Deep Learning for Visual Tracking: A Comprehensive Survey

Seyed Mojtaba Marvasti-Zadeh, Student Member, IEEE, Li Cheng, Senior Member, IEEE, Hossein Ghanei-Yakhdan, and Shohreh Kasaei, Senior Member, IEEE

Abstract—Visual target tracking is one of the most sought-after yet challenging research topics in computer vision. Given the ill-posed nature of the problem and its popularity in a broad range of real-world scenarios, a number of large-scale benchmark datasets have been established, on which considerable methods have been developed and demonstrated with significant progress in recent years—predominantly by recent deep learning (DL)-based methods. This survey aims to systematically investigate the current DL-based visual tracking methods, benchmark datasets, and evaluation metrics. It also extensively evaluates and analyzes the leading visual tracking methods. First, the fundamental characteristics, primary motivations, and contributions of DL-based methods are summarized from six key aspects of: network architecture, network exploitation, network training for visual tracking, network objective, network output, and the exploitation of correlation filter advantages. Second, popular visual tracking benchmarks and their respective properties are compared, and their evaluation metrics are summarized. Third, the state-of-the-art DL-based methods are comprehensively examined on a set of well-established benchmarks of OTB2013, OTB2015, VOT2018, and LaSOT. Finally, by conducting critical analyses of these state-of-the-art methods both quantitatively and qualitatively, their pros and cons under various common scenarios are investigated. It may serve as a gentle user guide for practitioners to weigh on when and under what conditions to choose which method(s). It also facilitates a discussion on ongoing issues and sheds light on promising research directions.

Index Terms—Visual tracking, deep learning, computer vision, appearance modeling.

1 INTRODUCTION

G enerically visual tracking aims to estimate the trajectory of an unknown visual target when only an initial state of the target (in a video frame) is available. Visual tracking is an open and attractive research field (see Fig. 1) with a broad extent of categories and applications; including self-driving cars [1]–[4], autonomous robots [5], [6], surveillance [7]–[10], augmented reality [11]–[13], unmanned aerial vehicle (UAV) tracking [14], sports [15], surgery [16], biology [17]–[19], ocean exploration [20], to name a few. The ill-posed definition of the visual tracking (i.e., model-free tracking, on-the-fly learning, single-camera, 2D information) is more challenging in complicated real-world scenarios which may include arbitrary classes of target appearance and their motion model (e.g., human, drone, animal, vehicle), different imaging characteristics (e.g., static/moving camera, smooth/abrupt movement, camera resolution), and changes in environmental conditions (e.g., illumination variation, background clutter, crowded scenes). Although traditional visual tracking methods utilize various frameworks—like discriminative correlation filters (DCF) [21]–[24], silhouette tracking [25], [26], Kernel tracking [27]–[29], point tracking [30], [31], and so forth—these methods cannot provide satisfactory results in unconstrained environments. The main reasons are the target representation by handcrafted features (such as the histogram of oriented gradients (HOG) [32] and Color-Names (CN)) [33] and inflexible target modeling. Inspired by deep learning (DL) breakthroughs [34]–[38] in ImageNet large scale visual recognition competition (ILSVRC) [39] and also visual object tracking (VOT) challenge [40]–[46], DL-based methods have attracted considerable interest in visual tracking community to provide robust visual trackers. Although convolutional neural networks

![Fig. 1: An overview of visual target tracking.](image-url)
(CNNs) have been dominant networks initially, the broad range of architectures such as Siamese neural networks (SNNs), recurrent neural networks (RNNs), auto-encoders (AEs), generative adversarial networks (GANs), and custom neural networks are currently investigated. Fig. 2 presents a brief history of the development of deep visual trackers in recent years. The state-of-the-art DL-based visual trackers have distinct characteristics such as exploitation of deep architecture, backbone network, learning procedure, training datasets, network objective, network output, types of exploited deep features, CPU/GPU implementation, programming language and framework, speed, and so forth. Besides, several visual tracking benchmark datasets have been proposed in the past few years for practical training and evaluating of visual tracking methods. Despite various properties, some of these benchmark datasets have common video sequences. Thus, a comparative study of DL-based methods, their benchmark datasets, and evaluation metrics are provided in this paper to facilitate developing advanced methods by the visual tracking community.

The visual tracking methods can be roughly classified into two main categories of before and after the revolution of DL in computer vision. The first category of visual tracking survey papers [47]–[50] mainly review traditional methods based on classical object and motion representations, and then examine their pros and cons systematically, experimentally, or both. Considering the significant progress of DL-based visual trackers, the reviewed methods by these papers are outdated. On the other hand, the second category reviews limited deep visual trackers [51]–[53]. The papers [51], [52] (two versions of a paper) categorize 81 and 93 handcrafted and deep visual trackers into the correlation filter trackers and non-correlation filter trackers, and then a further classification based on architectures and tracking mechanisms has applied. These papers study <40 DL-based methods with limited investigations. Although the paper [54] particularly investigates the network branches, layers, and training aspects of nine SNN-based methods, it does not include state-of-the-art SNN-based trackers (e.g., [55]–[57]) and the custom networks (e.g., [58]) which exploit SNNs, partially. The last review paper [55] has categorized the 43 DL-based methods according to their structure, function, and training. Then, 16 DL-based visual trackers are evaluated with different hand-crafted-based visual tracking methods. From the structure perspective, these trackers are categorized into 34 CNN-based (including ten CNN-Matching and 24 CNN-Classification), five RNN-based, and four other architecture-based methods (e.g., AE). Besides, from the network function perspective, these methods are categorized into the feature extraction network (FEN) or end-to-end network (EEN). While the FENs are the methods that exploit pre-trained models on different tasks, the EENs are classified in terms of their outputs; namely, object score, confidence map, and bounding box (BB). From the network training perspective, these methods are categorized into the NP-OL, IP-NOL, IP-OL, VP-OL, and VP-NOL categories, in which the NP, IP, VP, OL, and NOL are the abbreviations of no pre-trained, image pre-trained, video-pre-trained, online learning, and no online learning, respectively.

Despite all efforts, there is no comprehensive study to not only extensively categorize DL-based trackers, their motivations, and solutions to different problems, but also experimentally analyze the best methods according to different challenging scenarios. Motivated by these concerns, the main goal of this survey is to fill this gap and investigate the main present problems and future directions. The main differences of this survey and prior ones are described as follows.

**Differences to Prior Surveys:** Despite the currently available review papers, this paper focuses merely on 129 state-of-the-art DL-based visual tracking methods, which have been published in major image processing and computer vision conferences and journals. These methods include the HCF [59], DeepSRDCF [60], FCNT [61], CNN-SVM [62], DPST [63], CCOT [64], GOTURN [65], SiamFC [66], SINT [67], MDNet [68], HDT [69], STCT [70], RPNT [71], DeepTrack [72], CNT [73], CF-CNN [74], TCNN [75], RDLT [76], PTV [77], CREST [78], UCT/UCT-Lite [79], DSiam/DSiameseM [80], TN [81], WECO [82], ECO [83], RFL [84], IBCCF [85], DTO [86], SRT [87], F-RFCN [88], GNET [89], LST [90], VRCPF [91], DCDF [92], CFNet [93], ECO [94], DeepCSRDCF [95], McAF [96], BranchOut [97], DeepLMCF [98], Obli-RaFT [99], ACFN [100], SANet [101], DCNet/DCFNet2 [102], DET [103], DRN [104], DNT [105], STSGS [106], TripleLet [107], DSSL [108], UDPT [109], ACT [110], DaSiamRPN [111], RT-MDNet [112], StructSiam [113], MMLT [114], CPT [115], STP [116], Siam-MCF [117], Siam-BM [118], WAEF [119], TRAC [120], VITAL [121], DeepSTRCF [122], SiamRPN [123], SA-Siam [124], FlowTrack [125], DRT [126], LSART [127], RASN [128], MCC [129], DCPF [130], VDSR-SRT [131], FCFS [132], FRPN2T-Siam [133], FMFT [134], IMLCF [135], TGGAN [136], DAT [137], DCTN [138], FRPRNet [139], HCFIs [140], adaDDCF [141], YCNN [142], DeepFFP [143], CFCF [144], CSFRL [145], R2T [146], DDCF [147], HCFNet [148], LCTdeep [149], HSTC [150], DeepFWDCF [151], CF-CF-Siam [152], MGNet [153], ORHF [154], ASRCF [155], ATOM [156], C-RPN [157], GCT [158], RPCF [159], SPM [160], SiamDW [161], SiameseMask [162], SiamRPN++ [163], TADT [164], UDPT [165], DiMP [166], ADT [167], CODA [168], DRLR [169], SMART [170], MRCNN [171], MM [172], MTHCF [173], AEPFC [174], IMM-DFT [175], TAAT [176], DeepTACF [177], MAM [178], ADNet [179], C2F [180], DRL-IS [181], DRL [182], EAST [183], HP [184], P-Track [185], R-Track [186], and SINT++ [187].

The trackers include 73 CNN-based, 35 SNN-based, 15 custom-based (including AE-based, reinforcement learning (RL)-based, and combined networks), three RNN-based, and three GAN-based methods. One major contribution and
n Alrever, the recent visual trackers based on GAN and custom networks (which includes RL-based methods) are reviewed. Although the methods in this survey are categorized into the exploitation of off-the-shelf deep features and deep features for visual tracking (similar to the FENs and EENs in [45]), detailed characteristics of these methods such as pre-trained or backbone networks, exploited layer(s), training datasets, objective function, tracking speed, used features, types of tracking output, CPU/GPU implementation, programming language, DL framework are also presented. From the network training perspective, this survey independently studies deep off-the-shelf features and deep features for visual tracking. Because deep off-the-shelf features (i.e., extracted from FENs) are mostly pre-trained on the ImageNet for object recognition tasks, their training details are reviewed, independently. Hence, the network training for visual tracking purposes is categorized to the DL-based methods that exploit only offline training, only online training, or both offline and online training procedures. Finally, this paper comprehensively analyses different aspects of 45 state-of-the-art visual tracking methods on four visual tracking datasets.

The main contributions of this paper are summarized as follows:
1) State-of-the-art DL-based visual tracking methods are categorized based on their architecture (i.e., CNN, SNN, RNN, GAN, and custom networks), network exploitation (i.e., off-the-shelf deep features and deep features for visual tracking), network training for visual tracking (i.e., only offline training, only online training, both offline and online training), network objective (i.e., regression-based, classification-based, and both classification and regression-based), and exploitation of correlation filter advantages (i.e., DCF framework and utilizing correlation filter/layer/functions). Such a study covering all these aspects in detailed categorization of visual tracking methods has not been previously presented.
2) The main motivations and contributions of the DL-based methods to tackle the visual tracking problems are summarized. To the best of our knowledge, this is the first paper that investigates the primary problems and proposed solutions of visual tracking methods. This classification provides a proper insight in designing accurate and robust DL-based visual tracking methods.
3) Based on fundamental characteristics (including the number of videos, number of frames, number of classes or clusters, sequence attributes, absent labels, and overlap with other datasets), recent visual tracking benchmark datasets including OTB2013 [185], VOT [40] [46], ALOV [48], OTB2015 [186], TC128 [187], UAV123 [188], NUS-PRO [189], NSF [190], DTB [191], TrackingNet [192], OxUvA [193], BUAA-PRO [194], GOT10k [195], and LaSOT [196] are compared.
4) Finally, extensive quantitative and qualitative experimental evaluations are performed on well-known OTB2013, OTB2015, VOT2018, and LaSOT visual tracking datasets, and the state-of-the-art visual trackers are analyzed based on different aspects. Moreover, this paper specifies the most challenging visual attributes not only for the VOT2018 dataset, but also for the OTB2015 and LaSOT datasets for the first time. At last, the VOT toolkit [45] has been modified to qualitatively compare different methods according to the TraX protocol [197].

According to the comparisons, the following observations are made:
1) The SNN-based methods are the most attractive deep architectures due to their satisfactory balance between performance and efficiency for visual tracking. Moreover, the visual tracking methods recently attempt to exploit the advantages of RL and GAN methods to refine their decision making and alleviate the lack of training data. Based on these advantages, the recent visual tracking methods aim to design custom neural networks for visual tracking purposes.
2) The offline end-to-end learning of deep features appropriately adapts the pre-trained features for visual tracking. Although the online training of DNN increases the computational complexity such that most of these methods are not suitable for real-time applications, it considerably helps visual trackers to adapt with significant appearance variation, prevent from visual distractors, and improve the accuracy and robustness of visual tracking methods. Hence, exploiting both offline and online training procedures provides more robust visual trackers.
3) Leveraging deeper and wider backbone networks improves the discriminative power of distinguishing the target from its background.
4) The best visual tracking methods use both regression and classification objective functions not only to estimate the best target proposal but also to find the tightest BB for target localization.
5) The exploitation of different features enhances the robustness of the target model. For instance, most of the DCF-based methods fuse the deep off-the-shelf features and hand-crafted features (e.g., HOG and CN) for this reason. Also, the exploitation of complementary features such as temporal or contextual information has led to more discriminative and robust features for target representation.
6) The most challenging attributes for DL-based visual tracking methods are occlusion, out-of-view, and fast motion. Moreover, visual distractors with similar semantics may result in drifting problem.

The rest of this paper is as follows. Section 2 introduces our taxonomy of deep visual tracking methods. The visual tracking benchmark datasets and evaluation metrics are briefly compared in Section 3. Experimental comparisons of the state-of-the-art visual tracking methods are performed in Section 4. Finally, Section 5 summarizes the conclusions and future directions.

2 TAxonomy of Deep Visual Tracking Methods

In this section, three major components of: target representation/information, training process, and learning procedure are described. Then, the proposed comprehensive taxonomy of DL-based methods is presented.

One of primary motivations of DL-based methods is improving a target representation by utilizing/fusing deep
Deep Visual Tracking Methods
Network architecture
- CNN
- Siamese network
- Custom network
- RNN
- GAN

Network exploitation
- Deep off-the-shelf features
- Deep features for visual tracking

Network training for visual tracking
- Only offline pre-training
- Only online training
- Differs and online training

Classification-based
- DeepTracker, RST, CLUT, UCT/UCT_Lite, DSiam/DSiamM, CFNet, DCFNet/DCF2, FlowTrack, RASNet, adaDDCF, FICFNet, ATOM, TADT, UDT, MTHCF
- Attrack, TRACA, DCTN, CFSRL, DRRL, AEPCF, DRLT, EAST, P

Classification and regression-based
- SMART, TRACA, DCTN, CFSRL, DRRL, EAST, ADNet, C2FT, DRL

Regression-based
- Siamese network
- Custom network
- RNN
- GAN

Siamese network
- ACT, TRACA, DCTN, CFSRL, DRRL, EAST, P

Custom network
- ACT, TRACA, DCTN, CFSRL, DRRL, EAST, ADNet, C2FT, DRL

RNN
- FPRNet, REL, SAM

GAN
- VITAE, TOGAN, ADC

Deep off-the-shelf features
- DCF, DeepSiRDCF, FCNT, CNT/SMART, ACT, TRACA, DCTN-SVM, DEPT, COST, MDNet, HDT, SICT, RBPN, CBRT, C2F, DEEP, UCT/UCT_Lite, TSRT, P

Deep features for visual tracking
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Only offline pre-training
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Only online training
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Differs and online training
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Classification-based
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Classification and regression-based
- SMART, TRACA, DCTN, CFSRL, DRRL, EAST, P

Regression-based
- SMART, TRACA, DCTN, CFSRL, DRRL, EAST, P

Network output
- Unlinking correlation filters/advantages
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Network topology
- DCF-based methods
- DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

DeepTracker, RBPN, C2F, C2FT, DRLT, EAST, P

Unlinking correlation filters/advantages
- UCT/UCT_Lite, DSiam/DSiamM, DCFNet/DCF2, FlowTrack, RANNet, AdaDDCF, HSI, ATOM, TADT, UDT, MTHCF

Fig. 3: Taxonomy of DL-based visual tracking methods.
2.1.1 Convolutional Neural Network (CNN)

Motivated by CNN breakthroughs in computer vision tasks and some attractive advantages such as parameter sharing, sparse interactions, and dominant representations, a wide range of methods utilize CNNs for visual tracking. The CNN-based visual trackers are mainly classified according to the following motivations.

- Robust target representation: Providing a powerful target representation is the main advantage of employing CNNs for visual tracking. To achieve the goal of learning generic representations for target modeling and constructing a more robust target model, the main contributions of methods are classified into:
  - ii) designing specific deep convolutional networks instead of employing pre-trained models [63], [68], [72], [73], [75], [76], [80], [82], [89], [97], [100], [101], [104], [112], [116], [135], [137], [142], [144], [153], [165], [168], [169], [173],
  - iii) constructing multiple target models to capture variety of target appearances [75], [116], [127], [129], [130], [143], [146], [172],
  - iv) incorporating spatial and temporal information to improve model generalization [79], [82], [106], [119], [122], [137], [151], [153],
  - v) fusion of different deep features to exploit complementary spatial and semantic information [64], [101], [108], [109], [135],
  - vi) learning different target models such as relative model [104] or part-based models [116], [127], [146] to handle partial occlusion and deformation, and
  - vii) utilizing multi-stage regression to refine target representation [177].

- Computational complexity problem: Despite significant progress of CNNs in terms of target estimation accuracy, the CNN-based methods still suffer from high computational complexity. To reduce this limitation, CNN-based visual tracking methods exploit different solutions namely:
  - i) disassembling a CNN into several shrunk models [76],
  - ii) compressing or pruning training sample space [94], [115], [141], [153], [168] or feature selection [61], [154],
  - iii) feature computation via RoIAlign operation [112] (i.e., feature approximation via bilinear interpolation) or oblique random forest [99] for better data capturing,
  - iv) corrective domain adaption method [165],
  - v) lightweight structure [72], [73], [167],
  - vi) efficient optimization processes [98], [155],
  - vii) exploiting advantages of correlation filters [59]–[61], [64], [69], [74], [77], [80], [83], [85], [86], [92], [94], [96], [98], [100], [106], [108], [109], [115], [119], [122], [126], [127], [129], [131], [135], [140], [141], [143], [144], [149], [151], [155], [159], [165], [167], [171], [172], [174] for efficient computations,
  - viii) particle sampling strategy [96], and
  - ix) utilizing attentional mechanism [100].

2.1.2 Siamese Neural Network (SNN)

To learn similarity knowledge and achieve real-time speed, SNNs are widely employed for visual tracking purposes in the past few years. Given the pairs of target and search regions, these twin networks compute the same function to produce a similarity map. The common aim of SNN-based methods is to overcome the limitations of pre-trained deep CNNs and take full advantage of end-to-end learning for real-time applications.

- Discriminative target representation: The ability of visual tracker to construct a robust target model majorly relies on target representation. For achieving more discriminative deep features and improving target modeling, SNN-based methods propose:
  - i) learning distractor-aware [111] or target-aware features [161],
  - ii) fusing deep multi-level features [132], [157] or combining confidence maps [88], [90], [124],
  - iii) utilizing different loss functions in Siamese formulation to train more effective filters [57], [107], [161]–[163],
  - iv) leveraging different types of deep features such as context information [117], [124], [158] or temporal features/models [65], [81], [125], [133], [158], [175],
  - v) full exploring of low-level spatial features [132], [157],
  - vi) considering angle estimation of target to prevent from salient background objects [118],
  - vii) utilizing multi-stage regression to refine target representation [157], and
  - viii) using deeper and wider deep network as the backbone to increase receptive field of neurons which is equivalent to capturing the structure of the target [56].

- Adapting target appearance variation: Using only offline training of the first generation of SNN-based methods caused a poor generalization of these methods to adapt to target appearance variations. To solve it, recent SNN-based methods have proposed:
  - i) online update strategies [81], [90], [93], [103], [111], [152], [156], [165],
  - ii) background suppression [81], [111],
  - iii) formulating tracking task as a one-shot local detection
task \[111\], \[123\], and
iv) giving higher weights to important feature channels or
score maps \[88\], \[124\], \[128\], \[148\].
Alternatively, the DaSiamRPN \[111\] and MMLT \[114\] use a
local-to-global search region strategy and memory exploita-
tion to handle critical challenges such as full occlusion and
out-of-view and enhance local search strategy, respectively.

- **Balancing training data:** As a same problem for the
CNN-based methods, some efforts by SNN-based methods
have been performed to address imbalance distribution of
training samples. The main contributions of the SNN-based
methods are:
i) exploiting multi-stage Siamese framework to stimulate
hard negative sampling \[157\],
ii) adopting sampling heuristic such as fixed foreground-
to-background ratio \[157\] or sampling strategies such as
random sampling \[111\] or flow-guided sampling \[133\], and
iii) taking advantages of correlation filter/layer into Siamese
framework \[77\], \[78\], \[81\], \[93\], \[102\], \[111\], \[123\], \[125\], \[128\], \[148\], \[152\], \[154\], \[156\], \[161\], \[162\], \[170\].

### 2.1.3 Recurrent Neural Network (RNN)

Since visual tracking is related to both spatial and tempo-
ral information of video frames, RNN-based methods are
employed to consider target motion/movement, simulta-
neously. Because of arduous training and a considerable
number of parameters, the number of RNN-based methods
is limited. Almost all these methods try to exploit additional
information and memory to improve target modeling. Also,
the second aim of using RNN-based methods is to avoid
fine-tuning of pre-trained CNN models, which takes a lot of
time and is prone to over-fitting. The primary purposes of
these methods can be classified to the spatio-temporal repre-
sentation capturing \[84\], \[139\], \[175\], leveraging contextual
information to handle background clutters \[139\], exploiting
multi-level visual attention to highlight target as well as
background suppression \[175\], and use convolutional long
short-term memory (LSTM) as the memory unit of previous
target appearances \[84\]. Moreover, RNN-based methods
exploit pyramid multi-directional recurrent network \[139\] or
incorporate LSTM into different networks \[84\] to memorize
target appearance and investigate time dependencies during
the time. At last, the \[139\] encodes the self-structure of a
target to reduce tracking sensitivity related to similar
distractors.

### 2.1.4 Generative Adversarial Network (GAN)

Based on some attractive advantages such as capturing sta-
tistical distribution and generating desired training samples
without extensive annotated data, the GANs have been
intensively utilized in many research areas. Although GANs
are usually hard to train and evaluate, some DL-based
visual trackers employ them to enrich training samples and
target modeling. These networks can augment positive sam-
ples in feature space to address the imbalance distribution of
training samples \[121\]. Also, the GAN-based methods can
learn general appearance distribution to deal with the
self-learning problem of visual tracking \[136\]. Furthermore,
the joint optimization of regression and discriminative net-
works will lead to taking advantage of both regression and
classification tasks \[164\].

### 2.1.5 Custom Networks

Inspired by particular deep architectures and network lay-
ers, modern DL-based methods have combined a wide
range of networks such as AE, CNN, RNN, SNN, and also
deep RL for visual tracking. The primary motivation is to
compensate deficiencies of ordinary methods by exploiting
the advantages of other networks. The primary motivations
and contributions are classified as follows.

- **Computational complexity problem:** As stated before,
this problem limits the performance of online trackers in
real-time applications. To control computational complexity
of custom network-based visual trackers, the TRACA \[120\]
and AEPCF \[171\] methods employ AEs to compress raw
conventional deep features, the EAST \[181\] adaptively takes
either shallow features for simple frames for tracking or
expensive deep features for challenging ones \[181\], and the
TRACA \[120\], CFSRL \[145\], and AEPCF \[171\] exploit the
DCF computation efficiency.

- **Model update:** To maintain the stability of target model
during the tracking process, different update strategies
have been proposed; for instance, the CFSRL \[145\] updates
multiple models in parallel, the DRRL \[166\] incorporates
an LSTM to exploit long-range time dependencies, and the
AEPCF \[171\] utilizes long-term and short-term update
schemes to increase tracking speed. To prevent the erro-
neous model update and drift problem, the RDT \[184\] has
revised the visual tracking formulation to a consecutive
decision-making process about the best target template for
the next localization. Moreover, efficient learning of good
decision policies using RL \[183\] is another technique to take
either model update or ignore the decision.

- **Limited training data:** The soft and non-representative
training samples can disturb visual tracking if occlusion,
blurring, and large deformation happen. The AEPCF \[171\]
exploits a dense circular sampling scheme to prevent over-
fitting problem caused by limited training data. To make
diverse and challenging training data, the SINT++ \[58\] gen-
ergates positive and hard training samples by positive sample
generation network (PSGN) and hard positive transforma-
tion network (HPTN). To efficiently train DNNs without
a large amount of training data, partially labeled training
samples are utilized by an action-driven deep tracker \[176\],
\[177\]. Also, the P-Track \[183\] uses active decision-making
to interactively label videos while learning a tracker when
limited annotated data are available.

- **Search strategy:** From the definition, visual tracking
methods estimate the new target state in the search region
of the next frame given an initial target state in the first
frame. The selection of the best search region is depend-
ton the iterative search strategies that usually are not
only independent from video content but also heuristic,
brute-force, and hand-engineered. Despite classical search
strategies based on sliding windows, mean shift, or particle
filter, the state-of-the-art DL-based visual trackers exploit
RL-based methods to learn data-driven searching policies.
To exhaustively explore a region of interest and select the
best target candidate, action-driven tracking mechanisms
\[176\], \[177\] consider the target context variation and actively
pursues the target movement. Furthermore, the ACT and
DRRL have proposed practical RL-based search strategies
for real-time requirements by dynamic search process \[110\] and coarse-to-fine verification \[166\].

- **Exploiting additional information**: To enhance the target model by utilizing motion or contextual information, the DCTN \[138\] establishes a two-stream network, and the SRT \[87\] adopts multi-directional RNN to learn further dependencies of a target during visual tracking. To encode relevant information for better localization, previous semantic information and tracking proposals are modeled by a recurrent convolutional network \[180\]. Also, DRL-IS \[179\] has introduced an Actor-Critic network to estimate target motion parameters efficiently.

- **Decision making**: Online decision making has principal effects on the performance of DL-based visual tracking methods. The state-of-the-art methods attempt to learn online decision making by incorporating RL into the DL-based methods instead of hand-designed techniques. To gain effective decision policies, the P-Track \[183\] ultimately exploits data-driven techniques in an active agent to decide about tracking, re-initializing, or updating processes. Also, the DRL-IS \[179\] utilizes a principled RL-based method to select sensible action based on target status. Also, an action-prediction network has been proposed to adjust the continuous actions of a visual tracker to determine the optimal hyper-parameters for learning the best action policies and make satisfactory decisions \[182\].

### 2.2 Network Exploitation

Roughly speaking, there are two main exploitations of DNNs for visual tracking, including reusing a pre-trained model on partially related datasets or exploiting deep features for visual tracking, which is equivalent to train a DNN for visual tracking purposes.

#### 2.2.1 Model Reuse or Deep Off-the-Shelf Features

Exploiting deep off-the-shelf features is the simplest way to transfer the power of deep features into the traditional visual tracking methods. These features provide a generic representation of visual targets and help visual tracking methods to construct more robust target models. Regarding topologies, DNNs include either a simple multi-layer stack of non-linear layers (e.g., AlexNet \[34\], VGGNet \[53\], \[36\]) or a directed acyclic graph topology (e.g., GoogLeNet \[37\]). ResNet \[128\], SSD \[198\], Siamese convolutional neural network \[199\] which allows designing more complex deep architectures that include layers with multiple input/output. The main challenge of these trackers is how to benefit the generic representations completely. Different methods employ various feature maps and models that have been pre-trained majorly on large-scale still images of ImageNet dataset \[39\] for the object recognition task. Numerous methods have studied the properties of pre-trained models and explored the impact of deep features in traditional frameworks (see Table 1). As a result, simultaneous exploitation of both semantic and fine-grained deep features has been preferred by the DL-based methods \[59\], \[61\], \[64\], \[140\], \[157\], \[200\], \[201\]. Fusion of deep features is also another motivation of these methods which is performed by different techniques to utilize multi-resolution deep features \[59\]–\[61\], \[64\], \[69\], \[83\], \[109\], \[129\], \[130\], \[143\], \[152\], \[173\] and independent fusion of deep features with shallow ones at a later stage \[109\]. Exploiting motion information \[92\], \[106\], \[172\], \[202\], and selecting appropriate deep features for visual tracking tasks \[61\], \[154\] are two other interesting motivations for DL-based methods. The detailed characteristics of DL-based visual trackers based on deep off-the-shelf features are shown in Table 1. Needless to say, the network output for these methods are deep feature maps.

#### 2.2.2 Deep Features for Visual Tracking Purpose

One trending part of new methods is how to design and train DNNs for visual tracking. Using deep off-the-shelf features limits the visual tracking performance due to inconsistency among the objectives of different tasks. Also, offline learned deep features may not capture target variations and tend to over-fit on initial target templates. Hence, DNNs are trained on large-scale datasets to specialize the networks for visual tracking purposes. Besides, applying a fine-tuning process during visual tracking can adjust some network parameters and produce more refined target representations. However, the fine-tuning process is time-consuming and prone to over-fitting because of a heuristically fixed iteration number and limited available training data. As it is shown in Table 2 to Table 4, these DL-based methods usually train a pre-trained network (i.e., backbone network) by either offline training or online training or both.

### 2.3 Network Training

The state-of-the-art DL-based visual tracking methods mostly exploit end-to-end learning with train/re-train a DNN by applying gradient-based optimization algorithms. However, these methods have differences according to their offline network training, online fine-tuning, computational complexity, dealing with lack of training data, addressing overfitting problem, and exploiting unlabeled samples by unsupervised training. The network training section in the previous review papers \[51\]–\[53\] consider both FENs and EENs, although the FENs were only pre-trained for other tasks, and there is no training procedure for visual tracking. In this survey, DL-based methods are categorized into only offline pre-training, only online training, and both offline and online training for visual tracking purposes. The training details of these methods are shown in Table 2 to Table 4.

#### 2.3.1 Only Offline Training

Most of the DL-based visual tracking methods only pre-train their networks to provide a generic target representation and reduce the high risk of over-fitting due to imbalanced training data and fixed assumptions. To adjust the learned filter weights for visual tracking task, the specialized networks are trained on large-scale data to not only exploit better representation but also achieve acceptable tracking speed by preventing from training during visual tracking (see Table 2).

#### 2.3.2 Only Online Training

To discriminate unseen targets which may consider as the target in evaluation videos, some DL-based visual tracking methods use online training of whole or a part of DNNs to adapt network parameters according to the large variety of target appearance. Because of the time-consuming process
TABLE 1: Deep off-the-shelf features for visual tracking. The abbreviations are denoted as: confidence map (CM), saliency map (SM), bounding box (BB), votes (vt), deep appearance features (DAF), deep motion features (DMF).

| Method            | Pre-trained models | Exploited layers | Pre-training data | Pre-training dataset(s) | Exploited features | PC (CPU, RAM, Nvidia GPU) | Language Framework | Speed (fps) | Tracking output |
|-------------------|--------------------|------------------|-------------------|-------------------------|--------------------|--------------------------|---------------------|-------------|----------------|------------------|
| DeepSRDCF        | VGG-M              | Conv5            | Still images      | ImageNet                | HOG, DAF           | N/A, GPU                 | Matlab MatConvNet    | N/A         | CMCCOT          |
| PC                 |                    |                  |                   |                         |                    |                          |                     | CM          |                |
| GOTURN            | AlexNet            |                   | ILSVRC-DET, ALOV  | DAF                     | N/A, GTX Titan X GPU | C/C++ Caffe              | PyTorch             | 166         | BBSiamFC        |
| Intel I7-3.60GHz, 32GB RAM, GTX 1080 GPU |                      |                  |                   |                         |                    |                          |                     |             | CM              |
| SMART             | ZFNet              | DAF              | Only set the learning rates in conv1-conv3 to zero | Intel 3.10GHz CPU, 256 GB RAM, GTX Titan X GPU | N/A, GPU           | Matlab Caffe              | ...                 | 32          |                |
| SMFRPN2T-DSiam     | Custom             | DAF              | Only update fully-connected layers | Matlab Caffe | N/A, GPU | Matlab Caffe | ... | 4.15         | CMFRPN2T-DSiam |
| DRN               | AlexNet            | DAF              | ImageNet          | Yes                     | DAF                | N/A, K20 GPU             | Matlab Caffe         | 1.3         | CMDSiam/DSiamM |
| Only on the first frame |                      |                  |                   |                         |                    |                          |                     |             |                |
| 3 | Table 2: Only offline training for visual tracking. The abbreviations are denoted as: confidence map (CM), saliency map (SM), bounding box (BB), object score (OS), feature maps (FM), segmentation mask (SCM), rotated bounding box (RBB), action (AC), deep appearance features (DAF), deep motion features (DMF).

| Method            | Backbone network   | Offline training dataset(s) | Exploited features | PC (CPU, RAM, Nvidia GPU) | Language Framework | Speed (fps) | Tracking output |
|-------------------|--------------------|-----------------------------|--------------------|--------------------------|---------------------|-------------|----------------|
| GOTURN            | AlexNet            | ILSVRC-DET, ALOV            | DAF                | N/A, GTX Titan X GPU     | C/C++ Caffe         | 166         | BBSiamFC        |
| Intel I7-3.60GHz, 32GB RAM, GTX 1080 GPU |                      |                  |                   |                         |                    |                          |                     |             | CM              |
| SMART             | ZFNet              | DAF                         | Only set the learning rates in conv1-conv3 to zero | Intel 3.10GHz CPU, 256 GB RAM, GTX Titan X GPU | N/A, GPU | Matlab Caffe | ... | 32          | CMFRPN2T-DSiam |
| 3 | Table 3: Only online training for visual tracking. The abbreviations are denoted as: confidence map (CM), bounding box (BB), object score (OS), deep appearance features (DAF), deep motion features (DMF).

| Method            | Backbone network   | Exploited features | Strategy to alleviate the over-fitting problem | PC (CPU, RAM, Nvidia GPU) | Language Framework | Speed (fps) | Tracking output |
|-------------------|--------------------|--------------------|-----------------------------------------------|--------------------------|---------------------|-------------|----------------|
| SMART             | ZFNet              | DAF                | Only update fully-connected layers | Intel 3.10GHz CPU, 256 GB RAM, GTX Titan X GPU | N/A, GPU | Matlab Caffe | ... | 32          | CMFRPN2T-DSiam |
| 3 | Table 4: Both offline and online training for visual tracking. The abbreviations are denoted as: confidence map (CM), bounding box (BB), object score (OS), voting map (VM), action (AC), deep appearance features (DAF), deep motion features (DMF), compressed deep appearance features (CDAF).

| Method            | Backbone network   | Offline training(s) | Online network training | Exploited features | PC (CPU, RAM, Nvidia GPU) | Language Framework | Speed (fps) | Tracking output |
|-------------------|--------------------|---------------------|-------------------------|--------------------|--------------------------|---------------------|-------------|----------------|
| SMART             | ZFNet              | DAF                 | Only update fully-connected layers | Intel 3.10GHz CPU, 256 GB RAM, GTX Titan X GPU | N/A, GPU | Matlab Caffe | ... | 32          | CMFRPN2T-DSiam |
| 3 | Table 5: Both offline and online training for visual tracking. The abbreviations are denoted as: confidence map (CM), bounding box (BB), object score (OS), voting map (VM), action (AC), deep appearance features (DAF), deep motion features (DMF), compressed deep appearance features (CDAF).
of offline training on large-scale training data and insufficient discrimination of pre-trained models for representing tracking particular targets, the methods shown in Table 3 use directly training of DNNs and inference process alternatively online. However, these methods usually exploit some strategies to prevent over-fitting problem and divergence.

2.3.3 Both Offline and Online Training

To exploit the maximum capacity of DNNs for visual tracking, the methods shown in Table 3 use both offline and online training. The offline and online learned features are known as shared and domain-specific representations, which majorly can discriminate the target from foreground information or intra-class distractors, respectively. Because visual tracking is a hard and challenging problem, the DL-based visual trackers attempt to employ feature transferability and online domain adaption simultaneously.

2.4 Network Objective

After the training and inference stages, DL-based visual trackers localize the given target based on network objective function. Hence, the DL-based visual tracking methods are categorized into classification-based, regression-based, or both classification and regression-based methods as follows. This categorization is based on the objective functions of DNNs that have been used in visual tracking methods (see Fig. 3); hence, this sub-section does not include the methods that exploit deep off-the-shelf features because these methods do not design and train the networks and usually employ DNNs for feature extraction.

2.4.1 Classification-based Objective Function

Motivated by other computer vision tasks such as image detection, classification-based visual tracking methods employ object proposal methods to produce hundreds of candidate proposals extracted from the search region. These methods aim to select the high score proposal by classifying the proposals to the target and background classes. This two-class (or binary) classification involves visual targets from various classes and moving patterns, and also individual sequences, including challenging scenarios. Due to the main attention of these methods on inter-class classification, tracking a visual target in the presence of the same labeled targets is intensely prone to drift-problem. Also, tracking the arbitrary appearance of targets may lead to problems in recognizing different targets with varying appearances. Therefore, the performance of the classification-based visual tracking methods is also related to their object proposal method, which usually produces a considerable number of candidate BBs. On the other side, some recent DL-based methods utilize this objective function to take the optimal action on BB [58], [166], [176]–[179], [181].

2.4.2 Regression-based Objective Function

Due to the continuous instinct of estimation space of visual tracking, regression-based methods usually aim to directly localize target in the subsequent frames by minimizing a regularized least-squares function. Generally, extensive training data are needed to train these methods effectively. The primary goal of regression-based methods is to refine the formulation of L2 or L1 loss functions such as utilizing shrinkage loss in learning procedure [108], modeling both regression coefficients and patch reliability to optimize a neural network efficiently [127], or applying the cost-sensitive loss to enhance unsupervised learning performance [162].

2.4.3 Both Classification and Regression-based Objective Function

To take advantages of both foreground-background/category classification and ridge regression (i.e., regularized least-squares objective function), some methods employ both classification- and regression-based objective functions for visual tracking (see Fig. 3), which their goal is to bridge the gap between the recent tracking-by-detection and continuous localization process of visual tracking. Commonly, these methods utilize classification-based methods to find the most similar object proposal to target and then the estimated region will be refined by a BB regression method [55], [68], [75], [87], [101], [110], [112], [123], [137], [153], [168], [173]. To enhance efficiency and accuracy, the target regions are estimated by classification scores and optimized regression/matching functions [56], [57], [134], [145], [146], [156], [157], [160], [163], [164], [167], [179]. The classification outputs are mainly inferred for confidence scores of candidate proposals, foreground detection, response of candidate window, actions, and so forth.

2.5 Network Output

Based on their network outputs, the DL-based methods are classified into six main categories (see Fig. 3 and Table 2 to Table 4), namely confidence map (also includes score map, response map, and voting map), BB (also includes rotated BB), object score (also includes probability of object proposal, verification score, similarity score, and layer-wise score), action, feature maps, and segmentation mask. According to the network objective, the DL-based methods generate different network outputs to estimate or refine the estimated target location.

2.6 Exploitation of Correlation Filters Advantages

The DCF-based methods aim to learn a set of discriminative filters that an element-wise multiplication of them with a set of training samples in the frequency domain determines spatial target location. Since DCF has provided competitive tracking performance along with computational efficiency compared to sophisticated techniques, DL-based visual trackers use correlation filter advantages. These methods are categorized based on how they exploit DCF advantages by using either a whole DCF framework or some benefits, such as its objective function or correlation filters/layers. Considerable visual tracking methods are based on the integration of deep features in the DCF framework (see Fig. 3). These methods aim to improve the robustness of target representation against challenging attributes, while other methods attempt to benefit the computational efficiency of correlation filter(s) [95], correlation layer(s) [125], [141], [148], [161], [170], and the objective function of correlation filters [80], [81], [102], [128], [156], [162].
3 Visual Tracking Benchmark Datasets

Visual tracking benchmark datasets have been introduced to provide fair and standardized evaluations of single-object tracking algorithms. The tracking datasets contain video sequences that include not only diverse target categories but also have different time durations and challenging attributes. These datasets contain a large variety of the numbers of video sequences, frames, attributes, and classes (or clusters). The attributes include illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC), low resolution (LR), aspect ratio change (ARC), camera motion (CM), full occlusion (FOC), partial occlusion (POC), similar object (SOB), viewpoint change (VC), light (LI), surface cover (SC), peculiarity (SP), transparency (TR), shape (SH), motion smoothness (MS), motion coherence (MCO), confusion (CON), low contrast (LC), zooming camera (ZC), long duration (LD), shadow change (SHC), flash (FL), dim light (DL), camera shaking (CS), rotation (ROT), fast background change (FBC), motion change (MOC), object color change (OCO), scene complexity (SCO), absolute motion (AM), size (SZ), relative speed (RS), distractors (DI), length (LE), fast camera motion (FCM), and small/large objects (SLO). Table 5 compares the characteristics of visual tracking datasets, the existence of missing labeled data for unsupervised training, and the partial overlap of datasets. By different evaluation protocols, existing visual tracking benchmarks assess the accuracy and robustness of visual trackers in realistic scenarios. The homogenized evaluation protocols facilitate straightforward comparison and development of visual trackers. In the following, the most popular visual tracking benchmark datasets and evaluation metrics are briefly described.

3.1 Visual Tracking Datasets

One of the first object tracking benchmark datasets, called OTB2013 [185], is developed by 51 fully annotated video sequences to address the issues of reported tracking results based on a few video sequences or inconsistent initial conditions or parameters. The OTB2015 [186] is an extended OTB2013 dataset that includes 100 commonly used video sequences with the aim of unbiased performance comparisons. To provide the performance of visual tracking algorithms on color sequences, the Temple Color 128 (TC128) dataset [187] collected a set of 129 fully annotated video sequences that 78 ones are different from the OTB datasets; however, its attributes are annotated same as attributes of the OTB ones. The Amsterdam library of ordinary videos (ALOV) dataset [48] has been gathered to cover diverse video sequences and attributes. By emphasizing on challenging visual tracking scenarios, the ALOV dataset composes of 304 assorted short videos and ten longer ones. The video sequences are chosen from real-life YouTube videos and have thirteen difficulty degrees. The videos of ALOV have been categorized according to one of its attributes (Table 5), although in the OTB dataset each video has been annotated by several visual attributes.

The unmanned aerial vehicle 123 (UAV123) [188] provides a sparse and low altitude UAV tracking dataset which contains the realistic and synthetic HD video sequences captured by professional-grade UAV, a board-cam mounted on a small-low cost UAV, and UAV simulator. For tracking pedestrian and rigid objects, the NUS people and rigid objects (NUS-PRO) dataset [189] not only has been provided 365 video sequences from YouTube under twelve challenging factors but also annotated the level of occluded objects of each frame with no occlusion, partial occlusion, and full occlusion labels. It consists of five main categories (namely, face, pedestrian, sportsman, rigid object, and long sequences) and sixteen subcategories (including, hat, mask, interview, politician, sunglasses, basketball, gymnastics, handball, racing, soccer, tennis, airplane, boat, car, helicopter, and motorcycle) that majorly captured by moving cameras. By higher frame rate (240 FPS) cameras, the need for speed (NfS) dataset [190] provides 100 video sequences from real-world scenarios to systematically investigate trade-off bandwidth constraints related to real-time analysis of visual trackers. These videos are either recorded by hand-held iPhone/iPad cameras or from YouTube videos. Also, it contains a wide range of visual targets, which are the vehicle, person, face, animal, aircraft, and boat.

Motivated by inequality of large dataset with useful one, the VOT dataset [40–46] aims to provide a diverse and sufficiently small dataset from existing visual tracking datasets, and per-frame annotate them by rotatable BBs and visual properties. To evaluate different visual tracking methods fast and straightforward, the VOT includes visual tracking exchange (TraX) protocol [197] that not only prepares data, runs experiments, and performs analyses but also can detect tracking failures (i.e., losing the target) and re-initialize the tracker five frames after each failure to assess tracking robustness. The drone tracking benchmark (DTB) [191] is the dataset captured by UAVs or drones that consists of 70 diverse RGB videos with massive displacement of target location due to abrupt camera motion. This dataset primarily focuses on tracking people and cars and aims to concentrate on the performance of the motion model of visual trackers for non-stationary or slow-moving cameras, which have a higher degree of freedom. Despite some small and saturated tracking datasets in the wild, which are mostly provided for object detection task, the large-scale TrackingNet benchmark dataset [192] has been proposed to feed deep visual trackers properly. It includes 500 original videos, more than 14 million upright BB annotations, densely annotated data in time, rich distribution of object classes, and real-world scenarios by sampled YouTube videos. While the training and testing sets of the TrackingNet respectively consist of 30132 and 115 video sequences, it provides the same object class distributions in terms of video length, BB resolution, motion change, aspect ratio, and attribute distributions.

With the aim of long-term tracking of frequently disappearance targets, the OxUvA dataset [193] has selected 366 video sequences (14 hours of video) from YouTube-BoundingBoxes (or YTBB) [203] to provide development and test sets with continuous attributes. The annotated absent labels show that the target does not exist in the frame. Also, this dataset includes continuous attributes which their quantities have been measured by BB annotations and metadata. The BUAA-PRO dataset [194] is a segmentation-based
benchmarks dataset to address the problem of inevitable non-target elements in BBs. It consists of 150 video sequences from NUS-PRO with three main categories of the rigid object (namely, airplane, boat, car, helicopter, and motorcycle), sportsman (includes basketball, gymnastics, handball, racing, soccer, and tennis), and pedestrian. It exploits not only the same attributes of NUS-PRO but also the segmentation mask-based version of level-based occlusion attribute (i.e., no occlusion, partial occlusion, and full occlusion). The large-scale single object tracking (LaSOT) benchmark dataset [190] has developed to address the problems of visual tracking datasets such as small scale, lack of high-quality, dense annotations, short video sequences, and category bias. The object categories are from the ImageNet and a few visual tracking applications (such as drones) with an equal number of videos per each category. According to the Pareto principle (80% for training and 20% for the test), the LaSOT data are divided into the training and testing subsets, including 1120 (2.3M frames) and 280 (690K frames) video sequences, respectively. The large diversity benchmark dataset, called GOT-10k [195], includes more than ten thousand videos from the semantic hierarchy of WordNet [204] splitting to train, validation, and test sets. The video sequences are classified to 563 classes of moving objects and 87 classes of motion to cover as many challenging patterns in real-world scenarios as possible. The GOT-10k has informative attributes similar to the OxUvA.

### 3.2 Evaluation Metrics

To perform experimental comparisons on large-scale datasets, the visual tracking methods are evaluated by two fundamental evaluation categories of performance measures and performance plots. These metrics are briefly described as follows.

#### 3.2.1 Performance Measures

To reflect several views of a visual tracker, various performance measures have been proposed. These performance attempts to intuitively interpret performance comparisons in terms of complementary measures of accuracy, robustness, and tracking speed. In the following, these measures are concisely investigated.

- **Center location error (CLE):** The CLE is defined as the average Euclidean distance between the precise ground-truth locations of target and estimated locations by the visual tracking methods. The CLE is the oldest metric that not only is sensitive to dataset annotation and does not consider tracking failures but also ignores the targets BB and results in significant errors.

- **Accuracy:** To achieve visual tracking accuracy, first, the overlap score is calculated as $S = \frac{|b_t \cap b_g|}{|b_t \cup b_g|}$, which $b_t$, $b_g$, $\cap$, and $\cup$ represent the ground-truth BB, an estimated BB by a visual tracking method, intersection operator, union operator, and the number of pixels in the resulted region, respectively. By considering a certain threshold, the overlap score indicates the success of a visual tracker in one frame. Then, the accuracy is calculated by the average overlap scores (AOS) during the tracking when a visual tracker’s estimations have overlap with the ground-truth ones. This metric jointly considers both location and region to measure the drift rate of the estimated target up to its failure.

- **Robustness/failure score:** The robustness or failure score is defined as the number of required re-initializations when a tracker loses (or drifts) a visual target during the tracking task. The failure is detected when the overlap score drops to zero.

- **Expected average overlap (EAO):** This score is interpreted as the combination of accuracy and robustness scores. Given $N_s$ frames long sequences, the EAO score is calculated as $\bar{EAO} = \left(\frac{1}{N} \sum_{i=1}^{N_s} \frac{1}{i} \sum_{j=1}^{i} \Phi(j)\right)$, where $\Phi_i$ is defined as the average of per-frame overlaps until the end of sequences, even if failure leads to zero overlaps.

- **Area under curve (AUC):** The AUC score has defined the average success rates (normalized between 0 and 1) according to the pre-defined thresholds. To rank the visual tracking methods based on their overall performance, the AUC score summarizes the AOS of visual tracking methods across a sequence.

#### 3.2.2 Performance Plots

To figure out the performance of visual tracking methods, different methods are usually evaluated in terms of different thresholds to provide more intuitive quantitative comparisons. In the following, these metrics are summarized.

- **Precision plot:** Given the CLEs per different thresholds, the precision plot shows the percentage of video frames in which the estimated locations have at most the specific threshold, and the ground-truth locations. The precision plot shows the percentage of frames in which the estimated overlaps and the ground-truth ones have larger overlap than a certain threshold.

- **Success plot:** Given the calculated various accuracies per thresholds, success plot measures the percentage of frames in which the estimated overlaps and the ground-truth ones have larger overlap than a certain threshold.

- **Expected average overlap curve:** For an individual length of video sequences, the expected average overlap curve has resulted from the range of values in a specific interval $[N_{lo}, N_{hi}]$ as $\Phi = \frac{1}{N_{hi} - N_{lo}} \sum_{N_{lo}}^{N_{hi}} \Phi(N)$.

- **One-pass evaluation with restart (OPER):** The OPER is a supervised system that continuously measures a tracking
method in terms of accuracy to re-initialize it when a failure occurs. Also, the SRER does the same OPER for numerous evaluations of SRE.

4 Experimental Analyses

To analyze the performance of state-of-the-art visual tracking methods, 45 different methods are quantitatively compared on four well-known datasets OTB2013 [185], OTB2015 [186], VOT2018 [45], and LaSOT [196].

Due to the page limitation, all experimental results are publicly available on https://github.com/MMarvasti/Deep-Learning-for-Visual-Tracking-Survey. The included 45 DL-based trackers in the experiments are shown in Table 6. The ECO, CFNet, TRACA, DeepSTRCF, and C-RPN are considered as baseline trackers to compare performances on various datasets. All evaluations are performed on an Intel I7-9700K 3.60GHz CPU with 32GB RAM with the aid of MatConvNet toolbox [205] that uses an NVIDIA GeForce RTX 2080Ti GPU for its computations. The OTB and LaSOT toolkits evaluate the visual tracking methods in terms of the well-known precision and success plots and then rank the methods based on AUC score [185], [186]. For performance comparison on the VOT2018 dataset, the visual trackers have been assessed based on the TraX evaluation protocol [197] using three primary measures of accuracy, robustness, and EAO to provide the Accuracy-Robustness (AR) plots, expected average overlap curve, and ordering plots according to five challenging visual attributes [45], [206], [207].

4.1 Quantitative Comparisons

According to the results shown in Fig. 4, the top-5 visual tracking methods in terms of precision metric are the VITAL, MDNet, DAT, ASRCF, and SiamDW-SiamRPN on the OTB2013 dataset, the SiamDW-SiamRPN, ASRCF, VITAL, SiamRPN+++, and MDNet on the OTB2015 dataset, and the C-RPN, MDNet/VITAL, SiamFC/StructSiam, ASRCF, and DSiam on the LaSOT dataset, respectively. In terms of success metric, the ASRCF, VITAL, MDNet, DAT, and SiamRPN++ on the OTB2013 dataset, the SiamRPN++, SANet, ASRCF, VITAL, and MDNet on the OTB2015 dataset, and the C-RPN, MDNet, VITAL, ASRCF, and SiamFC on the LaSOT dataset have achieved the best performance, respectively. On the VOT2018 dataset (see Fig. 5 and Table 7), the top-5 visual trackers are the SiamMask, SiamRPN++, DaSiamRPN, C-RPN, and SiamDW-SiamRPN in terms of accuracy measure while the UPDT, LSART, DeepSTRCF, SiamMask, and SiamRPN+++/DRT have the best robustness, respectively. On the other hand, the best visual tracking methods based on both precision-success measures (see Fig. 4), the VITAL, MDNet, DAT, ASRCF, and SiamRPN++ on the OTB2013 dataset, the SiamRPN++, ASRCF, VITAL, SiamDW-SiamRPN, and MDNet on the OTB2015 dataset, and the C-RPN, MDNet, VITAL, SiamFC, and ASRCF/StructSiam on the LaSOT dataset. On the VOT2018 dataset, the SiamRPN++, SiamMask, UPDT, DRT, and DeepSTRCF are the best performing trackers based on the EAO score. Moreover, the SiamRPN++, UPDT, MCPF, LSART, and DeepSTRCF have achieved the best AUC scores while the SiamRPN, SiamRPN++, CFNet, DAT, and DCFCNet are the fastest visual trackers, respectively (see Table 7).

According to the results (i.e., Fig. 4, Fig. 5, and Table 7), the best visual tracking methods that repeated their desirable performance on different visual tracking datasets are the VITAL [121], MDNet [68], DAT [137], ASRCF [155], SiamDW-SiamRPN [56], SiamRPN++ [157], C-RPN [157], and DeepSTRCF on the LaSOT dataset. On the OTB2015 dataset, the SiamRPN++, ASRCF, VITAL, DAT, SiamFC/StructSiam, ASRCF, and DSiam on the LaSOT dataset. On the VOT2018 dataset, the SiamRPN++, SiamMask, UPDT, DRT, and DeepSTRCF are the best performing trackers based on the success metric. Also, the best-performing trackers to handle these attributes are investigated by Fig. 6 to Fig. 8 (see Table 6) shows the first to fifth challenging attributes for each benchmark dataset. Also, the best-performing trackers to handle these attributes are investigated by Fig. 5 to Fig. 7. Based on the results in Table 8, the most challenging attributes on the VOT2018 dataset are the OCC, SV, and IV, according to the accuracy metric and the OCC, MOC, and IV, according to the robustness metric. Based on the precision metric, the VITAL, OCC, and LR on the OTB2015 and the FM, OV, and DEF on the LaSOT dataset are the most challenging attributes for visual tracking methods. At last, the DEF, OV, and LR on the OTB2015 dataset and the FM, OV, and FOC on the LaSOT dataset are the most challenging ones based on the success metric. To sum up, the OCC, OV, FM, DEF, IV, and
LR are selected as the most challenging attributes that can effectively impact on the performance of DL-based visual tracking methods.

On the other hand, the most accurate visual tracking methods on the VOT2018 dataset, according to the OCC, SV, and IV are the SiamRPN++ [55], SiamMask [57], and SiamMask, respectively. Also, the DRT [126], UPDT [109], and SiamMask [57]/CFCF [144] visual trackers are the most robust trackers on the VOT2018 dataset according to the OCC, MOC, and IV, respectively. In terms of success metric, the SiamRPN++ [55] is the best visual tracking method to tackle the DEF and OV attributes while the Siam-MCF [117] is the best one to deal with the visual tracking in LR videos on the OTB2015 dataset. The ASRCF [155], ECO [94], and SiamDW-SiamRPN [56] are the best trackers in terms of precision metric to face with OV, OCC, and DEF attributes on the OTB-2015 dataset. Except for the FM attribute that the MDNet [68] is the best method in terms of precision metric, the C-RPN [15] has the superior performance on the other challenging attributes of the LaSOT benchmark dataset in terms of precision and success metrics. According to the overall and attribute-based comparisons, the C-RPN, MDNet, and VITAL are the top-3 trackers on the LaSOT dataset.

While the VOT2018 dataset provides frame-based attributes for individual video sequences, each video sequence has annotated with multiple attributes for the OTB and LaSOT datasets. According to this difference, the attribute-based comparisons on the VOT2018 are just investigated to infer the best strategies based on specific conditions. As shown in Fig. 5, the DCF-based methods have achieved fewer failures among the other methods, while the SNN-based methods have gained more overlap between the estimated BBs and ground-truth ones. The SiamRPN-based methods (i.e., [55], [57], [111]) accurately handle scenarios under each of CM, IV, MC, OCC, or SC
attributes by adopting deeper and wider backbone networks which include classification and regression branches, the following strategies lead to improve the robustness of DL-based methods under specific conditions of real-world scenarios. By considering the fusion of hand-crafted and deep features [109], [122], [126], temporal regularization term [122], reliability term [126], data augmentation [109], and exploitation of ResNet-50 model [109], the DCF-based methods have attained desirable robustness against CM attribute. To effectively deal with the IV attribute, focusing on the discrimination power between the target and its background is the primary problem. The strategies such as training a fully convolutional network for correlation filter cost function, spatial-aware KRR and spatial-aware CNN, and employing semi-supervised video object segmentation improve the robustness of DL-based trackers when significant IV occurs. To robustly deal with MC and OCC attributes, the DCF-based and CNN-based methods have performed the best. However, the SNN-based methods with the aid of region proposal subnetwork and proposal refinement can robustly estimate the tightest BB under severe scale changes.

4.3 Discussion

The overall best methods (i.e., VITAL [121], MDNet [68], DAT [137], ASRCF [155], SiamDW-SiamRPN [56], SiamRPN++ [55], C-RPN [157], StructSiam [113], SiamMask [57], DaSiamRPN [111], UPDT [109], LSART [127], DeepSTRCF [122], and DRT [126]) belong to a wide range of network architectures. For instance, the MDNet, LSART, and DAT (uses the MDNet architecture design) utilize CNNs to localize a visual target while the ASRCF (using both VGG-M and VGG-16), UPDT (using ResNet-50), DRT (using both VGG-M and VGG-16), and DeepSTRCF (using VGG-M) just...
Fig. 7: Performance comparison of state-of-the-art visual tracking methods in terms of the most challenging attributes of the LaSOT dataset.

exploit pre-trained CNN models for feature extraction. Besides the VITAL that is a GAN-based tracker, the others have the SNN architecture (i.e., C-RPN, StructSiam, SiamMask, DaSiamRPN, SiamDW, and SiamRPN++). Currently, the most attractive deep architecture for visual tracking is the SNN [55]–[57], [111], [113], [157] although the GAN-based and RL-based methods have been recently developing for some specific purposes such as addressing the imbalance distribution of training samples [121] or selecting an appropriate real-time search strategy [110], [166]. In addition to providing a desirable balance between the performance and speed of SNN-based methods, this architecture is modified to not only integrate with diverse deep structures, searching strategies, and learning schemes but also exploit fully convolutional networks, correlation layers, region proposal networks, video object detection modules (i.e., towards designing custom networks for visual tracking purposes). The interesting point is that five SNN-based methods including the SiamDW-SiamRPN, SiamRPN++, C-RPN, SiamMask, and DaSiamRPN are based on the fast SiamRPN method [123] which is consisted of Siamese subnetwork and region proposal subnetwork; these subnetworks are leveraged for feature extraction and proposal extraction on correlation feature maps to solve the visual tracking problem by one-shot detection task. The main advantages of the SiamRPN method are the time efficiency and precise estimations with integrating proposal selection and refinement strategies into a Siamese network.

Interestingly, the ASRCF, UPDT, DRT, and DeepSTRCF, which exploit deep off-the-shelf features, are among the top-performing visual tracking methods. Moreover, five methods of UPDT, DeepSTRCF, DRT, LSART, and ASRCF, which are among the best methods, take the advantages of the DCF framework. On the other side, most of the best performing visual trackers, namely the VITAL, MDNet, DAT, SiamDW, SiamRPN++, C-RPN, StructSiam, SiamMask, DaSiamRPN, and LSART exploit specialized deep features for visual tracking purpose. Although diversified backbone networks (i.e., AlexNet (for C-RPN, StructSiam, and DaSiamRPN), VGG-M (for MDNet, VITAL, and DAT), VGG-16 (for LSART), and ResNet (for SiamMask, SiamDW-SiamRPN, and SiamRPN++)) are employed for these methods, the state-of-the-art methods have been leveraging deeper neural network such as the ResNet-50 to strengthen discriminative power of target modeling. From the network training perspective, the SiamDW-SiamRPN, SiamRPN++, C-RPN, StructSiam, SiamMask, and DaSiamRPN use only offline training while the LSART utilizes only online training (online two-stream training network to effectively learn filters and in-plane rotation of target). Most of these methods aim to learn offline dominant representations to achieve real-time tracking speed. Handling significant appearance variations needs to adapt to network parameters during tracking, but online training has the over-fitting risk because of limited training samples. Hence, the VITAL, MDNet, and DAT by employing adversarial learning, domain-independent information, and attention maps as regularization terms benefit both offline and online training of DNNs. However, these methods provide tracking speed about one frame per second (FPS) that is not suitable for real-time applications. From the perspective of objective function of DNNs, the VITAL [121] and StructSiam [113] are classification-based, the LSART [127] is regression-based, and the other best-performing methods [55]–[57], [68], [111], [137], [157] are based on both classification and regression. For instance, five modified versions of the SiamRPN [123] (i.e., SiamDW-SiamRPN [56], SiamRPN++ [55], C-RPN [157], SiamMask [57], and DaSiamRPN [111]) have two branches for classification and regression.

Based on the motivation categorization of the best methods, the recent advanced methods rely on 1) alleviating the imbalanced distribution of visual training data by the data augmentation [109], [111] and generative network from adversarial learning [121], 2) efficient training and learning procedures by reformulating classification/regression problems [109], [111], [121], [122], [126], [127], [155] and providing specified features for visual tracking [55]–[57], [68], [111], [113], [121], [137], [157], 3) exploiting state-
of-the-art deeper and wider neural networks to provide more discriminative representations by leveraging ResNet models as the backbone networks [55]–[57], [109], and 4) extracting complementary features by employing additional information such as contextual [56], [109], [111], [113] or temporal information [111], [121], [122], [137]. The VITAL, DaSiamRPN, and UPDT attempt to alleviate the imbalanced distribution of positive and negative samples of training data and extract more discriminative features. The VITAL uses adversarial learning to not only augment positive samples and decrease simple negative samples but also preserve the most discriminative and robust features during tracking. Furthermore, the DaSiamRPN utilizes both data augmentation and negative semantic samples to consider visual distractors and improve visual tracking robustness. Finally, the UPDT uses standard data augmentation and a quality measure for estimated states to effectively fuse shallow and deep features.

To improve the learning process of the best DL-based methods, the UPDT, DeepSTRCF, DRT, LSART, and ASRCF revise the conventional ridge regression of DCF formulation. Moreover, the DaSiamRPN and VITAL utilize the distractor-aware objective function and reformulated objective function of GANs using a cost-sensitive loss to improve the training process of these visual trackers, respectively. Furthermore, training of DL-based methods on large-scale datasets adapts their network function for visual tracking purposes. The SiamDW, SiamRPN++, and SiamMask methods have aimed to leverage state-of-the-art deep networks as a backbone network of Siamese trackers. While these methods exploit ResNet models, the SiamDW proposes new residual modules and architectures to prevent significant receptive field increase and simultaneously improve feature discriminability and localization accuracy. Also, the ResNet-driven SNN-based tracker proposed by the SiamRPN++ includes different layer-wise and depth-wise aggregations to fill the performance gap between SNN-based and CNN-based methods.

Despite some other computer vision tasks (e.g., object detection or recognition), visual tracking is performed on video sequences that include both spatial and temporal information. In addition to the spatial information, the DAT and DeepSTRCF also consider temporal information in different ways to provide more robust features. The DAT and DeepSTRCF employ reciprocative learning and online passive-aggressive (PA) learning, respectively. While reciprocative learning scheme with attention regularization term acts as an attentive classifier to robustly select target region with the aid of temporal features, the spatial-temporal regularization of online PA learning helps reducing the sensitivity of tracker and better adapting to significant appearance variations. From the learning perspective, the four learning methods of the similarity learning (i.e., SiamDW, SiamRPN++, C-RPN, StructSiam, SiamMask, DaSiamRPN), multi-domain learning (i.e., MDNet, DAT), adversarial learning (i.e., VITAL), spatial-aware regressions learning (i.e., LSART), and DCF learning are utilized.

In the following, the best visual tracking methods are studied based on their advantages and disadvantage. Three SNN-based methods of the C-RPN, StructSiam, and DaSiamRPN exploit the shallow AlexNet as their backbone network (see Table 2), which is the main weakness of these methods according to their discriminative power. To improve tracking robustness in the presence of significant scale change and visual distractors, the C-RPN cascades multiple RPNs in a Siamese network to exploit from hard negative sampling (to provide more balanced training samples), multi-level features, and multiple steps of regressions. To decrease the sensitivity of SNN-based methods specifically for non-rigid appearance change and POC attributes, the StructSiam detects contextual information of local patterns and their relationships and matches them by a Siamese network in real-time speed. By adopting the local-to-global search strategy and the non-maximum suppression (NMS) to re-detect target and reduce potential distractors, the DaSiamRPN correctly handles the FOC, OV, POC, and BC challenges. In contrast, the SiamMask, SiamDW-SiamRPN, and SiamRPN++ exploit the ResNet models. To rely on rich target representation, the SiamMask uses three-branch architecture to estimate the target location by a rotated BB, which includes the binary mask of the target. The most failure reasons of SiamMask are the MB and OV attributes that produce erroneous target masks. To reduce the performance margin of the SNN-based methods with state-of-the-art visual tracking methods, the SiamDW-SiamRPN and SiamRPN++ study the exploitation of deep backbone networks to reduce the sensitivity of these methods to the most challenging attributes.

The MDNet and the other methods based on it (e.g., DAT) are still among the best visual tracking methods. Because of specialized offline and online training of these networks on large-scale visual tracking datasets, these methods can handle a diverse variety of challenging situations, hardly miss the visual targets, and have a satisfactory performance to track LR targets. However, these methods suffer from high computational complexity, intra-class discrimination of targets with similar semantics, and performing discrete space for scale estimation. The VITAL can tolerate massive DEF, IPR, and OPR because it focuses on hard negative samples by a high-order cost-sensitive loss. However, it does not have a robust performance in case of significant SV due to the producing a fixed size of weight mask via a generative network. The LSART utilizes the modified Kernelized ridge regression (KRR) by the weighted combination of patch-wise similarities to concentrate on reliable regions of the target. Due to the consideration of rotation information and online adaptation of CNN models, this method provides promising responses to tackle the DEF and IPR challenges.

The DeepSTRCF, ASRCF, DRT, and UPDT are the DCF-based methods that not only exploit deep off-the-shelf features but also fuse them with shallow ones (e.g., HOG and CN) to improve the robustness of visual tracking (see Table 1). To reduce the adverse impact of the OCC and OV attributes, the DeepSTRCF adds a temporal regularization term to the spatially regularized DCF formulation. The revisited formulation helps the DeepSTRCF to not only endure some appearance variations such as the IV, IPR, OPR, and POC. Using object-aware spatial regularization and reliability terms, the ASRCF and DRT methods attempt to optimize models to learn more adaptive correlation filters effectively. Both these methods have studied major imperfections of DCF-based methods such as circular shifted
sampling process, same feature space of localization and scale estimation processes, the strict focus on discrimination, and sparse and non-uniform distribution of correlation responses. Hence, these methods handle the DEF, BC, and SV, suitably. Finally, the UPDT focuses on enhancing the visual tracking robustness through independently training a shallow feature-based DCF and a deep off-the-shelf feature-based DCF and considering augmented training samples with an adaptive fusion model. Although these methods demonstrate the competitive performance of well-designed DCF-based trackers compared to more sophisticated trackers, they suffer from the limitations of pre-trained models (e.g., the computational complexity of deep features), aspect ratio variation, model degradation, and considerable appearance variation.

To qualitatively compare the performance of the best methods, Fig. 8 shows the tracking results of the SiamRPN++ [55], SiamMask [57], C-RPN [157], SiamDW-SiamRPN [56], ECO [94], LSART [127], DRT [126], UPDT [109], and DeepSTRCF [122] on different video sequences of the VOT2018 dataset. According to these results, all these methods have failed in challenging scenarios that consist of multiple critical visual attributes. For instance, the SiamMask misuses the semi-supervised video object segmentation when the OCC and SV co-occur, or the significant SV dramatically reduces the performance of the SiamRPN++. Despite considerable advances that are emerged in visual tracking, the state-of-the-art visual tracking methods are still unable to handle the real-world challenges; severe variations of target appearance, MOC, OCC, SV, CM, DEF, and even IV can not only have severe effects on their performance but also lead to the failure. These results demonstrate that the state-of-the-art methods are still not reliable for real-world application for the main reason of lacking the intelligence for scene understanding. Although visual tracking methods improve the ability of object-scene distinction, the DL-based visual trackers still cannot infer scene information, immediately recognize the global/configural structure of a scene or organize purposeful decisions based on space and acts within.

5 CONCLUSION AND FUTURE DIRECTIONS

The state-of-the-art DL-based visual tracking methods were categorized into a broad taxonomy based on network archi-
tectures, network exploitation, network training, network objective, network output, and the exploitation of correlation filter advantages. Moreover, the motivations and contributions of these methods were categorized according to the main problems and solutions that different methods have proposed. Furthermore, not only the recent visual tracking benchmark datasets and evaluation metrics were briefly investigated, but also the state-of-the-art visual tracking methods were compared in terms of various evaluation metrics on the OTB2013, OTB2015, VOT2018, and LaSOT benchmark datasets.

Recently, the DL-based visual tracking methods have investigated different exploitation of deep off-shelf features, fusion of deep features and hand-crafted features, various architectures and backbone networks, offline and online training of DNNs on large-scale datasets, update schemes, search strategies, contextual information, temporal information, and how to deal with lacking training data. However, many other problems need to be explored in the future. The current main concentration is to design custom neural networks for visual tracking, such that they may provide robustness, accuracy, and efficiency simultaneously. These trackers are primarily developed by integrating efficient network architectures with both classification and regression branches that not only are trained on large-scale datasets, but also are learned a robust representation against the most challenging visual attributes (e.g., OCC, OV, FM, IV, and LR). By analyzing different methods, the main problem is their deficiency is in scene understanding. The state-of-the-art visual tracking methods still cannot interpret dynamic scenes in a meaningful way, immediately recognize global structures, infer existing objects, and perceive basic level categories of different objects or events. Despite all efforts to devise sophisticated methods and time-consumed training procedures, these methods quickly lose the generic targets in real-world scenarios. For example, these methods still have the problems to simultaneously handle significant attributes such as OCC, DEF, SV, and FM. Although the SNN-based methods desirably reduce the computational complexity of visual trackers, these architectures can be modified to not only employ complementary features (e.g., temporal information) but also incorporate with the recent RL and adversarial learning. Finally, online training of DNNs can lead to a well adaption of the network filters to significant variations of target appearance. However, it requires more efficient strategies to reduce the computational complexity that is a necessity in real-time applications.

ACKNOWLEDGMENTS

We wish to thank Prof. Kamal Nasrollahi (Visual Analysis of People Lab (VAP), Aalborg University) for his beneficial comments.

REFERENCES

[1] M.-f. Chang, J. Lambert, P. Sangkloy, J. Singh, B. Slawomir, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3D tracking and forecasting with rich maps,” in Proc. IEEE CVPR, 2019, pp. 8746–8757.
[2] W. Luo, B. Yang, and R. Urtasun, “Fast and furious: Real-time end-to-end 3D detection, tracking and motion forecasting with a single convolutional network,” in Proc. IEEE CVPR, 2018, pp. 3569–3577.
[3] P. Girão, A. Asvadi, P. Peixoto, and U. Nunes, “3D object tracking in driving environment: A short review and a benchmark dataset,” in Proc. IEEE ITSVC, 2016, pp. 7–12.
[4] C. Li, X. Liang, Y. Lu, N. Zhao, and J. Tang, “RGB-T object tracking: Benchmark and baseline,” Pattern Recognit., vol. 96, 2019.
[5] H. V. Hoof, T. V. D. Zant, and M. Wiering, “Adaptive visual face tracking for an autonomous robot,” in Proc. Belgian/Netherlands Artificial Intelligence Conference, 2011.
[6] C. Robin and S. Lacroix, “Multi-robot target detection and tracking: Taxonomy and survey,” Autonomous Robots, vol. 40, no. 4, pp. 729–760, 2016.
[7] B. Risse, M. Mangan, B. Webb, and L. Del Pero, “Visual tracking of small animals in cluttered natural environments using a freely moving camera,” in Proc. IEEE ICCV, 2018, pp. 2840–2849.
[8] Y. Luo, D. Yin, A. Wang, and W. Wu, “Pedestrian tracking in surveillance video based on modified CNN,” Multimedia Tools Appl., vol. 77, no. 18, pp. 24041–24058, 2018.
[9] A. Brunetti, D. Buongiorno, G. F. Totta, and V. Bevilacqua, “Computer vision and deep learning techniques for pedestrian detection and tracking: A survey,” Neurocomputing, vol. 300, pp. 17–33, 2018.
[10] L. Hou, W. Wan, J. N. Hwang, R. Muhammad, M. Yang, and K. Han, “Human tracking over camera networks: A review,” EURASIP Journal on Advances in Signal Processing, vol. 2017, no. 1, 2017.
[11] G. Klein, “Visual tracking for augmented reality,” PhD Thesis, pp. 1–182, 2006.
[12] M. Klepschitz, G. Schall, D. Schmalstieg, and G. Reitmayr, “Visual tracking for augmented reality,” in Proc. IPIN, 2010, pp. 1–4.
[13] F. Ababsa, M. Maïd, J. Y. Didier, and M. Mallem, “Vision-based tracking for mobile augmented reality,” in Studies in Computational Intelligence. Springer, 2008, vol. 120, pp. 297–326.
[14] J. Hao, Y. Zhou, G. Zhang, Q. Lv, and Q. Wu, “A review of target tracking algorithm based on UAV,” in Proc. IEEE CCB, 2019, pp. 328–333.
[15] M. Manafífard, H. Ebadi, and H. Abrishami Moghaddam, “A survey on player tracking in soccer videos,” Comput. Vis. Image Und., vol. 159, pp. 19–46, 2017.
[16] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, “Vision-based and marker-less surgical tool detection and tracking: A review of the literature,” Medical Image Analysis, vol. 35, pp. 633–654, 2017.
[17] V. Ulman, M. Maïka, and et al., “An objective comparison of cell-tracking algorithms,” Nature Methods, vol. 14, no. 12, pp. 1141–1152, 2017.
[18] T. Fei, H. Mao, J. Guo, and Z. Yi, “Cell tracking using deep neural networks with multi-task learning,” Image Vision Comput., vol. 60, pp. 142–153, 2017.
[19] D. E. Hernandez, S. W. Chen, E. E. Hunter, E. B. Steager, and V. Kumar, “Cell tracking with deep learning and the Viterbi algorithm,” in Proc. MARS, 2018, pp. 1–6.
[20] J. Luo, Y. Han, and L. Fan, “Underwater acoustic target tracking: A review,” Sensors, vol. 18, no. 1, p. 112, 2018.
[21] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in Proc. IEEE CVPR, 2010, pp. 2544–2550.
[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, 2015.
[23] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, “Discriminative Scale Space Tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 8, pp. 1561–1575, 2017.
[24] S. M. Marvasti-Zadeh, H. Ghanei-Yakhdan, and S. Kasaei, “Rotation-aware discriminative scale space tracking,” in Iranian Conf. Electrical Engineering (ICEE), 2019, pp. 1272–1276.
[25] G. Boudoukh, I. Leichter, and E. Rivlin, “Visual tracking of object silhouettes,” in Proc. ICIP, 2009, pp. 3625–3628.
[26] C. Xiao and A. Yilmaz, “Efficient tracking with distinctive target colors and silhouette,” in Proc. ICIP, 2016, pp. 2728–2733.
[27] V. Bruni and D. Vitulano, “An improvement of kernel-based object tracking based on human perception,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 44, no. 11, pp. 1474–1485, 2014.
[28] W. Chen, B. Niu, H. Gu, and X. Zhang, “A novel strategy for kernel-based small target tracking against varying illumination with multiple features fusion,” in Proc. ICICT, 2018, pp. 135–138.
