The Visual Computer (2021) 37:993–1014
https://doi.org/10.1007/s00371-020-01848-y

SURVEY

A survey on online learning for visual tracking

Mohammed Y. Abbass[1,2] · Ki-Chul Kwon[1] · Nam Kim[1] · Safey A. Abdelwahab[2] · Fathi E. Abd El-Samie[3,5] · Ashraf A. M. Khalaf[4]

Published online: 15 May 2020

Abstract
Visual object tracking has become one of the most active research topics in computer vision, which has been growing in commercial development as well as academic research. Many visual trackers have been proposed in the last two decades. Recent studies of computer vision for dynamic scenes include motion detection, object classification, environment modeling, tracking of moving objects, understanding of object behaviors, object identification, and data fusion from multiple sensors. This paper provides an in-depth overview of recent object tracking research. Object tracking tasks in realistic scenario often face challenging problems such as camera motion, occlusion, illumination effect, clutter, and similar appearance. A variety of tracker techniques have been published, which combine multiple techniques to solve multiple visual tracking sub-problems. This paper also reviews the latest research trend in object tracking based on convolutional neural networks, which is receiving growing attention. Finally, the paper discusses the future challenges and research directions for the object tracking problems that still need extensive studies in coming years.

Keywords Object tracking · Convolutional neural networks · Online learning · Deep learning · Real-time computer vision · Particle filter

1 Introduction

One of the major tasks in the field of computer vision is to help machines, such as robots, computers, drones, and vehicles, and perform the main tasks of the human vision system, such as image comprehension, and motion analysis. To realize these functions of intelligent motion analysis, many works have attempted to track the visual objects, which became a high-demand research area in the real-time computer vision field. Basically, the main step of visual tracking is to evaluate the trajectory model (i.e., position, direction, shape, etc.) of a tracked object in each scene of a video sequence. A robust tracker appoints consistent markers to the target objects in successive scenes. In short, visual tracking is an operation that seeks to locate, detect, and define the dynamic configuration of one or more objects in the video sequence of one or different cameras.

In recent years, researchers worldwide have been influenced by a broad range of real-world applications, including human activity recognition, video monitoring, visual com-

✉ Nam Kim
namkim@chungbuk.ac.kr

Mohammed Y. Abbass
myehiaa@yahoo.com; mhmd_abb@yahoo.co.uk

Ki-Chul Kwon
kwon@osp.chungbuk.ac.kr

Safey A. Abdelwahab
safeyash@yahoo.com

Fathi E. Abd El-Samie
fathi_sayed@yahoo.com

Ashraf A. M. Khalaf
ashkhalaf@yahoo.com

[1] School of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, South Korea

[2] Engineering Department, Nuclear Research Center, Atomic Energy Authority, Cairo, Egypt

[3] Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia 32952, Egypt

[4] Electronics and Communications Department, Faculty of Engineering, Minia University, Minia, Egypt

[5] Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia

pression, traffic control surveillance, and human–computer interaction. There are three fundamental stages in scene analysis: moving object detection, object tracking from scene to scene, and object analysis to observe the behavior. Therefore, the development of object tracking is relevant to many purposes. For instance, object tracking has been employed to enhance the analysis of human activities [1]. In the field of intelligent traffic systems, many object tracking techniques have been proposed to address traffic systems, such as traffic surveillance [2]; accident avoidance, especially at traffic intersections [3, 4]; and pedestrian counting [5, 6]. Moreover, the standard of MPEG-4 video compression [7, 8] exploits object tracking techniques to provide more encoding bytes to moving objects in the scene and fewer encoding bytes for the remaining redundant background scene. Currently, the most active applications are human–computer interaction, such as hand gesture recognition [9], which needs a powerful visual tracking mechanism. Tracking is motivating several companies, such as Sony and Intel, which have developed cameras appropriate for visual monitoring, like omnidirectional cameras [10, 11] and smart cameras [12]. Additionally, visual tracking is used in modern medicine. The tracker is employed to observe the tracking path of protein stress granules in cells and discover the characteristics of the cell structure [13]. Furthermore, military guidance utilizes visual tracking [14], as in rocket steering, individual combat systems, unmanned aerial vehicle (UAV) flight control, and radar detection.

The aforementioned works demonstrate a wide and mounting benefit in visual tracking in successive video scenes. Moreover, we can directly observe that the applications strongly depend on the results achieved by an object tracking method. If such a tracking method yields inaccurate outputs and unstable results, it could not be used for such applications. Therefore, the key to grow these applications is to overcome the problems associated with visual tracking. In addition, online robust visual tracking techniques are in high demand and many works are being developed to deal with online performance. Unfortunately, several challenges make visual tracking of objects complex. To create a robust visual tracking system, some difficulties need to be considered, which are as follows:

(i) The appearance of the object can be changed by the position and viewing angle, and it shows a large range of dimensions and distances.
(ii) The object could be tracked in highly dynamic scenes. The camera and tracked object are in motion, which makes tracking and analysis of the movement difficult.
(iii) Real-time processing is one of the main difficulties. A system should have high-speed performance to work with live scene sequences.

This paper does not review all existing works in visual tracking, as many algorithms have been published every year since the 1990s. Moreover, comparing different trackers is a non-trivial task. For these reasons, the paper only evaluates and compares some state-of-the-art works on visual tracking. As such, the paper will help researchers, especially newcomers, understand the performance of most of the existing trackers they need in order to compare their tracker results in terms of current issues in visual tracking. Another goal of the paper is to highlight the status of visual tracking, provide the challenges associated with the trackers, and present the research direction of recent publications. In addition, we debate the quickly rising technique in the tracking community, which is deep learning, and more specifically, Convolutional Neural Network (CNNs). We cover many aspects, measurement analyses, classifications, in-demand implementations, and the upcoming potential of the techniques.

Our work investigates the approaches of online-learning tracking, for which the first frame must have a bounding box. Such approaches can exploit adaptive appearance models, which aim to expand to consider continuous target deformations. Moreover, these approaches should observe the drifting issue. The approaches of pre-trained tracking are not discussed, for which an object is identified earlier than at system startup. The achievement of pre-trained tracking relies on sequential video frames, as well as the training data, which can be considered another problem. Furthermore, offline tracking is not regarded by our work, which utilizes the total enhancement of the path by scanning forward and backward through the video. Offline tracking is generally based on the needs of medical applications, but we consider the broader implementation domains of online tracking.

The paper is organized as follows. In Sect. 2, the principle of tracking is briefly reviewed. In Sect. 3, the paper states common and milestone visual tracking techniques and discussions. Then, CNN-based tracking is discussed in Sect. 4. Finally, the concluding remarks and summary are given in Sect. 6.

2 Principle of tracking

This section discusses the challenges that impact visual tracking. In addition, we present the different metric methods that are commonly applied to test the output of visual tracking approaches. We further discuss the principles of the classifications of visual tracking based on its applications and methods [15].

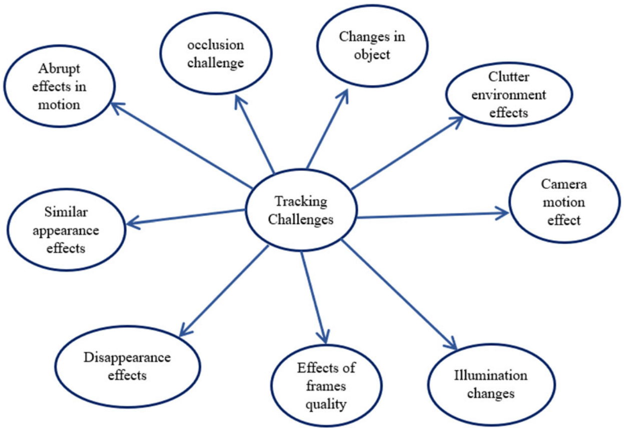


Fig. 1 Different tracking challenges

2.1 Challenges in visual tracking

Many issues appear when tracking through sequential frames that can potentially cause the failure of object tracking. Below, we consider a group of the most common challenges. Figure 1 summarizes these tracking issues.

Illumination changes. The light surrounding an object can continuously vary from area to area for many reasons, such as indoor (turning lights on or off) or outdoor (weather, time of day) lighting. Shadows also impact the lighting conditions. In the same spirit, reflectance or transparency factors can be shown in the scene, whose occurrence differs corresponding to incident light and angle of view. These types of variations create different color distributions over time on the tracked object, which disturbs the performance of the tracking mechanism.

Camera motion effects. Many applications require embedding one or more cameras, as in the case of vehicles, drones, and body-mounted systems. In such applications, it is difficult to distinguish the motion associated with the tracked object from the motion associated with the embedded camera. An embedded camera can also produce irregular motion that cannot be modeled. This motion, in some cases, causes motion blurring, which deforms the image details, and therefore, decreases the visual tracker performance.

Cluttered environment effects. Cluttered scenes are created by either additional objects, especially objects that are similar to the tracked object, or highly textured backgrounds. This confuses the visual tracking algorithm and results in output drift.

Changes in object model. The tracked object has some geometric degradation, because the frames are projected from the 3D space onto the 2D plane. In other words, the object shape is changed based on 3D-to-2D projection, and therefore, some information is lost.

Effects of frames quality. The sensors and acquisition conditions impact the quality of the consecutive video frames. When the video sequence has been compressed, block artifacts can be observed. This causes the visual tracker to yield an undesired output.

Occlusion effects. Other unwanted objects in the scene may occlude the tracked object. The tracker encounters difficulties in that case, because the tracked object can be hidden, either partially or completely, and sometimes, the most important parts of the object can be hidden from the tracker scene.

Disappearance effects. The tracked object may enter the scene, but it leaves temporally due to object motion. In another case, the object may be visible across two or more cameras without overlapping in the scene (for instance, a person can enter by one door and leave from another). In these cases, the visual tracker should memorize the tracked object and be able to find it upon its reappearance in the scene. The difference between occlusion and disappearance is that in the former, the object is covered by another unwanted object, but the object is still in the scene (e.g., the object is walking behind wall or the target person is standing behind tree). In the case of disappearance, the tracked object is removed completely from the scene for a while, but reappears in either the same or another camera scene.

Abrupt effects in motion. The motion rate of the tracked object can change abruptly over the time. This change can be unpredictable, and therefore, the tracker can misplace the object location due to incorrect location prediction.

Similar appearance effects. When the video sequence frames have objects similar in appearance to the target object, as in the case of tracking vehicles on the road, the differentiation between the correct object and the similar objects becomes a difficult challenge for the visual tracking algorithm.

A robust visual tracking technique is required to resist the above-mentioned issues appearing in the successive frames, which are strongly interesting for the researchers. Moreover, the quality of the tracker results must cost less in terms of computational and time efficiency. Today, based on our knowledge, no approach can satisfy all these requirements. Therefore, future tracking approaches are still concerned with these challenges, in different applications such as driver assistance systems, vehicle navigation, traffic surveillance, video player analysis, activity-based recognition, human—computer interfacing, and motion analysis. The visual tracking algorithms are classified based on their applications.

2.2 Classification of visual tracking algorithms

Classification by camera movement. Visual tracking algorithms can be categorized based on the condition of the camera, which is either stationary (static) or non-stationary (moving). The background is unchanging in the condition of a stationary camera; therefore, the foreground and background can be segmented simply. Several works have been presented to segment the foreground and background, such as mixture of Gaussian (MOG) [16, 17]. In the case of a moving camera, the segmentation process between foreground and background is a complex step, because both are changing.

Classification by scene. Visual tracking has two scene scenarios in existing applications. The first one is tracking using a single scene, and the second is tracking across multiple scenes. The single-scene tracking depends on one camera to track the target object, whereas multi-scene scenarios depend on an established network by multiple different cameras to track the object [18]. In the case of multiple scenes, a unique identifier for the object is determined and it is tracked the object continuously using fused images.

Classification by number of moving objects. Visual object tracking can be divided into two classes based on the number of moving objects: single object and multiple objects. Generally, multiple-object tracking is more difficult than single-object tracking. However, both should perform the steps of object detection correctly and object extraction precisely from the video frames. Several factors affect the result of the detection and extraction steps, such as noise, background clutter, and illumination. Multiple-object tracking should tolerate mergers, detection, and occlusion among these objects. In contrast, in the case of single-object tracking, we define one object as the target object and other objects as the background.

Evaluation metrics of visual tracking. The visual tracking algorithm can be compared based on qualitative metrics. Unfortunately, qualitative metrics are insufficient, especially when two or more approaches have similar results. Therefore, quantitative methods have been used and many quantitative metrics for testing the efficiency of trackers have been adopted. Typically, the visual tracking performance is compared against the ground truth. This sub-section presents different quantitative measures, namely the most general evaluations applied in object tracking. The three main types of error in tracking are deviations, false positives, and false negatives. In the deviation case, the deviation error of the location of the object is computed from the ground truth. In a false positive result, the object marked is not a target object. In a false negative result, the object is missed, but it is in the scene.

The overlap between the detected and ground-truth object is calculated based on PASCAL [19]:

$$\frac{T^i \cap GT^i}{T^i \cup GT^i} \geq 0.5 \tag{1}$$

where T^i is the tracked location in scene i and GT^i is the ground-truth location in the same scene. If Eq. (1) is realized, the tracking approach can be consistent with the ground truth. Many researchers have developed PASCAL overlap without a threshold, called Dice [20], which is similar to the similarity metric without a threshold. However, most studies apply a threshold, because it can be used to easily compute metrics on large videos.

Another popular metric comprises precision and recall.

$$\text{Precision} = \frac{n_{tp}}{\left(n_{tp} + n_{fp}\right)} \tag{2}$$

$$\text{Recall} = \frac{n_{tp}}{\left(n_{tp} + n_{fn}\right)} \tag{3}$$

Here, n_{tp}, n_{fp}, and n_{fn} are the number of true positives, false positives, and false negatives in a sequence, respectively. Precision and recall can be embedded in the F-score [21]:

$$F = 2 \cdot \frac{\text{Precision } \cdot \text{Recall}}{\text{Precision } + \text{Recall}} \tag{4}$$

$F1$-score [22] is considered, when the area is included:

$$F1 = \frac{1}{N_{\text{frames}}} \sum_i \left(2 \cdot \frac{p^i \cdot r^i}{p^i + r^i}\right) \tag{5}$$

Here, p^i and r^i are:

$$p^i = \frac{\left|T^i \cap GT^i\right|}{T^i} \tag{6}$$

$$r^i = \frac{\left|T^i \cap GT^i\right|}{GT^i} \tag{7}$$

Equation (7) considers the average coverage of the tracked object region and the ground-truth region.

We calculate the object tracking accuracy (OTA) metric as follows:

$$OTA = 1 - \frac{\sum_i \left(n^i_{fn} + n^i_{fp}\right)}{\sum_i g^i} \tag{8}$$

Here, g^i determines the number of ground-truth bounding boxes in sequence i. The OTA computes how much tracked object patches match with ground-truth patches. In the same manner, object tracking precision (OTP) can be expressed with precision similar to that of Dice [23]:

$$OTP = \frac{1}{|M_s|} \sum_{i \in M_s} \frac{\left|T^i \cap GT^i\right|}{\left|T^i \cup GT^i\right|} \tag{9}$$

Here, M_s refers the number of frames in a video, where the tracked object area overlaps the ground-truth area. The average tracking accuracy (ATA) is also similar to OTP [24].

$$ATA = \frac{1}{N_{frames}} \sum_{i \in M_s} \frac{\left|T^i \cap GT^i\right|}{\left|T^i \cup GT^i\right|} \tag{10}$$

The method in [25] applied Deviation, in which the central location error is used as a metric for tracking accuracy:

$$\text{Deviation} = 1 - \frac{\sum_{i \in M_s} d\left(T^i, GT^i\right)}{|M_s|} \tag{11}$$

Here, $d(T^i, GT^i)$ is the normalized distance between the centroids of patches T^i and GT^i [26].

3 Tracking methods based on online learning

The challenging task in visual tracking is handling the appearance variations of a target object. The appearance variations are categorized as intrinsic or extrinsic. Intrinsic appearance variations include shape deformation and position variation of a target object, whereas extrinsic variations include changes resulting from varying illumination, camera motion, camera viewpoints, and occlusion. These variations should be updated with adaptive mechanisms that have the ability to continuously adapt their modeling and representations. Thus, there is an essential necessity for an online mechanism that learns incrementally. Generally speaking, existing tracking methods are classified into generative methods and discriminative methods [27].

3.1 Tracking using generative online learning

Generative online learning approaches are used to track an object by searching for the areas that are most similar to the target model. The online learning approach is executed in the tracking algorithm to adapt the representative model of the target in response to appearance variations. Next, some recent improvements in online-learning-based generative tracking approaches are detailed. These approaches are inspired by the developments in appearance representation.

The deficiency of appropriate appearance representation is a crucial aspect that weakens the results of visual tracking algorithms. Classical template matching tracking procedures cannot handle appearance variations, because they use static models. Therefore, dynamic templates based on online learning are adopted to model the appearance variations of an object due to changes in illumination and posture.

Jepson et al. [28] applied the online expectation maximization (EM) technique and adopted a wavelet-based mixture model to improve the appearance representation and interpret the tracking factors efficiently. Zhou et al. [29] included the EM based algorithm for updated appearance representation with a particle filter to improve performance. This method has two EM processes, one for improving the appearance representation and the other for concluding the tracking factors. Tu and Tao [30] developed an online EM method to calculate the appearance representation characteristics and improve the histogram space through incoming observations. The EM has efficient characteristics in terms of stability and simplicity. However, the high number of iterations can cause the result to easily reach a local optimum, and the slow convergence can cause target loss and tracking failure.

Fussenegger et al. [31] presented an online method that can adapt a shape model of fewer dimensions by utilizing incremental principal component analysis (IPCA). This method preserves the latest model of the object to learn variations in the object itself and variations in the surrounding observation. However, in each update, only one sample is dealt with. Yang et al. [32] converted the scene to the grids of histograms of oriented gradients (HOG). Therefore, the IPCA–HOG descriptor has been developed to allow the tracking process to address variations in the appearance as well as the cluttered scene. Chiverton and Xie [33] developed an online updating method based on an active contour shape and a bootstrapping stage. The bootstrapper has been used to obtain the shape characteristics repeatedly from successive scenes. Chiverton et al. [34] applied a memory for object features in a high-dimensional shape coordinate to adapt high-level shape information online. These shapes have been utilized to determine templates. The essential limitation of this technique is that it is unsuitable for real-time processing due to its inappropriate computational speed. Furthermore, similar to many active contour tracking methods, effective tracking follows an empirical selection of factors that adjust the relative contribution of the different model components.

Liu et al. [35] applied a tracking method based on online learning with hybrid models that contain several forms of features, including sketch, texture, and flatness. The hybrid model is learned by computing the most discriminative features from the foreground. Then, the model is developed by modifying the feature confidences and removing the older, less discriminative features with the newer, more discriminative ones from the current scene. Several types of features are used with each other to more fully represent the target than a single feature could. One main weakness of these approaches is the lack of clear removal of the invalid patches during template preservation. To address this issue, Xu et al. [36] derived a hyper model, using HOG, center-symmetric local binary patterns, and color histograms to depict the local statistics of edges, textures, and flatness of objects, which are automatically adjusted online by combining new, efficiently selected patches computed from the fusion of the matching templates and the prospective set.

Kwon and Lee [37] divided the observational paradigm into multiple fundamental observational paradigms to compute the appearance of the object. Each fundamental observational paradigm contains a certain feature of the appearance of the target and is created dynamically at each time step by sparse principal component analysis (SPCA). The global tracker is started from various fundamental trackers corresponding to the fundamental observational template and motion templates. Each tracker covers a specific variation in the object or its surroundings and increases the method stability to numerous variations. However, this technique is insufficient for difficult tracking tasks with severe variations between scenes, because of the fixed number of basic trackers.

To address this issue, Kwon and Lee [38] used a tracker sampler to determine multiple suitable trackers from the tracker space dynamically to update to certain variations. The result of this method is very good, even in a real scene. However, compared with the multi-feature model, the computation of effective templates increases the cost of the computation. Thus, without further enhancement, the algorithm cannot be applied to real-time tracking.

Instead of applying a simple methodology to derive the appearance model for tracking, an online learned subspace representation is employed to provide a compact representation of the object and indicate appearance variations through tracking. The subspace probabilistic model provides an effective calculation.

Ross et al. [39] developed an adaptive probabilistic tracking method based on an inference probabilistic Markov model to adapt the templets of an object by means of incremental eigenbasis updates. Then, in consideration of the changing sample mean over time, an incremental mean update has been incorporated into the learning method [40]. Owing to intrinsic and extrinsic parameters, the appearance of the object has been learned in order to handle variations based on a low-dimensional eigen space representation. Moreover, an impact-lessening parameter was added into the incremental subspace update process to decrease the impacts of earlier observations on the existing appearance model based on IPCA. This technique adapts the appearance pattern more suitably in order to increase the overall tracking result. An online incremental method based on an appearance class was developed by Lee and Kriegman [41]. It is implemented by a combination of sub-groups and the connectivity between them. Each group is described by a principal component analysis (PCA) domain. This method uses a previous appearance model of a class of objects into the appearance model of an object of this class by incrementally learning online from the successive frames, including the target instance. Because of the use of hyper structures, this method has better strategies than the online update method to obtain a more accurate appearance model. The limitation of this method is that a previous model is demanded. In other words, the algorithm tracks an object if it has a model of the object class being tracked.

In all of the above-mentioned frameworks, the tracking relies on image-as-vector representations, which do not obviously use the spatial information within the image pixels.

Researchers introduced image-as-matrix methods or high-order tensors to form representations of image pixels. Li et al. [42] developed a three-dimensional (3D) temporal representation for incremental learning using adaptive updates of the sample mean and eigenbasis. This method succeeded in representing the appearance of an object more informatively. Wen and Gao [43] combined the retinex image with the original image by defining a weighted subspace representation of an object to consider the illumination variations. The online learning mechanism adaptes the appearance model and the different illumination due to the light reflectance. This approach does not update, but empirically determines the weight based on the representation. For more informative modeling, the target features to construct covariance matrices in five modes are applied to consider both spatial and statistical parameters of the object appearance [44]. Each mode of the object updates the eigen-basis and sample mean online to incrementally learn an eigenspace representation to handle appearance variations. The covariance calculation has a computational burden and cannot be embedded in real-time applications. Wu et al. [45] introduced a framework to lower the computational burden using incremental covariance representation updates. The current covariance model computation depends on weighting to give newer samples greater impact.

Lu et al. [46] proposed a subspace learning method that depends on exploitation of a locally-connected graph (LCG). The semantic subspace representation is trained by creating a supervised graph with some labeled object features. The LCG integrates the objects with minor negative features to have a robust subspace through the projection, which is built before the tracking process. Features of the object are categorized based on semantic details into some categories such as illumination, occlusion, and rotation. The LCG uses added label rules to define the subgraph of each class to generate a better informative and reasonable graph to tackle the drifting issue [47].

The appearance template of the target is defined in sparse constraints by a linear combination of only a few basis vectors. The tracking is derived by comparing features with sufficient accuracy in a learned template subspace.

In Mei and Ling [48], the object can be defined as a linear combination of the online updated object samples and negative samples. Then, tracking is considered as a sparse calculation task. The sparsity is realized by determining a least squares problem. However, partial occlusion, appearance changes, and other challenging cases are regarded through an error vector represented by the group of negative samples. This approach showed a stable tracking outcome through experiments. However, it does not handle abrupt pose variations or full occlusion of the object. Liu et al. [49] developed a two-step sparse enhancement method for tracking (Two-step Sparsity Tracking, TST). A sparse set of samples is adopted to decrease the target remodeling error and increase the discriminative power. The template set and the training set are adapted online to improve the efficiency of the tracker. This approach does not address partial occlusion in the case of modeling of the target as a single entity.

Liu et al. [50] applied a basis distribution that updates automatically online and a fixed sparse dictionary to represent the appearance of the object. Chen et al. [51] represented the appearance of the object with the actual intensity pixels of the object region. A similarity measure is used to compute the distance between a tracked object and the updated appearance template. The maximum a posteriori approximation is employed to calculate the object conditions in each frame over time, depending on Bayesian inference.

The model of Jia et al. [52] learns online, depending on sparse representation and incremental subspace learning (ISL) to account for the partial occlusion and drifting issue. The learned framework reinforces the tracker to tolerate the changes in an object appearance. Lu et al. [53] used the geometrical information of the object template set based on sparse representation. This approach is called non-local self-similarity regularized coding, and it utilizes K-nearest neighbors (KNNs) to model the structural features of the object. In this model, the weights of the templates are then learned to account for the appearance variations. It has a robust performance, but the tracking speed restricts its application.

3.2 Tracking using discriminative online learning

Discriminative online learning frameworks, called tracking by detection, handle object tracking as a classification task. They simultaneously exploit features of the object and the background. A binary classifier is used to discriminate the object from its background, and it is trained online to address variations in the environment and appearance. This classifier utilizes features from both the object and the background. Next, the various discriminative tracking frameworks that are dependent on online learning are presented by category, according to where the online update technique is applied.

The discriminative tracking frameworks depend directly on the feature space employed. If the features of an object are readily distinguished from its surroundings, the tracker will usually be able to track it. The updated online feature space is applied for visual tracking, instead of applying a static group of features that is specified a priori. These frameworks adaptively rank the features, and the highest ranked discriminative features are used in the tracking mechanism.

In [54], training features were extracted from raw images using RGB coordinates. Then, the color transform function was used to convert the RGB space into different color spaces, such as normalized RGB, XYZ, YCbCr, and YIQ. Finally, linear discriminant analysis (LDA) is used to build a histogram for tracking using a single-color coordinate, which is decided online.

Collins et al. [55] developed an adjustable online framework to improve and update the proper features for tracking. All features are ranked by computing the distinctions between the object and the background features over time, to determine the most appropriate features to handle the appearance. Then, the selected features are used to identify pixels in the current frame for association with either the object or background category. Both of these two previous approaches utilize color information, which does not have a perfect discrimination property under many conditions, to account for the object and background. For example, the tracker loses the object when the object and background areas have identical or very similar color information over successive frames. Nguyen and Smeulders [56] embedded textural features to enhance the modeling of the object and the background. Changes in foreground appearance are exploited with features extracted from Gabor filters. This framework is generally robust, but textural features increase dimensionality, which makes it invalid for real-time purposes.

In Wang et al. [57], the feature selection method in the particle-filtering process has the advantage of using the current background particles. The Fisher discriminant technique is applied to determine the online discriminative features in a large feature space. However, this approach is also inefficient in real time, because of the number of features and the characteristics of the particle filter. Li et al. [58] used 2D LDA to compute a 2D image matrix instead of transforming 2D images into vectors. The method recursively computes an improved projection subspace, by only updating the model at specific frame intervals rather than for every frame, resulting in less computation time. However, tracking failure may occur, when there is large variation between successive updates, such as an abrupt occlusion or a change in the other side of the face.

Specific features of an object can be used to train an online binary classifier. We cannot have sufficient appearance information of the object; therefore, the binary classifier should be continuously updated to compensate for the insufficiency of training features. The dependence of the classifier on online learning techniques is considered. The object initial location is determined in the current frame, and then, the classifier computes various probable locations in the surrounding area for successive frames. Avidan [59] applied adaptive boosting (AdaBoost) to build a robust classifier by integrating an ensemble of weak classifiers. Each individual classifier is developed online with several training samples based on an 11D histogram of feature coordinates, including a local orientation histogram and pixel colors. Then, the pixels in the following frames are assigned using a strong classifier, labeling them as either object or background and generating a confidence map. The new object location is defined using the highest confidence score in the map. This approach requires a small amount of computational time; however, it is sensitive to noise samples disturbing the tracker performance.

The methods discussed so far handle variations in appearance, cluttered backgrounds, and short-term occlusion. However, drifting can occur due to the accumulated errors in accuracy. To address this issue, the semi-supervised AdaBoost classifier has been developed [60]. The classifier updating process is controlled by a second classifier trained on the first frame. The method categorizes features extracted from the first frame only, and subsequent training features are uncategorized. The performance is unsatisfactory due to tracking errors as a result of extracting sub-optimal positive features.

The online multiple-instance learning (MIL) approach [61] has been developed to solve this difficulty. The classifier is updated, when the existing tracker patch is captured as positive features and the surroundings as negative features, because the object may not be fully present in the bounding box, or it may dominate most of the background. An object area is considered with additional bounding boxes within close range to generate a positive set. Multiple negative sets are extracted using bounding boxes with a distant range. Next, the Haar method is applied, as follows. Prospective bounding boxes are prescribed uniformly in a circular region around the original area. The maximum classification score is used to define the updated location of the object in the MIL, and the classifier coefficients are updated with the new data points. The MIL framework is computationally expensive due to ambiguity between samples of positive sets. Batch-mode adaptive MIL (Li et al. [62]) was designed to reduce the computation time, by separating training sets into batches instead of applying them all at once, allowing real-time tracking.

For long-term tracking, Kalal et al. [63] developed an efficient method that divides the process into tracking, learning, and detection (TLD). For the tracking part, a short-term approach is used, based on the Kanade–Lucas–Tomasi method, and random forests are used in the detection stage [64]. A positive–negative (P–N) learning module computes false positives and false negatives. The object is defined in the first frame, and then, the pattern is observed by the detector using two-bit binary patterns differentiated from surrounding background patterns. In the subsequent frame, the detector determines the locations of the top 50 scores, and then each potential candidate window is computed using normalized cross-correlation. Next, the prospective window with the greatest similarity to the object is labeled as the same object.

Positive samples are considered to be in the vicinity of the object after the new location is determined, and negative samples are considered to be further away from the object. The main advantage of this method is the ability to learn a new appearance and to avoid repeating mistakes. However, it also has several challenges. For example, TLD cannot provide good results in the case of a rotation out of the original plane. For the case where an object leaves the field of view, Hare et al. [65] developed a method named Struck, which relies upon a multi-structured output support vector machine (SVM). It explicitly learns a prediction function to directly compute the object transformation in-between output frames. Alternatively, to address the drift issue, Zhang et al. [66] proposed a multi-expert restoration structure instead of learning one classifier only.

Bolme et al. [67] added correlation filters (CFs) into the tracking process, and proposed the minimum output sum of squared error (MOSSE) filter. This process converts the essential convolution operations in the time domain into simple additions and multiplications in the frequency domain. The MOSSE performs favorably against state-of-the-art trackers with 600 frames per second. The CF-based tracking algorithms have grown in popularity within the tracking community. Heriques et al. [68] introduced a circular structure kernel (CSK) algorithm that adopts a dense sampling training pattern created by circular shifts of an input image patch. The CSK tracker relies on illumination intensity features. It has been developed to use more robust features such as the histogram of oriented gradients (HOG) in kernelized CFs (KCFs) [69], and color attributes or color names (CNs) [70].

Danelljan et al. [71] combined two independent CFs for robust scale computation. Prior to this, CF-based trackers could not handle a scale change in the target. Li and Zhu [72] applied a multi-resolution extension of a KCF (denoted SAMF) for scale changes. Danelljan et al. [73] simultaneously computed the scale and translation of the target object while minimizing the search space. In another study [74], the color histogram was proposed, and channel and spatial reliability were developed. HOG attributes are not sensitive to motion blur or variation in illumination, while CN features maintain robustness to shape deformation. Combining the advantages of using HOG and CN, Bertinetto et al. [75] created the STAPLE tracker, which produces outstanding results compared to state-of-the-art methods.

3.3 Tracking using combined online learning

Generative approaches can efficiently handle the object appearance, but they have poor performance in complex backgrounds. On the contrary, discriminative approaches have the ability to model complex backgrounds and significant appearance variations. Nevertheless, the discriminative approaches cannot handle high noise and generally do not incur drifting. Furthermore, the discriminative approaches can be interrupted by other objects that have a similar appearance. Thus, many researchers have developed frameworks to combine the advantages of both approaches to create a robust tracker.

Lin et al. [76] proposed a discriminative generative framework. The generative method is applied online to track the observational model and the discriminative method is adopted to compute the next position of the object. Zhang et al. [77] introduced a framework based on graph-based discriminative tracking. This framework integrates Fisher discriminant analysis (FDA) and ISL for tracking. The target subspace and the pattern models of graphs are learned online concurrently to collect the appearance variations and derive the object from its background. This framework attempts to preserve within-class compactness to adjust the position. However, it suffers from the drifting error that is accumulated as the track progresses.

Yu et al. [78] proposed a co-training technique to incorporate both generative and discriminative models. The generative model depends on subspace features in online learning to model the object appearance and learn different appearance variations. The discriminative model depends on the continuously updated support vector machine (SVM) classifier with HOG features. The SVM is updated and trained to capture the new appearance. This framework is robust and efficient, but abrupt appearance and occlusion disrupt the tracker performance. Yin and Collins [79] introduced a framework to mitigate the accumulated pixel classification inaccuracy. This framework applies the global shape and region-based probability of the object boundary. Yang et al. [80] proposed a novel tracker that depends on a particle filter and sparse representation. Each object can be modeled by object templates and surrounding templates with an additional representation error to learn appearance variations. Both templates of the object and its surroundings are embedded into a voting method to differentiate between the object and the background.

4 CNN-based tracking

Machine learning has been revolutionized by deep-learning methods, and so, the tracking community has been working to glean from this subject area to improve visual tracking methods. In general, traditional tracking techniques, including online learning techniques, employ man-made features to improve robustness. Over the last 5 years, deep-learning techniques [81] have produced good results in feature extraction via multi-layer nonlinear transformations in numerous applications. These include computer vision [82, 83], speech recognition [84, 85], and natural language processing [86, 87]. This means that deep-learning processes automatically obtain groups of features from the given images [88]. Preprocessing steps may be used, such as the pyramidal technique [89]. As first described in 2006 [90], the key feature of a deep-learning model is its layers. Essentially, they depend on the multi-layered architecture of data representation that is performed within the neural network, and they extract the characteristics directly from the raw input. For image analyses, architecture layers learn from the adjacent chain, e.g., pixels, then edges, then groups of edges, then shapes.

A deep-learning model is configured by several neurons in various hidden layers. The hidden layers represent the inputs to higher-level mapping. Generally, the aim of implementing deep learning within tracking is to distinguish patterns more quickly and accurately than a human does, thereby enhancing the efficiency of video applications.

The main advantages of deep-learning methods are as follows:

(i) Development of efficient representations and producing new architectures to update and learn these representations from large-scale unlabeled data.
(ii) The ability to directly deduce a complex set of features at a high level of abstraction.
(iii) The ability to learn low-level features from minimally processed input samples.
(iv) The ability to make decisions using a large number of datasets.

Typically, deep-learning networks are divided into five main groups: convolutional neural networks (CNNs), deep belief networks (DBNs), stacked auto-encoders, deep Boltzmann machines (DBMs), and deep residual learning (DRL) networks.

4.1 History of deep learning

Historically, deep learning has been used since the inception of artificial neural networks (ANNs) [91]. ANN methods have brilliantly dominated in the previous decades for recognition, segmentation, enhancement, and prediction in the areas of industry, biology, finance, robotics, marketing, medicine, manufacturing, and detection of moving objects [92]. In 1980, the perception of deep learning began when the neocognitron model was suggested by Fukushima [91]. LeCun et al. [93] designed a method to address the recognition issue of hand-written ZIP codes by utilizing the back-propagation technique in a deep neural network. However, this method had significant drawbacks which made it impractical to use, as the training time was unreasonable. However, deep neural networks have been applied to speech recognition for several years [94].

Over the next two decades, many research groups attempted to reduce the time cost of the tracking. Hinton [90] achieved great outputs for training multi-layer DBNs by applying an unsupervised restricted Boltzmann machine to pre-training of a single layer at a time. Then, a supervised backpropagation method was exploited for additional improvements. After that, several research domains implemented a primitive deep-learning model to handle various

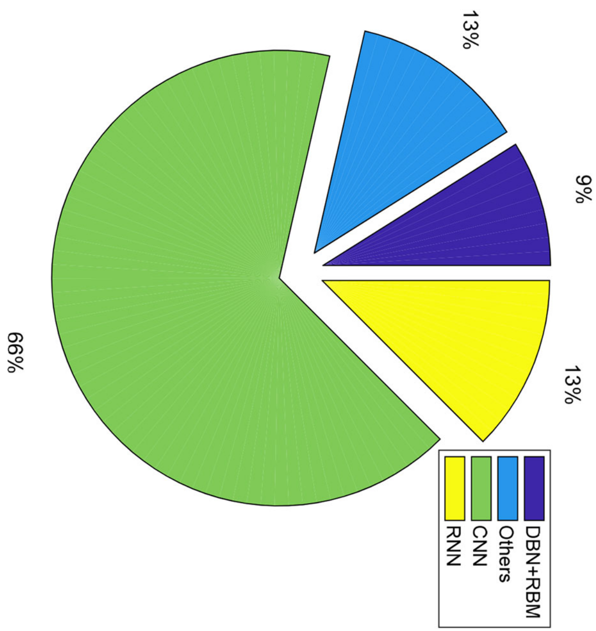


Fig. 2 Percentages of different deep-learning architectures

issues. The parallel configurations of hardware and software have resulted in magnificent achievements in deep-learning methods in recent developments [95, 96]. Several models of deep learning have been proposed. Milestone models are CNNs [97], stacked autoencoders [98], and recurrent neural networks (RNNs) [99].

4.2 CNN-based tracking

The CNN model attracts most researchers with its impressive performance, particularly in the computer vision area. Figure 2 shows the percentage of different deep-learning architectures in recently published works for object detection, recognition, and tracking. We can easily observe that the CNN technique is used in 66% of the applications [100].

The CNNs are composed of a multi-layered artificial neural network architecture. Each layer contains several neurons. During the first step, the convolutional layers apply a convolution operation between filters and patches of the input image, outputting a feature map. In the second step, each convolutional layer feeds into a layer that applies a nonlinear function to the feature map. The third step is to down-sample the feature map to decrease its features. Down-sampling can be done in several ways, such as minimum, average, or maximum pooling. Based on the application, these three steps are continuously iterated until the desired high-level feature map is extracted. Finally, the fully-connected output layer generates a certain number of class outputs. The main advantages of the CNN are that it is simple to train, and less dependent on earlier iterations of the model and on human knowledge compared to other methods. It directly receives the 2D input data structure, and can also receive the 3D input.

Table 1 Comparison results in terms of average center error (unit pixels) on different attributes (the bold indicates the best performance)

Sequence	TLD [63]	MEEM [66]	CSK [68]	KCF [69]	SAMF [72]	DSST [73]	CSRDCF [74]	STAPLE [75]	CF2 [101]	CNN-SVM [103]	DeepSRDCF [104]	CCOT [107]	ECO [108]
Deer	1.07e+02	9.0094	7.34	23.386	13.4873	5.327	4.5004	20.04	7.207	5.484	6.6351	**4.4619**	4.621
Tiger1	1.12e+02	12.237	73.276	**8.257**	53.66	9.17	9.753	9.886	51.66	57.82	58.239	56.97	13.35
Toy	13.441	**12.501**	45.096	20.4368	16.277	16.4031	15.667	15.4943	17.9305	19.282	19.5493	15.5191	14.208
BlurCar1	3.034e+02	10.759	11.436	1.6e+02	7.352	12.4952	18.942	64.465	9.92	11.436	**6.2154**	6.2794	6.297
BlurFace	19.2242	8.988	1.8053e+02	1.8044e+02	9.413	7.6518	8.2585	9.212	6.95688	**6.8865**	11.9618	8.6041	10.5732
Box	67.142	1.552e+02	1.19e+02	86.788	**15.348**	93.913	1.022e+02	82.415	1.0258e+02	35.0079	32.4419	15.9194	15.3481
Skating1	1.67e+02	79.72	22.93	22.744	17.818	23.07	1.441e+02	70.911	23.578	41.9729	64.349	61.1809	**22.42**
Crowds	2.838e+02	5.6131	3.6906	3.065	4.332	4.5126	4.3613	2.9042	**2.789**	7.21966	4.60359	3.47099	3.776
Basketball	2.4709e+02	8.32161	6.527	7.8894	10.3717	11.73	11.0261	19.0974	**3.56209**	20.8885	30.5855	6.19374	19.24
David3	2.37e+02	5.43262	56.4581	5.518	3.984	**3.6503**	4.6035	3.9205	5.5622	5.6829	4.6136	3.998	3.7
Football1	1.019e+02	4.31498	15.7262	5.1921	6.01707	2.94103	5.72648	4.0352	**2.56727**	9.9171	5.0553	5.167	5.336
Rubik	27.3184	14.025	27.35	22.218	14.244	17.19	26.294	14.54713	21.76253	15.43915	11.11562	**9.816**	10.38
DragonBaby	1.37e+02	9.54213	90.8	51.649	26.38	60.48	31.7133	23.08	17.954	11.9442	12.9765	**9.038**	11.40

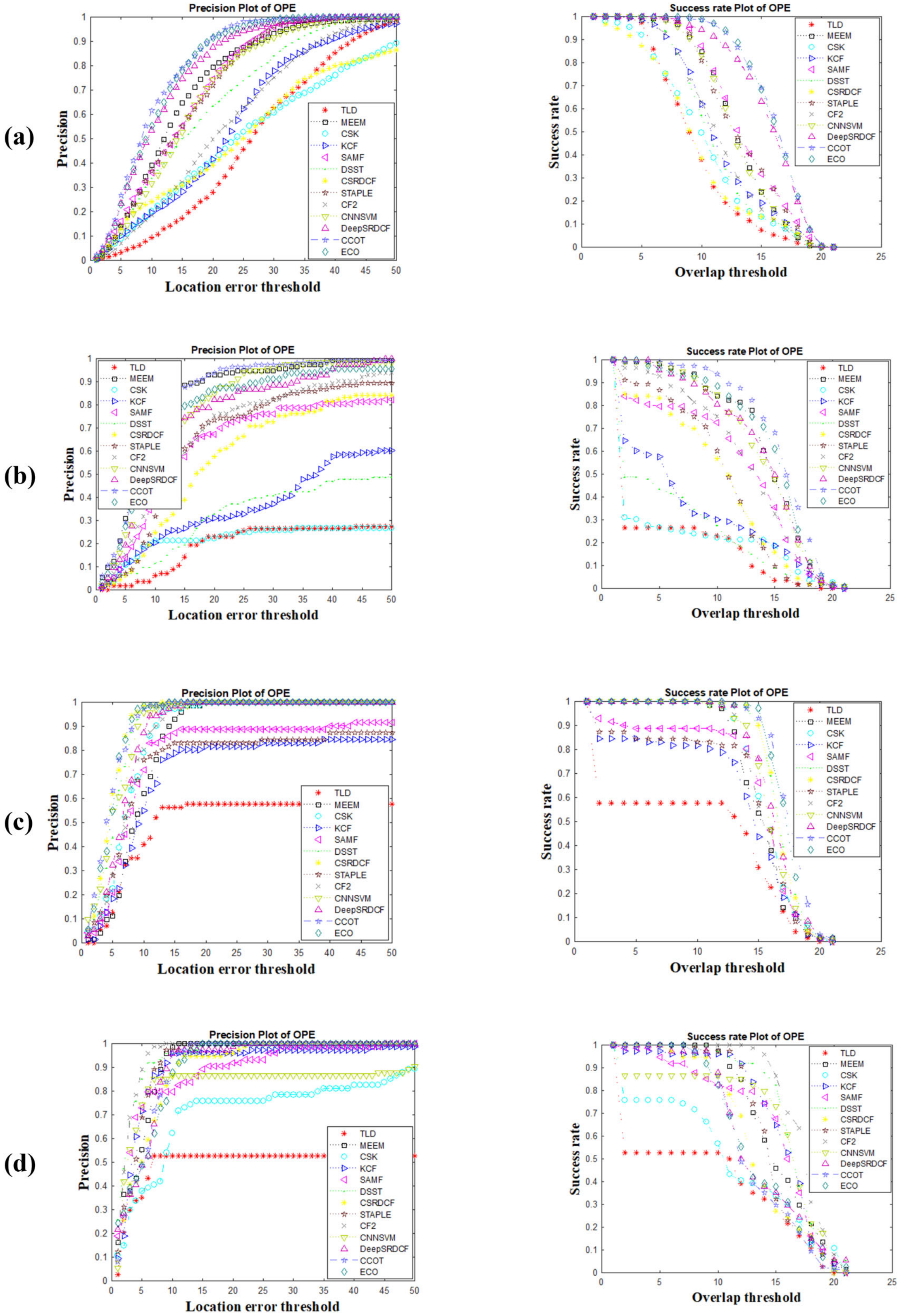


Fig. 3 Precision and success plots of compared trackers on the basis of the OPE manner over 13 video sequences for challenging attributes, **a** FM, **b** MB, **c** IV, and **d** BC

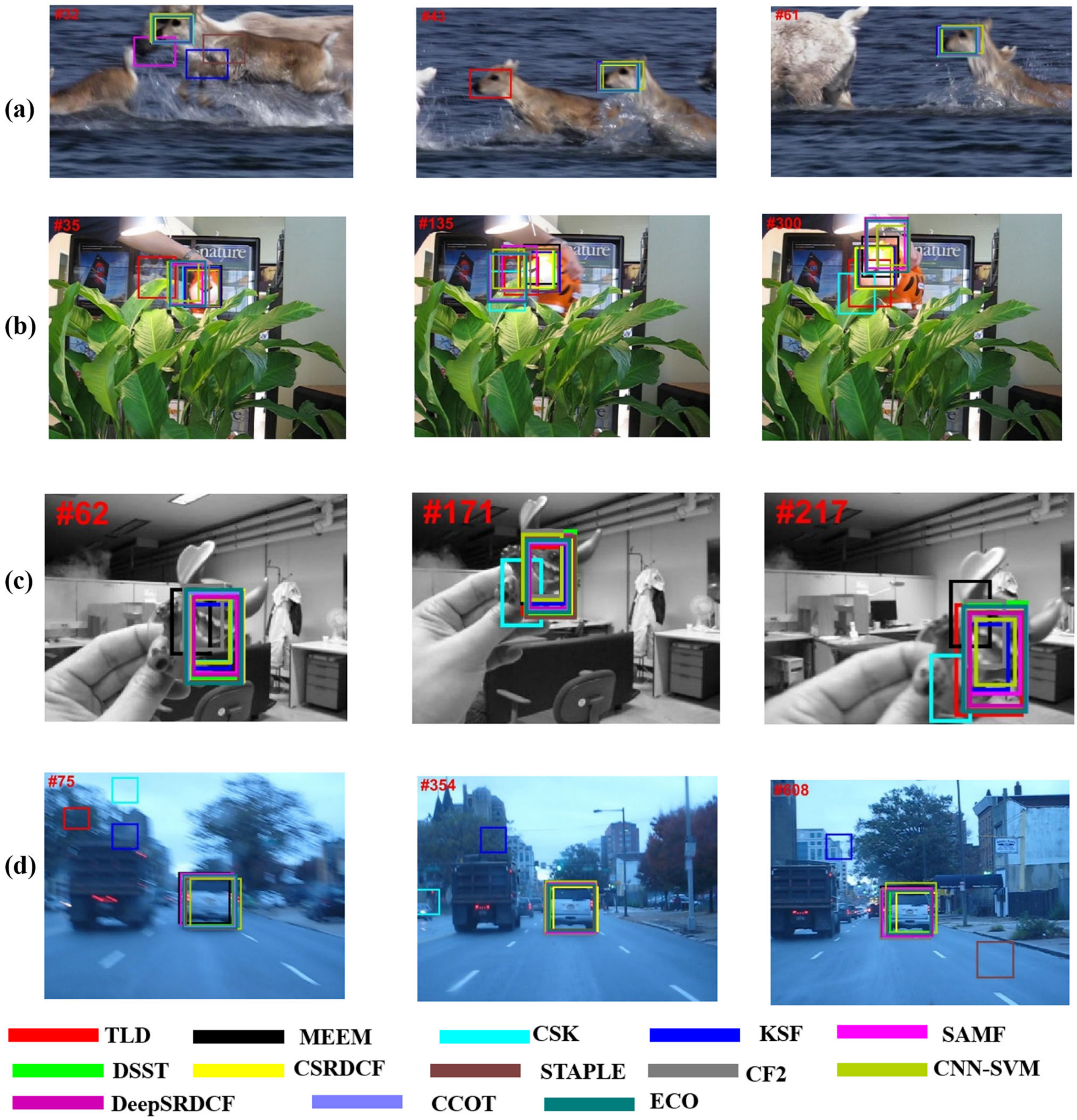


Fig. 4 Qualitative comparison of selected trackers on the **a** Deer, **b** Tiger1, **c** Toy, and **d** BlurCar1 sequences

The CNN-based tracking techniques are either generative or discriminative, similar to conventional trackering techniques. The generative techniques use a similarity metric to estimate object template matching within a certain search area. The discriminative techniques apply binary classification in the CNN scheme to effectively distinguish the object from its background. To utilize a CNN-based tracker, a convenient and simple approach is to substitute hand-crafted features with deep features, captured from the CNN using a popular tracking method, such as the CF.

4.3 CNN-based classification tracking

The spatial information in the last convolutional layer cannot accurately determine the object position. In contrast, earlier CNN layers define an accurate position, but have less robustness from the appearance model perspective. Therefore, Ma

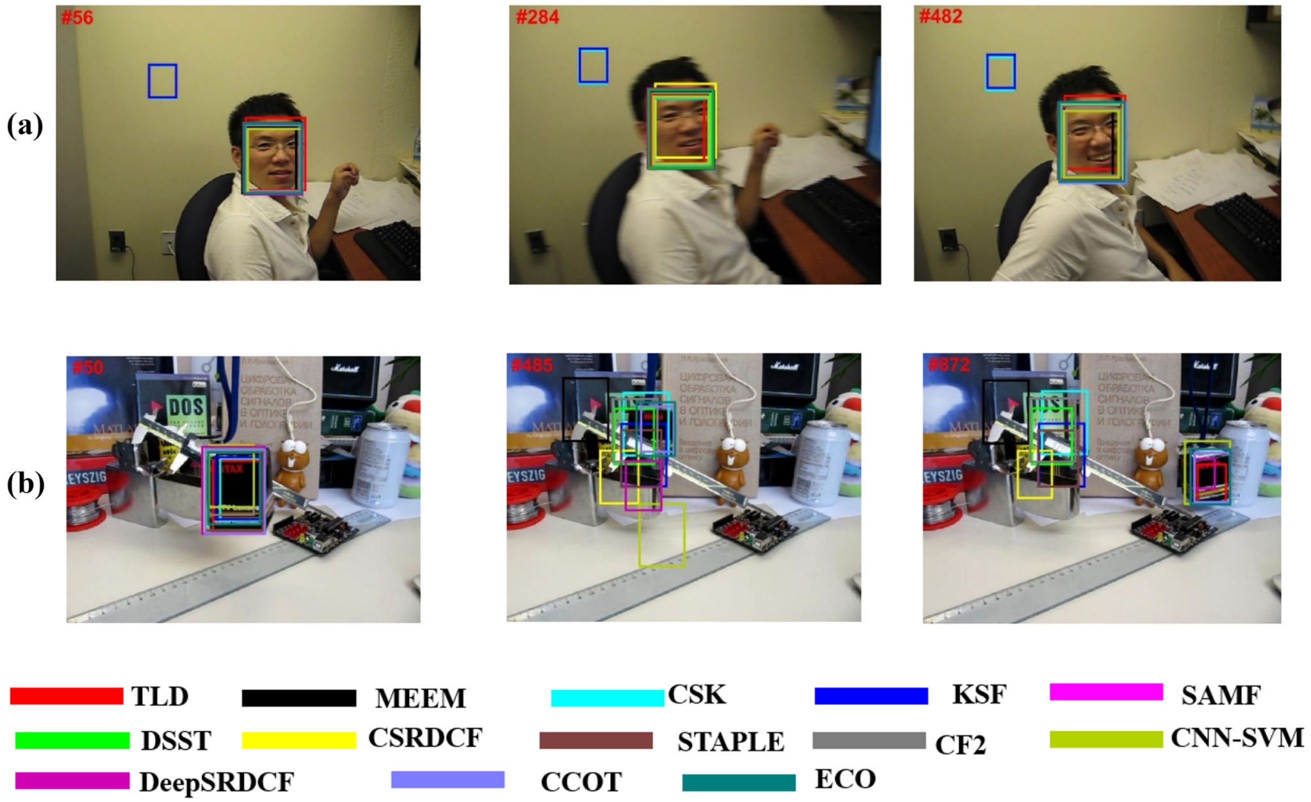


Fig. 5 Qualitative comparison of selected trackers on the **a** BlurFace, and **b** Box sequences

et al. [101] used a visual geometry group network (VGGNet) [102] to capture the features of the last convolutional layer. These features represent the semantic resolution that competes with coarse appearance changes. They introduced three different convolutional layers (Conv3–4, Conv4–4, and Conv5–4) with three CFs and then collected their associated output maps to derive the object.

Hong et al. [103] extracted discriminative saliency maps using the CNN, then embedded them into an online SVM to update and account for appearance variations. Moreover, the deep layers are implemented not only to capture features, but also to classify them. The deep spatially regularized discriminative CF (DeepSRDCF) [104] approach exploits the feature map from the CNN layer to the framework of SRDCF [105]. Zhu et al. [106] applied the CNN layers in a similar manner to a faster region CNN (FR-CNN) to create object patterns, which are combined into an online SVM to compute the object appearance. The CCOT [107] and the ECO [108] form a tracker based on a continuous convolution filter. In the CCOT, the tracker adopts features from three convolutional layers by applying a pre-trained VGGNet and updating a discriminative continuous convolution operator to increase robustness. The ECO incorporates deep features: along with handcrafted features HOG and CN, and the convolution operator is factorized to improve the number of parameters. The UPDT model [109] developed the fusion of shallow and deep features to fully exploit the benefits of CNNs.

4.4 CNN-based matching tracking

Recently, the Siamese neural network has gained attention in the area of object tracking. Many researchers have utilized the CNN architecture to learn robust matching methods. In one previous study [110], the developers of the GOTURN modified the Siamese neural network to carry out object tracking for pairs of consecutive frames and adopted a feed-forward network without online training through regression. The developers of SINT [111] introduced a Siamese neural network architecture to compute the similarity between an object pattern in the first frame and candidates in the next frame. This method performs visual tracking as a validation task, which determines the optimal conditions based on the maximum matching score. Bertinetto et al. [112] proposed a fully-connected Siamese model to correlate the object pattern with the recent search region in a CNN. Chen et al. [113] developed a generic framework using an efficient two-flow CNN model to combine two inputs, one for the object image patch and the other for the search region patch. The method estimates the appearance of the object within a certain area. Valmadre et al. [114] applied CFs to the different

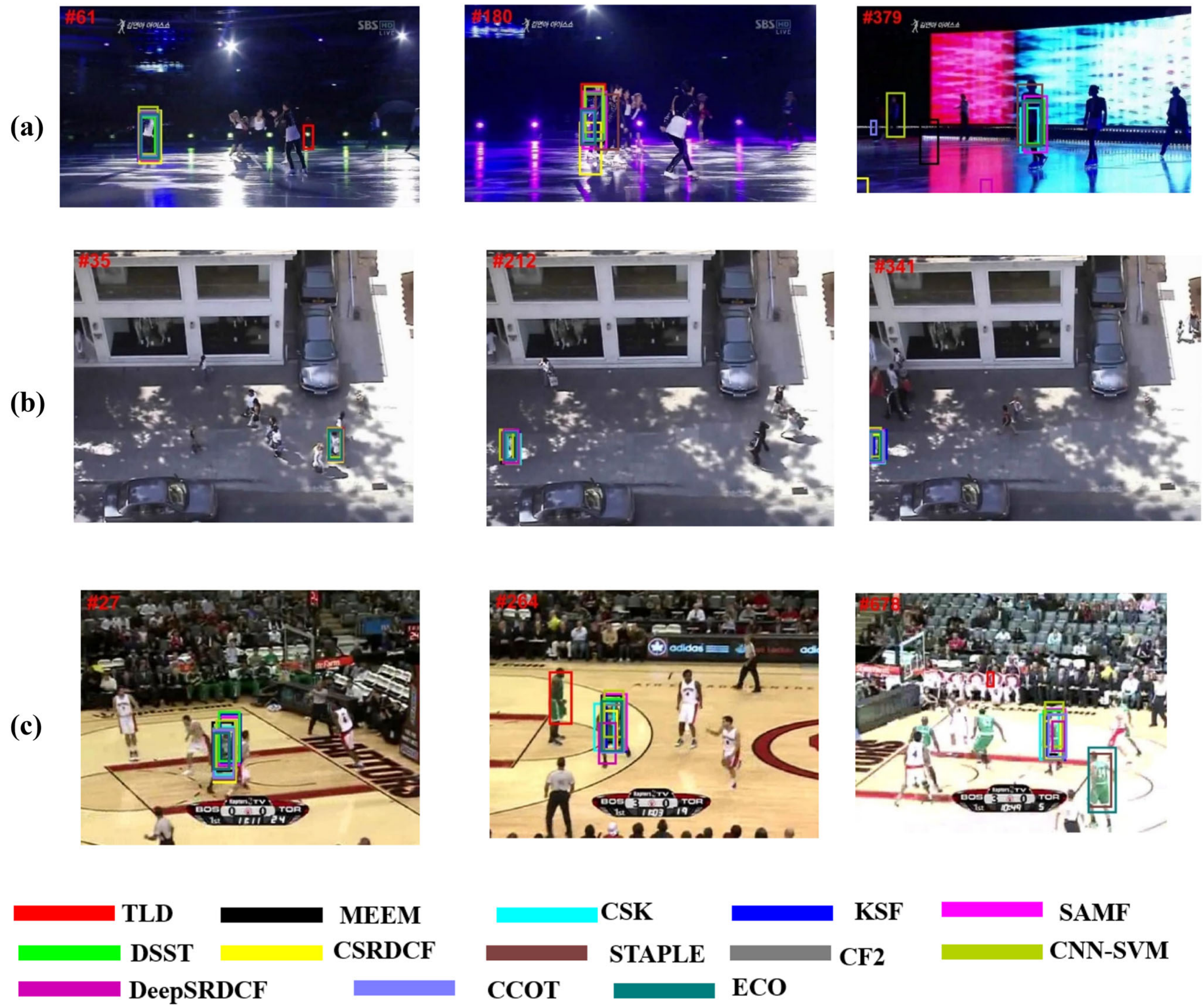


Fig. 6 Qualitative comparison of selected trackers on the **a** Skating1, **b** Crowds, and **c** Basketball sequences

output features of a layer within the Siamese architecture. The FlowTrack [115] introduces rich traffic information in successive frames to increase feature coding and tracking performance.

The developers of SiamFC [116] were the first to adopt CFs to the Siamese network. The developers of CFNet [117] enhanced SiamFC by applying a CF to the exemplar branch to learn the template representation, which makes the Siamese network more robust to appearance variations. Kuai et al. [118] used target object and target template models to improve the efficiency of SiamFC. The developers of Re3 [119] introduced a recurrent network to extract enhanced features created by exemplar branches. In DSiam [120], appearance changes in the target and background can be updated and learned from previous frames online. Dong and Shen [121] adopted a triplet loss operation to increase the robustness of SiamFC and CFNet. The developers of SiamRPN [122] used a Siamese region proposal network (RPN) to compute the bounding boxes of targets.

Deeper and wider Siamese networks [123] depend on a deeper and broader CNN to improve the efficiency of tracking. This method reduces the negative effects of padding, while managing perceptual domain size and network stride. The architecture of this design is very lightweight, and the output is enhanced while ensuring real-time performance. The developers of SiamMask [124] used a simple framework that is able to perform both tracking and segmentation simultaneously in real time. This method improves the full convolutional Siamese neural network by adding a mask branch for target tracking and segmentation.

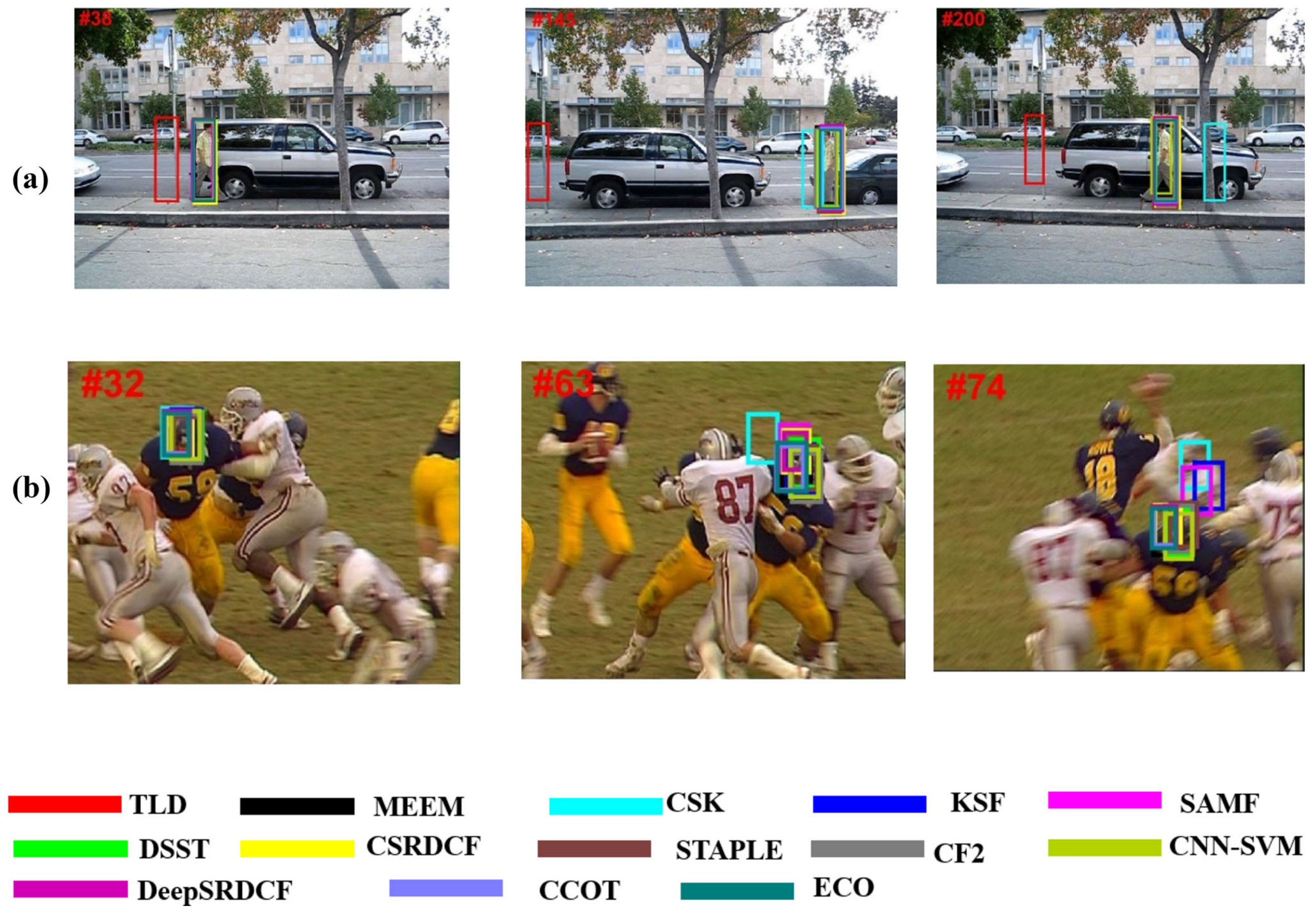


Fig. 7 Qualitative comparison of selected trackers on the **a** David3, and **b** Football1 sequences

5 Experiments and analysis

In this section, we evaluate popular benchmark OTB [125] and highlight advantages of using different visual trackers. We outline the trackers used, and then analyze, compare, and discuss the experimental output. Finally, we summarize our conclusions. All trackers were run in MATLAB on a Desktop PC with 2.9 GHz CPU and a GTX 1080 Ti GPU.

5.1 Tracking algorithm

We considered 13 visual trackers: TLD [63], MEEM [66], CSK [68], KCF [69], SAMF [72], DSST [73], CSRDCF [74], STAPLE [75], CF2 [101], CNN-SVM [103], DeepSRDCF [104], CCOT [107], and ECO [108], all of which have displayed good performance with popular benchmarks. We ran the source codes published by the authors and used tracking results for experimental comparisons. The trackers were compared in the presence of fast motion (FM), motion blur (MB), illumination variation (IV), background clutter (BC), and occlusion (Figs. 4, 5, 6, 7, 8). Table 1 displays these results quantitatively, with the results of different tracking algorithms listed for different sequences. Figure 3 presents the precision and success plots of the one-pass-evaluation (OPE) measurement.

Trackers using deep learning-based clearly outperformed the traditional trackers. The ECO, CNN-SVM, CF2, and CCOT trackers did much better than the others. Based on these results, we conclude that the utilization of deep-learning features substantially improves tracking over human-developed methods. This may be related to the CNN layers, which depend on parameters sharing local connectivity make the image feature extraction more useful.

Many deep-learning-based methods utilize convolutional features from a single layer, while others, such as trackers, utilize a combination of multiple convolutional layers. The deep-learning models for the trackers in this study were pre-trained prior to tracking, and were not updated during the tracking process. An important benefit of this process is that the deep-learning models do not require additional computation and memory space. Therefore, research into improving the performance of combining pre-training and online learning

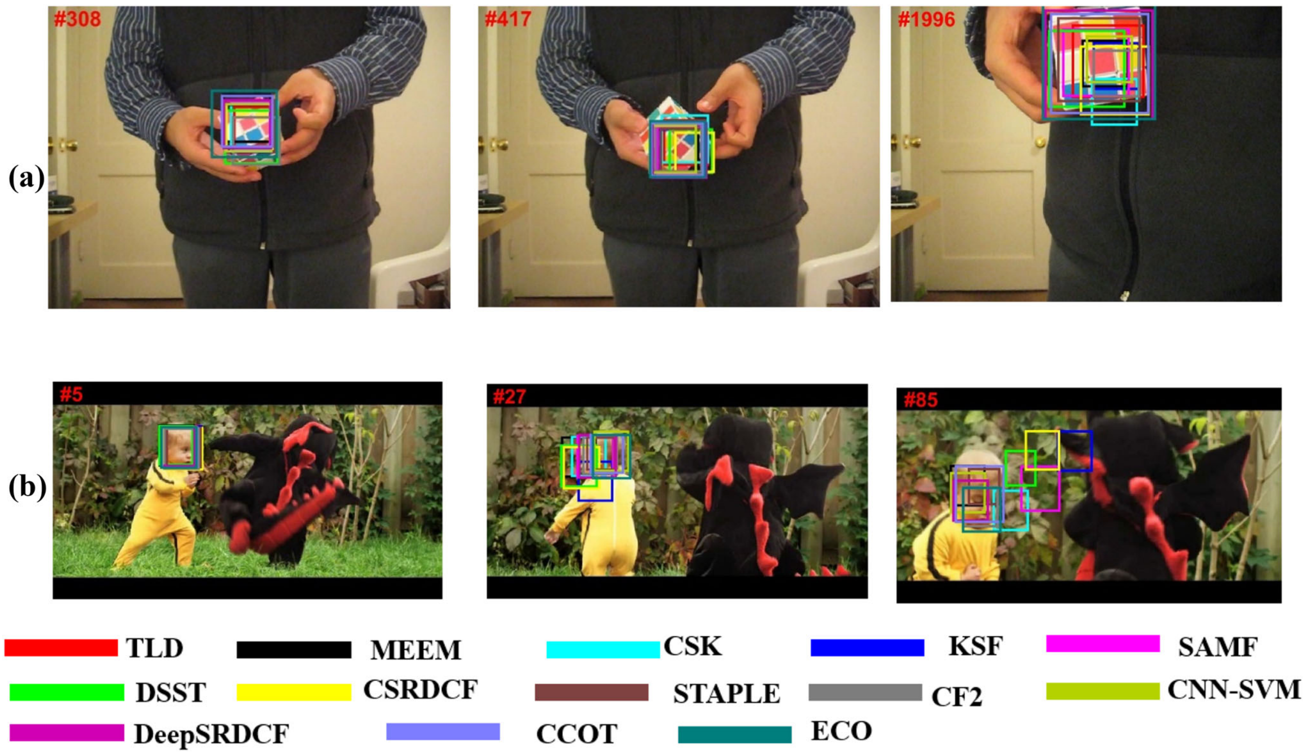


Fig. 8 Qualitative comparison of selected trackers on the **a** Rubik, and **b** DragonBaby sequences

within the deep-learning models could be extremely valuable.

5.2 Qualitative comparison on different attributes

Fast motion. Qualitative comparison results on the fast motion sequences: the challenging Deer, Tiger1, Toy, and BlurCar1 sequences, are presented in Fig. 4.

Motion Blur. Qualitative comparison results on the motion blur sequences: the challenging BlurFace, and Box sequences, are presented in Fig. 5.

Illumination variation. Qualitative comparison results on the illumination variation sequences: the challenging Skating1, Crowds, and Basketball sequences, are presented in Fig. 6.

Background clutter. Qualitative comparison results on the background clutter sequences: the challenging Skating1, Crowds, and Basketball sequences, are presented in Fig. 7.

Occlusion. Qualitative comparison results on the occlusion sequences: the Rubik, and DragonBaby sequences, are presented in Fig. 8.

5.3 Discussion and analysis

It is clear that the deep-learning trackers have much smaller average central errors than traditional trackers, for most of the sequences (Table 1), and the ECO tracker performed the best overall in terms of central error. As shown in Table 1, to test each tracker, we considered Deer, Tiger1, Toy, BlurCar1, BlurFace, Box, Skating1, Crowds, Basketball, David3, Football1, Rubik, and DragonBaby sequences, which have various challenges as outlined in the following.

DeepSRDCF enhances tracking outputs using convolutional features and spatial regularization penalties. However, it did not successfully tackle deformation (Basketball) or occlusion (DragonBaby). The CF2 utilizes different convolutional layers to train multiple CFs. However, it did not successfully handle an object with FM and in-plane rotation (BlurFace). The CCOT tracked objects in most of the selected video sequences

The objects in Deer (Fig. 4a) and Tiger1 (Fig. 4b) have abrupt motions, along with appearance changes caused by motion blur, which makes them difficult to track. Nonetheless, most trackers handled the Deer sequence with some drift, but the TLD tracker did not perform well. Due to the weak re-initialization mechanism in the TLD, it may detect a non-target object with a similar shape as the target (Deer 43). For the Tiger1 sequence, the deep trackers (ECO, CF2, and CNN-SVM) tracked well, whereas the other trackers (MEEM, CSK, KCF, SAMF, DSST, CSRDCF, and STAPLE) did not. This is attributed to the repetitive motion in the sequence, along with the fact that the latter trackers do not have a re-initialization mechanism, and hence they cannot locate a target after failure.

Table 2 Comparison of trackers speeds

Trackers	Speed (FPS)
TLD [63]	12
MEEM [66]	17
CSK [68]	151
KCF [69]	141
SAMF [72]	11
DSST [73]	20
CSRDCF [74]	15
STAPLE [75]	43
CF2 [101]	10
CNN-SVM [103]	9
DeepSRDCF [104]	5
CCOT [107]	8
ECO [108]	7

Figure 6 shows the results from three complicated sequences where illumination is variable. In the Skating1 sequence, a drastic lighting change occurs when the skater moves around the lights. As a result, MEEM, CSK, KCF, and SAMF suffered from severe drift at frames 180 and 379, while the CNN-SVM, DeepSRDCF, CCOT, and ECO trackers performed well.

5.4 Tracking speed analysis

Several parameters influence the computational speed of trackers, aside from different user platforms. These include the bounding box size of the target object, the number of features, the searching bounding box, the number of iterations, and the type of the classifier. As an example, classification trackers perform faster than matching trackers. Most existing deep-learning trackers adopt a CNN to model the variations in appearance. Some use a CNN to separate the object from its background, while others use it to match candidates with the object. Classification of positive features with negative ones is faster and simpler than matching two features. Therefore, CNN-based classification trackers have faster performance than CNN-based matching trackers, because most CNN-based trackers adopt the Siamese neural network to represent prior information, instead of fine-tuning online. In the MIL tracker, when the number of Haar features becomes larger, the frame rate is lower, but the robustness increases. The average speeds of all the 13 trackers are listed in Table 2.

6 Conclusion

In tracking, the main critical issue is appearance variations that prevent the tracker from localizing the object efficiently and correctly. Therefore, online learning techniques are being developed to combat sharp appearance variations during tracking. In this paper, the milestone visual object tracking methods based on online learning were discussed considering generative and discriminative methods. The main concepts and features of these frameworks have been presented at the beginning of Sect. 3. To summarize, generative methods consider only the object appearance without background details. In contrast, the discriminative methods compute a boundary region to differentiate the object from its surroundings by considering details on both the object and the background.

In the visual tracking community, the number of successfully-tracked objects and the error of the average position are used to quantitatively assess the tracking result. Generally, discriminative methods have better results than generative ones, if they have sufficient instances. However, if the number of samples embedded in the training step is small or inadequate, generative methods often have better performance than discriminative ones. Finally, the issue of finding an innovative framework that combines effectiveness and precision or adaptation and balance is still an open and vital issue in visual tracking. With the large improvement that CNNs have provided in recent years, impressive CNN architectures can be applied to perform the visual tracking tasks. It is hoped that this presentation of tracking frameworks, which are dependent on online updating, will provide a valuable orientation to new-comers and researchers in related domains.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) (No. NRF-2018R1D1A3B07044041), under the ITRC (Information Technology Research Center) support program supervised by the IITP (Institute for Information & communications Technology Promotion) (IITP-2020-2015-0-00448), and under Industrial Technology Innovation Program (No.20002655), grant funded by Korea Government.

References

1. Wang, X., Chen, D., Yang, T., Hu, B., Zhang, J.: Action recognition based on object tracking and dense trajectories. In: IEEE International Conference on Automatica (ICA-ACCA) (2016). https://doi.org/10.1109/ica-acca.2016.7778391
2. Foresti, G.L., Snidaro, L.: (2005) Vehicle detection and tracking for traffic monitoring. In: Roli, F., Vitulano, S. (eds) Image Analysis and Processing—ICIAP 2005. ICIAP 2005. Lecture Notes in Computer Science, vol. 3617. Springer, Berlin. https://doi.org/10.1007/11553595_147
3. Hui, Z., Yaohua, X., Lu M, Jiansheng, F.: Vision-based real-time traffic accident detection. In: 2014 11th World Congress on Intelligent Control and Automation (WCICA). https://doi.org/10.1109/wcica.2014.7052859
4. Kamijo, S., Matsushita, Y., Ikeuchi, K., Sakauchi, M.: Traffic monitoring and accident detection at intersections. IEEE Trans. Intell. Trans. Syst. **10**(1109/6979), 880968 (2000)
5. Sidla, O., Lypetskyy, Y., Brandle, N., Seer, S.: Pedestrian detection and tracking for counting applications in crowded situations. In: IEEE International Conference on Video and Signal Based

Surveillance. AVSS'06 (2006). https://doi.org/10.1109/AVSS.2006.91

6. Li, X., Zhao, H., Zhang, L.: Pedestrian counting system based on multiple object detection and tracking. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S. (eds) Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science, vol 10636. Springer, Cham, https://doi.org/10.1007/978-3-319-70090-8_9
7. Wang, Y., Doherty, J. E., Van Dyck, R. E.: Moving object tracking in video. In: Proceedings. 29th Applied Imagery Pattern Recognition Workshop (2000). https://doi.org/10.1109/aiprw.2000.953609
8. Kim, C., Hwang, J.-N.: Fast and automatic video object segmentation and tracking for content-based applications. IEEE Trans. Circuits Syst. Video Technol. (2002). https://doi.org/10.1109/76.988659
9. Lu, G., Shark, L. K., Hall, G.: Dynamic hand gesture tracking and recognition for real-time immersive virtual object manipulation. In: International Conference on CyberWorlds, 2009. CW'09 (2009). https://doi.org/10.1109/CW.2009.22
10. Boult, T.: Frame-rate multi-body tracking for surveillance. In: Proceedings of the DARPA Image Understanding Workshop, Monterey, CA, pp. 305–308 (1998)
11. Basu, A., Southwell, D.: Omni-directional sensors for pipe inspection. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3107–3112 (1995)
12. Kemeny, S. E., Panicacci, R., Pain, B., Matthies, L., Fossum, E. R.: Multi-resolution image sensor. In: IEEE Transactions on the Circuits System Video Technology, vol. 7, pp. 575–583 (1997)
13. Gress, O., Posch, S.: Trajectory retrieval from Monte Carlo data association samples for tracking in fluorescence microscopy images. In: 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI). IEEE, pp. 374–377 (2012)
14. Mian, A.S.: Real time visual tracking of aircrafts. Digital Image Comput Tech Appl (2008). https://doi.org/10.1109/dicta.2008.33
15. Li, P., Wang, D., Wang, L., Huchuan, L.: Deep visual tracking: review and experimental comparison. Pattern Recogn. **76**, 323–338 (2018)
16. Yan, C., Li, L., Zhang, C., Liu, B., Zhang, Y., Dai, Q.: Cross-modality bridging and knowledge transferring for image understanding. IEEE Trans Multimed **21**, 2675–2685 (2019)
17. Abbass, M.Y., Kwon, K., Kim, N. et al.: Efficient object tracking using hierarchical convolutional features model and correlation filters. Vis. Comput. (2020). https://doi.org/10.1007/s00371-020-01833-5
18. Hao, X., Zhang, Y., Dai, Q.: A fast uyghur text detector for complex background images. IEEE Trans Multimed **20**, 3389–3398 (2018)
19. Everingham, M., Gool, L.J.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The Pascal visual object classes VOC challenge. IJCV **88**(2), 303–338 (2010)
20. Nghiem, A. T., Bremond, F., Thonnat, M., Valentin, V.: Etiseo, performance evaluation for video surveillance systems. In: Proceedings of the AVSS, London, UK, pp. 476–481 (2007)
21. Kwon, J., Lee, K. M.: Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In: Proceedings of the IEEE CVPR, Miami, FL, USA (2009)
22. Kwon, J., Lee, K.: Tracking of abrupt motion using Wang Landau Monte Carlo estimation. In: Proceedings of the 10th ECCV, Marseille, France (2008)
23. Salti, S., Cavallaro, A., di Stefano, L.: Adaptive appearance modeling for video tracking: survey and evaluation. IEEE Trans. Image Process. **21**(10), 4334–4348 (2012)
24. Karasulu, B., Korukoglu, S.: A software for performance evaluation and comparison of people detection and tracking methods in video processing. MTA **55**(3), 677–723 (2011)
25. Maggio, E., Cavallaro, A.: Tracking by sampling trackers. In: Proceedings of the IEEE ICCV, Barcelona, Spain, pp. 1195–1202 (2011)
26. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2014)
27. Liu, Q., Zhao, X., Hou, Z.: Survey of single-target visual tracking methods based on online learning. IET Comput. Vis. **8**(5), 419–428 (2014)
28. Jepson, A. D., Fleet, D. J., El-Maraghi, T. F.: Robust online appearance models for visual tracking. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, pp. 415–422 (2001)
29. Zhou, S., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Process. **13**(11), 1491–1506 (2004)
30. Tu, J. L., Tao, H.: Online updating appearance generative mixture model for meanshift tracking. In: Proceedings of the Asian Conference Computer Vision (ACCV), Hyderabad, India, pp. 694–703 (2006)
31. Fussenegger, M., Roth, P., Bischof, H., Deriche, R., Pinz, A.: A level set framework using a new incremental, robust active shape model for object segmentation and tracking. Image Vis. Comput. **27**(8), 1157–1168 (2009)
32. Yang, H. X., Song, Z., Chen, R. N.: An incremental PCA-HOG descriptor for robust visual hand tracking. In: Proceedings of the International Symposium Visual Computing (ISVC), Las Vegas, Nevada, USA, pp. 687–695 (2010)
33. Chiverton, J., Xie, X.H.: Automatic bootstrapping and tracking of object contours. IEEE Trans. Image Process. **21**(3), 1231–1245 (2012)
34. Chiverton, J., Mirmehdi, M., Xie, X. H.: On-line learning of shape information for object segmentation and tracking. In: Proceedings of the British Machine Vision Conference (BMVC), London, UK, pp. 1–11 (2009)
35. Liu, X.B., Lin, L., Yan, S.C., Jin, H., Jiang, W.B.: Adaptive object tracking by learning hybrid template online. IEEE Trans. Circuits Syst. Video Technol. **21**(11), 1588–1599 (2011)
36. Xu, Y. L., Zhou, H. F., Wang, Q., Lin, L.: Real time object of interest tracking by learning composite patch-based templates. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, pp. 389–392 (2012)
37. Kwon, J., Lee, K. M.: Visual tracking decomposition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, pp. 1269–1276 (2010)
38. Kwon, J., Lee, K. M.: Tracking by sampling trackers. In: Proceedings of the IEEE Conference on Computer Vision (ICCV), Barcelona, Spanish, pp. 1195–1202 (2011)
39. Ross, D., Lim, J., Yang, M. H.: Adaptive probabilistic visual tracking with incremental subspace update. In: Proceedings of the European Conference on Computer Vision (ECCV), Prague, Czech Republic, pp. 470–482 (2004)
40. Lim, J., Ross, D., Lin, R.S., Yang, M.H.: Incremental learning for visual tracking. In: Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, pp. 793–800. MTI Press, Boca Raton (2005)
41. Lee, K., Kriegman, D.: Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, USA, pp. 852–859 (2005)
42. Li, X., Hu, W. M., Zhang, Z. F.: Robust visual tracking based on incremental tensor subspace learning. In: Proceedings of the IEEE Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, October 2007, pp. 1–8

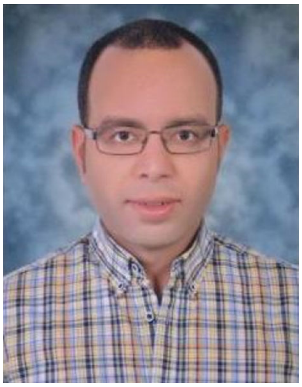
43. Wen, J., Gao, X.: Incremental learning of weighted tensor subspace for visual tracking. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), San Antonio, TX, USA, pp. 3688–3693 (2009)
44. Li, X., Hu, W., Zhang, Z., Zhang, X., Luo, G.: Visual tracking via incremental log-Euclidean Riemannian subspace learning. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, USA, pp. 1–8 (2008)
45. Wu, Y., Cheng, J., Wang, J., Lu, H.: Real-time visual tracking via incremental covariance tensor learning. In: Proceedings of the IEEE Conference on Computer Vision (ICCV), Kyoto, Japan, pp. 1631–1638 (2009)
46. Lu, K., Ding, Z.M., Ge, S.: Locally connected graph for visual tracking. Neurocomputing **120**, 45–53 (2013)
47. Matthews, L., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 810–815 (2004)
48. Mei, X., Ling, H.B.: Robust visual tracking and vehicle classification via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2259–2272 (2011)
49. Liu, B., Yang, L., Huang, J., Meer, P., Gong, L., Kulikowski, C. A.: Robust and fast collaborative tracking with two stage sparse optimization. In: Proceedings of the European Conference on Computer Vision (ECCV), Grete, Greece, pp. 624–637 (2010)
50. Liu, R., Huang, J. Z., Yang, L., Kulikowsk, C. A.: Robust tracking using local sparse appearance model and K-selection. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, pp. 1313–1320 (2011)
51. Chen, F., Wang, Q., Wang, S., Zhang, W.D., Xu, W.L.: Object tracking via appearance modeling and sparse representation. Int. J. Image Vis. Comput. **29**, 787–796 (2011)
52. Jia, X., Lu, H., Yang, M. H.: Visual tracking via adaptive structural local sparse appearance model. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, USA, pp. 1822–1829 (2012)
53. Lu, X.Q., Yuan, Y., Yan, P.K.: Robust visual tracking with discriminative sparse learning. Pattern Recogn. **46**(7), 1762–1771 (2013)
54. Stern, H., Efros, B.: Adaptive color space switching for face tracking in multi-colored lighting environments. In: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, pp. 236–241 (2002)
55. Collins, R.T., Liu, Y.X., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1631–1643 (2004)
56. Nguyen, H. T., Smeulders, A.: Tracking aspects of the foreground against the background. In: Proceedings of the European Conference on Computer Vision (ECCV), Prague, Czech Republic, pp. 446–456 (2004)
57. Wang, J., Chen, X., Gao, W.: Online selecting discriminative tracking features using particle filter. In: Proceedings of the IEEE Conference Vision and Pattern Recognition (CVPR), San Diego, CA, USA, pp. 1037–1042 (2005)
58. Li, G., Liang, D., Huang, Q., Jiang, S. Q., Gao, W.: Object tracking using incremental 2D-LDA learning and Bayes inference. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), San Diego, California, USA, pp. 1568–1571 (2008)
59. Avidan, S.: Ensemble tracking. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, pp. 494–501 (2005)
60. Leistner, C., Granber, H., Bischof, H.: Semi-supervised boosting using visual similarity learning. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska, USA, pp. 1–8 (2008)
61. Babenko, B., Yang, M. H., Belongie, S.: Visual tracking with online multiple instance learning. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Miami, Florida, USA, pp. 983–990 (2009)
62. Li, W., Duan, L.X., Tsang, I.W., Xu, D.: Batch mode adaptive multiple instance learning for computer vision tasks. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, USA, pp. 2368–2375 (2012)
63. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
64. Kalal, Z., Matas, J., Mikolajczyk, K.: Online learning of robust object detectors during unstable tracking. In: Proceedings of the IEEE Conference on Computer Vision Workshop (ICCV Workshop), Kyoto, Japan, pp. 1417–1424 (2009)
65. Hare, S., Saffari, A., Torr, P. H. S.: Struck: structured output tracking with kernels. In: Proceedings of the ICCV, Barcelona, Spain, pp. 263–270 (2011)
66. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Proceedings of the ECCV, Zürich, Switzerland, pp. 188–203 (2014)
67. Bolme, D. S., Beveridge, J. R., Draper, B. A., Lui, Y. M.: Visual object tracking using adaptive correlation filters. In: Proceedings of the CVPR, San Francisco, CA, USA, pp. 2544–2550 (2010)
68. Henriques,J. F., Rui, C., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Proceedings of the ECCV, Firenze, Italy, pp. 702–715 (2012)
69. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
70. Danelljan, M., Khan, F. S., Felsberg, M., van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Proceedings of the CVPR, Columbus, OH, USA, pp. 1090–1097 (2014)
71. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: Proceedings of the British Machine Vision Conference (BMVC), Nottingham, UK, pp. 1–11 (2014)
72. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Proceedings of the ECCV Workshop, pp. 254–265 (2014)
73. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1561–1575 (2017)
74. Lukei, A., Voji, T., Zajc, L.C., Matas, J., Kristan, M.: Discriminative correlation filter tracker with channel and spatial reliability. Int. J. Comput. Vis. **126**, 671–688 (2018)
75. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P. H. S.: Staple: complementary learners for real-time tracking. In: Proceedings of the CVPR, Las Vegas, NV, USA, pp. 1401–1409 (2016)
76. Lin, R.S., Ross, D., Lim, J., Yang, M.H.: Adaptive discriminative generative model and its applications. Adv. Neural. Inf. Process. Syst. **17**, 801–808 (2004)
77. Zhang, X. Q., Hu, W. M., Maybank, S., Li, X.: Graph based discriminative learning for robust and efficient object tracking. In: Proceedings of the IEEE Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, pp. 1–8 (2007)
78. Yu, Q., Dinh, T. B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, pp. 678–691 (2008)
79. Yin, Z., Collins, R. T.: Shape constrained figure-ground segmentation and tracking. In: Proceedings of the IEEE Conference Com-

puter Vision and Pattern Recognition (CVPR), Miami, Florida, USA, pp. 731–738 (2009)

80. Yang, M., Wu, Y., Lao, S.: Intelligent collaborative tracking by mining auxiliary objects. In: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, pp. 697–704 (2006)
81. Le Cun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**, 436–444 (2015)
82. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 142–158 (2016)
83. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
84. Kim, S., Hori, T., Watanabe, S.: Joint ctc-attention based end-to-end speech recognition using multi-task learning. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pp. 4835–4839 (2017)
85. Wu, Z., Valentini-Botinhao, C., Watts, O., King, S.: Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pp. 4460–4464 (2015)
86. Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., Hinton, G. E.: Grammar as a foreign language. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 2773–2781 (2015)
87. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. Clinical Orthopaedics and Related Research. https://arxiv.org/abs/1409.0473
88. Bora, K., Chowdhury, M., Mahanta, L. B., Kundu, M. K., Das, A. K.: Pap smear image classification using convolutional neural network. In: Tenth Indian Conference on Computer Vision, Graphics and Image Processing, p. 55 (2016)
89. Han, X.-H., Lei, J., Chen, Y.-W.: HEp-2 Cell Classification Using k-Support Spatial Pooling in Deep CNNs. Deep Learning and Data Labeling for Medical Applications, pp. 3–11. Springer, Berlin (2016)
90. Hinton, G., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
91. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. J. Biol. Cybern. **36**(4), 193–202 (1980). https://doi.org/10.1007/bf00344251
92. Ramírez-Quintana, J.A., Chacon-Murguia, M.I., Chacon-Hinojos, J.F.: Artificial neural image processing applications: a survey. Eng Lett **20**(1), 68–80 (2012)
93. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. J. Neural Comput. **1**(4), 541–551 (1989). https://doi.org/10.1162/neco.1989.1.4.541
94. Padmanabhan, J., Premkumar, M.J.J.: Machine learning in automatic speech recognition: a survey. IETE Tech. Rev. **32**(4), 240–251 (2015). https://doi.org/10.1080/02564602.2015.1010611
95. Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proceedings part I of the 13th European conference computer vision (ECCV'14), Zurich, Switzerland, pp. 818–833 (2014). https://doi.org/10.1007/978-3-319-10590-153
96. Wang, L., Sng, D.: Deep learning algorithms with applications to video analytics for a smart city: a survey. In: CoRR, https://arxiv.org/abs/1512.03131 (2015)
97. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
98. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'06). MIT Press, Canada, pp 153–160 (2006)
99. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649 (2013). https://doi.org/10.1109/icassp.2013.6638947
100. Abbas, Q., Ibrahim, M. E. A., Jaffar, M. A.: Artif. Intell. Rev. (2018). https://doi.org/10.1007/s10462-018-9633-3
101. Ma, C., Huang, J., Yang, X., Yang, M.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082 (2015)
102. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Clin. Orthop. Rel. Res. (2014). https://arxiv.org/abs/1409.1556
103. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: Proceedings of the International Conference on Machine Learning, pp. 597–606 (2015)
104. Danelljan, M., Häger, G., Khan, F. S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)
105. Galoogahi, H. K., Sim, T., Lucey, S.: Multi-channel correlation filters. In: ICCV, pp. 7–25 (2013)
106. Zhu, G., Porikli, F., Li, H.: Robust visual tracking with deep convolutional neural network based object proposals on pets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1265–1272 (2016)
107. Danelljan, M., Robinson, A., Khan, F. S., Felsberg, M.: Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. Springer, Cham (2016)
108. Danelljan, M., Bhat, G., Khan, F. S., Felsberg, M.: ECO: efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on the Computer Vision Pattern Recognition (CVPR), pp. 6931–6939 (2017)
109. Bhat, G., Johnander, J., Danelljan, M., Khan, F. S., and Felsberg, M.: Unveiling the power of deep tracking. In: Proceedings of the European Conference on the Computer Vision (ECCV), Munich, Germany, pp. 483–498 (2018)
110. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks. In: Proceedings of the European Conference on the Computer Vision (ECCV), Amsterdam, The Netherlands, pp. 749–765 (2016)
111. Tao, R., Gavves, E., Smeulders, A. W. M.: Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1420–1429 (2016)
112. Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., Torr, P. H. S.: Fully-convolutional siamese networks for object tracking. In: Proceedings of the European Conference on Computer Vision Workshops, pp. 850–865 (2016)
113. Chen, K., Tao, W.: Once for all: a two-flow convolutional neural network for visual tracking. Clin. Orthop. Rel. Res. (2016). https://arxiv.org/abs/1604.07507
114. Yan, C., Tu, Y., Wang, X., Zhang, Y., Hao, X., Zhang, Y., Dai, Q.: STAT: spatial-temporal attention mechanism for video captioning. IEEE Trans. Multimed. **22**, 830–830 (2019)
115. Zhu, Z., Wu, W., Zou, W., Yan, J.: End-to-end_ow correlation tracking with spatial-temporal attention. In: Proceedings of the IEEE Conference on the Computer Vision Pattern Recognition (CVPR), pp. 548–557 (2018)

116. Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., Torr, P. H. S.: Fully-convolutional siamese networks for object tracking. In: Proceedings of the European Conference on the Computer Vision (ECCV), Amsterdam, The Netherlands, pp. 850–865 (2016)
117. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P. H. S.: End-to-end representation learning for correlation filter based tracking. In: Proceedings of the European Conference on the Computer Vision Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 5000–5008 (2017)
118. Kuai, Y., Wen, G., Li, D.: Masked and dynamic siamese network for robust visual tracking. Inf. Sci. **503**, 169–182 (2019). https://doi.org/10.1016/j.ins.2019.07.004
119. Gordon, D., Farhadi, A., Fox, D.: Re3: real-time recurrent regression networks for visual tracking of generic objects, https://arxiv.org/abs/1705.06368 (2017)
120. Guo, Q., Wei, F., Zhou, C., Rui, H., Liang, W., Song, W.: Learning dynamic siamese network for visual object tracking. In: Proceedings of the IEEE Conference on Computer Vision (ICCV), Venice, Italy, vol. 1, pp. 1781–1789 (2017)
121. Dong, X., Shen, J.: Triplet loss in siamese network for object tracking. In: Proceedings of the European Conference Computer Vision (ECCV), Munich, Germany, pp. 472–488 (2018)
122. 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 Pattern Recognition (CVPR), Salt Lake City, UT, USA, pp. 8971–8980 (2018)
123. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision Pattern Recognition (2019)
124. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P. H. S.: Fast online object tracking and segmentation: a unifying approach. In: Proceedings of the IEEE International Conference on the Computer Vision Pattern Recognition, pp. 1328–1338 (2019)
125. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Mohammed Y. Abbass received the B.S. and M.Sc. degrees in Electronics and Electrical Communications Engineering. He is currently pursuing the Ph.D. degree in Electronics and Communications Engineering. His research interests include computer vision, video analysis, machine learning and deep learning in the applications of super-resolution, object tracking and detection in video.

Ki-Chul Kwon received his Ph.D. degree in Information and Communication Engineering from Chungbuk National University in 2005. Since 2014, he has been a research professor in the School of Electrical Engineering and Computer Science, Chungbuk National University. His current research area of interest is eye surgery using a microscope three-dimensional visualization system, medical image processing, and computer vision.

Nam Kim received his Ph.D. degree in Electronic Engineering from Yonsei University, Seoul, Korea, in 1988. Since 1989, he has been a professor in the School of Information and Communication Engineering, Chungbuk National University. From 1992 to 1993, he spent 1 year as a visiting researcher in Dr. Goodman's group at Stanford University. In addition, he attended Caltech as a visiting professor from 2000 to 2001. Currently, he is also a director of IT Research Center on Hologram Convergence Technology in Chungbuk National University. He is interested in the 3D display and visualization systems, 3D medical imaging systems, 3D image processing and applications based on stereoscopic, holography and integral imaging techniques, diffractive optics, and optical security systems.

Safey A. Abdelwahab received his B.Sc. in Electronics and Communications from the Faculty of Engineering, Cairo University in 1992. He received his M.Sc. and Ph.D. in Systems and Computers Engineering from the Faculty of Engineering, Al-Azhar University in 1998, and 2003, respectively. His research interests include digital image and digital signal processing, fuzzy logic, design of microcontroller-based instruments, design of radiation measurement instruments, software programming for interfacing and data acquisition, embedded systems, developing ICT-based materials, and design of FPGA-based instrument.

Fathi E. Abd El-Samie received the B.Sc. (Honors), M.Sc., and Ph.D. from the Faculty of Electronic Engineering, Menoufia University, Menouf, Egypt, in 1998, 2001, and 2005, respectively. He joined the teaching staff of the Department of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menoufia University, Menouf, Egypt, in 2005. He has received the most cited paper award from Digital Signal Processing journal for 2008. His current research areas of interest include image enhancement, image restoration, image interpolation, super-resolution reconstruction of images, data hiding, multimedia communications, medical image processing, optical signal processing, and digital communications.

Ashraf A. M. Khalaf (Ph.D.) received his B.Sc. and M.Sc. degrees in electrical engineering from Minia University, Egypt, in 1989 and 1994, respectively. He received the degree "Doctor of Engineering in System Science and Engineering" from Graduate School of Natural Science and Technology, Kanazawa University, Japan, in March 22, 2000 - PhD degree in Egypt. He is currently an associate professor at Electronics and Communications Engineering Department, Faculty of Engineering, Minia University, Egypt. His current research areas of interest include Adaptive systems, filtering, signal and image processing, neural networks, deep learning, biomedical signal processing, and optical communications.