Journal of Big Data

METHODOLOGY Open Access

Robust visual tracking using very deep generative model

Eman R. AlBasiouny[1,2*], Abdel-Fattah Attia[1], Hossam E. Abdelmunim[2] and Hazem M. Abbas[2]

*Correspondence:
eng_eman_2008@eng.kfs.edu.eg

[1] Computer Science and Engineering Department Faculty of Engineering, Kafrelsheikh University, Kafrelsheikh, Egypt
[2] Computer and Systems Engineering Department Faculty of Engineering, Ain Shams University, Cairo, Egypt

Abstract

Deep learning algorithms provide visual tracking robustness at an unprecedented level, but realizing an acceptable performance is still challenging because of the natural continuous changes in the features of foreground and background objects over videos. One of the factors that most affects the robustness of tracking algorithms is the choice of network architecture parameters, especially the depth. A robust visual tracking model using a very deep generator (RTDG) was proposed in this study. We constructed our model on an ordinary convolutional neural network (CNN), which consists of feature extraction and binary classifier networks. We integrated a generative adversarial network (GAN) into the CNN to enhance the tracking results through an adversarial learning process performed during the training phase. We used the discriminator as a classifier and the generator as a store that produces unlabeled feature-level data with different appearances by applying masks to the extracted features. In this study, we investigated the role of increasing the number of fully connected (FC) layers in adversarial generative networks and their impact on robustness. We used a very deep FC network with 22 layers as a high-performance generator for the first time. This generator is used via adversarial learning to augment the positive samples to reduce the gap between the hungry deep learning algorithm and the available training data to achieve robust visual tracking. The experiments showed that the proposed framework performed well against state-of-the-art trackers on OTB-100, VOT2019, LaSOT and UAVDT benchmark datasets.

Keywords: Deep learning, Generative adversarial network, Fully connected layers, Visual tracking

Introduction

Visual tracking is the process of locating a sequence of locations of a target object in each frame of a video, given its position in the first frame only. Visual tracking is one of the most attractive research areas in the computer vision field because it is applied to videos instead of fixed images, and also it has widespread use in different applications, like self-driving cars [1], surveillance and security [2], handwritten recognition [3], surgery [4], and augmented reality [5], to name a few.

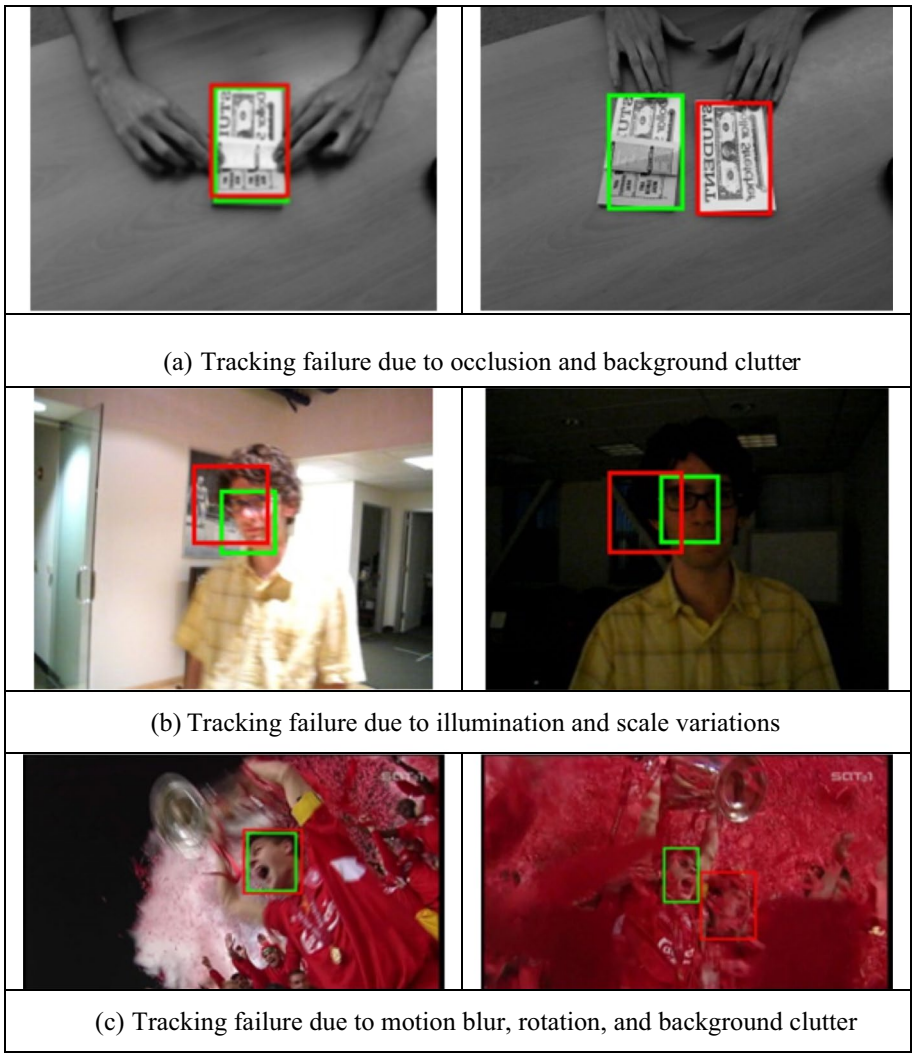
The only way to achieve the most robust visual tracking results is to win the race to extract the features that precisely describe the target object. Over many years, machine learning (ML) algorithms have been used to extract different types of features such as

color, texture, shape, thermal, and audio. Researchers have mainly concentrated on integrating two or more ML algorithms to cover most features and achieve good tracking results [6]. In the last few years, researchers have shifted their procedure to deep learning (DL) algorithms, which have achieved remarkable performance compared to ML algorithms. There are three main reasons why DL approaches are more robust than ML approaches are. First, deep features are extracted automatically through the layers of the deep neural network; usually, spatial features are extracted in the lower layers and semantic features are extracted in the higher layers. This is unlike handcrafted features, which need to be collected from different decisions from different integrated algorithms. Second, deep features can present the multi-level characteristics of the target. Therefore, they are more invariant to diverse appearance variations, which are considered the most significant challenge the tracking algorithms suffer. Third, deep features are more discriminative because they can capture high-level dynamic information than handcrafted features, which can only capture low-level static information.

Problem and motivation

Although DL-based tracking approaches have achieved significant success in recent years, they face two significant challenges. The first challenge is the dynamic nature of the environment in visual tracking applications. The appearance of both the foreground and background objects continuously changes throughout the video. The tracker's primary purpose is to extract the features of the target object and hence recognize the object and localize it. The dynamic nature of videos makes the features partially different from one frame to another. Therefore, the visual tracker must follow different feature maps that belong to the same object, depending on the standard one given in the first frame. As shown in Fig. 1, various appearance variations significantly affect tracking robustness. If we extracted the feature maps of the images on the left and right sides, they would mostly be different. Our mission is to discover whether the two different feature maps belong to the same object.

The second challenge faced by DL tracking approaches is a lack of training data. Although DL approaches are more robust than ML approaches, DL algorithms are more data hungry. They require a large amount of training data to realize their expected roles. One of the most well-known architectures exploited by DL tracking methods is CNN [7, 8]. Despite being extensively used in several DL tracking algorithms, CNNs have not achieved in visual tracking the same success in image classification [9]. CNN tracking algorithms have only one source of data, positive samples in the first video frames. Therefore, this sample in each video does not contain sufficient data for the hungry DL tracking algorithms. One of the most successful CNN structures attempts is MDNet [10], which has achieved state-of-the-art performance in most tracking benchmark studies. It presents a multi-domain convolutional neural network consisting of five shared layers in the first stage and one domain-specific layer in the second stage. However, two problems have not yet been resolved. First, it faces the problem of model degradation caused by online sample updates, which the MDNet algorithm exploits during the classification stage. These updates are noisy and unreliable for robust tracking purposes. Second, the training samples are not sufficiently diverse to face the challenge of changing the appearances of the sample



(a) Tracking failure due to occlusion and background clutter

(b) Tracking failure due to illumination and scale variations

(c) Tracking failure due to motion blur, rotation, and background clutter

Fig. 1 Three examples show the bad effect of different appearance variations on the tracking results

during the tracking operation, such as occlusion and object deformation. This lack of data diversity severely degrades the performance of visual tracking algorithms.

Our RTDG model uses generative adversarial learning to solve this lack of data diversity by producing feature maps that contain the most robust and long-lasting properties of the tracked object. Unlike the discriminator networks that just differentiate between the synthesized and the real data distributions, the generators should represent all the information inside the scene to approximate the target distribution. Because the spaces of deep representations are more informative than those of pixels to capture semantic aspects of images, we used a very deep generator model (22 layers). This deep architecture makes the generator network more able to identify the most discriminative features required to achieve hopeful tracking robustness. The proposed model realized a competitive performance against the state-of-the-art trackers on three benchmark datasets.

Related work

In many comprehensive surveys, visual tracking is one of the most computer vision topics extensively studied in the last decade [11–13]. This section first reviews some deep visual tracking approaches in the literature and discusses visual tracking methods that use generative adversarial networks (GANs).

Deep visual tracking

According to the network structure, DL visual trackers are categorized into two main types of model: generative (or one-stage regression) and discriminative (or two-stage detection) methods. The generative models consider only the information about the target object, whereas the discriminative models consider the information about both the foreground objects and their background.

The most well-known generative deep trackers are correlation filter (CF)-based networks and Siamese neural networks. In [14], the authors applied trained correlation filters in each convolutional layer in a CNN to encode the appearances of the tracked object. They then used the maximum response in each layer to locate an object. Danelljan et al. [15] used a new formulation to overcome the restriction of using discriminative convolution filters with single-resolution feature maps by integrating multi-resolution feature maps for more robust tracking results. Galoogahi et al. [16] introduced a computationally efficient and robust visual tracker by exploiting a correlation filter that is aware of foreground and background objects using handcrafted features. Spatial temporal regularized correlation filter (STRCF) can handle the unwanted boundary effects by integrating temporal and spatial regularization [17]. Li et al. presented a dual-regression framework that fuses a discriminative fully convolutional module and fine-grained correlation filter component to realize robust and accurate visual tracking results [18]. After the widespread use of correlation filters, Siamese neural networks have become the focus of generative tracking approaches in recent years owing to their high performance and efficiency. Some Siamese based networks exploit the output of two parallel networks to indicate the location of an object [19, 20]. In [21], the authors integrated correlation filters and a Siamese network to propose a self-supervised learning-based visual tracker. They applied a multi-cycle consistency loss as self-supervised information to learn the feature extractor from adjacent video frames. Li. B. et al. proposed the SIAMRPN++model, which uses a very deep neural network, ResNet [22]. In [23], the authors introduced a novel noise-aware (NA) window customized for visual tracking and used the particle filter to improve the signal-to-noise ratio (SNR) of windowed region of interest (ROIs). In [24], they exploited deep convolutional features with a small number of particles in a novel hierarchical particle filter, which formulates correlation filters as observation models and decomposes the standard particle filter framework into two constituent particle layers. In [25], the authors presented a multi-level similarity model, one for the global semantic similarity and the other for the local structural similarity of the thermal infrared object. This model was based on the Siamese framework. In [26–28], the authors presented three thermal infrared (TIR) tracking methods which treat the tracking problem as a classification task. Fan et al. integrated alignment and aggregation modules into a Siamese-based network [29]. The feature-alignment module

calibrates the search region to handle severe pose variations. A shallow-level and high-level aggregation module was developed to handle the severe appearance variations of an object.

In contrast, tracking-by-detection methods discriminate the boundaries of the target object from the background after excluding negative candidate samples. A CNN is one of the main structures representing this type of deep tracker. The general structure of a CNN comprises two parts: feature extraction and classification. The output of the feature extraction stage is a two-dimensional (2D) plane, called a feature map. It contains features that the network can extract to represent the target object in the input image. This feature map is fed into the classifier to generate a score for each candidate sample, and finally indicates the target object. Shunli et al. integrated a fuzzy least-squares support vector machine (SVM) with metric learning to improve the adaptation of an appearance model to different video sequences [30]. A convolutional network without training (CNT) tracker [31] is an adaptive algorithm that uses a particle filter framework to adapt to the appearance variation during the tracking process. Hong et al. constructed a target-specific saliency map using a CNN pre-trained on a large-scale repository with SVM guidance [32]. They proved their method's effectiveness based on a classification dataset, which is unreliable for the tracking task. One of the most successful networks used for visual tracking is the MDNet [10]. It consisted of five shared layers (three convolutional layers and two fully connected (FC) layers) and one domain-specific FC layer. The shared layers extract general representation features, and the domain-specific layer is responsible for identifying a particular target in a specific domain. In [33], the authors presented a model-free tracking system that can automatically locate many objects with the same spatial and motion structure, and update the structure without previous acknowledgement. Yang, Y. et al. suggested enhancing tracking accuracy through online training [34]. On the one hand, duplicated training data were compressed by examining the dataset's distribution in low-level feature space. In contrast, they developed statistically-based losses to enhance inter-class distance while minimizing intra-class variation for high-level semantic characteristics. In [35], they developed an attribute-based CNN with numerous branches, each of which is responsible for classifying the target according to a particular attribute. In [36], they suggested adaptively employing the level set segmentation and bounding box regression techniques to achieve a tight enclosing box, and designing a CNN to determine if the target is occluded. Recent attempts [37–41] have been made to realize higher performance of object tracking by using new schemes.

Despite the strengths of CNNs and their wide use in computer vision applications, there is still a large gap between the amount of labeled data required by the tracking frameworks and the amount of training data used by CNNs. CNNs mainly suffer from two problems: the high spatial overlap between the positive candidate samples and the lack of diversity in the training data, which is required for the different appearance variations that occur in the object during the video. One approach that was recently used to provide the required amount and diversity of data is adversarial learning.

Deep visual tracking using GANs

Generative adversarial networks (GANs) were introduced by Goodfellow et al. [42] in 2014. The most remarkable point with this network is that it is not consistent with the

existing amount and diversity of real data. In contrast, it attempts to mimic real-world data distribution and generate similar fake samples. This network consists of two subnetworks: a generator and discriminator. The generator learns to produce synthetic instances by mapping from the latent space to a particular distribution that belongs to real data. The adversarial discriminator distinguishes between real and fake data, and sends feedback to the generator to generate more realistic samples. The promising performance of GANs has encouraged researchers to propose different improvements, such as WGAN [43], DCGAN [44], StarGAN [45], cGAN [46], StyleGAN [47], cycleGAN [48], and many other models in the GAN family. They have been widely utilized in recent years to achieve better performance in different computer vision applications, such as object detection [49, 49], image-to-image translation [51], super-resolution [52], and object tracking [53–57].

GANs are not familiar with visual tracking because visual tracking is a supervised learning algorithm that uses labeled data, whereas GANs are unsupervised learning algorithms that use unlabeled data. Although their use with visual tracking is not widespread, GANs are used as CNN assistants to achieve high-performance results. The VITAL network [53] utilized adversarial learning via a CNN to augment the positive samples in the feature space by generating a wide variety of appearances of the same mode over a temporal span. However, the architecture of their generator had only two layers which was insufficient to cover all the details of the input images. Therefore, we averted this drawback by replacing their architecture with a very deep one (22 layers). Also, we changed number of neurons in each layer and used the LeakyReLU activation function instead of ReLU to take into account both positive and negative weights. The SINT + + framework [54] proposed a massive amount of deformation in hard positive samples, and the results were then optimized by deep reinforcement learning. Zhao et al. [55] introduced a framework for both regression and classification. They used a fully convolutional SNN for regression and discriminative classifier for classification. They then used adversarial learning to optimize both results. Han et al. [56] utilized GANs once in the sample space, which carries a diversity of deformation and motion blur, and once in the feature space that uses occlusion masks. Yin et al. [57] integrated GANs into a tracking-by-detection network to enrich the extracted convolutional samples to capture a variety of object appearances.

Contributions

The main task of visual tracking is to search for features that are most similar to the ground-truth features in each frame in the video. In this study, we introduce a novel network that integrates GAN with a tracking CNN model to extract a particular distribution of unlabeled data via adversarial learning to produce further synthesized training data to reduce the overfitting effect on the tracking results. We discuss an important aspect related to the architecture of generative models in GANs, namely, its depth. We fixed all parameters related to the architecture and increased the number of fully connected (FC) layers to determine their impact on robustness. After several trials, we found that 22 FC layers were the best architecture for the generator to obtain the best training results. Using this strong generator, we produced masks applied to the input feature maps to obtain other versions of the feature maps containing more diversified

appearances. Subsequently, our model chooses a mask that helps to include the most robust features that last for a long time. This training scheme makes our framework robust to expected appearance circumstances. We evaluated the proposed framework on OTB-100, VOT2019, LaSOT and UAVDT benchmark datasets and found that it performed well against state-of-the-art trackers.

Paper organization

The rest of the paper is organized as follows. "Deep fully connected network architecture" describes the properties of the FC network structure and the effect of the size and depth of layers on the network's robustness. "Methodology" introduces the proposed method during both the training and the tracking phases. The implementation and validation of the proposed method based on two state-of-the-art benchmark datasets are presented in "Experimental results and validation". Finally, conclusions and suggested future search trends are presented in "Conclusions and future work".

Deep fully connected network architecture

A fully connected network [58] is a feed-forward neural network architecture in which all nodes in each layer are connected to all nodes in the adjacent layers. It consists of three types of layers: input, hidden, and output. The input layer receives the input image to be processed and the output layer is responsible for the classification task. The hidden layers are the real computational engines in the FC network, and most of the processing is performed through them. FCs were designed to solve non-linear mapping problems. Therefore, each node in the network performs a nonlinear activation function. In our case, we used the LeakyReLU activation function for all hidden nodes, except the last one.

The hidden layers did not have specific sizes or depths. Their sizes depended on the tasks performed. No theory yet indicates how many layers or nodes are required to perform a certain task. Traditionally, the most common method for selecting the hyperparameters of the hidden layers is based on trial and error. However, choosing a suitable depth is one of the most critical factors that lead to high-performance deep networks, as proven in [59–61]. Our architecture achieved the best robustness with 22 hidden layers, which is considered a very deep fully connected neural network. It is common to use very deep convolutional networks to enhance classification performance, but we are using for the first time a very deep, fully connected network to improve adversarial learning performance. [62, 63] represented a detailed visualization and understanding of what happened in the neurons of the deep generator networks. Their study showed that the neurons of the first layers in the generator had information about the small parts which composed the objects. The neurons of the middle layers had semantic information about the objects in the scene. The later layers had low-level information about the materials, textures and colors of the items in the scene. Therefore, the very deep generator could be more robust in generating the semantic output with more low-level details.

During the training phase, the hidden layers perform the primary role of generalizing features learned from the input features. In addition, they can later recognize any feature belonging to a feature class that is generalized from the input features. Therefore, we used the FC network as the generator model, which is responsible for generating masks

during the adversarial learning process in our model. Traditionally, some of the existing generator architectures use convolutional layers only and eliminate or minimize fully connected layers as in the DCGAN [44]. Typically, conventional networks use one fully connected layer in the generator to receive input noise and one layer in the discriminator to map the extracted features to a lower-dimensional space for classification. Unlike convolutional layers that extract spatial features, FC layers can extract general information that lasts for a long time span [64]. In addition, FC layers can generate subtle variations in the input features in different spatial zones which can cover the entire image because their mapping is non-spatial. These properties make it better to generate different variations of the target object in the feature space, thereby augmenting the training data. Thus, it is considered an essential step to capture the diverse appearances that occur through the video, and as a result, achieve more robust tracking.

Methodology

As discussed in Sec. I: GANs are unfamiliar with visual tracking. Therefore, it is always used as an assistant in CNN frameworks to achieve better results. In our framework (RTDG), we used the VGG-M network [61] as the backbone CNN network. A generator network was inserted between the feature extraction and classification stages. The generator augments the input feature maps by producing masks that represent different variations in appearance via adversarial learning. The classifier then distinguishes between the discriminative features in individual frames and the features that last for a long time. Our framework comprises three main stages during the training process: feature extraction, adversarial feature generation, and binary classification.

Feature extraction

Feature extraction in visual tracking tasks acts as the first gate through which the algorithm's success passes. If the extracted features fail to describe the tracked object properly, the next steps in the tracking framework cannot compensate for this failure. Feature extraction converts raw data into a numerical form, which machine learning and deep learning algorithms can deal with. Thus, it extracts a group of pixels with similar spectral, spatial, or textural attributes. The architecture of the feature extractor in our framework is similar to that used in a VGG-M network [12], as shown in Fig. 2. We used only the first three convolutional layers in the VGG-M model with internal pooling layers-similar to the feature extractor in the MDNet model [10]. We used different sizes of feature maps that were adjusted using our 107×107 input image, as shown in Fig. 2. The convolutional layers are equipped with a rectification (ReLU) function with filter sizes of 7, 5, and 3 for Conv.1, Conv. 2, and Conv.3 layers, respectively. The max-pooling layers were performed over 3×3 pixel filters with a stride of 2. Each layer worked with 256 positive and negative samples that were used for better adversarial learning results. The output of the feature extraction stage was a 3×3 feature map with 512 channels. Map C is then flattened to 4608 elements, which will be the input to the following core stage: the adversarial learning. The following two subsections explain the adversarial learning stage during the training phase.

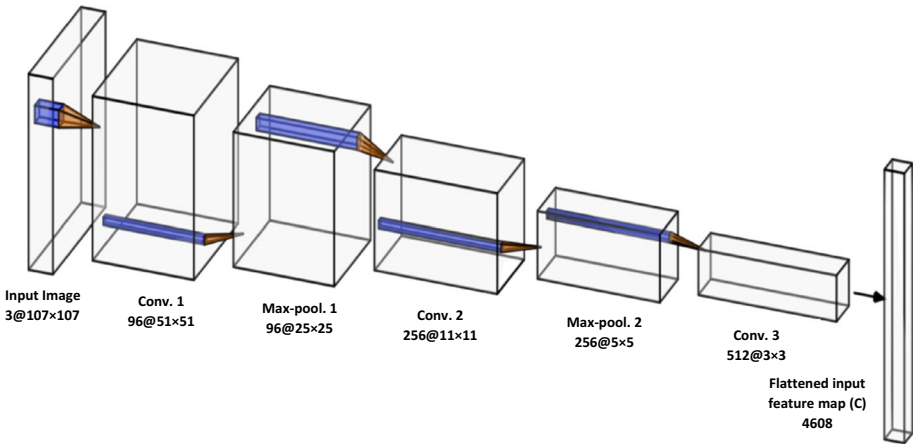


Fig. 2 The architecture of the feature extraction stage

Adversarial feature generation

Feature map (C) generated by the pre-trained feature extractor mentioned in the previous subsection contains both robust and discriminative features. Hence, the natural question is what the difference between robust and discriminative features is. Robust features are general features that describe an object over a long temporal span. These are the main features that do not change from frame to frame. However, discriminative features are specific features that describe the object precisely only in individual situations, and they do not last for more frames. They are related to particular conditions of the object owing to some appearance variations. Overfitting would occur if the classifier depended on them. Most tracking-by-detection algorithms cannot extract robust features and leave discriminative features. The adversarial learning procedure can collect the most robust features by augmenting positive samples by generating diverse input variations in the feature space. Using these augmented feature maps makes the algorithm more robust to variations in appearance during tracking.

We placed the adversarial feature generator network (G) between the feature extraction and classification networks, as shown in Fig. 3. The G network uses the feature map C of the first frame of the video as an input and generates nine masks (G(C) or M*). All masks had the same size as the input feature map (3×3) with only one channel, and each mask represented one of the appearance variations. The nine different masks cover almost all expected appearance variations. The mask is split into nine equal parts, where only one part is assigned 1, and the others are assigned 0 s in turn. This operation was performed to ensure that the generated masks were applied to the diversified versions of each input feature. The weights of these masks were randomly initialized and then gradually updated during training. The generated masks (M*) were applied to the extracted features (C) to create a C^O feature map. (C^O) is defined in (1), and this operation is defined as the dropout operation.

$$\mathrm{C}^{O}_{ijk} = \mathrm{C}_{ijk} M^{*}_{ij} \tag{1}$$

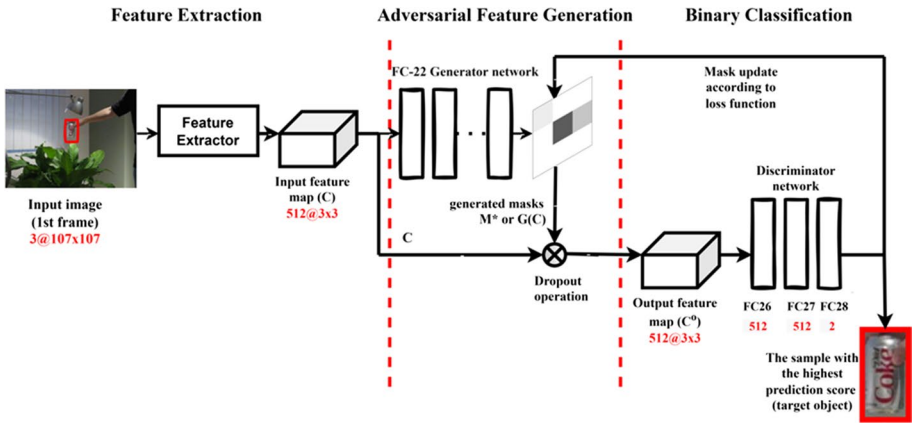


Fig. 3 The architecture of RTDG tracking framework which uses 22-FC generative model in the training

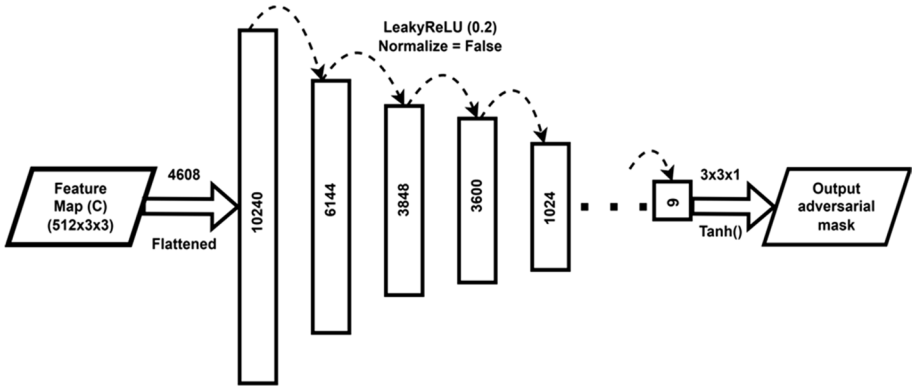


Fig. 4 The architecture of the FC-22 (RTDG) generator network

where (C^{O}_{ijk}) is the feature (C) after the dropout operation with the corresponding generated mask (M*) of the element (i,j). Mask M* has 3×3 values, zeros, and ones after thresholding. M^{*}_{ij} represents the value at the i-th row and j-th column in the mask, and C_{ijk} represents the value at locations i and j in channel k in the input feature map. C^{O}_{ijk} contains only features in C_{ijk} whose corresponding values are ones in M^{*}_{ij}. This operation reduces the weights of the most discriminative features, which is a known method for solving the regularization problem and for reducing overfitting. (C^{O}) can be considered a modified feature map that is passed to the classifier or discriminative network (D). During the training of G, a mask is gradually recognized, which reduces the performance of the classifier. G is optimized by using the mean square error loss (MSE) to measure the difference between the estimated map (positive and negative samples of images based on the output probability of the generator) and the ground truth map (positive and negative samples of images based on the output probability of the discriminator).

Traditionally, very deep convolutional networks have been used in deep learning algorithms to achieve more robust results in classification and tracking tasks. The VITAL model uses only two fully connected layers as the generator network. In our model, we used 22 FC layers in the generator, as illustrated in Fig. 4. The architecture of our

generator consists of 22 FC hidden layers equipped with a LeakyReLU activation function with a slope of 0.2, except for the final layer, which uses the Tanh function. We used the trial-and-error method to obtain the best number of layers that yield the best tracking robustness. The sizes of the 22 layers are as following: 10240, 6144, 3848, 3600, 3352, 3104, 2856, 2608, 2360, 2112, 1864, 1616, 1368, 1120, 872, 624, 376, 128, 256, 512, 1024, and 9. We used stochastic gradient descent (SGD) as the optimizer with the generator. Batch normalization was not used because it did not improve the results.

Binary classification

GANs use semi-supervised learning, which means that they can extract structures from unlabeled data to augment data sets with additional training data to regularize the classifier. In traditional adversarial learning, G uses a random noise vector z from an easy-to-sample distribution $P_{noise}(z)$ as an input and outputs an image G(z). The discriminator D takes either a real image x with a distribution $P_{data}(x)$ or G(z) as an input and outputs the classification probability to detect whether it is a real or fake image. Specifically, the loss function of the GAN calculates the similarity between the generated data distribution p_{noise} and the real sample distribution p_{data} using the Jensen-Shannon (JS) divergence equation as follows in (2):

$$\min_{G} \max_{D} L(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_{noise}(z)}[\log(1 - \mathrm{D}(\mathrm{G}(z)))] \tag{2}$$

This function is based on cross-entropy loss theory. D and G were trained synchronously. In the proposed model, D is trained once, and G is trained based on the results of training D. In each iteration, the feature map C is entered into G to generate mask M*, which is used in a dropout operation. The result of this operation is then entered into D, which uses this unlabeled data as additional training data to be invariant to appearance variations. The architecture of the discriminator in our model consisted of three fully connected layers, as shown in Fig. 3.

D measured the prediction scores for positive samples (target object) and negative samples (background) using (C^O). The prediction score was used to calculate the loss that was derived from the objective function in (2). This loss was optimized to minimize using a stochastic gradient descent (SGD) optimizer. This loss is described in (3).

$$\begin{aligned} L = \min_{G} \max_{D} & E_{(C,M)\sim P_{(C,M)}}[logD(M.C)] \\ & + E_{C\sim P_{(C)}}[log(1 - D(G(C).C))] \\ & + \lambda E_{(C,M)\sim P_{(C,M)}} \|G(C) - M\|^2 \end{aligned} \tag{3}$$

The discriminator is implemented by optimizing the maximum $E_{(C,M)\sim P_{(C,M)}}[\log D(M.C)]$ and $E_{C\sim P_{(C)}}[\log(1 - D(G(C).C))]$. On the other hand, G is trained to maximize the probability that D produces a fake example; consequently, to minimize the two terms, $E_{C\sim P_{(C)}}[\log(1 - D(G(C).C))]$ and $\lambda E_{(C,M)\sim P_{(C,M)}} \|G(C) - M\|^2$. Once the generator is trained to its optimum value ($p_g = p_r$), the discriminator's loss reaches the highest value. In this case, the corresponding mask is the actual mask, M, which is used to update M in (3).

We compute the cost-sensitive loss (4) to reduce the effect of a large number of easy negative samples and reformulate our final loss as in (5). This final loss function receives the discriminator output prediction score as the input and computes the loss value that is optimized for the lowest. Based on the cost-sensitive loss equation, which is based on entropy loss,

$$L(p, y) = -\big(y.(1-p).log(p) + (1-y).p.log(1-p)\big) \tag{4}$$

we reformulate the loss function in (3) as:

$$\begin{aligned} L = \underset{G}{min}\ \underset{D}{max}\ & E_{(C,M)\sim P_{(C,M)}}[K_1.logD(M.C)] \\ & + E_{C\sim P_{(C)}}[K_2.log(1 - D(G(C).C))] \\ & + \lambda E_{(C,M)\sim P_{(C,M)}} \|G(C) - M\|^2 \end{aligned} \tag{5}$$

where $K_1 = 1 - D(M.C)$ and $K_2 = D(G(C).C)$ are the modulating factors that balance training sample loss.

Online tracking

After the training phase was completed, the test phase was started. In this phase, the generator network is eliminated, and the model becomes an ordinary tracking-by-detection model consisting of feature extraction and classification networks. In the training stage, we pre-trained the network offline based on the positive and negative samples used in the MDnet model [10]. The training stage is applied only to the first frame of the input sequence. Throughout the training period, hard-negative samples were excluded.

In the tracking scheme, every time a new frame arrives, positive and negative sample candidates are generated around the previous target position and fed into the binary classifier to generate a prediction score for each of them. Finally, the candidate with the highest prediction score is chosen as the new state of the target. We continuously update our model with a short-term tracking update (when a failure occurs) and a regular long-term tracking update (every 10 frames).

Experimental results and validation

This section presents an empirical analysis of the impact of increasing the number of FC layers in the generator network. We study how this impact increases the robustness of a tracking-by-detection network augmented with a GAN via adversarial learning. We compared the proposed framework with state-of-the-art frameworks, such as VITAL [53] and MDNet [10], which are two of the top frameworks in the survey paper published in 2021 [12]. The VITAL algorithm uses a GAN model similar to ours to augment the positive samples with synthesized ones except that they use only two fully connected layers generators. In our experiment, we tried different architectures of the generator to show the impact of using very deep FC layers compared with non-deep networks. Therefore, we studied the robustness of the tracker based on the following architectures for the generator: two FC layers, as in the VITAL network (VITAL), five FC layers (FC-5),

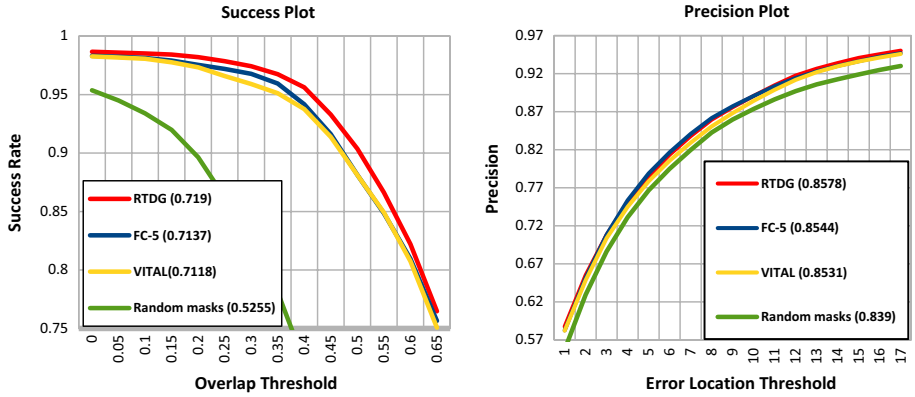


Fig. 5 Ablation Study. Precision and success plots on the OTB-2015 dataset using OPE evaluation

and 22 FC layers (RTDG). The learning rates for training the generator and discriminator networks in all the models were 0.2×10^{-3} and 0.5×10^{-3}, respectively.

Hardware

All implementations were written in PyTorch and run on a Fluidstack-based Linux cloud server. The cloud server had the following specifications: Ubuntu 18.04.5 LT, RAM: 114 GB, GPU: Nvidia RTX 2080, CPU: Intel(R) Xeon(R) Silver 4208 CPU @ 2.10G.

For more effective experimentation, we show the results of three of the most common tracking datasets: OTB-100 [65], VOT2019 [66], LaSOT [67] and UAVDT [68].

OTB evaluation

The OTB-100 dataset [65] contained approximately 100 videos. For supervised learning, the videos were labeled with bounding box annotations. This dataset includes illumination variations, low resolution, scale variation, fast motion, background clutter, deformation, occlusion, out-of-view, motion blur, in-plane rotation, and out-of-plane rotation, among other appearance variations that challenge visual tracking applications (11 attributes). The training samples in the OTB dataset are the first frames in each of the 100 sequences, as is a typical practice. As a result, we obtained 100 photos for the training set and augmented them to be more robust via an adversarial learning process. All the remaining frames from the 100 sequences were included in the test set.

Ablation study

In RTDG, we used 22 FC layers in a generator via adversarial learning to produce synthesized positive samples that help the tracking model to be more robust to appearance variations. We introduce this study in Fig. 5 to validate the influence of using both adversarial learning and a very deep FC generator. First, we performed random generation of masks without adversarial learning and found that the success and precision on the OTB-100 dataset were severely affected, leading to inferior performance. As shown in Fig. 5, the performance in the case of using adversarial learning is increased by 18.63%. We then tried different depths of FC generators to observe the impact of increasing the number of FC layers in the generator network. We tested FC-2 (VITAL), FC-5, and

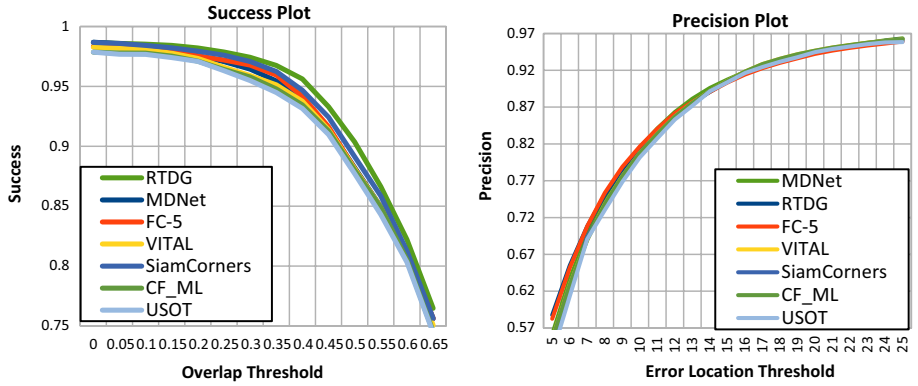


Fig. 6 Precision and success plots on the OTB-2015 dataset using OPE evaluation

Table 1 Comparison of the state-of-the-art trackers in terms of precision, AUC, and FPS on OTB dataset

Tracker	VITAL	FC-5	MDNet	SiamCorners	CF_ML	USOT	RTDG (ours)
Precision	0.853	0.854	**0.861**	0.832	0.853	0.813	0.857
AUC	0.711	0.714	0.714	0.715	0.699	0.682	**0.719**
FPS	3.351	3.841	4.051	44.3	**58.9**	37.7	3.197

The bolded values are the best ones and the underlined values are in the second order

FC-22 (RTDG) generators, where 2, 5, and 22 refer to the number of layers in the generator. When we increased the number of FC layers, the performance in terms of both success and precision improved. The performance in the case of FC-5 increased by 0.19% compared with FC-2, and RTDG increased by 0.53% compared with FC-5. This inference implies that the very deep generator can produce distributions that are more similar to the input feature distributions and has an excellent capability to capture robust non-spatial features. As a result, RTDG was the best for augmenting the positive samples and realizing the best tracking robustness compared to shallow FC generators.

Quantitative evaluation

We used standard evaluation metrics followed by the OTB-100 benchmark dataset. We employed one-pass evaluation (OPE), which starts tracking the ground-truth state in the first frame and provides the average precision and success rates in subsequent frames.

The precision of the trackers was measured using the Euclidean distance between the centers of the estimated bounding box and manually labeled ground-truth bounding box. However, when the target was lost, the distance was calculated randomly. As a result, it is preferable to count the number of successful frames in which the distance between the assessed and ground-truth bounding boxes is less than a certain threshold (x-axis of the plot, in pixels), as shown in the precision plot (Fig. 6). To show the trackers' overall performance, we use the area under the curve (AUC) or success rate. Success plots are preferred over precision plots because precision only considers bounding box positions and ignores the size and overlap. The success rate changes when the overlap score threshold on the x-axis fluctuates between 0 and 1, and the resultant curve is

Table 2 The success scores of the state-of-the-art trackers with different appearnce variations in OTB-100 dataset

Tracker							
Attribute	**VITAL**	**FC-5**	**MDNet**	**Siam-corners**	**CF_ML**	**USOT**	**RTDG (ours)**
IV (25)	529.714	547.347	531.187	538.865	545.348	521.704	**573.497**
SV (28)	509.614	506.877	501.294	501.434	495.876	505.348	**533.942**
OCC (29)	478.240	510.816	513.198	487.930	527.209	514.452	**535.058**
DEF (19)	309.759	316.120	327.642	323.448	319.457	317.357	**330.967**
MB (12)	413.080	394.773	389.127	405.435	375.984	417.279	**421.667**
FM (17)	**361.024**	326.807	315.745	327.832	302.560	345.812	344.812
IPR (31)	**461.357**	427.843	427.071	456.096	409.289	416.374	445.652
OPR (39)	467.530	481.221	485.065	478.489	453.698	457.053	**503.952**
OV (6)	519.680	529.673	520.047	509.233	491.024	526.285	**544.707**
BC (21)	432.522	467.596	454.673	475.680	452.982	447.764	**490.763**
LR (4)	**446.804**	432.323	376.085	436.342	367.921	428.354	440.225

The bolded values are the best ones and the underlined values are in the second order

illustrated in Fig. 6. In addition, we calculated the average numerical values of precision and success for the 100 videos, which are listed in Table 1. We used in our comparison some state-of-the-art trackers like VITAL and MDNet, and some recent visual trackers like SiamCorners [69], CF_ML [70] and USOT [71].

As shown in Fig. 6 and Table 1, the overall performance of the RTDG architecture is superior to that of state-of-the-art trackers. It is at the first place according to the success rate and second place according to the precision. Our RTDG framework has the capability to capture a variety of lasting features, which gives the classifier the ability to recognize an object even if it is deformed under several appearance circumstances. It is clear from the results that the depth of the FC generator is an essential factor that severely affects tracking robustness. As shown, the results of the HDTG generator (22 FC layers) are better than the 5 FC layers generator, and the 5 FC layers generator has better results than 2 FC layers generator (VITAL). In addition, the MDNet framework has a lower success rate than ours, indicating that our tracker is more robust. Table 2 shows values of success rates against each of the OTB attributes for a detailed analysis of the capabilities of the state-of-the-art tackers against the different appearance variations. This indicates that the HDTG tracker was the most robust to the following eight attributes: IV, SV, OCC, DEF, MB, OPR, OV, and BC. It is also second with the remaining three attributes: FM, IPR, and LR.

Although the deep generator network is more robust than the shallow ones, but this depth negatively affects the speed of the tracker. As shown in Table 1, the proposed model is slow compared with the other models which have less number of layers.

Qualitative evaluation

Figures 7, 8, 9, 10, 11 show the qualitative evaluation of the five sequences of the OTB dataset covering most of the attributes. The MDNet tracker does not perform well in severe deformation cases, such as occlusion, out-of-view, and out-of-plane attributes, which require considerable coverage in the training phase using an extensive and

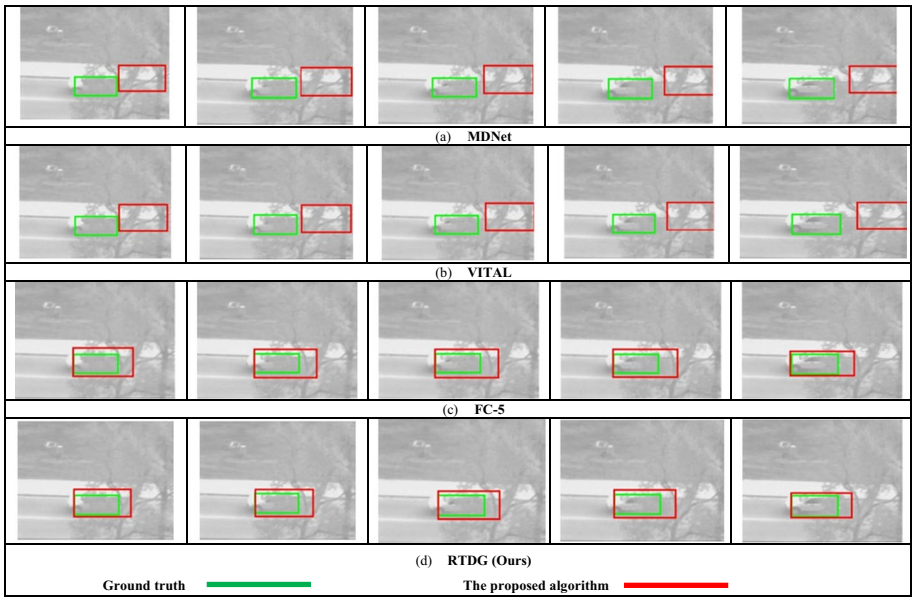


Fig. 7 Tracking results of the compared trackers on the 'Suv' video (frames: 562 to 566)

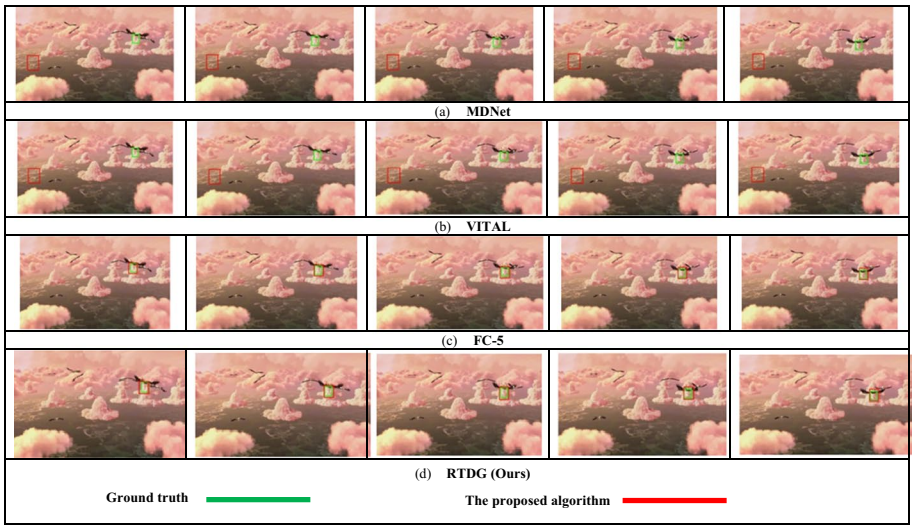


Fig. 8 Tracking results of the compared trackers on the 'Bird 1' video (frames: 280 to 284)

diversified training set. It is apparent in Figs. 7, 8, 11 that MDNet lost the target completely, and it was not sufficiently accurate to locate the object in Figs. 9, 10. The training samples in the MDNet method are insufficiently diversified to meet the severe changes in the samples' appearances or even get out of the scene during the tracking operation, which leads to an overfitting problem. The other three trackers all use adversarial learning by augmenting positive samples, which is performed by generating masks using FC generators. They can extract the most stable features to be more robust to changes in appearance, but it is obvious that the depth of the FC generator affects the ability to extract robust features. The VITAL tracker, which has a 2 FC generator, completely lost the target, as shown in Figs. 9, 10, and did not identify the precise locations and

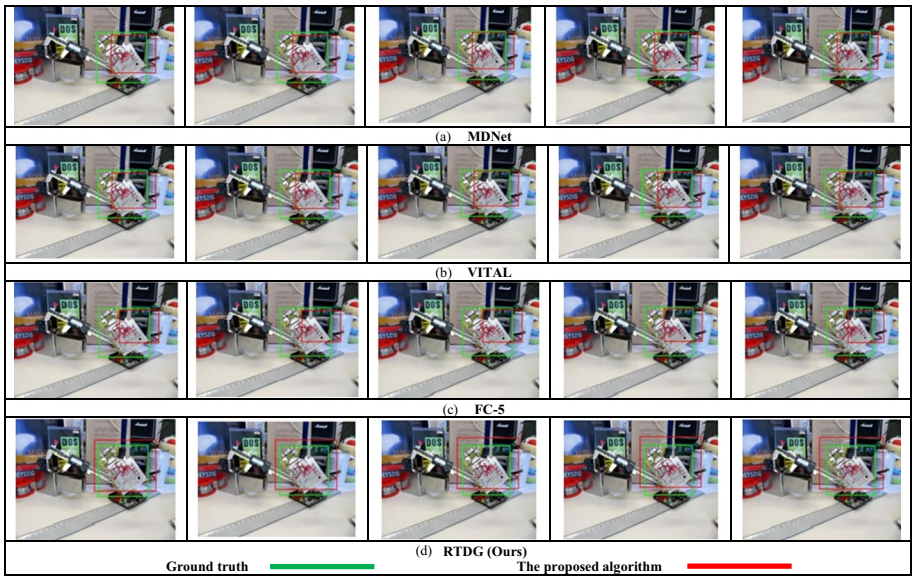


Fig. 9 Tracking results of the compared trackers on the 'Board' video (frames: 547 to 551)

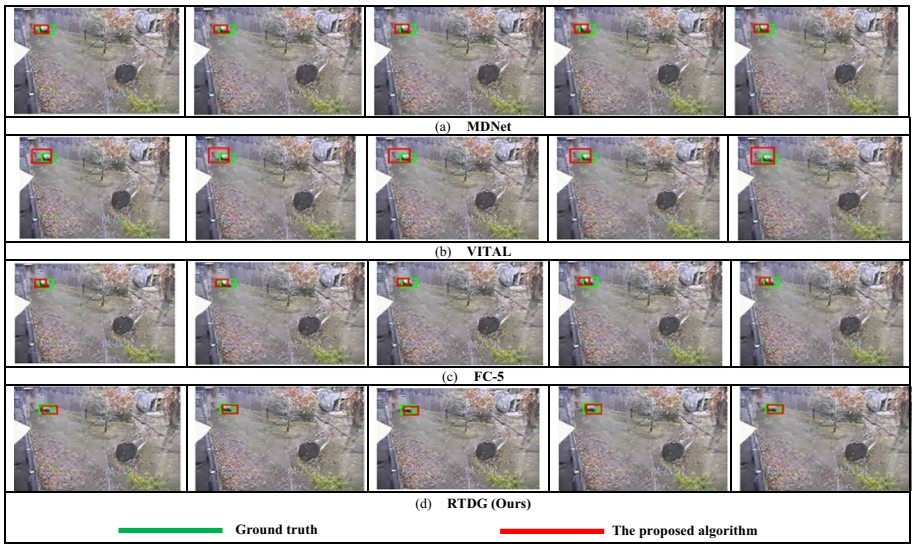


Fig. 10 Tracking results of the compared trackers on the 'Panda' video (frames: 739 to 743)

sizes in Figs. 9, 10. The framework with the FC-5 generator performed slightly better than VITAL. However, its performance was still lower than that of the proposed FC-22 (RTDG), as shown in Figs. 9, 10. Finally, it is evident from the figures that the proposed RTDG, which has the deepest FC generator, is the best among the state-of-the-art trackers on the five sequences, which have the most challenging attributes.

VOT2019 evaluation

The VOT2019 dataset is considered the seventh visual object tracking dataset in the VOT family. There are four different challenges in VOT2019 dataset based on the nature

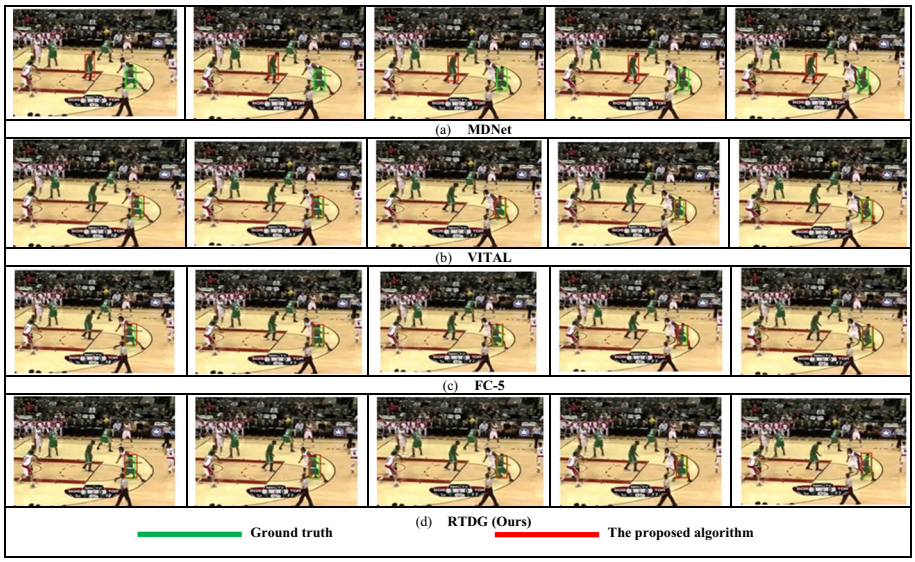


Fig. 11 Tracking results of the compared trackers on the 'Basketball' video (frames: 484 to 488)

of the tracker. Our tracker operates with short-term sequences. Therefore, we tested our tracker against state-of-the-art trackers on the VOT-ST2019 challenge. It contains 60 sequences with the following attributes: occlusion, illumination change, motion change, size change, and camera motion. The training sample set contains 60 images which are the first frames in the 60 videos. As this training set is not big enough, we used the adversarial learning by generating masks help to train the tracking model on the most robust features. First, the tracker is initialized on the first frame of a sequence, and it resets each time the overlap between the expected and ground truth bounding boxes is reduced to zero. Subsequently, the accuracy (A), robustness (R), and expected average overlap (EAO) measures were calculated. Accuracy (A) measures how closely the tracker's anticipated bounding box overlaps with the ground-truth bounding box. In a frame, the overlap is defined as the intersection over the union between the calculated and ground-truth bounding boxes. Robustness (R) is a measure of the frequency of tracker failures. The evaluation process was as follows: at the start of the sequence, a tracker was initialized and allowed to track until the overlap between the predicted region and the ground-truth annotation was larger than zero. After five frames, the tracker is re-initialized when the overlap decreases to zero, which is considered a tracker failure. The number of failures is counted over all sequences in the dataset and denoted as F, whereas the total number of frames is denoted as M. Robustness (R) was defined as follows:

$$R = exp(-S * (F/M)) \tag{6}$$

where S is the sensitivity parameter, and robustness denotes the probability that the tracker will not fail after S frames. Finally, EAO is a combination of accuracy and robustness metrics, and is the primary metric that specifies the performance of the tracker. It evaluates average overlaps across a

Table 3 Comparison of the state-of-the-art trackers on VOT2019 dataset

Tracker	VITAL	FC-5	MDNet	A3CTD	TADT	SiamRPNpp	CSRDCF	RTDG (Ours)
Accuracy	0.51	0.51	0.51	0.451	0.516	**0.599**	0.496	<u>0.52</u>
Robustness	<u>85</u>	95	99	243	117	100	132	**83**
EAO	**0.294**	0.264	0.274	0.165	0.207	0.285	0.201	<u>0.286</u>

The bolded values are the best ones and the underlined values are in the second order

large number of N_s-frame-long sequences over a range of sequence lengths, including zero overlaps ($\varnothing_{N_s}$). The overall EAO ($\varnothing$) is computed by averaging the EAO values in the range of [N_{lo}; N_{hi}] frames in short-term videos.

$$\varnothing = \frac{1}{N_{hi} - N_{lw}} \sum\nolimits_{N_s = N_{lo}:N_{hi}} \varnothing_{N_s} \tag{7}$$

The range boundaries are the places closest to the left and right of the mode, where $p(N_{lo}) \approx p(N_{hi})$ and the integral of the probability density function is within the range of 0.5. Table 3 presents the results of RTDG, VITAL, FC-5, A3CTD [72], TADT [73], SiamRPNpp [22], CSRDCF [74] and MDNet on the VOT2019 dataset in terms of accuracy, robustness, and EAO. The results show that RTDG has the best robustness, and the second-best in terms of EAO and accuracy.

LaSOT evaluation

We evaluated the proposed tracker with the state-of-the-art trackers on a high-quality benchmark for large-scale single object tracking (LaSOT) [67]. This dataset is for long-term tracking videos. It contains 1400 videos that had between 1000 and 11397 frames per video. The test dataset comprises 280 videos and the training dataset contains 1120 videos.

Table 4 shows the results of the proposed framework (RTDG) compared with VITAL, FC-5, A3CTD, TADT, SiamRPNpp, CSRDCF, MDNet, SiamCorners, CF_ML and USOT on the LaSOT dataset in terms of success, precision, and normalized precision. RTDG scored the second one in terms of precision and the third position in terms of success and normalized precision, making it a strong competitor to SiamRPNpp tracker, although it does not use long-term strategies.

UAVDT evaluation

The unmanned aerial vehicle for detection and visual tracking benchmark dataset (UAVDT) [68] is a large scale benchmark which contains 100 sequences that consist of about 80,000 frames with over 0.8 million bounding boxes selected from 10 h raw videos. The dataset is interested in vehicles (cars, buses and trucks) over urban areas and focuses on complex scenarios (e.g., flying altitude, weather condition, camera view and occlusion). This UAVDT dataset was collected for three computer vision tasks: object detection, single object tracking and multiple object tracking. For single object tracking, the one related to our method, there are eight challenging attributes: background clutter (BC), camera rotation (CR), Object rotation (OR), small object (SO), illumination variation (IV), object blur (OB), scale variation (SV), and large occlusion (LO). It uses the

Table 4 Comparison of the state-of-the-art trackers on LaSOT dataset

Tracker	VITAL	FC-5	MDNet	A3CTD	TADT	SiamRPNpp	CSRDCF	Siam-corners	CF_ML	USOT	RTDG (ours)
Success	0.390	0.392	0.397	0.415	0.397	**0.495**	0.244	<u>0.480</u>	0.305	0.358	0.433
Precision	0.360	0.368	0.373	0.368	0.370	**0.493**	0.220	0.406	0.287	0.340	<u>0.409</u>
Normalized Precision	0.453	0.455	0.460	0.454	0.457	**0.570**	0.254	<u>0.555</u>	0.341	0.398	0.481

The bolded values are the best ones and the underlined values are in the second order

Table 5 Comparison of the State-of-the-art trackers on UAVDT dataset

Tracker	VITAL	FC-5	MDNet	SiamFC	KCF	STRCF	CREST	ECO	MS-faster	MRCF	Dr^{2}track	RTDG (ours)
Success	44.8	44.9	**46.4**	44.7	29.0	41.1	39.6	45.1	44.5	<u>45.9</u>	44.9	45.8
Precision	71.8	71.6	<u>72.5</u>	68.1	57.0	62.9	64.9	70.2	71.0	71.9	71.2	**72.8**

The bolded values are the best ones and the underlined values are in the second order

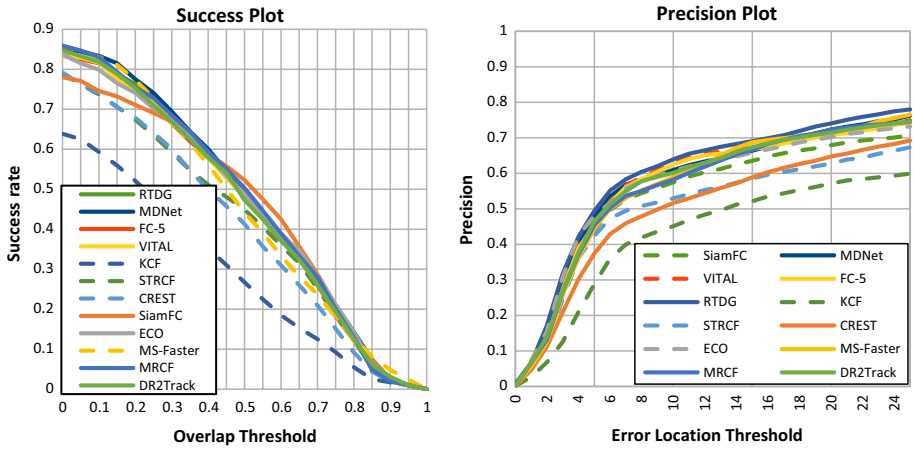


Fig. 12 Precision and success plots on the UAVDT dataset using OPE evaluation

common one-pass evaluation (OPE) scheme used by OTB and LaSOT datasets that calculates success and precision scores to evaluate the tracking performance. A comparison was performed on some state-of-the-art trackers like MDNet, VITAL, SiamFC [75], KCF [76], STRCF [17], CREST [77] and ECO [78], and other three of the most recent trackers; MS-Faster [79], MRCF [80] and DR^{2}Track [81]. Table 5 and Fig. 12 show that our RTDG method has a favorable performance compared with the state-of-the-art trackers on UAVDT dataset. It achieved the best precision and the third one in terms of success after MDNet and MRCF trackers. Accordingly, RTDG is a significant competitor to state-of-the-art trackers on OTB-100, VOT2019, LaSOT and UAVDT datasets.

Conclusions and future work

In recent years, adversarial learning has achieved significant success in the field of deep learning. In this study, we introduce a novel deep FC structure of the generator that is used in the tracking network via an adversarial learning process. We augmented a tracking-by-detection framework using our RTDG generator that produces masks in the feature space that can distinguish between the most robust features and the discriminative features in individual frames. This augmentation enriches the training set from the perspective of sample diversity to decrease the gap between the data-hungry deep learning algorithms and ordinary CNN networks. According to the experiments that we performed on four datasets to compare the RTDG algorithm with other state-of-the-art algorithms, the empirical results showed that RTDG has an effective robustness compared to the others. In addition, we compared different depths of FC generators and found that depth is a vital factor that influences generator performance. As the depth increases, the ability to generate more robust masks that train the framework to be robust against different appearance variations also increases.

The GAN family has a new birth of new models with more capabilities every day. Therefore, the selection of a suitable architecture for GANs integrated with CNNs

remains an open research area. In addition, the generated masks can be designed to have more advanced properties and sizes.

Abbreviations

RTDG	Robust visual tracking using a very deep generative model
CNN	Convolutional neural network
GAN	Generative adversarial network
FC	Fully connected
ML	Machine learning
DL	Deep learning
CF	Correlation filter
NA	Noise-aware
SNR	Signal-to-noise
ROI	Region of interest
TIR	Thermal infrared
LaSOT	Large-scale single-object tracking
STRCF	Spatial temporal regularized correlation filter
SVM	Support vector machine
CNT	Convolutional network without training
ReLU	Rectification function
MSE	Mean square error
SGD	Stochastic gradient descent
JS	Jensen-Shannon loss divergence
BCE	Binary cross-entropy loss
OPE	One-pass evaluation
AUC	Area under the curve
EAO	Expected average overlap
UAVDT	The unmanned aerial vehicle for detection and visual tracking benchmark dataset
CR	Camera rotation
OR	Object rotation
SO	Small object
OB	Object blur
LO	Large occlusion

Acknowledgements
Not applicable.

Author contributions
ERA designed the approach, performed the software, wrote the methodology, experimental results and the literature review. HEA reviewed the methodology and experimental results and provided valuable ideas for better article framework. AA and HMA double checked the manuscript. All authors read and approved the final manuscript.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and materials
The data that support the findings of this study are publicly available. OTB-100 dataset is available at: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html, VOT2019 dataset is available at: https://www.votchallenge.net/vot2019/, LaSOT dataset is available at: http://vision.cs.stonybrook.edu/~lasot/ and UAVDT dataset is available at: https://sites.google.com/view/grli-uavdt/%E9%A6%96%E9%A1%B5.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 16 May 2022 Accepted: 25 December 2022
Published online: 11 January 2023

References

1. Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, Hays J. Argoverse: 3d tracking and forecasting with rich maps. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019; 8748–8757.
2. Ali A, Jalil A, Niu J, Zhao X, Rathore S, Ahmed J, Aksam Iftikhar M. Visual object tracking—classical and contemporary approaches. Front Comp Sci. 2016;10(1):167–88.
3. Yang W, Jin L, Tao D, Xie Z, Feng Z. DropSample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition. Pattern Recogn. 2016;58:190–203.
4. Bouget D, Allan M, Stoyanov D, Jannin P. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. Med Image Anal. 2017;35:633–54.
5. Klopschitz M, Schall G, Schmalstieg D, Reitmayr, G. Visual tracking for augmented reality. In 2010 International conference on indoor positioning and indoor navigation. 2010; 1–4.
6. Kumar A, Walia GS, Sharma K. Recent trends in multicue based visual tracking: a review. Expert Syst Appl. 2020;162: 113711.
7. Wang L, Ouyang W, Wang X, Lu H. Visual tracking with fully convolutional networks. Proc Int Conf Comp Vision. 2015;1:3119–27.
8. Li H, Li Y, Porikli F. DeepTrack: learning discriminative feature representations by convolutional neural networks for visual tracking. In BMVC. 2014;1(2):3.
9. Chen Y, Jiang H, Li C, Jia X, Ghamisi P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans Geosci Remote Sens. 2016;54(10):6232–51.
10. Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking. Proc IEEE Conf Comp Vision Pattern Recog. 2016;1:4293–302.
11. Smeulders AW, Chu DM, Cucchiara R, Calderara S, Dehghan A, Shah M. Visual tracking: an experimental survey. IEEE Trans Pattern Anal Mach Intell. 2013;36(7):1442–68.
12. Marvasti-Zadeh SM, Cheng L, Ghanei-Yakhdan H, Kasaei, S. Deep learning for visual tracking: a comprehensive survey. IEEE trans intell transp syst. May 2022;23(5):3943–68.
13. Li P, Wang D, Wang L, Lu H. Deep visual tracking: review and experimental comparison. Pattern Recogn. 2018;76:323–38.
14. Touil DE, Terki N, Medouakh S. Hierarchical convolutional features for visual tracking via two combined color spaces with SVM classifier. SIViP. 2019;13(2):359–68.
15. Danelljan M, Robinson A, Shahbaz Khan F, Felsberg M. Beyond correlation filters: Learning continuous convolution operators for visual tracking In European conference on computer vision. Cham: Springer; 2016.
16. Kiani Galoogahi H, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking. In Proc Int Conf Comp Vision. 2017;1:1135–43.
17. Li F, Tian C, Zuo W, Zhang L, Yang MH. Learning spatial-temporal regularized correlation filters for visual tracking. Proc Conf Comp Vision Pattern Recog. 2018;1:4904–13.
18. Li X, Liu Q, Fan N, Zhou Z, He Z, Jing XY. Dual-regression model for visual tracking. Neural Netw. 2020;132:364–74.
19. Li B, Yan J, Wu W, Zhu Z, Hu X. High performance visual tracking with siamese region proposal network. Proceed Conf Comp Vision Pattern Recog. 2018;1:8971–80.
20. Li Y, Zhang X. SiamVGG: Visual tracking using deeper siamese networks. arXiv preprint 2019 arXiv:1902.02804.
21. Yuan D, Chang X, Huang PY, Liu Q, He Z. Self-supervised deep correlation tracking. IEEE Trans Image Process. 2020;30:976–85.
22. Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. Proc Conf Comp Vision Pattern Recog. 2019;1:4282–91.
23. Li S, Zhao S, Cheng B, Chen J. Noise-aware framework for robust visual tracking. IEEE Trans Cybern. Feb. 2022;52(2):1179–92.
24. Li S, Zhao S, Cheng B, Zhao E, Chen J. Robust visual tracking via hierarchical particle filter and ensemble deep features. IEEE Trans Circuits Syst Video Technol. 2018;30(1):179–91.
25. Liu Q, Li X, He Z, Fan N, Yuan D, Wang H. Learning deep multi-level similarity for thermal infrared object tracking. IEEE Trans Multimedia. 2020;23:2114–26.
26. Li X, Liu Q, Fan N, He Z, Wang H. Hierarchical spatial-aware siamese network for thermal infrared object tracking. Knowl-Based Syst. 2019;166:71–81.
27. Liu Q, Lu X, He Z, Zhang C, Chen WS. Deep convolutional neural networks for thermal infrared object tracking. Knowl-Based Syst. 2017;134:189–98.
28. Liu Q, Yuan D, Fan N, Gao P, Li X, He Z. Learning dual-level deep representation for thermal infrared tracking. IEEE Trans Multimed. 2022. https://doi.org/10.1109/TMM.2022.3140929.
29. Fan J, Song H, Zhang K, Yang K, Liu Q. Feature alignment and aggregation siamese networks for fast visual tracking. IEEE Trans Circuits Syst Video Technol. 2020;31(4):1296–307.
30. Zhang S, Lu W, Xing W, Zhang L. Using fuzzy least squares support vector machine with metric learning for object tracking. Pattern Recogn. 2018;84:112–25.
31. Zhang K, Liu Q, Wu Y, Yang MH. Robust visual tracking via convolutional networks without training. IEEE Trans Image Process. 2016;25(4):1779–92.
32. Hong S, You T, Kwak S, Han B. June). Online tracking by learning discriminative saliency map with convolutional neural network. Int Conf Mach Learn. 2015;1:597–606.
33. Qi Y, Yao H, Sun X, Sun X, Zhang Y, Huang Q. Structure-aware multi-object discovery for weakly supervised tracking. In 2014 IEEE International Conference on Image Processing (ICIP). 2014:466–70. IEEE. https://doi.org/10.1109/ICIP.2014.7025093.
34. Yang Y, Li G, Qi Y, Huang Q. Release the power of online-training for robust visual tracking. Proceed Conf on Art Intel. 2020;34(07):12645–52.
35. Qi Y, Zhang S, Zhang W, Su L, Huang Q, Yang MH. Learning attribute-specific representations for visual tracking. Proc Conf Art Intel. 2019;33(01):8835–42.
36. Qi Y, Qin L, Zhang S, Huang Q, Yao H. Robust visual tracking via scale-and- state-awareness. Neurocomputing. 2019;329:75–85.

37. Borsuk V, Vei R, Kupyn O, Martyniuk T, Krashenyi I, Matas J. (2021). FEAR: Fast, efficient, accurate and robust visual tracker. arXiv preprint arXiv:2112.07957.
38. Mayer C, Danelljan M, Bhat G, Paul M, Paudel DP, Yu F, Van Gool L. Transforming model prediction for tracking. Proc Conf Comp Vision Pattern Recog. 2022;1:8731–40.
39. Shah RA, Urmonov O, & Kim H. Improving Performance of CNN Based Vehicle Detection and Tracking by Median Algorithm. In 2021 IEEE International Conference on Consumer Electronics-Asia(ICCEA-Asia), 2021:1–3. https://doi.org/10.1109/ICCE-Asia53811.2021.9641942.
40. Duan R, Fu C, Alexis K, Kayacan E. Online recommendation-based convolutional features for scale-aware visual tracking. Int Conf Rob Auto. 2021;1:4206–14212.
41. Lu X, Li F. Study of robust visual tracking based on traditional denoising methods and CNN. In 2021 International Conference on Security, Pattern Analysis, and Cybernetics. 2021; 392–396.
42. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Bengio Y. Generative adversarial nets. Adv Neural Inform Process Sys. 2014;1:27.
43. Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. Int Conf Machine Learn. 2017;1:214–23.
44. Yu Y, Gong Z, Zhong P, Shan, J. (2017). Unsupervised representation learning with deep convolutional neural network for remote sensing images. In International conference on image and graphics. Springer: Cham.
45. Choi Y, Choi M, Kim M, Ha JW, Kim S, Choo J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. Proceed ConfComp Vision Pattern Recog. 2018;1:8789–97.
46. Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint 2014 arXiv:1411.1784.
47. Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. Proc Conf Comp Vision Pattern Recog. 2019;1:4401–10.
48. Zhu JY, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proc Intern Conf Comp Vision. 2017;1:2223–32.
49. Bai Y, Zhang Y, Ding M, Ghanem B. Sod-mtgan: Small object detection via multi-task generative adversarial network. Proc Eur Conf Comp Vision. 2018;1:206–21.
50. Sampath V, Maurtua I, Aguilar Martín JJ, Gutierrez A. A survey on generative adversarial networks for imbalance problems in computer vision tasks. J Big Data. 2021;8(1):1–59.
51. Zhang Z, Yang L, Zheng Y. Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network. Proc Conf Comp Vision Pattern Recog. 2018;1:9242–51.
52. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Shi W. Photo-realistic single image super-resolution using a generative adversarial network. Proc Conf Comp Vision Pattern Recog. 2017;1:4681–90.
53. Song Y, Ma C, Wu X, Gong L, Bao L, Zuo W, Yang MH. Vital: Visual tracking via adversarial learning. Proc Conf Comp Vision Pattern Recog. 2018;2018:8990–9.
54. Wang X, Li C, Luo B, Tang J. Sint++: Robust visual tracking via adversarial positive instance generation. Proc Conf Comp Vision Pattern Recog. 2018;1:4864–73.
55. Zhao F, Wang J, Wu Y, Tang M. Adversarial deep tracking. IEEE Trans Circuits Syst Video Technol. 2018;29(7):1998–2011.
56. Han Y, Zhang P, Huang W, Zha Y, Cooper GD, Zhang Y. Robust visual tracking based on adversarial unlabeled instance generation with label smoothing loss regularization. Pattern Recogn. 2020;97: 107027.
57. Yin Y, Xu D, Wang X, Zhang L. Adversarial feature sampling learning for efficient visual tracking. IEEE Trans Autom Sci Eng. 2019;17(2):847–57.
58. Taud H, Mas JF. Multilayer perceptron (MLP) In Geomatic approaches for modeling land change scenarios. Cham: Springer; 2018.
59. Liu S, Deng W. Very deep convolutional neural network based image classification using small training sample size. In 2015 3rd IAPR Asian conference on pattern recognition (ACPR). 2015;730–34. https://doi.org/10.1109/ACPR.2015.7486599.
60. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Rabinovich A. Going deeper with convolutions. Pro Conf Comp Vision Pattern Recog. 2015;2015:1–9.
61. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint 2014 arXiv:1409.1556.
62. Nguyen A, Dosovitskiy A, Yosinski J, Brox T, Clune J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. Adv Neural Inform Proc Sys. 2016;29:1.
63. Bau D, Zhu JY, Strobelt H, Zhou B, Tenenbaum JB, Freeman WT, Torralba A. Visualizing and understanding generative adversarial networks. Int Conf Learning Repre. 2019;1:10.
64. Barua S, Erfani SM, Bailey J. FCC-GAN: A fully connected and convolutional net architecture for GANs. arXiv Preprint. 2019 arXiv:1905.02417.
65. Wu Y, Lim J, Yang M. Object tracking benchmark. IEEE Trans Pattern Anal Mach Intell. 2015;37(9):1834–48. https://doi.org/10.1109/TPAMI.2014.2388226.
66. Kristan M, Matas J, Leonardis A, Felsberg M, Pflugfelder R, Kamarainen JK, Hak Ki B. The seventh visual object tracking vot2019 challenge results. In proceedings of the IEEE/CVF international conference on computer vision workshops. 2019; 0–0.
67. Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, Ling H. Lasot: A high-quality benchmark for large-scale single object tracking. Proc Conf Compr Vision Pattern Recog. 2019;1:5374–83.
68. Du D, Qi Y, Yu H, Yang Y, Duan K, Li G, Tian Q. The unmanned aerial vehicle benchmark: Object detection and tracking. Proceedings of the European Conference on Computer Vision (ECCV). 2018;1:370–86.
69. Yang K, He Z, Pei W, Zhou Z, Li X, Yuan D, Zhang H. SiamCorners: siamese corner networks for visual tracking. IEEE Trans Multimedia. 2021;24:1956–67.
70. Zhao H, Yang G, Wang D, Lu H. Deep mutual learning for visual object tracking. Pattern Recogn. 2021;112: 107796.
71. Zheng J, Ma C, Peng H, Yang X. Learning to track objects from unlabeled videos. Proc Intern Conf Comp Vision. 2021;1:13546–55.

72. Dunnhofer M, Martinel N, Luca Foresti G, Micheloni C. Visual tracking by means of deep reinforcement learning and an expert demonstrator. In proceedings of The IEEE/CVF international conference on computer vision workshops. 2019;0–0.
73. Li X, Ma C, Wu B, He Z, Yang MH. Target-aware deep tracking. Proc Conf Comp Vision Pattern Recog. 2019;1:1369–78.
74. Kart U, Kamarainen JK, Matas J. How to make an rgbd tracker?. In proceedings of the european conference on computer vision (ECCV) Workshops. 2018;0–0.
75. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH. Fully-convolutional siamese networks for object tracking. In European conference on computer vision. Cham: Springer; 2016.
76. Henriques JF, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell. 2014;37(3):583–96.
77. Song Y, Ma C, Gong L, Zhang J, Lau RW, Yang MH. Crest: Convolutional residual learning for visual tracking. Proc Intern Conf Comp Vision. 2017;1:2555–64.
78. Danelljan M, Bhat G, Shahbaz Khan F, Felsberg M. Eco: Efficient convolution operators for tracking. Proc Conf Comp Vision Pattern Recog. 2017;1:6638–46.
79. Avola D, Cinque L, Diko A, Fagioli A, Foresti GL, Mecca A, Piciarelli C. MS-Faster R-CNN: multi-stream backbone for improved faster R-CNN object detection and aerial tracking from UAV images. Remote Sensing. 2021;13(9):1670.
80. Ye J, Fu C, Lin F, Ding F, An S, Lu G. Multi-regularized correlation filter for UAV tracking and self-localization. IEEE Trans Industr Electron. 2021;69(6):6004–14.
81. Fu C, Ding F, Li Y, Jin J, Feng C. Learning dynamic regression with automatic distractor repression for real-time UAV tracking. Eng Appl Artif Intell. 2021;98: 104116.

Publisher's Note