability

Table 12 Summary of robust appearance modeling approaches, their strengths and weaknesses as well as the common deep learning architectures used for their implementation

Approach	Main aim	Architectures	Strength	Weakness
Data augmentation	Expand training data	GANs, AE, VAE	Can be the only effective method when small or no data is available	Inflates data, leading to higher computational resource requirements
Compositional part modeling	Represent target objects by their constituent parts	DCNN, DPM	Strong against occlusions and deformations	Not applicable to objects without distinctive parts
Similarity learning	Predict by comparing the similarity between a given target and template image(s)	Pairwise deep CNNs	Simplicity; can be trained on large-scale image data offline	Difficult to update online
Attention and memory	Selectively process and retains only useful visual information	RNN, LSTM, GRU, transformer, external memory	Highly efficient; can encode contextual relationships; can update model online using previously learned information	Functions at low (pixel) level; limited memory capacity; memory access can slow performance
Embedded units for geometry learning	Explicitly model spatial transformations of real-world objects	STN, DCN, Astrous convolutions	Explainable; general (i.e., object-agnostic)	Introduces additional complexity and computational overheads

of appearance models. These fusing methods could allow multiple and diverse inference engines to be modeled as computational primitives within deep learning frameworks and be fused to enable predictions in a manner that is consistent with high-level real-world contexts.

- *The use of automated machine learning (AutoML) techniques:* The emerging area of Automated Machine Learning (AutoML) [325], especially Neural Architecture Search (NAS) [326–328], has already produced impressive deep learning models for many visual recognition problems. However, it remained under-explored in visual tracking tasks. An important dimension of future research would potentially involve the exploitation of these techniques to develop more advanced detectors and trackers. The configuration of these machine-generated frameworks could fundamentally differ from existing architectures.

11 Conclusion

Appearance modeling is the most important task in visual object tracking and is generally solved by extracting visual features from sample data of the target objects into sets of invariant feature vectors, and subsequently making inference based on the encoded representations. In this paper, we extensively survey the most important deep learning techniques for learning robust visual representations for object detection and tracking. The main motivations, key functional principles, implementation issues and application scenarios of these algorithms are thoroughly discussed. In addition, common datasets, performance evaluation metrics and quantitative results of state-of-the-art models surveyed in this paper are presented.

As we have noted earlier in the survey, owing to the enormous complexity of real-world visual tracking scenarios, there is still a lot of room for further improvement of appearance models with regard to their robustness and accuracy in challenging detection and tracking tasks. State-of-the-art deep learning techniques still fare poorly in visual tracking as compared to other machine vision tasks. Nevertheless, with the wide diversity of approaches at their disposal, developers and researchers have a lot of leverage and flexibility in developing appearance models that meet the requirements of specific applications. One of the main tasks for developers will be in defining the most suitable approaches for each given application scenario and adaptively fusing appropriate models for optimum performance.

References

1. Zhu, H., Wei, H., Li, B., Yuan, X., Kehtarnavaz, N.: A review of video object detection: datasets, metrics and methods. Appl. Sci. **10**(21), 7834 (2020)
2. Voigtlaender, P., Luiten, J., Torr, P.H., Leibe, B.: Siam r-cnn: visual tracking by re-detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6578–6588 (2020)
3. Elharrouss, O., Almaadeed, N., Al-Maadeed, S., Bouridane, A., Beghdadi, A.: A combined multiple action recognition and sum-

marization for surveillance video sequences. Appl. Intell. **51**(2), 690–712 (2021)
4. Najeeb, H.D., Ghani, R.F.: A survey on object detection and tracking in soccer videos. MJPS **8**(1), 1–13 (2021)
5. Siddique, A., Medeiros, H.: Tracking passengers and baggage items using multi-camera systems at security checkpoints. arXiv preprint arXiv:2007.07924 (2020)
6. Krishna, V., Ding, Y., Xu, A., Höllerer, T.: Multimodal biometric authentication for VR/AR using EEG and eye tracking. In: Adjunct of the 2019 International Conference on Multimodal Interaction, pp. 1–5 (2019)
7. D'Ippolito, F., Massaro, M., Sferlazza, A.: An adaptive multirate system for visual tracking in augmented reality applications. In: IEEE 25th International Symposium on Industrial Electronics (ISIE), vol. 2016, pp. 355–361. IEEE (2016)
8. Guo, Z., Huang, Y., Hu, X., Wei, H., Zhao, B.: A survey on deep learning based approaches for scene understanding in autonomous driving. Electronics **10**(4), 471 (2021)
9. Moujahid, D., Elharrouss, O., Tairi, H.: Visual object tracking via the local soft cosine similarity. Pattern Recognit. Lett. **110**, 79–85 (2018)
10. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3101–3109 (2015)
11. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. **4**(4), 1–48 (2013)
12. Dutta, A., Mondal, A., Dey, N., Sen, S., Moraru, L., Hassanien, A.E.: Vision tracking: a survey of the state-of-the-art. SN Comput. Sci. **1**(1), 1–19 (2020)
13. Walia, G.S., Kapoor, R.: Recent advances on multicue object tracking: a survey. Artif. Intell. Rev. **46**(1), 1–39 (2016)
14. Manafifard, M., Ebadi, H., Moghaddam, H.A.: A survey on player tracking in soccer videos. Comput. Vis. Image Underst. **159**, 19–46 (2017)
15. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X, et al.: Multiple object tracking: a literature review. arXiv preprint arXiv:1409.7618 (2014)
16. SM, J.R., Augasta, G.: Review of recent advances in visual tracking techniques. Multimed. Tools Appl. **16**, 24185–24203 (2021)
17. Ciaparrone, G., Sánchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., Herrera, F.: Deep learning in video multi-object tracking: a survey. Neurocomputing **381**, 61–88 (2020)
18. Marvasti-Zadeh, S.M., Cheng, L., Ghanei-Yakhdan, H., Kasaei, S.: Deep learning for visual tracking: a comprehensive survey. IEEE Trans. Intell. Transp. Syst. **23**, 3943–3968 (2021)
19. Xu, Y., Zhou, X., Chen, S., Li, F.: Deep learning for multiple object tracking: a survey. IET Comput. Vis. **13**(4), 355–368 (2019)
20. Li, P., Wang, D., Wang, L., Lu, H.: Deep visual tracking: review and experimental comparison. Pattern Recognit. **76**, 323–338 (2018)
21. Sun, Z., Chen, J., Liang, C., Ruan, W., Mukherjee, M.: A survey of multiple pedestrian tracking based on tracking-by-detection framework. IEEE Trans. Circuits Syst. Video Technol. **31**, 1819–1833 (2020)
22. Fiaz, M., Mahmood, A., Jung, S.K.: Tracking noisy targets: a review of recent object tracking approaches. arXiv preprint arXiv:1802.03098 (2018)
23. Sugirtha, T., Sridevi, M.: A survey on object detection and tracking in a video sequence. In: Proceedings of International Conference on Computational Intelligence, pp. 15–29. Springer (2022)
24. Brunetti, A., Buongiorno, D., Trotta, G.F., Bevilacqua, V.: Computer vision and deep learning techniques for pedestrian detection and tracking: a survey. Neurocomputing **300**, 17–33 (2018)
25. Ravoor, P.C., Sudarshan, T.: Deep learning methods for multi-species animal re-identification and tracking—a survey. Comput. Sci. Rev. **38**, 100289 (2020)
26. Kamble, P.R., Keskar, A.G., Bhurchandi, K.M.: Ball tracking in sports: a survey. Artif. Intell. Rev. **52**(3), 1655–1705 (2019)
27. Fahmidha, R., Jose, S.K.: Vehicle and pedestrian video-tracking: a review. In: 2020 International Conference on Communication and Signal Processing (ICCSP), pp. 227–232. IEEE (2020)
28. Shukla, A., Saini, M.: Moving object tracking of vehicle detection: a concise review. Int. J. Signal Process. Image Process. Pattern Recognit. **8**(3), 169–176 (2015)
29. Karuppuchamy, S., Selvakumar, R.: A Survey and study on "vehicle tracking algorithms in video surveillance system". In: 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4. IEEE (2017)
30. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., et al.: The sixth visual object tracking vot2018 challenge results. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
31. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., et al.: The seventh visual object tracking vot2019 challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
32. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., et al.: The eighth visual object tracking VOT2020 challenge results. In: European Conference on Computer Vision, pp. 547–601. Springer (2020)
33. Dendorfer, P., Osep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., et al.: Motchallenge: a benchmark for single-camera multiple target tracking. Int. J. Comput Vis. **129**(4), 845–881 (2021)
34. Lan, L., Wang, X., Zhang, S., Tao, D., Gao, W., Huang, T.S.: Interacting tracklets for multi-object tracking. IEEE Trans. Image Process. **27**(9), 4585–4597 (2018)
35. Milan, A., Schindler, K., Roth, S.: Multi-target tracking by discrete-continuous energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2054–2068 (2015)
36. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
37. Li, H., Li, Y., Porikli, F., et al.: DeepTrack: learning discriminative feature representations by convolutional neural networks for visual tracking. In: BMVC, vol. 1, p. 3 (2014)
38. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., et al.: Ssd: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016)
39. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W., Yang, M.H.: Crest: convolutional residual learning for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2555–2564 (2017)
40. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
41. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: International Conference on Machine Learning, pp. 597–606. PMLR (2015)
42. Tao, Q.Q., Zhan, S., Li, X.H., Kurihara, T.: Robust face detection using local CNN and SVM based on kernel combination. Neurocomputing **211**, 98–105 (2016)
43. Niu, X.X., Suen, C.Y.: A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recognit. **45**(4), 1318–1325 (2012)

44. Li, H., Li, Y., Porikli, F.: Deeptrack: learning discriminative feature representations online for robust visual tracking. IEEE Trans. Image Process. **25**(4), 1834–1848 (2015)
45. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: Advances in Neural Information Processing Systems (2013)
46. Zhou, K., Yang, Y., Hospedales, T., Xiang, T.: Deep domain-adversarial image generation for domain generalisation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13025–13032 (2020)
47. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M., Visual object tracking using adaptive correlation filters. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2010, pp. 2544–2550. IEEE (2010)
48. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 58–66 (2015)
49. Zhang, F., Ma, S., Qiu, Z., Qi, T.: Learning target-aware background-suppressed correlation filters with dual regression for real-time UAV tracking. Signal Process. **191**, 108352 (2022)
50. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
51. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1401–1409 (2016)
52. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: European Conference on Computer Vision, pp. 254–265. Springer (2014)
53. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European Conference on computer vision, pp. 472–488. Springer (2016)
54. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3539–3548 (2017)
55. Kieritz, H., Hubner, W., Arens, M.: Joint detection and online multi-object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1459–1467 (2018)
56. Wang, Z., Zheng, L., Liu, Y., Wang, S.: Towards real-time multi-object tracking. arXiv preprint arXiv:1909.12605 (2019)
57. Sultana, F., Sufian, A., Dutta, P.: A review of object detection models based on convolutional neural network. In: Image Processing Based Applications, Intelligent Computing, pp. 1–16 (2020)
58. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multi-object tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 1428–1437 (2018)
59. Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: TransMOT: spatial-temporal graph transformer for multiple object tracking. arXiv preprint arXiv:2104.00194 (2021)
60. Yu, E., Li, Z., Han, S., Wang, H.: RelationTrack: relation-aware multiple object tracking with decoupled representation. arXiv preprint arXiv:2105.04322 (2021)
61. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell T, et al.: Quasi-dense similarity learning for multiple object tracking. arXiv preprint arXiv:2006.06664 (2020)
62. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: FairMOT: on the fairness of detection and re-identification in multiple object tracking. arXiv preprint arXiv:2004.01888 (2020)
63. Ullah, M., Cheikh, F.A.: Deep feature based end-to-end transportation network for multi-target tracking. In: 25th IEEE International Conference on Image Processing (ICIP), vol. 2018, pp. 3738-3742. IEEE (2018)
64. Ren, L., Lu, J., Wang, Z., Tian, Q., Zhou, J.: Collaborative deep reinforcement learning for multi-object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 586–602 (2018)
65. Leal-Taixé, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: siamese CNN for robust target association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 33–40 (2016)
66. Zhang, S., Gong, Y., Huang, J.B., Lim, J., Wang, J., Ahuja, N., et al.: Tracking persons-of-interest via adaptive discriminative features. In: European Conference on Computer Vision, pp. 415–433. Springer (2016)
67. Chen, L., Ai, H., Shang, C., Zhuang, Z., Bai, B.: Online multi-object tracking with convolutional neural networks. In: 2017 IEEE international conference on image processing (ICIP), pp. 645–649. IEEE (2017)
68. Xu, Y., Osep, A., Ban, Y., Horaud, R., Leal-Taixé, L., Alameda-Pineda, X.: How to train your deep multi-object tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6787–6796 (2020)
69. Yoon, Y.C., Kim, D.Y., Song, Y.M., Yoon, K., Jeon, M.: Online multiple pedestrians tracking using deep temporal appearance matching association. Inf. Sci. **561**, 326–351 (2021)
70. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 941–951 (2019)
71. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European Conference on Computer Vision, pp. 474–490. Springer (2020)
72. Jia, Y.J., Lu, Y., Shen, J., Chen, Q.A., Chen, H., Zhong, Z., et al.: Fooling detection alone is not enough: adversarial attack against multiple object tracking. In: International Conference on Learning Representations (ICLR'20) (2020)
73. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3038–3046 (2017)
74. Lu, Z., Rathod, V., Votel, R., Huang, J.: Retinatrack: online single stage joint detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14668–14678 (2020)
75. Wu, J., Cao, J., Song, L., Wang, Y., Yang, M., Yuan, J.: Track to detect and segment: an online multi-object tracker. arXiv preprint arXiv:2103.08808 (2021)
76. Chaabane, M., Zhang, P., Beveridge, J.R., O'Hara, S.: DEFT: detection embeddings for tracking. arXiv preprint arXiv:2102.02267 (2021)
77. Sampath, V., Maurtua, I., Martín, J.J.A., Gutierrez, A.: A survey on generative adversarial networks for imbalance problems in computer vision tasks. J. Big Data. **8**(1), 1–59 (2021)
78. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress Artif. Intell. **5**(4), 221–232 (2016)
79. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 101–117 (2018)
80. Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., et al.: Vital: visual tracking via adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8990–8999 (2018)
81. Bhat, G., Johnander, J., Danelljan, M., Khan, F.S., Felsberg, M.: Unveiling the power of deep tracking. In: Proceedings of the

European Conference on Computer Vision (ECCV), pp. 483–498 (2018)
82. Wang, Y., Wei, X., Tang, X., Shen, H., Ding, L.: CNN tracking based on data augmentation. Knowl.-Based Syst. **194**, 105594 (2020)
83. Neuhausen, M., Herbers, P., König, M.: Synthetic data for evaluating the visual tracking of construction workers. In: Construction Research Congress 2020: Computer Applications, pp. 354–361. American Society of Civil Engineers Reston, VA (2020)
84. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4340–4349 (2016)
85. Shermeyer, J., Hossler, T., Van Etten, A., Hogan, D., Lewis, R., Kim, D.: Rareplanes: synthetic data takes flight. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 207–217 (2021)
86. Han, Y., Zhang, P., Huang, W., Zha, Y., Cooper, G., Zhang, Y.: Robust visual tracking using unlabeled adversarial instance generation and regularized label smoothing. Pattern Recognit. 1–15 (2019)
87. Cheng, X., Song, C., Gu, Y., Chen, B.: Learning attention for object tracking with adversarial learning network. EURASIP J. Image Video Process. **2020**(1), 1–21 (2020)
88. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)
89. Han, Y., Zhang, P., Huang, W., Zha, Y., Cooper, G.D., Zhang, Y.: Robust visual tracking based on adversarial unlabeled instance generation with label smoothing loss regularization. Pattern Recognit. **97**, 107027 (2020)
90. Yin, Y., Xu, D., Wang, X., Zhang, L.: Adversarial feature sampling learning for efficient visual tracking. IEEE Trans. Autom. Sci. Eng. **17**(2), 847–857 (2019)
91. Wang, F., Wang, X., Tang, J., Luo, B., Li, C.: VTAAN: visual tracking with attentive adversarial network. Cognit. Comput. **13**, 646–656 (2020)
92. Javanmardi, M., Qi, X.: Appearance variation adaptation tracker using adversarial network. Neural Netw. **129**, 334–343 (2020)
93. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
94. Kim, H.I., Park, R.H.: Siamese adversarial network for object tracking. Electron. Lett. **55**(2), 88–90 (2018)
95. Wang, X., Li, C., Luo, B., Tang, J.: Sint++: Robust visual tracking via adversarial positive instance generation. In: Proceedings of the IEEE Conference on Computer Vision and pattern recognition, pp. 4864–4873 (2018)
96. Guo, J., Xu, T., Jiang, S., Shen, Z.: Generating reliable online adaptive templates for visual tracking. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 226–230. IEEE (2018)
97. Wu, Q., Chen, Z., Cheng, L., Yan, Y., Li, B., Wang, H.: Hallucinated adversarial learning for robust visual tracking. arXiv preprint arXiv:1906.07008 (2019)
98. Kim, Y., Shin, J., Park, H., Paik, J.: Real-time visual tracking with variational structure attention network. Sensors **19**(22), 4904 (2019)
99. Lin, C.C., Hung, Y., Feris, R., He, L.: Video instance segmentation tracking with a modified vae architecture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13147–13157 (2020)
100. Cheng, X., Zhang, Y., Zhou, L., Zheng, Y.: Visual tracking via auto-encoder pair correlation filter. IEEE Trans. Ind. Electron. **67**(4), 3288–3297 (2019)
101. Wang, L., Pham, N.T., Ng, T.T., Wang, G., Chan, K.L., Leman, K.: Learning deep features for multiple object tracking by using a multi-task learning strategy. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 838–842. IEEE (2014)
102. Liu, P., Li, X., Liu, H., Fu, Z.: Online learned Siamese network with auto-encoding constraints for robust multi-object tracking. Electronics **8**(6), 595 (2019)
103. Xu, L., Niu, R.: Semi-supervised visual tracking based on variational siamese network. In: International Conference on Dynamic Data Driven Application Systems, pp. 328–336. Springer (2020)
104. Tao, R., Gavves, E., Smeulders, AW.: Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1420–1429 (2016)
105. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3018–3027 (2017)
106. Wei, L., Zhang, S., Gao, W., Tian, Q.: Person transfer gan to bridge domain gap for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 79–88 (2018)
107. Li, K., Zhang, Y., Li, K., Fu, Y.: Adversarial feature hallucination networks for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13470–13479 (2020)
108. Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Feris, R., et al.: Delta-encoder: an effective sample synthesis method for few-shot object recognition. arXiv preprint arXiv:1806.04734 (2018)
109. Amirkhani, A., Barshooi, A.H., Ebrahimi, A.: Enhancing the robustness of visual object tracking via style transfer. Comput. Mater. Contin. **70**(1), 981–997 (2022)
110. López-Sastre, R.J., Tuytelaars, T., Savarese, S.: Deformable part models revisited: A performance evaluation for object category pose estimation. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1052–1059. IEEE (2011)
111. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2147–2156 (2016)
112. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. Int. J. Comput. Vis. **61**(1), 55–79 (2005)
113. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 269–286 (2018)
114. Rad, M., Lepetit, V.: Bb8: a scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3828–3836 (2017)
115. Wang, B., Wang, L., Shuai, B., Zuo, Z., Liu, T., Luk Chan, K., et al.: Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8 (2016)
116. Uricár, M., Franc, V., Hlavác, V.: Facial landmark tracking by tree-based deformable part model based detector. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 10–17 (2015)
117. Crivellaro, A., Rad, M., Verdie, Y., Moo Yi, K., Fua, P., Lepetit, V.: A novel representation of parts for accurate 3D object detection and tracking in monocular images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4391–4399 (2015)

118. Li, J., Wong, H.C., Lo, S.L., Xin, Y.: Multiple object detection by a deformable part-based model and an R-CNN. IEEE Signal Process. Lett. **25**(2), 288–292 (2018)
119. De Ath, G., Everson, R.M.: Part-based tracking by sampling. arXiv preprint arXiv:1805.08511 (2018)
120. Liu, W., Sun, X., Li, D.: Robust object tracking via online discriminative appearance modeling. EURASIP J. Adv. Signal Process. **2019**(1), 1–9 (2019)
121. Wang, G., Yuan, Y., Chen, X., Li, J., Zhou, X.: Learning discriminative features with multiple granularities for person re-identification. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 274–282 (2018)
122. Tian, Y., Luo, P., Wang, X., Tang, X.: Deep learning strong parts for pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1904–1912 (2015)
123. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1971–1978 (2014)
124. Gao, J., Zhang, T., Yang, X., Xu, C.: P2t: part-to-target tracking via deep regression learning. IEEE Trans. Image Process. **27**(6), 3074–3086 (2018)
125. Lim, J.J., Dollar, P., Zitnick III, C.L.: Learned mid-level representation for contour and object detection. Google Patents; 2014. US Patent App. 13/794,857
126. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: 2011 International Conference on Computer Vision, pp. 1323–1330. IEEE (2011)
127. Lee, S.H., Jang, W.D., Kim, C.S.: Tracking-by-segmentation using superpixel-wise neural network. IEEE Access **6**, 54982–54993 (2018)
128. Yang, F., Lu, H., Yang, M.H.: Robust superpixel tracking. IEEE Trans. Image Process. **23**(4), 1639–1651 (2014)
129. Verelst, T., Blaschko, M., Berman, M.: Generating superpixels using deep image representations. arXiv preprint arXiv:1903.04586 (2019)
130. Jampani, V., Sun, D., Liu, M.Y., Yang, M.H., Kautz, J.: Superpixel sampling networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 352–368 (2018)
131. Yang, F., Sun, Q., Jin, H., Zhou, Z.: Superpixel segmentation with fully convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13964–13973 (2020)
132. Yang, X., Wei, Z., Wang, N., Song, B., Gao, X.: A novel deformable body partition model for MMW suspicious object detection and dynamic tracking. Signal Process. **174**, 107627 (2020)
133. Liu, W., Song, Y., Chen, D., He, S., Yu, Y., Yan, T., et al.: Deformable object tracking with gated fusion. IEEE Trans. Image Process. **28**(8), 3766–3777 (2019)
134. Girshick, R., Felzenszwalb, P., McAllester, D.: Object detection with grammar models. Adv. Neural Inf. Process. Syst. **24**, 442–450 (2011)
135. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2009)
136. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: European Conference on Computer Vision, pp. 836–849. Springer (2012)
137. Ouyang, W., Wang, X.: Single-pedestrian detection aided by multi-pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3198–3205 (2013)
138. Nam, H., Baek, M., Han, B.: Modeling and propagating cnns in a tree structure for visual tracking. arXiv preprint arXiv:1608.07242 (2016)
139. Wang, J., Fei, C., Zhuang, L., Yu, N.: Part-based multi-graph ranking for visual tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). IEEE; 2016. p. 1714–1718
140. Du, D., Wen, L., Qi, H., Huang, Q., Tian, Q., Lyu, S.: Iterative graph seeking for object tracking. IEEE Trans. Image Process. **27**(4), 1809–1821 (2017)
141. Du, D., Qi, H., Li, W., Wen, L., Huang, Q., Lyu, S.: Online deformable object tracking based on structure-aware hyper-graph. IEEE Trans. Image Process. **25**(8), 3572–3584 (2016)
142. Wang, L., Lu, H., Yang, M.H.: Constrained superpixel tracking. IEEE Trans. Cybern. **48**(3), 1030–1041 (2017)
143. Jianga, B., Zhang, P., Huang, L.: Visual object tracking by segmentation with graph convolutional network. arXiv preprint arXiv:2009.02523 (2020)
144. Parizi, S.N., Vedaldi, A., Zisserman, A., Felzenszwalb, P.: Automatic discovery and optimization of parts for image classification. arXiv preprint arXiv:1412.6598 (2014)
145. Li, Y., Liu, L., Shen, C., Van Den Hengel, A.: Mining mid-level visual patterns with deep CNN activations. Int. J. Comput. Vis. **121**(3), 344–364 (2017)
146. Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 437–446 (2015)
147. Sun, Y., Zheng, L., Li, Y., Yang, Y., Tian, Q., Wang, S.: Learning part-based convolutional features for person re-identification. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 902–917 (2019)
148. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., et al.: Hedged deep tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4303–4311 (2016)
149. Mordan, T., Thome, N., Henaff, G., Cord, M.: End-to-end learning of latent deformable part-based representations for object detection. Int. J. Comput. Vis. **127**(11), 1659–1679 (2019)
150. Zhang, Z., Xie, C., Wang, J., Xie, L., Yuille, A.L.: Deepvoting: a robust and explainable deep network for semantic part detection under partial occlusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1372–1380 (2018)
151. Mordan, T., Thome, N., Cord, M., Henaff, G.: Deformable part-based fully convolutional network for object detection. arXiv preprint arXiv:1707.06175 (2017)
152. Jifeng, D., Yi, L., Kaiming, H., Jian, S.: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
153. Ouyang, W., Zeng, X., Wang, X., Qiu, S., Luo, P., Tian, Y., et al.: DeepID-Net: object detection with deformable part based convolutional neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(7), 1320–1334 (2016)
154. Yang, L., Xie, X., Li, P., Zhang, D., Zhang, L.: Part-based convolutional neural network for visual recognition. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 1772–1776. IEEE (2017)
155. Wang, J., Xie, C., Zhang, Z., Zhu, J., Xie, L., Yuille, A.: Detecting semantic parts on partially occluded objects. arXiv preprint arXiv:1707.07819 (2017)
156. Wang, J., Zhang, Z., Xie, C., Premachandran, V., Yuille, A.: Unsupervised learning of object semantic parts from internal states of cnns by population encoding. arXiv preprint arXiv:1511.06855 (2015)
157. Li, Y., Liu, L., Shen, C., van den Hengel, A.: Mid-level deep pattern mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 971–980 (2015)

158. Zhang, Q., Wu, Y.N., Zhu, S.C.: Interpretable convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8827–8836 (2018)
159. Stone, A., Wang, H., Stark, M., Liu, Y., Scott Phoenix, D., George, D.: Teaching compositionality to cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5058–5067 (2017)
160. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2056–2063 (2013)
161. Zhu, F., Kong, X., Zheng, L., Fu, H., Tian, Q.: Part-based deep hashing for large-scale person re-identification. IEEE Trans. Image Process. **26**(10), 4806–4817 (2017)
162. Wu, G., Lu, W., Gao, G., Zhao, C., Liu, J.: Regional deep learning model for visual tracking. Neurocomputing **175**, 310–323 (2016)
163. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
164. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer (2014)
165. Dinov, I.D. Black box machine-learning methods: Neural networks and support vector machines. In: Data Science and Predictive Analytics, pp. 383–422. Springer (2018)
166. Mozhdehi, R.J., Medeiros, H.: Deep convolutional particle filter for visual tracking. In: IEEE International Conference on Image Processing (ICIP), vol. 2017, pp. 3650–3654. IEEE (2017)
167. Yang, B., Hu, X., Wang, F.: Kernel correlation filters based on feature fusion for visual tracking. J. Phys. Conf. Ser. **1601**, 052026 (2020)
168. Yang, Y., Liao, S., Lei, Z., Li, S.: Large scale similarity learning using similar pairs for person verification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
169. Hirzer, M., Roth, P.M., Köstinger, M., Bischof, H.: Relaxed pairwise learned metric for person re-identification. In: European Conference on Computer Vision, pp. 780–793. Springer (2012)
170. Kulis, B., et al.: Metric learning: a survey. Found. Trends Mach. Learn. **5**(4), 287–364 (2012)
171. Jia, Y., Darrell, T.: Heavy-tailed distances for gradient based image descriptors. Adv. Neural Inf. Process. Syst. **24**, 397–405 (2011)
172. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning local feature descriptors using convex optimisation. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1573–1585 (2014)
173. Tian, S., Shen, S., Tian, G., Liu, X., Yin, B.: End-to-end deep metric network for visual tracking. Vis. Comput. **36**(6), 1219–1232 (2020)
174. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P, et al. Supervised contrastive learning. arXiv preprint arXiv:2004.11362 (2020)
175. Zhao, R., Ouyang, W., Wang, X.: Learning mid-level filters for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 144–151 (2014)
176. Paisitkriangkrai, S., Shen, C., Van Den Hengel, A.: Learning to rank in person re-identification with metric ensembles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1846–1855 (2015)
177. Yang, W., Liu, Y., Zhang, Q., Zheng, Y.: Comparative object similarity learning-based robust visual tracking. IEEE Access **7**, 50466–50475 (2019)
178. Zhou, Y., Bai, X., Liu, W., Latecki, L.J.: Similarity fusion for visual tracking. Int. J. Comput. Vis. **118**(3), 337–363 (2016)
179. Ning, J., Shi, H., Ni, J., Fu, Y.: Single-stream deep similarity learning tracking. IEEE Access **7**, 127781–127787 (2019)
180. Chicco, D.: Siamese neural networks: an overview. In: Artificial Neural Networks, pp. 73–94 (2021)
181. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. Adv. Neural Inf. Process. Syst. **6**, 737–744 (1993)
182. Vaquero, L., Brea, V.M., Mucientes, M.: Tracking more than 100 arbitrary objects at 25 FPS through deep learning. Pattern Recognit. **121**, 108205 (2022)
183. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., et al.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2015)
184. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision, pp. 850–865. Springer (2016)
185. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2805–2813 (2017)
186. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: European Conference on Computer Vision, pp. 749–765. Springer (2016)
187. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
188. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7952–7961 (2019)
189. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4834–4843 (2018)
190. Zha, Y., Wu, M., Qiu, Z., Yu, W.: Visual tracking based on semantic and similarity learning. IET Comput. Vis. **13**(7), 623–631 (2019)
191. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4353–4361 (2015)
192. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, pp. 84–92. Springer (2015)
193. Liu, Y., Zhang, L., Chen, Z., Yan, Y., Wang, H.: Multi-stream siamese and faster region-based neural network for real-time object tracking. IEEE Trans. Intell. Transp. Syst. **22**, 7279–7292 (2020)
194. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 459–474 (2018)
195. Li, K., Kong, Y., Fu, Y.: Visual object tracking via multi-stream deep similarity learning networks. IEEE Trans. Image Process. **29**, 3311–3320 (2019)
196. Jeany, S., Mooyeol, B., Cho, M., Han, B.: Multi-Object Tracking with Quadruplet Convolutional Neural Networks. IEEE Computer Society (2017)
197. Son, J., Baek, M., Cho, M., Han, B.: Multi-object tracking with quadruplet convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5620–5629 (2017)
198. Zhang, D., Zheng, Z.: Joint representation learning with deep quadruplet network for real-time visual tracking. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)

199. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 403–412 (2017)
200. Dike, H.U., Zhou, Y.: A robust quadruplet and faster region-based CNN for UAV video-based multiple object tracking in crowded environment. Electronics **10**(7), 795 (2021)
201. Wu, C., Zhang, Y., Zhang, W., Wang, H., Zhang, Y., Zhang, Y., et al.: Motion guided siamese trackers for visual tracking. IEEE Access **8**, 7473–7489 (2020)
202. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.; Learning dynamic siamese network for visual object tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1763–1771 (2017)
203. Yang, T., Chan, A.B.: Learning dynamic memory networks for object tracking. In: Proceedings of the European Conference on computer vision (ECCV), pp. 152–167 (2018)
204. Shi, T., Wang, D., Ren, H.: Triplet network template for siamese trackers. IEEE Access **9**, 44426–44435 (2021)
205. Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S.: SiamCAR: siamese fully convolutional classification and regression for visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6269–6277 (2020)
206. Kim, M., Alletto, S., Rigazio, L.: Similarity mapping with enhanced siamese network for multi-object tracking. arXiv preprint arXiv:1609.09156 (2016)
207. Ma, C., Yang, C., Yang, F., Zhuang, Y., Zhang, Z., Jia, H., et al.: Trajectory factory: tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2018)
208. Lee, S., Kim, E.: Multiple object tracking via feature pyramid siamese networks. IEEE Access **7**, 8181–8194 (2018)
209. Liang, Y., Zhou, Y.: LSTM multiple object tracker combining multiple cues. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 2351–2355. IEEE (2018)
210. Ma, L., Tang, S., Black, M.J., Van Gool, L.: Customized multi-person tracker. In: Asian Conference on Computer Vision, pp. 612–628. Springer (2018)
211. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. arXiv preprint arXiv:1406.6247 (2014)
212. Jenni, S., Jin, H., Favaro, P.: Steering self-supervised feature learning beyond local pixel statistics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6408–6417 (2020)
213. Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., Yu, N.: Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4836–4845 (2017)
214. Fiaz, M., Mahmood, A., Baek, K.Y., Farooq, S.S., Jung, S.K.: Improving object tracking by added noise and channel attention. Sensors **20**(13), 3780 (2020)
215. Kim, C., Li, F., Rehg, J.M.: Multi-object tracking with neural gating using bilinear lstm. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 200–215 (2018)
216. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
217. Zhao, F., Zhang, T., Wu, Y., Tang, M., Wang, J.: Antidecay LSTM for siamese tracking with adversarial learning. IEEE Trans. Neural Netw. Learn. Syst. **32**, 4475–4489 (2020)
218. Chen, X., Gupta, A.: Spatial memory for context reasoning in object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4086–4096 (2017)
219. Li, J., Wei, Y., Liang, X., Dong, J., Xu, T., Feng, J., et al.: Attentive contexts for object detection. IEEE Trans. Multimed. **19**(5), 944–954 (2016)
220. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018)
221. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755 (2014)
222. Kosiorek, A.R., Bewley, A., Posner, I.: Hierarchical attentive recurrent tracking. arXiv preprint arXiv:1706.09262 (2017)
223. Cui, Z., Xiao, S., Feng, J., Yan, S.: Recurrently target-attending tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1449–1458 (2016)
224. Milan, A., Rezatofighi, S.H., Dick, A., Reid, I., Schindler, K.: Online multi-target tracking using recurrent neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
225. Quan, R., Zhu, L., Wu, Y., Yang, Y.: Holistic LSTM for pedestrian trajectory prediction. IEEE Trans. Image Process. **30**, 3229–3239 (2021)
226. Shu, X., Tang, J., Qi, G., Liu, W., Yang, J.: Hierarchical long short-term concurrent memory for human interaction recognition. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1110–1118 (2019)
227. Fang, K., Xiang, Y., Li, X., Savarese, S.: Recurrent autoregressive networks for online multi-object tracking. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 466–475. IEEE (2018)
228. Zhang, S., Yang, J., Schiele, B.: Occluded pedestrian detection through guided attention in cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6995–7003 (2018)
229. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
230. Stollenga, M., Masci, J., Gomez, F., Schmidhuber, J.: Deep networks with internal selective attention through feedback connections. arXiv preprint arXiv:1407.3068 (2014)
231. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: International Conference on Machine Learning, pp. 1319–1327. PMLR (2013)
232. Zhao M, Okada K, Inaba M. TrTr: Visual tracking with transformer. arXiv preprint arXiv:2105.03817 (2021)
233. Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: TransCenter: transformers with dense queries for multiple-object tracking. arXiv preprint arXiv:2103.15145 (2021)
234. Zeng, F., Dong, B., Wang, T., Chen, C., Zhang, X., Wei, Y.: MOTR: end-to-end multiple-object tracking with transformer. arXiv preprint arXiv:2105.03247 (2021)
235. Sun, P., Jiang, Y., Zhang, R., Xie, E., Cao, J., Hu, X., et al. Transtrack: multiple-object tracking with transformer. arXiv preprint arXiv:2012.15460. (2020)
236. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
237. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: multi-object tracking with transformers. arXiv preprint arXiv:2101.02702 (2021)
238. Chen, Y., Cao, Y., Hu, H., Wang, L.: Memory enhanced global-local aggregation for video object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10337–10346 (2020)
239. Xiao, F., Lee, Y.J.: Spatial-temporal memory networks for video object detection. arXiv preprint arXiv:1712.06317 (2017)
240. Deng, H., Hua, Y., Song, T., Zhang, Z., Xue, Z., Ma, R., et al.: Object guided external memory network for video object detec-

tion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6678–6687 (2019)
241. Wang, L., Zhang, L., Wang, J., Yi, Z.: Memory mechanisms for discriminative visual tracking algorithms with deep neural networks. IEEE Transactions on Cognitive and Developmental Systems. **12**(1), 98–108 (2019)
242. Jeon, S., Kim, S., Min, D., Sohn, K.: Parn: pyramidal affine regression networks for dense semantic correspondence. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 351–366 (2018)
243. Xie, Y., Shen, J., Wu, C.: Affine geometrical region CNN for object tracking. IEEE Access **8**, 68638–68648 (2020)
244. Vu, H.T., Huang, C.C.: A multi-task convolutional neural network with spatial transform for parking space detection. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 1762–1766. IEEE (2017)
245. Zhou, Q., Zhong, B., Zhang, Y., Li, J., Fu, Y.: Deep alignment network based multi-person tracking with occlusion and motion reasoning. IEEE Trans. Multimed. **21**(5), 1183–1194 (2018)
246. Li, Y., Bozic, A., Zhang, T., Ji, Y., Harada, T., Nießner, M.: Learning to optimize non-rigid tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4910–4918 (2020)
247. Li, C., Dobler, G., Feng, X., Wang, Y.: Tracknet: simultaneous object detection and tracking and its application in traffic video analysis. arXiv preprint arXiv:1902.01466 (2019)
248. Zhu, H., Liu, H., Zhu, C., Deng, Z., Sun, X.: Learning spatial-temporal deformable networks for unconstrained face alignment and tracking in videos. Pattern Recognit. **107**, 107354 (2020)
249. Zhang, M., Wang, Q., Xing, J., Gao, J., Peng, P., Hu, W., et al.: Visual tracking via spatially aligned correlation filters network. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 469–485 (2018)
250. Zhang, X., Lei, H., Ma, Y., Luo, S., Spatial, Fan X.: Tracking, transformer part-based siamese visual. In: 39th Chinese Control Conference (CCC), vol. 2020, pp. 7269–7274. IEEE (2020)
251. Qian, Y., Yang, M., Zhao, X., Wang, C., Wang, B.: Oriented spatial transformer network for pedestrian detection using fish-eye camera. IEEE Trans. Multimed. **22**(2), 421–431 (2019)
252. Luo, H., Jiang, W., Fan, X., Zhang, C.: Stnreid: deep convolutional networks with pairwise spatial transformer networks for partial person re-identification. IEEE Trans. Multimed. **22**(11), 2905–2913 (2020)
253. Li, D., Chen, X., Zhang, Z., Huang, K.: Learning deep context-aware features over body and latent parts for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 384–393 (2017)
254. Zhang, Y., Tang, Y., Fang, B., Shang, Z.: Multi-object tracking using deformable convolution networks with tracklets updating. Int. J. Wavelets Multiresolut. Inf. Process. **17**(06), 1950042 (2019)
255. Wu, H., Xu, Z., Zhang, J., Jia, G.: Offset-adjustable deformable convolution and region proposal network for visual tracking. IEEE Access **7**, 85158–85168 (2019)
256. Cao, W.M., Chen, X.J.: Deformable convolutional networks tracker. In: DEStech Transactions on Computer Science and Engineering (iteee) (2019)
257. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. arXiv preprint arXiv:1506.02025 (2015)
258. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., et al.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)
259. Mumuni, A., Mumuni, F.: CNN architectures for geometric transformation-invariant feature representation in computer vision: a review. Manuscript accepted for publication, SN Computer Science **2**(5), 1–23 (2021)
260. Wang, X., Shrivastava, A., Gupta, A.: A-fast-rcnn: hard positive generation via adversary for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2606–2615 (2017)
261. Lin, C.H., Yumer, E., Wang, O., Shechtman, E., Lucey, S.: St-gan: spatial transformer generative adversarial networks for image compositing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9455–9464 (2018)
262. Zhang, D., Zheng, Z., Wang, T., He, Y.: HROM: learning high-resolution representation and object-aware masks for visual object tracking. Sensors **20**(17), 4807 (2020)
263. Johnander, J., Danelljan, M., Khan, F.S., Felsberg, M.: DCCO: towards deformable continuous convolution operators for visual tracking. In: International Conference on Computer Analysis of Images and Patterns, pp. 55–67. Springer (2017)
264. Araujo, A., Norris, W., Sim, J.: Computing receptive fields of convolutional neural networks. Distill **4**(11), e21 (2019)
265. Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R.: Siamese box adaptive network for visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6668–6677 (2020)
266. Jiang, X., Li, P., Zhen, X., Cao, X.: Model-free tracking with deep appearance and motion features integration. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 101–110. IEEE (2019)
267. Dequaire, J., Rao, D., Ondruska, P., Wang, D., Posner, I.: Deep tracking on the move: Learning to track the world from a moving vehicle using recurrent neural networks. arXiv preprint arXiv:1609.09365 (2016)
268. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122. (2015)
269. Li, Y., Zhang, X., Chen, D.: Csrnet: dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1100 (2018)
270. Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., et al.: Understanding convolution for semantic segmentation. In: IEEE Winter Conference on Applications of Computer Vision (WACV), vol. 2018, pp. 1451–1460. IEEE (2018)
271. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: Object-aware anchor-free tracking. arXiv preprint arXiv:2006.10721 (2020)
272. Weng, X., Wu, S., Beainy, F., Kitani, K.M.: Rotational rectification network: enabling pedestrian detection for mobile vision. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1084–1092. IEEE (2018)
273. Marcos, D., Volpi, M., Tuia, D.: Learning rotation invariant convolutional filters for texture classification. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 2012–2017. IEEE (2016)
274. Jacobsen, J.H., De Brabandere, B., Smeulders, A.W.: Dynamic steerable blocks in deep residual networks. arXiv preprint arXiv:1706.00598 (2017)
275. Tarasiuk, P., Pryczek, M.: Geometric transformations embedded into convolutional neural networks. J. Appl. Comput. Sci. **24**(3), 33–48 (2016)
276. Henriques, J.F., Vedaldi, A.: Warped convolutions: Efficient invariance to spatial transformations. In: International conference on machine learning, pp. 1461–1469. PMLR (2017)
277. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
278. Yang, L., Han, Y., Chen, X., Song, S., Dai, J., Huang, G.: Resolution adaptive networks for efficient inference. In: Proceedings

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2369–2378 (2020)
279. Tamura, M., Horiguchi, S., Murakami, T.: Omnidirectional pedestrian detection by rotation invariant training. In: IEEE winter conference on Applications of Computer Vision (WACV), vol. 2019, pp. 1989–1998. IEEE (2019)
280. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
281. Coors, B., Condurache, A.P., Geiger, A.: Spherenet: learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 518–533 (2018)
282. Rashed, H., Mohamed, E., Sistu, G., Kumar, V.R., Eising, C., El-Sallab, A., et al.: Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2272–2280 (2021)
283. Hao, Z., Liu, Y., Qin, H., Yan, J., Li, X., Hu, X.: Scale-aware face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6186–6195 (2017)
284. Yang, Z., Xu, Y., Dai, W., Xiong, H.: Dynamic-stride-net: deep convolutional neural network with dynamic stride. In: Optoelectronic Imaging and Multimedia Technology VI, vol. 11187, p. 1118707. International Society for Optics and Photonics (2019)
285. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.C., Qi, H., et al.: UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking. Comput. Vis. Image Underst. **193**, 102907 (2020)
286. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
287. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., et al.: The visual object tracking vot2015 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1–23 (2015)
288. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., et al.: The visual object tracking VOT2016 challenge results. In: Computer Vision—ECCV 2016 Workshops, pp. 777–823 (2016)
289. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., et al.: The visual object tracking vot2017 challenge results. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1949–1972 (2017)
290. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
291. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: a benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
292. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., et al.: Mot20: a benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
293. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
294. Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: a benchmark for higher frame rate object tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1125–1134 (2017)
295. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: European Conference on Computer Vision, pp. 445–461. Springer (2016)
296. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: algorithms and benchmark. IEEE Trans. Image Process. **24**(12), 5630–5644 (2015)
297. Huang, L., Zhao, X., Huang, K.: Got-10k: a large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intelligence. **43**, 1562–1577 (2019)
298. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., et al. Lasot: a high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5374–5383 (2019)
299. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: a large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 300–317 (2018)
300. Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., et al.: The unmanned aerial vehicle benchmark: object detection and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 370–386 (2018)
301. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
302. Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V.: Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5296–5305 (2017)
303. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
304. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4591–4600 (2019)
305. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6638–6646 (2017)
306. Yang, T., Chan, A.B.: Visual tracking via dynamic memory networks. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 360–374 (2019)
307. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4282–4291 (2019)
308. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: accurate tracking by overlap maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4660–4669 (2019)
309. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6182–6191 (2019)
310. Lukezic, A., Matas, J., Kristan, M.: D3S-A discriminative single shot segmentation tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7133–7142 (2020)
311. Xie, F., Yang, W., Zhang, K., Liu, B., Wang, G., Zuo, W.: Learning spatio-appearance memory network for high-performance visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2678–2687 (2021)
312. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: on the fairness of detection and re-identification in multiple object tracking. Int. J. Comput. Vis. **129**, 3069–3087 (2021)
313. Zheng, L., Tang, M., Chen, Y., Zhu, G., Wang, J., Lu, H.: Improving multiple object tracking with single object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2453–2462 (2021)
314. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.H.: Online multi-object tracking with dual matching attention net-

works. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 366–382 (2018)
315. Brasó, G., Leal-Taixé, L.: Learning a neural solver for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6247–6257 (2020)
316. Saleh, F., Aliakbarian, S., Rezatofighi, H., Salzmann, M., Gould, S.: Probabilistic tracklet scoring and inpainting for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14329–14339 (2021)
317. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., et al.: ByteTrack: multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021)
318. Wang, Q., Zheng, Y., Pan, P., Xu, Y.: Multiple object tracking with correlation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3876–3886 (2021)
319. Liang, C., Zhang, Z., Zhou, X., Li, B., Lu, Y., Hu, W.: One more check: making "fake background" be tracked again. arXiv preprint arXiv:2104.09441 (2021)
320. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J. Image Video Process. **2008**, 1–10 (2008)
321. Wu, S., Xu, Y.: DSN: a new deformable subnetwork for object detection. IEEE Trans. Circuits Syst. Video Technol. **30**(7), 2057–2066 (2019)
322. Liu, Y., Duanmu, M., Huo, Z., Qi, H., Chen, Z., Li, L., et al.: Exploring multi-scale deformable context and channel-wise attention for salient object detection. Neurocomputing **428**, 92–103 (2021)
323. Lee, H., Choi, S., Kim, C.: A memory model based on the siamese network for long-term tracking. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
324. Fiaz, M., Mahmood, A., Jung, S.K.: Learning soft mask based feature fusion with channel and spatial attention for robust visual object tracking. Sensors **20**(14), 4021 (2020)
325. Lee, D.J.L., Macke, S., Xin, D., Lee, A., Huang, S., Parameswaran, A.G.: A Human-in-the-loop Perspective on AutoML: milestones and the road ahead. IEEE Data Eng Bull. **42**(2), 59–70 (2019)
326. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)
327. Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., Xing, E.: Neural architecture search with bayesian optimisation and optimal transport. arXiv preprint arXiv:1802.07191 (2018)
328. Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., et al.: Nsga-net: neural architecture search using multi-objective genetic algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 419–427 (2019)