Chapter 4

Robust Outdoor Vehicle Visual Tracking Based on k-Sparse Stacked Denoising Auto-Encoder

Jing Xin, Xing Du, Yaqian Shi, Jian Zhang and
Ding Liu

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.80089

Abstract

Robust visual tracking for outdoor vehicle is still a challenging problem due to large object appearance variations caused by illumination variation, occlusion, and fast motion. In this chapter, k-sparse constraint is added to the encoder part of stacked auto-encoder network to learn more invariant feature of object appearance, and a stacked k-sparse-auto-encoder–based robust outdoor vehicle tracking method under particle filter inference is further proposed to solve the problem of appearance variance during the tracking. Firstly, a stacked denoising auto-encoder is pre-trained to learn the generic feature representation. Then, a k-sparse constraint is added to the stacked denoising auto-encoder, and the encoder of k-sparse stacked denoising auto-encoder is connected with a classification layer to construct a classification neural network. Finally, confidence of each particle is computed by the classification neural network and is used for online tracking under particle filter framework. Comprehensive tracking experiments are conducted on a challenging single-object tracking benchmark. Experimental results show that our tracker outperforms most state-of-the-art trackers.

Keywords: visual tracking, k-sparse stacked denoising auto-encoder, classification neural network, robust visual tracking, particle filter

1. Introduction

The purpose of the visual tracking for outdoor vehicle is to estimate the state of outdoor vehicle and provide current traffic state accurately and comprehensively. At present, it has become an important part of intelligent transport system (ITS). However, robust tracking for outdoor vehicle is still a challenging problem due to the complex and varying outdoor environment. Many researchers proposed solutions to the different challenging environment. Rad [1] proposed a strategy that can solve the problem of occlusion during the tracking process of moving vehicles on highway. But, the tracking accuracy of this method will be greatly reduced when the lighting conditions change sharply. Zhang et al. [2] proposed a multi-layer occlusion detection and processing framework that can be used to deal with the problem of mutual occlusion between two vehicles. Faro et al. [3] further improved [2] by introducing curvature scale space to segment occluded region accurately. Xin et al. [4] proposed adaptive multiple cues integration for robust outdoor vehicle visual tracking in the particle filter framework. This method has strong robustness against color interference and partial or complete occlusion of vehicles. Although these existing methods have achieved certain progress in outdoor vehicles visual tracking, these methods can only deal with the occlusion problem between two vehicles or the occlusion of vehicles by other objects. However, in the actual traffic scene, the mutual occlusion between multiple vehicles often occurs and faces the challenges of complex outdoor environments such as illumination variation (IV), cluttered background (BC), and fast motion (FM). Therefore, the robust outdoor vehicles visual tracking remains a thorny issue.

Existing visual object tracking algorithms are mainly divided into two major categories that include generative model and discriminative model. The generative model learns the appearance representation of the object and searches the candidate area that most closely matches the object appearance template as the location of the object in the new frame. The discriminative model treats the object tracking as a binary classification problem, using the learned characteristics to distinguish the object and background information. Therefore, the extraction of robust features is the key to the success of the object tracking technology. Traditional visual object tracking methods rely on artificial features; the low-dimensional artificial features are not robust to large appearance variation of object. Recently, deep learning shows promising performance in automatic extracting feature that outperforms pre-defined handcraft feature methods. Deep learning can map the original feature space to another feature space to learn more abundant features. Recently, deep learning has been widely applied to image processing, speech recognition, natural language processing, health care, robotics, and other fields for its powerful feature learning capability. It has been proved that feature representation when learnt in a deep learning way encourages sparsity. And k-sparse constraint can guarantee that each input for a certain sparsity. At the same time, some scholars have applied it to video object tracking technology. Due to the powerful feature representation ability of deep learning, the robustness of visual object tracking technology has been greatly improved. Wang et al. [5] proposed a fully convolutional networks tracker (FCNT) that uses convolutional neural networks to learn the characteristics of objects from large-scale classification datasets and further analyses performance of the extracted features in the object tracking aspect. High-level features are good at distinguishing different kinds of objects and are very robust to the appearance variation of the object. Low-level features more focus on the local details of the object and can be used to distinguish similar distractors in the background. FCNT can effectively prevent object tracking drift based on the effective use of different layers of convolutional neural network (CNN) features. Nam et al. [6] subsequently proposed the Multi-Domain Convolutional Neural Networks (MDNet). Unlike [5], MDNet directly uses the tracking video data sets train, the deep learning model to obtain the universal feature representation of the object, and then fine-tunes the network parameters for each particular video sequence in online tracking to achieve more robust tracking. However, the tracking speed of MDNet is slow and cannot meet the requirements of real-time performance. Existing research has demonstrated that sparseness is encouraged when deep learning learns feature representations. Because sparse representation can reduce the complexity of the representation, which is crucial to improve the speed of the object tracking algorithm, sparse constraints can be used to further optimize the deep network [7, 8] and can make the original signal express more meaningful, which has been verified by independent principal component analysis and sparse coding algorithm [9]. In general, there are two ways to add sparse constraints into the deep network for sparse representation: sparseness of the hidden layer response and weight sparseness between the hidden layer and the input layer. In this chapter, we adopt the first method for sparse representation. At the same time, we perform k-sparse constraint in neural network to keep only k highest activities in hidden layers, which can maintain the sparse representation of each input [10]. In other words, we add the k sparse constraint to the original stacked denoising auto-encoder (SDAE) hidden layer unit and form kSSDAE, which is used as a feature extractor in the target tracking to better learn the invariant feature of the object appearance. Therefore, the application of kSSDAE in object tracking can overcome poor robustness problem and further improve the robustness of visual tracking. The main contributions of this chapter are as follows.

- We propose a new auto-encoder–based tracking method, namely kSSDAE tracker, to solve the robust tracking for outdoor vehicles in complex environments, such as occlusion, clutter background, illumination variation, and so on.
- We add the k-sparse constraint into the encoder part of stacked auto-encoder network to learn more invariant feature of object appearance during the tracking.
- We evaluate our method on a challenging single-object tracking benchmark with 51 video sequences and 11 attributes.

2. Related works

Deep learning has exhibited powerful automatic feature extraction capability in computer vision tasks such as image classification, object detection, and so on. Visual object tracking is one of the important research contents in the field of computer vision. The performance of the tracker can be greatly improved due to the applications of the deep learning. Currently, two kinds of deep learning models including convolution neural network and deep auto-encoder are mainly used in the visual object tracking to perform automatic feature extraction.

2.1. Convolutional neural network for object tracking

Convolutional neural network (CNN) is a multi-layered supervised learning feedforward neural network. A typical CNN structure includes convolutional layer, pooling layer, and full connection layer. Specially, the automatic feature extraction function of the CNN is mainly realized through the convolution layer and pooling layer. The structure of the CNN determines that it has natural advantages for image processing, and it also shows a competitive performance in visual tracking. In order to solve the problem of object drift caused by similar or clutter background in visual tracking, Fan [11] et al. use CNN to learn spatial and temporal invariance features between adjacent frames. Jin [12] combine a CNN with two convolutional layers and two pooling layers and radial basis function (RBF) to perform feature extraction so that it can better learn the invariable features of the object appearance in visual tracking. Hong [13] use an offline-trained CNN to extract the distinctive feature map of the object in visual tracking. Wang [14] train a two-level CNN by offline way and use it for online object tracking. The network pays more attention to the learning of motion invariant features. Unlike most CNN used for object tracking, the network designed by Wang [15] et al. is not a binarized output classification result but instead generates a probability map to represent the potential area of the object. The use of CNN greatly improves the accuracy of visual tracking, but high computational complexity is still a limitation. In order to improve the real-time performance of the tracking algorithm, Doulamis et al. [16] proposed a fast adaptive supervised algorithm for object tracking and classification. In addition, although the pooling operation in CNN can obtain invariant features to drop the recognition effect caused by the change of the object appearance, however, it reduces the resolution of the image and leads to spatial information loss. The loss of information of pooling operations is crucial for tracking [17]. Zhang et al. [18] combined convolutional neural networks with spatial-temporal saliency-guided sampling for object tracking in a correlated filter framework. The algorithm establishes an optimization function to locate object positions based on significant region detection and significant motion estimation. Different from other object tracking algorithms whose location estimation is based on the last layer of the convolutional neural network, this algorithm combines intra-frame appearance correlation information and inter-frames motion saliency information to ensure accurate target location. All in all, the object tracking algorithm based on the convolutional neural network can effectively track object, but the network structure is relatively complex, consumes a lot of training time, and requires a large number of labeled training samples, and it is difficult to achieve a balance between tracking accuracy and tracking speed.

2.2. Deep auto-encoder for object tracking

The basic idea of the deep auto-encoder (DAE) is to encode the input signal and then use the decoder to reconstruct the original signal. The goal of the one is to minimize the reconstruction error between the reconstructed signal and original signal. Compared with the method of visual tracking using CNN, a DAE compresses the original signal by coding, removes redundancy, and can reflect the more primitive nature of the original signal in a more concise manner. Therefore, visual tracking using DAE has a lower calculation cost and is more suitable for some occasions with high real-time requirements. In 2013, Wang et al. [19] proposed a novel deep learning tracker (DLT), which for the first time uses a DAE for tracking. DLT considers the object tracking task as a two-category problem. Firstly, using Tiny Images data set to offline train a stacked denoising auto-encoder (SDAE) in an unsupervised manner to obtain a universal image feature representation for object and then use it for online tracking. The classification neural network is constructed and is fine-tuned in the tracking process to distinguish the target from the background. Soon after, many improved versions of the DLT methods have been proposed. For example, Zhou [20] combined online AdaBoost feature selection framework with SDAE for object tracking to effectively solve complex and dramatic changes of the object appearance. Cheng et al. [21] used the SDAE network to implement adaptive target tracking in an incremental deep learning approach under the dual particle filter framework. Cheng et al. [22] implemented an object tracking algorithm based on enhanced group tracker and SDAE in the framework of the popular tracking-learning-detection (TLD) algorithm in order to solve the object drift of the tracking method based on the appearance model. Due to the Haar-like features in the multi-instance learning (MIL) tracking algorithm are difficult to reflect the shortcomings of the object itself and the external changes, Cheng et al. [23] introduced SDAE to extract the effective features of the example image to achieve higher precision tracking. In order to further improve the application performance of the stacked denoising auto-encoder in video object tracking, some scholars have proposed many improved tracking algorithms based on a stacked denoising auto-encoder. Dai et al. [24] proposed a local patch tracking algorithm based on a stacked denoising auto-encoder. The algorithm partitions the input image; then a feature extractor combining multiple stacked denoising auto-encoder is used to describe the feature information of local patch and fuse their local features to achieve object tracking. The local feature extraction greatly reduces the computation complexity compared with the global feature representation. In the tracking process, the weight of each patch of the object candidate region can be adaptively adjusted according to the confidence of the corresponding network. Hua et al. [25] proposed a new visual tracking algorithm based on the multi-level feature learning capability of the stack denoising auto-encoder under the particle filter framework.

The training of the stacked auto-encoder network includes two stages of hierarchical pre-training and online tracking. In the hierarchical pre-training stage, a description of multi-level image features is obtained. In the online tracking stage, the network parameters are back-propagated through the genetic algorithm to fine-tuning. The use of genetic algorithm in network parameter adjustment effectively avoids the deficiency of traditional BP algorithm and further enhances the robust performance of the network. These trackers can use SDAE for unsupervised feature learning on data that lacks tagging, improving the problem of insufficient training data for deep neural networks (DNNs). However, in some challenging and complex environments, these trackers will fail to track the object. Therefore, we can further enhance the feature expression capabilities of deep neural networks (DNNs) for more robust tracking.

In this chapter, we add the K-sparse constraint into the coding part of the SDAE to learn more invariant feature of object appearance and propose a staked k-sparse-auto-encoder–based robust tracking algorithm for outdoor vehicle under particle filter framework to solve the problem of large appearance variations during the tracking.

3. The kSSDAE-based tracker

Overall structure of the proposed kSSDAE-based robust tracking algorithm for outdoor vehicle is shown in **Figure 1**. The tracking system mainly includes three parts as follows: offline

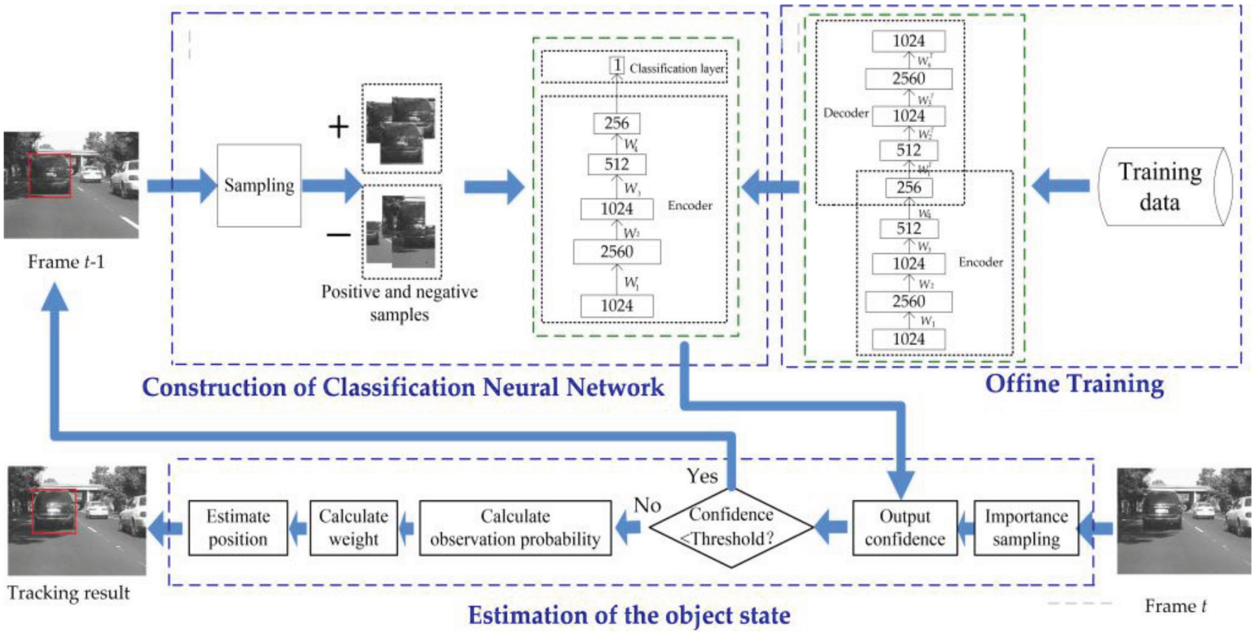


Figure 1. Overall structure of the proposed kSSDAE-based robust tracking algorithm for outdoor vehicle.

training of SDAE, construction of classification neural network, and estimation of object state. The basic idea of the algorithm is: firstly, we adopt the pre-trained SDAE model proposed in DLT [15] to learn the generic feature representation. Training data of the model are obtained through sampling randomly 1 million images from Tiny Images data set [26]. Tiny Images data set contains many kinds of the scene image. Before offline training, we need to preprocess the input data with 32 × 32. Offline training way of the SDAE is unsupervised. Secondly, we propose a kSSDAE model to learning more invariant feature of object appearance during tracking and train a classification neural network to compute the confidence of each particle. This is the key step to achieve robust tracking. Without the kSSDAE, the input cannot be guaranteed to have a sparse representation to extract more effective features to adapt the object appearance change. Finally, we estimate the object state under the particle filter framework, that is, the object state of the current frame can be represented by the particle with maximum confidence, which is calculated by classification neural network.

The specific implementation of the two main parts of the proposed tracking method will be stated in detail in the next section.

3.1. Construction of classification neural network

The main function of this module is to compute the confidence of each particle during the online tracking. Here, confidence is used for evaluating every particle's reliability. In this chapter, the classification neural network can be constructed by connecting the encoder of the well-trained kSSDAE with a classification layer as shown in **Figure 2**.

In feedforward phase, the hidden activities function z can be computed as

$$z = W^T x + b \tag{1}$$

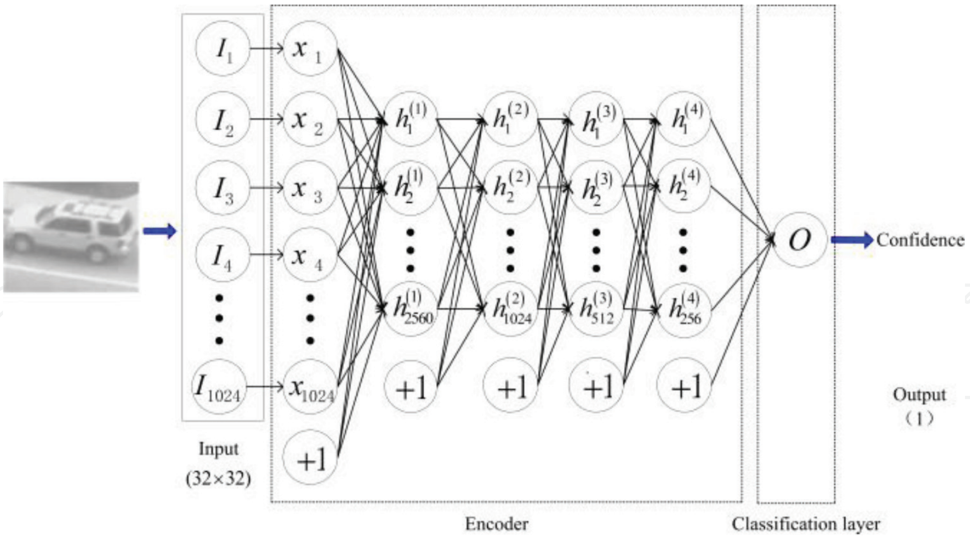


Figure 2. Architecture of classification neural network (1024-2560-1024-512-256-1).

where, x is the input vector, W is weight, and b is bias. We keep the k as largest hidden units and set others to zero.

Reconstruction error can be computed using the sparsified z as follows:

$$E = \left\| x - \left(Wz + b' \right) \right\|_2^2 \tag{2}$$

In back propagate phase, weights can adjusted by the k highest activities back propagating the reconstruction. The confidence computed by classification neural network reflects the credibility of decision in feature vector space of classifier. Ref. [27] has proved that when we use mean square error or cross-entropy as the cost function, the output expectation of multi-layer neural network is posterior probability of each class.

Let o_i be the output of the neural network corresponding to the k_i class, the output expectation can be computed by the posterior probability

$$E(o_i) = P(k_i|x) \tag{3}$$

Generally, the class with maximum probability is taken as a decision. So, the confidence can be obtained from the maximum output of the classification neural network.

$$c(x) = E(\max o_i) \tag{4}$$

At the beginning of the visual tracking, we select the object to be tracked and fine-tune the classification neural network using positive and negative samples. In the process of online tracking, in order to adapt specific object appearance changes, we need to fine-tune the classification neural network again when the confidence, which is calculated by the classification neural network, is lower than the predefined threshold.

3.2. Estimation of the object state

Object state can be estimated by the object tracking algorithm, which can be viewed as a problem to estimate the posterior distribution $p\left(s_t^i|y_{1:t}\right)$ of state s_t at time t according to dynamic system $p\left(s_t^i|s_{t-1}\right)$ of the object state. In this chapter, object state s_t is represented by six affine transformation parameters corresponding to horizontal translation, vertical translation, scale angle, aspect ratio, and skewness, and the state transition distribution $p\left(s_t^i|s_{t-1}^i\right)$ of each dimension can be modeled as a zero-mean normal distribution. The purpose of visual object tracking is to estimate the object state s_t (location, scale, etc.) from image sequences given all observations by any appropriate loss function, for example, maximum a posteriori (MAP) estimation or minimum mean square error (MMSE) estimation. The main online tracking steps under the particle filter framework are as follows.

3.2.1. Computing observation probability

Each particle represents a possible instantiation of the state of the object being tracked. Most likely, particle represents the object state at time t. Confidence c_t^i of each particle can be calculated by the classification neural network. When the maximum confidence is lower than the predefined threshold τ, that is, if $\max\left(c_t^i\right) \le \tau$, we will fine-tune classification neural network by reselecting positive and negative training samples. If $\max\left(c_t^i\right) > \tau$, we calculate the observation probability by normalizing confidence

$$p\left(y_t|s_t^i\right) \propto c_t^i, i = 1, 2, \ldots, n \tag{5}$$

3.2.2. Updating weight

The weights for each particle can be updated according to the observation probability

$$w_t^i = w_{t-1}^i \cdot \frac{p\left(y_t|s_t^i\right)p\left(s_t^i|s_{t-1}^i\right)}{q\left(s_t|s_{t-1}, y_{1:t}\right)} \tag{6}$$

where, $q\left(s_t|s_{t-1}, y_{1:t}\right)$ is importance distribution and is often assumed to follow a first-order Markov process in which the state transition is independent of the observation. So the weights are updated as $w_t^i = w_{t-1}^i \cdot p\left(y_t|s_t^i\right)$.

Finally, object state can be estimated by taking the particle with the largest weight at each time step.

The implementation process of the proposed kSSDAE-based tracker is given as follows:

```
Algorithm Outdoor Vehicle Tracking
Input: Training samples; Video frame $t$.
  Training SDAE offline;
  Constructing classification neural network;
```

```
    Connecting the encoder part of kSSDAE and a classification layer as shown in **Figure 2**;
    Adding k sparse constraint into classification neural network;
**For** $t = 1, 2, \ldots, N$ frame number **do**
    Sampling particles $S_t = \left\{s_t^i\right\}_{i=1}^{n}$;
    Calculate confidence for each particle by (4);
  **If** $t = 1$
    Sampling positive and negative samples;
    Fine-tuning classification neural network;
  **end**
  **If** $\max\left(c_t^i\right) \le \tau$
    Sampling positive and negative samples;
    Fine-tuning classification neural network.
  **Else**
    Calculating observation probability by (5);
    Updating weights by (6);
    $t = t + 1$.
  **end**
  **end**
**Output**: Object state
```

4. Experiments

In this section, we conducted a quantitative experiment to evaluate the proposed tracker (kSSDAE-T) on a popular single-object online tracking benchmark [28]. The benchmark data set provides 51 fully annotated video sequences that have the 11 challenging attributes. Most of these attributes exist in the real scene of outdoor vehicles. In order to better demonstrate the performance of our tracker, we compare our tracker with other three popular trackers, including deep learning tracker (DLT) [15], multi-task tracker (MTT) [29], and Circulant Structure of Tracking-by-Detection with Kernels (CSK) [30].

The main related parameters in our experiment are set as follows.

- Learning rate is set to 0.2; sparsity k is set to 40.

- Standard deviation of the conservation likelihood σ is set to 0.001.
- The number of particles is 1000, and the particle's confidence threshold τ is set to 0.8.
- We use momentum gradient method to optimize the network parameters, and the momentum parameter is set to 0.5.

4.1. Quantitative evaluation

In this chapter, we adopt two quantitative evaluation indicators: one-pass evaluation (OPE) of tracking precision and success rate [28]. The precision takes the position error as the benchmark, and the precision plot shows the percentage of frames whose estimation position error is less than the given threshold, and the horizontal axis of the precision plot is scaled to the range [0,50]. The success rate is based on the overlap rate, and the success plot counts the number of successful frames whose overlap is greater than the given threshold, and the horizontal axis of the success rate is scaled to the range [0,1]. We use the score for the threshold = 20 pixels of each precision plot and the area under curve (AUC) of each success plot to rank trackers and one-pass evaluation (OPE) for robustness evaluation. The scores and rankings of precision and success rate for four trackers on the overall performance and the 11 attributes performance are shown in **Table 1**, and the best tracking results corresponding to the overall performance and the 11 attributes are marked in bold, and the ranking score is shown after "\." In **Table 1**, SV: scale variation, OV: out-of-view, OPR: out-of-plane rotation, OCC: occlusion, LR: low resolution, IPR: in -plane rotation, IV: illumination variation, DEF: deformation, MB: motion blur, BC: background clutters, FM: fast motion. The precision plot and success rate plot of four trackers on overall performance is shown in **Figure 3**. The precision plots and success plots of

	kSSDAE-T (Ours)		**DLT**		**CSK**		**MTT**	
	Precision	**Success rate**	**Precision**	**Success rate**	**Precision**	**Success rate**	**Precision**	**Success rate**
Overall	**0.585\1**	**0.522\1**	0.550\2	0.499\2	0.545\3	0.443\4	0.475\4	0.445\3
SV	0.597\2	0.535\2	**0.602\1**	**0.547\1**	0.503\3	0.352\4	0.461\4	0.398\3
OV	**0.571\1**	0.537\2	0.526\2	**0.552\1**	0.379\3	0.410\3	0.374\4	0.392\4
OPR	**0.576\1**	**0.492\1**	0.527\3	0.464\2	0.540\2	0.439\3	0.473\4	0.423\4
OCC	**0.545\1**	**0.504\1**	0.532\2	0.502\2	0.500\3	0.404\4	0.426\4	0.422\3
LR	0.383\3	0.358\3	0.309\4	0.297\4	0.411\2	0.397\2	**0.510\1**	**0.506\1**
IPR	**0.551\1**	**0.479\1**	0.502\4	0.439\4	0.547\2	0.457\3	0.522\3	0.463\2
IV	**0.543\1**	**0.480\1**	0.514\2	0.472\2	0.481\3	0.388\3	0.351\4	0.337\4
DEF	**0.500\1**	**0.422\1**	0.433\3	0.389\2	0.476\2	0.370\3	0.332\4	0.334\4
MB	**0.359\1**	0.309\3	0.328\3	0.321\2	0.342\2	**0.336\1**	0.308\4	0.288\4
BC	0.528\2	0.440\2	0.455\3	0.398\4	**0.585\1**	**0.491\1**	0.424\4	0.411\3
FM	**0.460\1**	**0.421\1**	0.417\2	0.418\2	0.381\4	0.380\4	0.401\3	0.385\3

Table 1. Tracking performance on four trackers.

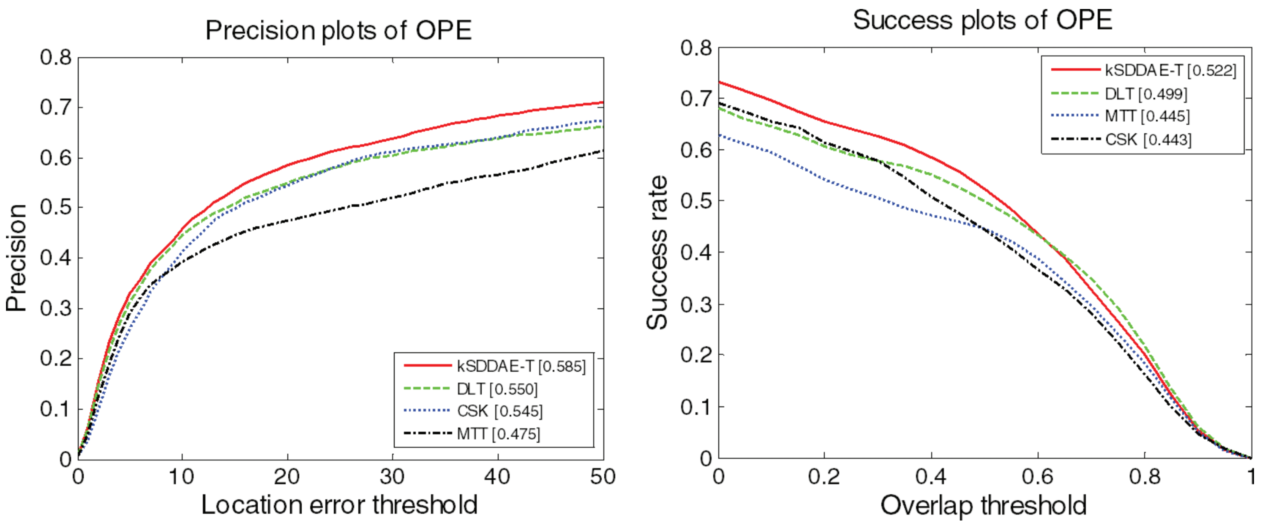


Figure 3. The overall performance of precision plot and success rate plot on four trackers.

four trackers on 11 attributes performance are shown in **Figure 4**. In order to analyze the performance of the tracker in every challenging attribute, [28] has marked the characteristics of each sequence and constructed subsets of the sequences with different saliency characteristics. For example, the OCC subset includes 29 sequences; it can be used for analyzing the ability of the tracker to handle occlusion problem. In **Figure 4**, the number that appears in the legend of each graph represents the ordinal number of sequence subset.

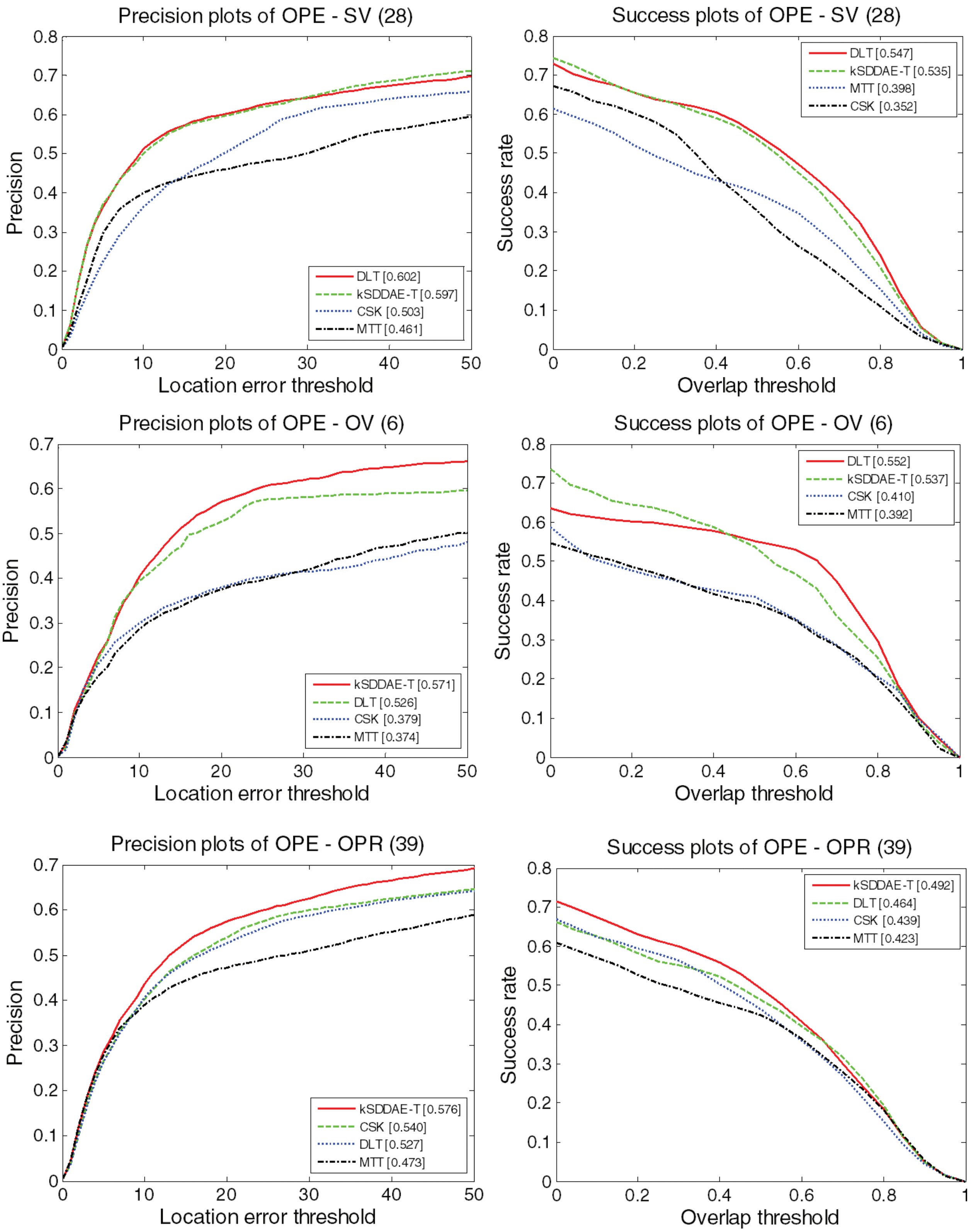
In overall performance of precision and success rate, our tracker is significant higher than the other three trackers. The performance of our tracker ranks first in 8 out of 11 attributes on precision as shown in **Table 1**. At the same time, the performance of our tracker ranks first in 6 out of 11 attributes on success rate. For the other attributes, except for LR attribute, our tracker has the success rate very close to the best on SV and BC attributes. The success rate of our tracker ranks 3 on MB attribute, but it is only lower than the best (CSK), 2.7%. Therefore, it can be concluded that the proposed kSSDAE-T tracker is the best compared with DLT, CSK, and MTT.

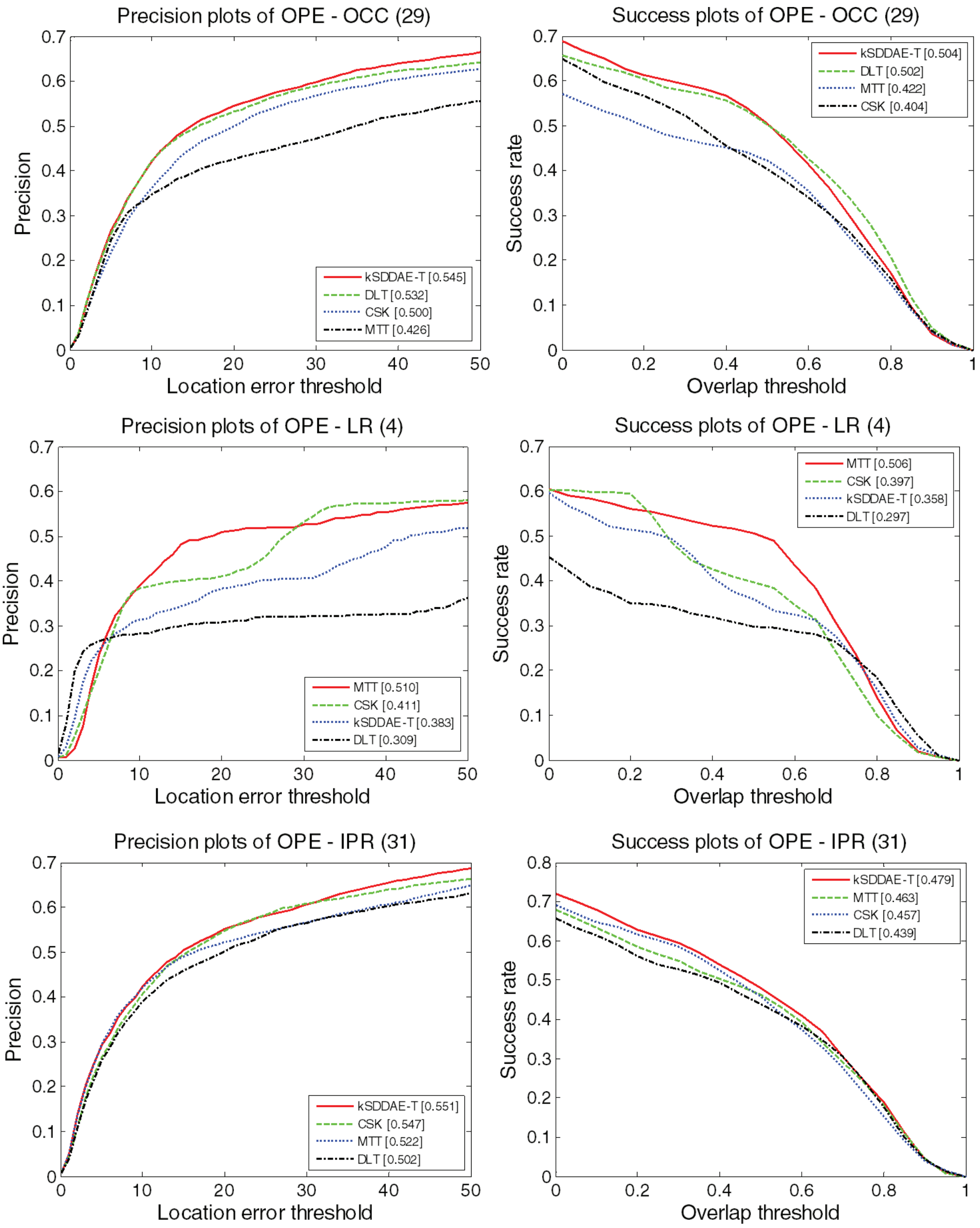
According to the precision plot and success rate plot of the four properties based on OCC, IV, MB, and FM attributes, we can see that our tracker can handle the appearance changes caused by most of outdoor environmental factors.

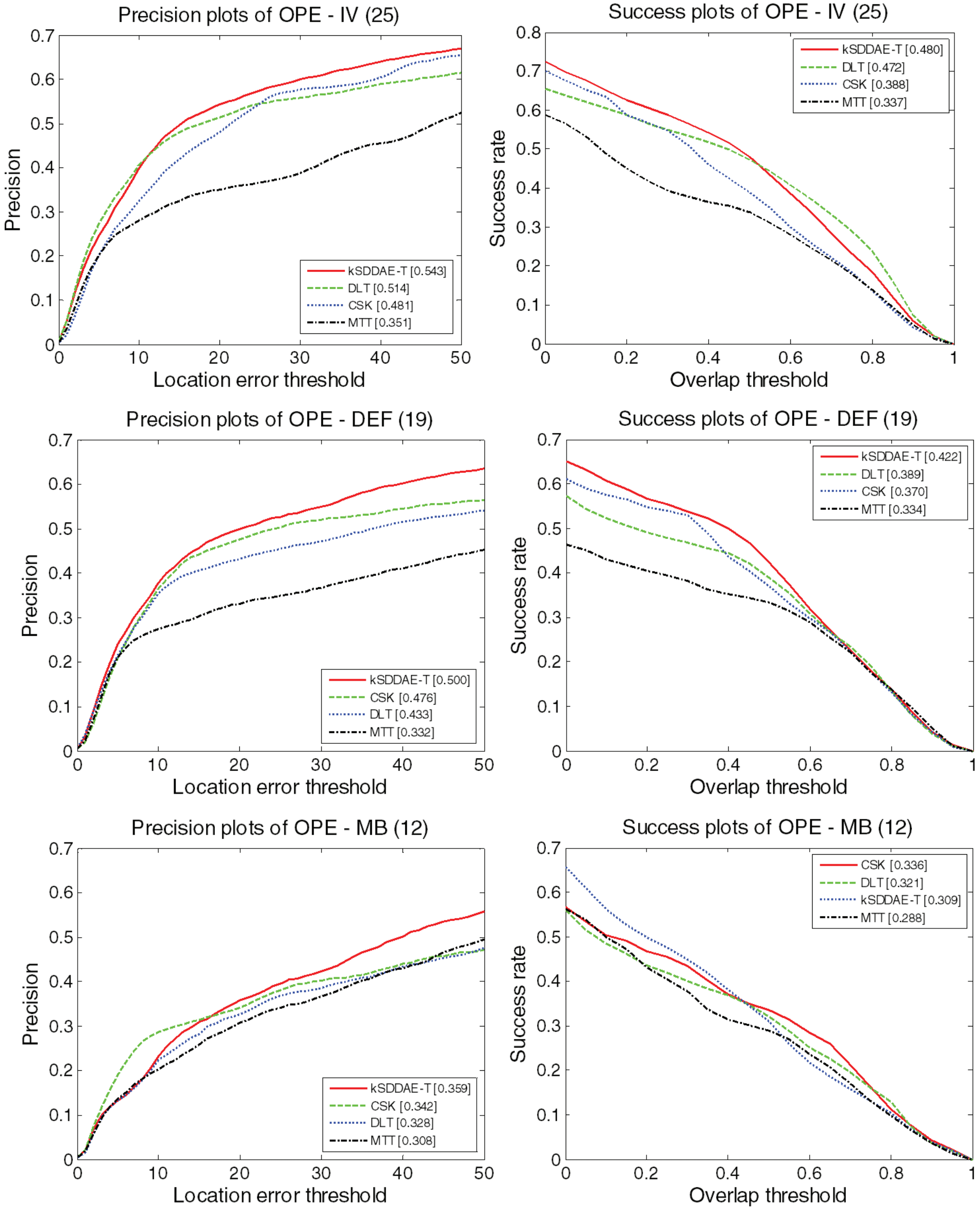
All in all, the proposed tracker can learn the invariable features of the object appearance and deal with the problem of object appearance changing caused by most of the outdoor complex environments. It can achieve better tracking results under most outdoor challenging conditions.

4.2. Qualitative evaluation

In order to further verify the effectiveness of the proposed tracking method in real scenarios, we compared the four trackers (proposed kSSDAE-based tracker, DLT, CSK, and MTT) on four outdoor vehicle sequences in real scenarios (Car4, CarDark, CarScale, and Suv). The attributes


Precision plots of OPE - SV (28)
Precision
Location error threshold
DLT [0.602]
kSDDAE-T [0.597]
CSK [0.503]
MTT [0.461]
Success plots of OPE - SV (28)
Success rate
Overlap threshold
DLT [0.547]
kSDDAE-T [0.535]
MTT [0.398]
CSK [0.352]
Precision plots of OPE - OV (6)
Precision
Location error threshold
kSDDAE-T [0.571]
DLT [0.526]
CSK [0.379]
MTT [0.374]
Success plots of OPE - OV (6)
Success rate
Overlap threshold
DLT [0.552]
kSDDAE-T [0.537]
CSK [0.410]
MTT [0.392]
Precision plots of OPE - OPR (39)
Precision
Location error threshold
kSDDAE-T [0.576]
CSK [0.540]
DLT [0.527]
MTT [0.473]
Success plots of OPE - OPR (39)
Success rate
Overlap threshold
kSDDAE-T [0.492]
DLT [0.464]
CSK [0.439]
MTT [0.423]
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
10
20
30
40
50
1


Precision plots of OPE - OCC (29)
Precision
Location error threshold
kSDDAE-T [0.545]
DLT [0.532]
CSK [0.500]
MTT [0.426]
Success plots of OPE - OCC (29)
Success rate
Overlap threshold
kSDDAE-T [0.504]
DLT [0.502]
MTT [0.422]
CSK [0.404]
Precision plots of OPE - LR (4)
Precision
Location error threshold
MTT [0.510]
CSK [0.411]
kSDDAE-T [0.383]
DLT [0.309]
Success plots of OPE - LR (4)
Success rate
Overlap threshold
MTT [0.506]
CSK [0.397]
kSDDAE-T [0.358]
DLT [0.297]
Precision plots of OPE - IPR (31)
Precision
Location error threshold
kSDDAE-T [0.551]
CSK [0.547]
MTT [0.522]
DLT [0.502]
Success plots of OPE - IPR (31)
Success rate
Overlap threshold
kSDDAE-T [0.479]
MTT [0.463]
CSK [0.457]
DLT [0.439]


Precision plots of OPE - IV (25)
Precision
Location error threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
kSDDAE-T [0.543]
DLT [0.514]
CSK [0.481]
MTT [0.351]
Success plots of OPE - IV (25)
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
0.2
0.4
0.6
0.8
1
kSDDAE-T [0.480]
DLT [0.472]
CSK [0.388]
MTT [0.337]
Precision plots of OPE - DEF (19)
Precision
Location error threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
kSDDAE-T [0.500]
CSK [0.476]
DLT [0.433]
MTT [0.332]
Success plots of OPE - DEF (19)
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
0.2
0.4
0.6
0.8
1
kSDDAE-T [0.422]
DLT [0.389]
CSK [0.370]
MTT [0.334]
Precision plots of OPE - MB (12)
Precision
Location error threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50
kSDDAE-T [0.359]
CSK [0.342]
DLT [0.328]
MTT [0.308]
Success plots of OPE - MB (12)
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
0.2
0.4
0.6
0.8
1
CSK [0.336]
DLT [0.321]
kSDDAE-T [0.309]
MTT [0.288]

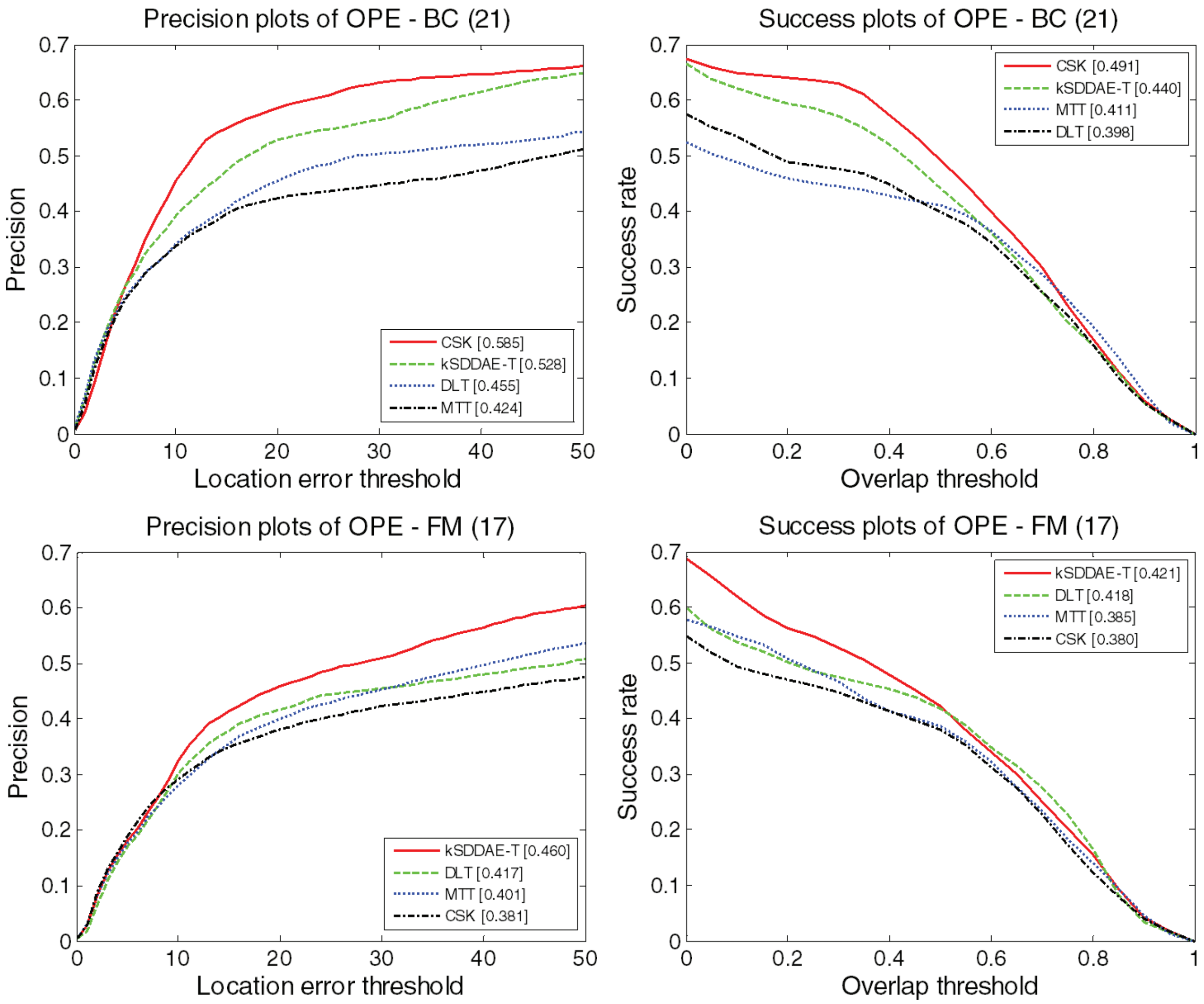


Figure 4. The precision plots and success plots of four trackers on 11 attributes performance (SV, OV, OPR, OCC, LR, IPR, IV, DEF, MB, BC, and FM).

Sequence	Attribute
Car4	IV, SV
CarDark	IV, BC
CarScale	SV, OCC, FM, IPR, OPR
Suv	OCC, IPR, OV

Table 2. Attributes of four sequences.

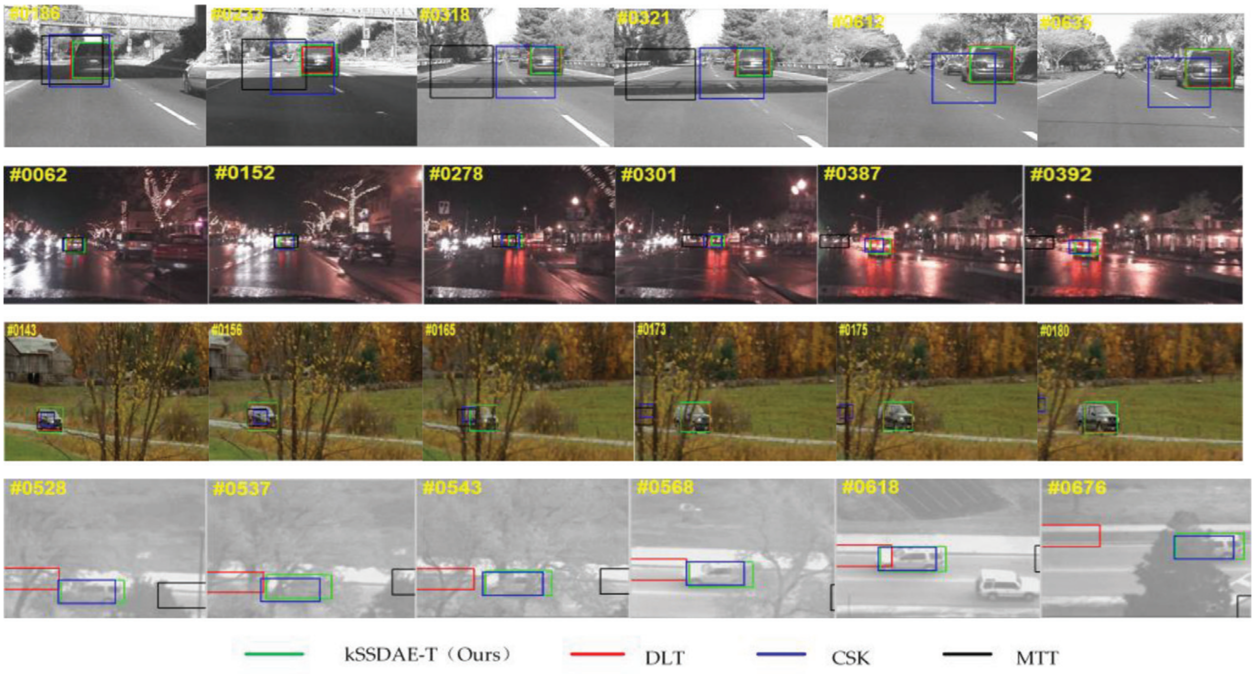


Figure 5. The sampled tracking results. Frame numbers are shown in the top left of each figure.

of four sequences are listed in **Table 2**. The partial tracking results of the four video sequences are shown in **Figure 5**.

In the video sequence Car4, when emerging IV and OCC near the #186, #233, and #318 frames, it can be seen from the tracking results that the CSK and the MTT tracker have different degrees of object drift. But, our tracker and DLT have achieved better tracking results. In addition, our tracker can also accurately track the target vehicle when SV emerges in #321, #612, and #635 frames. In the video sequence CarDark, our tracker can still perform effective tracking when the IV and BC emerging in #62, #152, #278, #301, #387, and #392 frames, while the MTT and CSK trackers have track drift at #301 frame, and at #387 frame, they completely lost the target vehicle. In the video sequence CarScale, our tracker can still show great performance when OCC was occurred in #165 and #175 frames, but CSK tracker failed. In the video sequence Suv, despite the OCC and similar background interference, our tracker can still accurately track the target vehicle.

To summarize, the proposed kSSDAE-based tracker can perform well in most complex outdoor environment.

5. Conclusion

In this chapter, we propose an improved auto-encoder–based approach for robust outdoor vehicle visual tracking. Our tracker can adapt the change of the object appearance during the tracking. The quantitative analysis on a standard evaluation platform shows that our tracker has a better tracking performance compared with the other three state-of-the-art trackers and has higher tracking precision in most of the outdoor vehicle tracking challenges. The qualitative analysis on four outdoor vehicle sequences in real scenarios shows that our tracker can work well in most complex outdoor environment.

The unsupervised training of kSSDAE requires that bottom image cannot be too large, otherwise it will consume a lot of training time. The training data are obtained by down-sampling directly from a full-sized image leading to information loss. In order to avoid loss of input image information, we can further improve the performance of outdoor vehicle tracking algorithms by using stacked convolutional auto-encoder (SCAE) [31] to take the outdoor vehicle tracking algorithm into application of life and industry.

Note: this chapter is an extended version of [32].

Acknowledgements

This work is supported by the Chinese National Natural Science Foundation under Grant No. 61873200,61833013,61573282.

Author details

Jing Xin[1]*, Xing Du[1], Yaqian Shi[1], Jian Zhang[2] and Ding Liu[1]

*Address all correspondence to: xinj@xaut.edu.cn

1 Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi’an University of Technology, Xi’an, P.R. China

2 Faculty of Engineering and Information Technology, University of Technology (UTS), Sydney, Australia

References

[1] Rad R et al. Real time classification and tracking of multiple vehicles in highways. Pattern Recognition Letters. 2005;**26**(10):1597-1607

[2] Zhang W, Wu QMJ, et al. Multilevel framework to detect and handle vehicle occlusion. IEEE Transactions on Intelligent Transportation Systems. 2008;**9**(1):161-174

[3] Faro A, Giordano D, et al. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. IEEE Transactions on Intelligent Transportation Systems. 2011;**12**(4):1398-1412

[4] Xin J, Liu XD, Ran BJ, et al. Adaptive multiple cues integration for robust outdoor vehicle visual tracking. In: The 34th Chinese Control Conference; July 28–30, Chengdu, China; 2015. pp. 4913-4918

[5] Wang L, Ouyang W, Wang X, et al. Visual tracking with fully convolutional networks. In: IEEE International Conference on Computer Vision. IEEE; 2015. pp. 3119-3127

[6] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking. Computer Vision and Pattern Recognition. 2016:4293-4302

[7] Olshausen BA, Field DJ. Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research. 1997;**37**(23):3311-3325

[8] Rehn M, Sommer FT. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. Journal of Computational Neuroscience. 2007;**22**(2):135-146

[9] Zheng Y et al. Deep learning and its new progress in object and behavior recognition. Journal of Image and Graphics. 2014;**19**(2):175-184

[10] Makhzani A, Frey B. k-sparse autoencoders. In: International Conference on Learning Representations, ICLR, 2014. pp. 235-241

[11] Fan J et al. Human tracking using convolutional neural networks. IEEE Transactions on Neural Networks. 2010;**21**(10):1610-1623

[12] Jin J, Dundar A, Bates J, et al. Tracking with deep neural networks. In: Conference on Information Sciences and Systems. 2013. pp. 1-5

[13] Hong S, You T, et al. Online tracking by learning discriminative saliency map with convolutional neural network. Computer Science. 2015. pp. 597-606

[14] Wang L, Liu T, et al. Video tracking using learned hierarchical feature. IEEE Transactions on Image Processing. 2015;**24**(4):1424-1435

[15] Wang N, Li S, Gupta A, et al. Transferring rich feature hierarchies for robust visual tracking. arXiv preprint arXiv:1501.04587, 2015

[16] Doulamis N, Voulodimos A. FAST-MDL: Fast adaptive supervised training of multilayered deep learning models for consistent object tracking and classification[C]. In: IEEE International Conference on Imaging Systems and Techniques (IST), Chania; 2016. pp. 318-323

[17] Guan H, Xue XY, An ZY. Advances on application of deep learning for video object tracking. Acta Automatica Sinica. 2016;**42**(6):834-847

[18] Zhang P, Zhuo T, Huang W, et al. Online object tracking based on CNN with spatial-temporal saliency guided sampling. Neurocomputing. 2017:**17**(4):1-13

[19] Wang N, Yeung DY. Learning a deep compact image representation for visual tracking. In: International Conference on Neural Information Processing Systems; 2013. pp. 809-817

[20] Zhou X, Xie L, Zhang P, Zhang Y. An ensemble of deep neural networks for object tracking. In: IEEE International Conference on Image Processing; 2014. pp. 843-847

[21] Cheng S, Sun JX, et al. Target tracking based on incremental deep learning. Optics and Precision Engineering. 2015;**23**(4):1161-1170

[22] Cheng S, Cao YG, Sun JX, et al. Target tracking based on enhanced flock of tracker and deep learning. Journal of Electronics and Information Technology. 2015;**37**(7):1146-1153

[23] Cheng S, Sun JX, et al. Target tracking based on multiple instance deep learning. Journal of Electronics and Information Technology. 2015;**37**:12

[24] Bo D, Hou ZQ, Yu WS, et al. Local patch tracking algorithm based on stacked denoising autoencoder. Control Theory & Applications. 2017;**34**(6):829-836

[25] Hua W, Mu D, Guo D, et al. Visual tracking based on stacked denoising autoencoder network with genetic algorithm optimization. Multimedia Tools & Applications. 2017;**2**:1-17

[26] Torralba A et al. 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008; **30**(11):1958-1970

[27] Richard MD et al. Neural network classifiers estimate Bayesian a posterior probabilities. Neural Computation. 1991;**3**(4):461-483

[28] Wu Y, Lim J. Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition. 2013;**9**(4):2411-2418

[29] Ahuja N. Robust visual tracking via multi-task sparse learning. In: IEEE Conference on Computer Vision and Pattern Recognition. 2012;**157**:2042-2049

[30] Henriques JF, Rui C, et al. Exploiting the circulant structure of tracking-by-detection with kernels. Computer Vision. 2012;**7575**:702-715

[31] Masci J, Meier U, Dan C, et al. Stacked convolutional autoencoders for hierarchical feature extraction. Artificial Neural Networks and Machine Learning-ICANN 2011. In: International Conference on Artificial Neural Networks; 2011. pp. 52-59

[32] Xin J, Du X, Zhang J. Deep learning for robust outdoor vehicle visual tracking. In: IEEE International Conference on Multimedia and Expo (ICME 2017); July 10–14, 2017. Hong Kong, China. pp. 613-618