IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 30, 2019, accepted August 5, 2019, date of publication August 20, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936433

Classifier Adaptive Fusion: Deep Learning for Robust Outdoor Vehicle Visual Tracking

JING XIN, XING DU, AND YAQIAN SHI
Shaanxi Key Laboratory of Integrated and Intelligent Navigation, Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an 710048, China

Corresponding author: Jing Xin (xinj@xaut.edu.cn)

This work was supported in part by the Chinese National Natural Science Foundation under Grant 61873200 and Grant 61833013, and in part by the Open Project of the Shaanxi Key Laboratory of Integrated and Intelligent Navigation under Grant SKLIIN-20180205.

ABSTRACT Deep auto-encoder (DAE) models have been successfully used in object tracking due to its strong capability of feature representation. However, single deep auto-encoder model would not be robust enough to represent the appearance model of outdoor vehicle for its harsh working environment, such as illumination variation, occlusion, cluttered background and so on. In this paper, a novel multiple-DAE-based tracking approach, that is, classifier adaptive fusion for robust outdoor vehicle visual tracking approach is proposed under particle filter framework. Firstly, two deep auto-encoders are offline trained by gray-scale image and gradient image of the raw training images, respectively to obtain the stronger feature representation of gray-scale image and gradient image. Secondly, two classifiers are constructed using the encoder of the two well-trained deep auto-encoders and the output of the each classifier is used to compute the confidence of the corresponding particles. Finally, the confidence output of the two classifiers is fused and applied in online tracking, where, the fusion weight of the each classifier is computed according to the distribution of particles represented by different classifier. Extensive tracking experiments conducted on visual tracking benchmark (VTB) show that the proposed tracking algorithm outperforms 9 popular tracking algorithms in the challenge scenes of outdoor vehicle tracking such as illumination variation, occlusion, cluttered background and scale variation.

INDEX TERMS Classifier adaptive fusion (CAF), multiple deep auto-encoder, outdoor vehicle visual tracking, particle filter.

I. INTRODUCTION

Video object tracking is an important research issue in computer vision. It has been widely used in intelligent transportation system (ITS) for obtaining the state information of the outdoor vehicle. However, outdoor vehicle tracking is facing severe challenges due to the complex and changeable outdoor environments, such as illumination variation, occlusion, cluttered background and so on [1], [2]. These challenges are attributed to dramatic changes of object appearance. The quality of the object appearance model heavily depends on the performance of feature extraction. Feature extraction can effectively encode the appearance of the object and map it from the original image space to a feature space, which provides the basis for the representation of the object appearance model. Generally, the features of video object tracking algorithm are divided into two categories which are artificial features and learning features, respectively [3].Artificial feature is a kind of simple and intuitive feature representation, but it needs to be redesigned according to different tasks, and it heavily relies on the knowledge and skills of the designers. Deep learning model is a powerful automatic feature extraction method, which learns high-level abstract features which are high-dimensional and distinguishable from the underlying features such as color and edge through multiple nonlinear transformations.

At present, deep learning model has been successfully applied to video object tracking. Wang and Yeung *et al.* [4] firstly proposed a deep learning tracker (DLT) in 2013. It learned general image feature representation offline by stacked denoising auto-encoder (SDAE), and applied the learned feature representation to online tracking by transfer learning. Deep-learning-based tracking algorithms often use grayscale image of raw image to train deep neural networks, which can obtain deeper feature representation of grayscale image. Generally speaking, features extracted by grayscale

The associate editor coordinating the review of this article and approving it for publication was Sabah Mohammed.

image are more robustness to object rotation, non-rigid deformation and occlusion, yet sensitive to illumination variation. In the outdoor vehicle tracking process, strong illumination variation often leads to huge changes in object appearance model, which reduces the performance of the tracking algorithm. Complementing the deep feature information enables the deep learning model to express the object appearance model more effectively, thereby achieving more robust tracking in many complex environments.

In recent years, researchers have proposed a variety of fusion strategies to complement the deep feature information, which shows better performance in various computer vision tasks. Huang and Yeung [5] designed a densely connected structure to implement shallow feature reuse of convolutional neural networks. Direct connections are adopted between any two layers in a dense convolutional neural network. The feature map learned in the current layer will be directly transmitted to all subsequent layers as input, so it can more effectively fuse and utilize the features of each layer in the deep network. Subsequently, Wang *et al.* [6] used densely concatenated convolutional neural networks for human pose estimation, and achieved high accuracy on the human pose estimation data sets MPII and LSP. Li and Zhou [7] designed a feature fusion module to fuse features from different layers of convolutional neural network, which effectively improved the accuracy of single object detection. In this fusion method, the features learned by different layers in deep neural network are fused directly and effectively to achieve the complementation of deep feature information. In addition, another way of complementing the multi-deep information is to utilize multiple deep neural network models to learn the deep features and then fuse their learning results. Dan *et al.* [8] fused the classification results predicted by multiple convolutional neural network models in average manner for traffic identification, and improves the recognition accuracy to 99.46%. Stolar *et al.* [9] and Zhang *et al.* [10] respectively used fixed weight fusion and average fusion of multi-convolutional neural networks for emotion recognition and face alignment. Subsequently, Yu *et al.* [11] used fusion strategies of average, maximum, majority and median to fuse multi-convolution neural networks for classification of medical images. Compared with a single deep neural network, the manner of multi-deep neural network fusion improves the accuracy of image classification to a certain extent. In the video classification task, in order to effectively utilize both temporal and spatial information of video frames, Peng *et al.* [12] decomposes the input video into frame images and optical flow images, and then learns the static and dynamic features of the input video using two spatial-temporal attention models composed of convolutional neural networks. Finally, the prediction scores of the frame images network and the optical flow images network are fused by the static-motion collaborative model. This fusion method makes use of frame images and optical flow images to complement the static and dynamic information of video, which greatly improves the accuracy of video classification.

The above two fusion methods can realize complementarity of the deep feature information and improve the performance of various computer vision tasks. For object tracking tasks, it is more suitable for the second multi-deep learning model fusion method. Because the second method can utilize other models to maintain the stability of algorithm when one model of them fails due to a great changes of the object appearance, it can improve the robustness of tracking in complex environments. The existing multiple deep learning models fusion methods mostly adopt strategy with fixed weight fusion or average fusion. This fusion strategy can't adaptively adjust the fusion weight between the models in time when a model changes significantly, so that the fusion result becomes unreliable and even causes the algorithm to fail. In order to solve the problem, Agostinelli *et al.* [13] proposed an adaptive fusion strategy, which first uses the quadratic program to predict the weight of each deep learning model and then learns how to predict the fusion weight by using the radial basis function (RBF), and finally use the multiple deep neural network with adaptive weight fusion for image denoising. This method solves the shortcomings of multiple model fusion with the fixed weight, but it needs a lot of computing time so that it is not suitable for the object tracking problem with strong real-time requirements. However, compared with other fixed weights fusion or average fusion strategies of multi-deep neural network, this adaptive fusion strategy of multi-deep learning model provides a new idea to solve the challenges in outdoor vehicle tracking.

In order to solve the problem that dramatic changes of object appearance caused by challenging environment factors in the outdoor vehicle tracking process, an outdoor vehicle tracking algorithm based on multi-deep learning model adaptive fusion is proposed in this paper. The main contributions of the proposed algorithm are as follows:

- A new multi-deep learning model fusion method is proposed, which fuses the results of the classifier trained with gray image and the classifier trained with gradient image to achieve multiple model information complementation for solving the outdoor vehicle tracking problem under the challenging environmental such as illumination variation, occlusion, rotation, and fast motion.
- Under the particle filter framework, a new classifier adaptive fusion tracking algorithm is proposed. That is, the fusion weight of the classifier is adaptively calculated according to the distribution of the particles characterized by the classifier, so that the fusion weight of each model can be adjusted in time to improve the robustness of the tracking algorithm when the object appearance greatly changes.
- Overall performance and attribute-based quantitative experimental results conducted on the all 50 video sequences with most of the outdoor challenging factors (such as IV, SV, OCC, etc.) on the OTB50 data set of the object tracking evaluation benchmark VTB2013 and the qualitative experimental results on the 4 representative

challenging outdoor vehicle sequences show that the proposed algorithm exhibits good tracking performance compared to the other 9 state-of-the-art trackers.

The remainder of this paper is organized as follows. In Section 2, we review the related works about existing deep-learning-model-based video object tracking algorithms. In Section 3, we give a brief introduction on system overview of the proposed multi-deep-auto-encoder-adaptive-fusion-based tracking approach, and a detailed introduction on the principle of the 3 main part of the system, including the generic feature representation, classifier adaptive fusion, and online tracking. The experimental results and performance analysis are provided in Section 4, where the algorithm performance is verified based on quantitative evaluation, qualitative evaluation, and fusion weight variation analysis. Finally, summary and future works are given in Section 5.

II. RELATED WORKS

Recently, researches have successfully applied the deep learning model to the video object tracking field and proposed many deep-learning-model-based object tracking methods. These methods mainly follow two ideas [3]: (1) Tracking combined offline training with online fine tuning, first train the deep neural networks by offline manners on video sequence data sets or large-scale natural image data sets, and then use online data to fine-tune deep neural network in online tracking; (2) Purely online tracking, the network structure is simplified so that the video sequence can be directly tracked online without relying on offline training.

A. TRACKING COMBINED OFFLINE TRAINING WITH ONLINE FINE TUNING

It is well know that deep network often needs a large amount of training data for effective learning, but only the bounding-box of the first frame is provided as training data in the video object tracking, which is very difficult for the training of deep learning model. At present, there are two main ways to solve this problem. One is to use tracking data sets for offline training of deep learning network, and then fine-tune the network in online tracking to track specific objects adaptively. The other is to use non-tracking data sets such as classification and detection to train the network offline, and then use the idea of transfer learning to apply the results of offline training to tracking tasks and fine-tune the network to adapt to object appearance changes.

In the way of using tracking data sets to train deep network offline, Kuen *et al.* [14] emphasizes temporal correlation learning, using labeled video sequences train deep auto-encoder to learn invariant features. Nam and Han [15] proposes a multi-domain network (MDNet), which offline trains and tests convolutional neural networks alternately using OTB100 data sets and VOT data sets that do not coincide with each other. The MDNet mainly includes shared layers and domain-specific layers. The shared layers are used to learn the general feature representations of the object in the tracking sequence, and the domain-specific layer solves the inconsistent problem of classification objects in different training sequences. The CNN pretrained by multi-domain is fine-tuned in the new sequence to adaptively track specific objects. MDNet achieved high tracking accuracy, but its speed is too slow to meet the real-time requirements of tracking tasks. Subsequently, Tao *et al.* [16] used the tracking video sequence data sets ALOV300 to train the siamese network offline to learn a matching function, that is, after obtaining object information in the first frame, all subsequent frames are sampled and matched with the first frame object information, and the highest score sample is the tracking result of current frame. The network does not need to fine tune the parameters during the online tracking, thus greatly improving the speed of online tracking. However, this method may cause mismatching when the object is occluded or similar background appears.

The above method of using the tracking data sets offline training deep learning model for visual tracking has solved the problem of lack of initial training data in the object tracking to some extent, but the number of existing tracking data sets is still far from enough compared with the massive training data required for deep learning. In order to make up for this shortcoming, some researchers try to use other non-tracking datasets to conduct offline training of deep learning models. Then, according to the idea of transfer learning, the general image features learned offline are applied to online tracking to achieve object tracking tasks. Wang and Yeung [4] proposed the deep learning tracker (DLT), which firstly performs unsupervised offline pre-training on the stacked denoising auto-encoder by using a large-scale natural image data sets to obtain general feature representation capability, and then uses the offline training result for online tracking. In online tracking, a small number of positive and negative samples are collected to fine-tune network parameters. DLT achieves a good tracking effect on OTB50 data sets, but the tracking effect will be greatly affected when the object appearance changes greatly. In order to further improve the tracking accuracy, in our previous work, considering that the response of each neuron in the neural network for visual information is sparse, we proposed a robust outdoor vehicle tracking method based on k-sparse stacked denoising auto-encoder in reference [17]. In this method, k-sparse restriction is introduced into the classification neural network to learn the invariant features of the input image, thereby to enhance the ability of the network to represent the object appearance model. This method improves the accuracy of DLT tracker to a certain extent, but it uses single deep network offline trained by gray image, and tracking drift still occurs in complex environment. Zhou *et al.* [18] incorporated a deep denoising auto-encoder (SDAE) with an online AdaBoost framework for object tracking. This method fuses multiple networks to compensate a single network susceptible to noise interference for the problem of tracking failure, but it increased computational complexity. Dai *et al.* [19] used deep auto-encoder for real-time tracking by simplifying network structure and designing model update strategies. The speed of tracking is improved, but it uses a simpler strategy to update the object appearance model during the online fine-tuning phase, the tracking result is not ideal when the object occurs strong illumination variation. In order to make the tracking results more reliable, in the online fine-tuning stage, Wang *et al.* [20] update network parameters by combining the two CNN of long-term $\mathrm{CNN_L}$ and short-term $\mathrm{CNN_S}$. $\mathrm{CNN_S}$ update frequently, it can respond to changes of object appearance in time. $\mathrm{CNN_L}$ update less, it can be more robustness for error tracking results. It take the most confident results as output by combining $\mathrm{CNN_S}$ and $\mathrm{CNN_L}$. Different from the above, Hua *et al.* [21] optimizes the network parameters through genetic algorithm in the online tracking phase. The use of genetic algorithm in network parameter adjustment helps to avoid the shortcomings of traditional BP algorithm and further enhances the robust performance of the network.

All of the above are object tracking strategies using ''tracking combined offline training with online fine tuning'', that is, firstly the deep neural network model is trained offline, and then the different update strategies are used to fine tune the parameters of the network in actual tracking to achieve more robust tracking. In addition, many scholars have attempted to extend the object state information of the initial frame in the visual tracking or to simplify the deep network structure to achieve purely online tracking.

B. PURELY ONLINE TRACKING

Purely online tracking directly use the tracking video sequence to train the deep neural network. It can learn the effective feature representation of the object in an online manner, and then fine tune the network parameters during subsequent frame tracking. Li *et al.* [22]–[24] proposed an online convolutional neural network for object tracking under the particle filter framework, which does not rely on offline training. It preprocesses the original image into the regularized images and a gradient images with different parameters to obtain more online training samples. In order to further solve the problem of insufficient label training samples in purely online tracking, Zhou *et al.* [25] constructed adaptive appearance model based on convolutional neural network to generate training samples changed over time, and combined convolutional neural network with Metropolis-Hastings re-sampling for online tracking under the particle filter framework. Unlike reference [22]–[25], Zhang *et al.* [26] proposed a convolution neural network model tracking framework (CNT) with two convolution layers by simplifying the structure of convolutional neural networks. The model is simple in structure, and the k-means method and soft shrinkage method are used to directly extract the robust representation of the object appearance model from many normalized image patches of the object area. The tracking effect is better when the object is occluded or deformed, but the tracking performance decreases when the object appearance model changes greatly due to the object moves fast or blurs. Wang *et al.* [27] trained convolutional neural networks through online video images rather than offline images to learn complex motion features for object tracking. The network pays attention to learn the invariant features of object motion, which effectively improves the problem of tracking drift caused by the large changes in appearance during the object motion. In addition, Hu *et al.* [28] proposed a deep metric learning (DML) method under the particle filter framework to further solve the problem that the object appearance emerge the great changes in complex environments. The DML tracker uses a feed-forward deep neural network to learn a nonlinear distance metric, thereby projecting the object appearance template and particles into the same feature space to better classify the object area and background area. The DML tracker adapts a normalized random initialization strategy to initialize weights and biases of networks, so it doesn't need to train the network offline and the initialized network parameters are updated online directly using the tracked video sequence to adapt to the change of the object appearance. Generally speaking, the online training of the network in the purely online tracking only depends on the position information of the object in the previous frame, and the information may emerge random noise to cause the model to be over-fitting. In order to solve this problem, Li *et al.* [29] combines multi-task convolution neural network with bagging for purely online object tracking. The algorithm can effectively deal with the over-fitting problem caused by the sample noise and the uncertainty of the random strategy training in the purely online tracking, thus further improving the tracking robustness.

Although researchers have taken various approaches to achieve purely online tracking, it needs to train deep learning model directly online, so it is difficult to achieve a good balance between tracking accuracy and tracking speed. At present, most fast trackers still need to rely on offline training. However, the key to the success of visual tracking method based on offline training is how to use the powerful feature extraction ability of deep learning model to express the tracking object appearance model more robustly.

Considering that the features extracted from gray image are robust to rotation, non-rigid transformation and occlusion of the object, but sensitive to illumination variation. The features extracted from gradient image are not sensitive to background and illumination variation, but are sensitive to motion. Therefore, a multi-deep-auto-encoder-adaptive-fusion-based outdoor vehicle visual tracking algorithm is proposed in this paper. Firstly, the two deep auto-encoder models are unsupervised trained by the gray-scale image and the gradient image of the raw training data, respectively. Then, the two classifiers are constructed and adaptively fused according to the training results. Finally, the fusion result is used in the object tracking under the particle filter framework. Extensive tracking experimental results show that the proposed multi-deep-auto-encoder-adaptive-fusion-based outdoor vehicle visual tracking algorithm can realize the robust tracking of outdoor vehicles in complex environments.

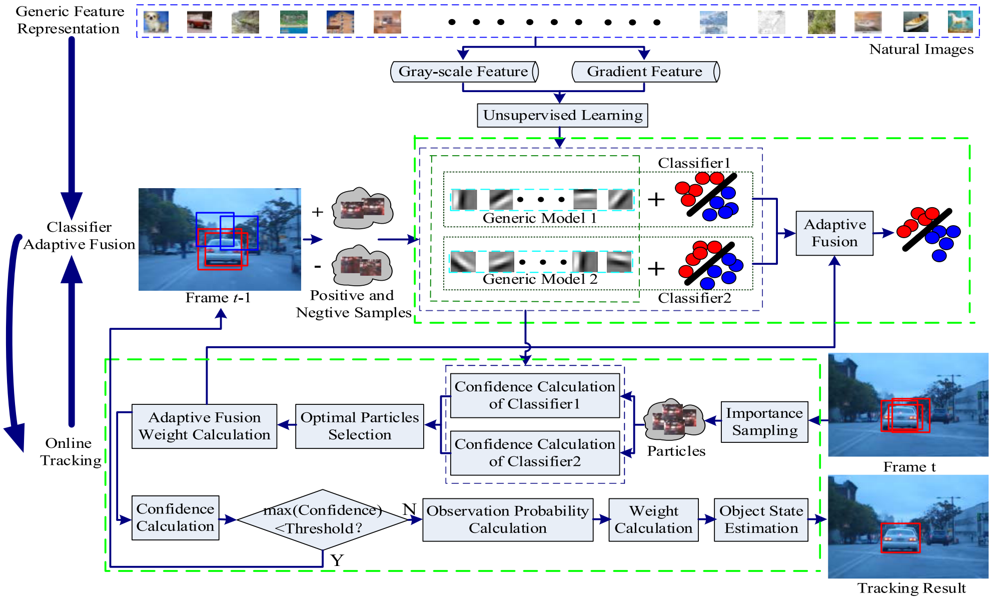

Generic Feature Representation
Natural Images
Gray-scale Feature
Gradient Feature
Unsupervised Learning
Classifier1
Generic Model 1
Generic Model 2
Classifier2
Adaptive Fusion
Classifier Adaptive Fusion
Frame *t*-1
+
-
Positive and Negtive Samples
Online Tracking
Confidence Calculation of Classifier1
Confidence Calculation of Classifier2
Adaptive Fusion Weight Calculation
Optimal Particles Selection
Particles
Importance Sampling
Frame t
Confidence Calculation
max(Confidence) <Threshold?
N
Y
Observation Probability Calculation
Weight Calculation
Object State Estimation
Tracking Result

FIGURE 1. Overview of the proposed multi-deep-auto-encoder-adaptive-fusion-based outdoor vehicle visual tracking algorithm.

III. OUR TRACKER

The overview of the proposed multi-deep-auto-encoder-adaptive-fusion-based outdoor vehicle visual tracking algorithm is shown in Fig. 1. We firstly train the two deep auto-encoders using the gray-scale image and gradient image of the raw training images in an unsupervised and offline way and construct the two classifiers according to the corresponding training results, and then compute adaptive fusion weight on the basis of particle distribution represented by corresponding classifier. Finally, we apply the fusion result to the online tracking.

The whole algorithm is consisted of three parts which are generic feature representation, classifier adaptive fusion, online tracking, respectively.

The detailed description of the function and implementation of each part are as following:

A. GENERIC FEATURE REPRESENTATION

The main purposes of the generic feature representation part is to learn the object general feature representation by training samples, namely, mapping the image space to a feature space by deep neural network, and converting the image to the expression that is conducive to demand. In this paper, we train the two deep auto-encoders using the gray-scale image and the gradient image of the raw training images, which can obtain more essential feature representation of gray-scale image and gradient image. The architecture of the deep auto-encoder is shown in Fig. 2.

We select 0.3 million images from Tiny Images data set randomly for the offline training of deep auto-encoder. The data set contains 80 million images, each of them has a size of 32 × 32, 32 × 32 and most of them exist in real scene. We adopt two ways for data pre-processing: (1) Obtaining gray-scale images from original images, then vectoring and normalizing the gray-scale images; (2) Calculating gradient on the foundation of gray-scale images, and vectoring and normalizing the gradient magnitude. In this paper, we use the above two processed images for the unsupervised training of two deep auto-encoders, respectively.

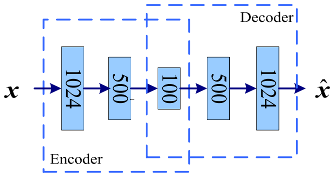


FIGURE 2. Architecture of deep auto-encoder (1024-500-100-500-1024).

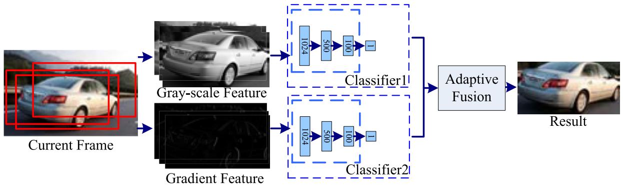


FIGURE 3. Framework of classifier adaptive fusion.

B. CLASSIFIER ADAPTIVE FUSION

In this section, we will continue to introduce the proposed multi-classifier adaptive fusion strategy. Each classifier is constructed by connecting the encoder of the well-trained deep auto-encoder with a classifier layer, and it needs to be fine-tuned using the positive and negative training samples in the first frame to adapt to the change of object appearance in tracking process. The framework of the proposed classifier adaptive fusion is shown in Fig. 3.

The weight of each classifier represents its importance in tracking. Under the particle filtering framework, the reliability of each classifier for object state estimation is mainly reflected in the distribution of particle represented by classifier. The more concentrated particles are in space, the smaller space variance of particles is, the closer particle mean is to object position, and the more particles can reflect the real situation, the greater weight of the classifier has. So the fusion weight can be determined by the distribution of particles represented by different classifiers.

We sort the confidence according to NN particles for each classifier j, and select several particles with maximum confidence as optimal particles. For each classifier j, we chose the first 10% particles with maximum confidence as the optimum particles $\left\{\mathrm{s}_{\mathrm{t}}^{\mathrm{i}}\right\}_{\mathrm{i}=1}^{\mathrm{n}}$, $n= 10\% \times N$. The state mean and state variance of the optimum particles are calculated according to Eq. (1) and Eq. (2).

$$u_j = \frac{1}{n}\sum_{i=1}^{n} s_t^{i,j} \tag{1}$$

$$\sigma_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(s_t^{i,j} - u_j\right)^T \left(s_t^{i,j} - u_j\right) \tag{2}$$

where, u_j is the state mean of optimum particles represented by classifier j, σ_j is the state variance of optimum particles represented by classifier j, and $j= 1, 2, 3 \ldots, M$, we select $M= 2$.

The overall mean u of the optimum particles can be calculated by the following Eq. (3).

$$u = \frac{1}{2}\sum_{j=1}^{2} u_j \tag{3}$$

The weight a_j of each classifier can be calculated and normalized by following Eq. (4) and Eq. (5), respectively.

$$\hat{a}_j = \frac{1}{\left|u_j - u\right| \cdot \sigma_j^2} \tag{4}$$

$$a_j = \frac{\hat{a}_j}{\sum_{j=1}^{2} \hat{a}_j} \tag{5}$$

The confidence $c_t^{i,1}$ and $c_t^{i,2}$ are calculated by propagating particles through two classifiers, and the fusion weight of the each classifier a_j are online obtained according to the Eq. (5), which will be further applied to the online tracking.

C. ONLINE TRACKING

This part mainly completes outdoor vehicle tracking under particle filter framework. The particle-filter-based tracking problem can be regarded as a problem that predicting state at time t when given observation $y_{1:t-1} = \{y_1, y_2, \cdots, y_{t-1}\}$ at time $t-1$.It can be calculated according to Eq. (6).

$$s_t = \arg\max \int p\left(s_t|s_{t-1}\right) p\left(s_{t-1}|y_{1:t-1}\right) ds_{t-1} \tag{6}$$

where, s_t and y_t represent state value and observation value at time t. $p\left(s_t|s_{t-1}\right)$ is state transition probability between sequential frames, and $p\left(s_t|s_{t-1}\right)$ is modeled by a zero-mean Gaussian distribution. The state variable is represented as six affine transformation parameters: translation, scale, aspect, ratio, rotation, and skewness. The particle set $s_t = \left\{s_t^i\right\}_{i=1}^N$ is sampled from importance distribution $\mathrm{q}\left(s_t|s_{1:t-1}, y_{1:t}\right)$.

When a new observation y_t is available, the posterior distribution of state variable is updated by following Eq. (7).

$$p\left(s_t|y_{1:t}\right) = \frac{p\left(y_t|s_t\right) \cdot p\left(s_t|y_{1:t-1}\right)}{p\left(y_t|y_{1:t-1}\right)} \tag{7}$$

The posterior probability distribution $p\left(s_t|y_{1:t}\right)$ can be approximated by the true state of N particles. In particle-filter-based tracking, the predicted value and current state of the system are corrected by observation y_t. In this paper, a new observation probability is computed by the classifiers.

Firstly, particles $s_t = \left\{s_t^i\right\}_{i=1}^N$ are sampled near the object position in the previous frame, then these particles $s_t = \left\{s_t^i\right\}_{i=1}^N$ are propagated in two classification neural networks to calculate the confidence of particles in each network. Next, the output particles confidence of each classification neural networks are adaptively fused according to the Eq. (8).

$$c_t^i = \sum_{j=1}^{2} a_j c_t^j \tag{8}$$

If max $\left(c_t^i\right)$ less than the pre-defined threshold τ, 10 positive samples and 100 negative samples are selected from the previous frames and the two classifiers are fine-tuned using the batch gradient descent method, otherwise the observation probability is calculated by the fusion confidence according to the Eq. (9).

$$\mathrm{p}\left(y_t|s_t^i\right) = e^{\frac{c_t^i - min\left(c_t^i\right)}{\sigma}} \Big/ \sum_{i=1}^{N} e^{\frac{c_t^i - min\left(c_t^i\right)}{\sigma}} \tag{9}$$

where, σ is the standard deviation of observe likelihood function.

The particle weight $W_t = \left\{w_t^i\right\}_{i=1}^N$ is updated by observation probability by Eq. (10).

$$w_t^i = w_{t-1}^i \cdot \frac{\mathrm{p}\left(y_t|s_t^i\right) \mathrm{p}\left(s_t^i|s_{t-1}^i\right)}{\mathrm{q}\left(s_t|s_{1:t-1}, y_{1:t}\right)} \tag{10}$$

where, $\mathrm{q}(s_t|s_{1:t-1}, y_{1:t})$ represents the importance distribution of particles. It is assumed to follow a first-order Markov process.

So the weight in Eq. (10) is updated by following Eq. (11).

$$w_t^i = w_{t-1}^i \cdot \mathrm{p}\left(y_t|s_t^i\right) \tag{11}$$

The proposed algorithm is described as following algorithm 1.

IV. EXPERIMENTAL TESTING

In this section, a comprehensive experimental analysis is stated in detail. We conduct the performance comparison between our tracking algorithm (CAF) and other 9 state-of-the-art tracking algorithms (DLT [4], TLD [30], L1APG [31], IVT [32], MIL [33], OAB [34], Frag [35], MTT [36],

Algorithm 1 The CAF Tracker

```
Input: Training data; Deep auto-encoder structure; Training parameter.
Extract gray-scale images and gradient images from training images.
Offline training two deep auto-encoders by the above two image, respectively.
Construct classifier.
For $t = 1, 2, \ldots,$ frame number
    Generate particles $s_t = \left\{s_t^i\right\}_{i=1}^N$ according to importance sampling.
    Output confidence $c_t^j = \left\{c_t^{i,j}\right\}_{i=1}^N$ by putting forward particles through two classifiers.
    Compute adaptive fusion weight according to Eq. (1-5).
    Fuse confidence of two classifiers according to Eq. (8) to generate fusion confidence.
    Compare fusion confidence with pre-defined threshold $\tau$ to decide whether updating parameter of two classifiers.
    Calculate observation probability $\mathrm{p}\left(y_t|s_t^i\right)$ according to Eq. (9).
    Calculate the weight $w_t^i$ according to Eq. (11).
    Estimate the optimal state $s_t^i$ with maximum weight.
    $t = t + 1$.
End
Output:The predicted tracking position.
```

CSK [37]) on OTB50 data set of the standard object tracking evaluation benchmark VTB [38]. The effectiveness of our tracking algorithm is demonstrated by quantitative evaluation and qualitative evaluation. At the end of this part, the variation curve of each model fusion weight of our tracking algorithm in the tracking process is further given.

A. PARAMETER SETTING

The proposed tracking algorithm was implemented by MATLAB R2012b, and used NVIDIA GeForce, GTX 980Ti for GPU acceleration in the offline training and the online tracking. We used stochastic gradient descent for parameters training. The iteration times was set to 20 in the offline training. The mini-batch size was set to 100. The learning rate was set to 1. The penalty term coefficient was set to 1e-4. The iteration times was set to 5 for the fine-tuning classification neural network in online tracking. The positive and negative samples were set to 10 and 100, respectively. The sampling particle number was set to 1000. The confidence threshold was set to 0.8. The standard deviation of the conservation likelihood σ was set to 0.001.

B. QUANTITATIVE EVALUATION

VTB2013 designed a unified evaluation benchmark for the different tracking algorithms, which contains 50 fully annotated video sequences and 3 different evaluation criteria. The 50 sequences were labeled by 11 attributes, namely illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC), low resolution (LR). 3 evaluation criteria include one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE). In this paper, precision and success rate on OPE were used for quantitative evaluation of tracking algorithms. The precision shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. The success rate counts the number of successful frames whose overlap is larger than the given threshold. The threshold of precision and success rate adopted pre-defined threshold 20 pixel and area under curve (AUC) for ranking tracking algorithms, respectively. The performances on precision and success rate of 10 tracking algorithms are shown in Table 1 and Table 2, respectively, and the best result is shown in bold, and the ranking is shown after '/'. The overall and 11 attribute-based performances on precision and success rate of 10 tracking algorithms are shown in Fig. 4 and Fig. 5, respectively.

In overall performance, the precision and success rate of CAF rank first by 0.637 and 0.536, respectively.

TABLE 1. The performance on precision of 10 tracking algorithms.

	CAF(Ours)	DLT	TLD	MIL	Frag	IVT	L1APG	OAB	MTT	CSK
Overall	**0.637/1**	0.550/3	0.608/2	0.475/8	0.471/10	0.499/6	0.485/7	0.504/5	0.475/8	0.545/4
IV	**0.543/1**	0.514/3	0.537/2	0.349/8	0.326/10	0.418/5	0.341/9	0.388/6	0.351/7	0.481/4
OPR	**0.618/1**	0.527/4	0.596/2	0.466/8	0.444/10	0.464/9	0.478/6	0.503/5	0.473/7	0.540/3
SV	**0.665/1**	0.602/3	0.606/2	0.471/8	0.407/10	0.494/6	0.472/7	0.541/4	0.461/9	0.503/5
OCC	**0.598/1**	0.532/3	0.563/2	0.427/9	0.475/6	0.455/8	0.461/7	0.483/5	0.426/10	0.500/4
DEF	**0.520/1**	0.433/7	0.512/2	0.455/6	0.468/5	0.409/8	0.383/9	0.470/4	0.332/10	0.476/3
MB	0.380/2	0.328/7	**0.518/1**	0.357/5	0.288/9	0.222/10	0.375/3	0.360/4	0.308/8	0.342/6
FM	0.462/2	0.417/3	**0.551/1**	0.396/6	0.364/9	0.220/10	0.365/8	0.416/4	0.401/5	0.381/7
IPR	**0.594/1**	0.502/6	0.584/2	0.453/9	0.401/10	0.457/8	0.518/5	0.471/7	0.522/4	0.547/3
OV	0.518/3	0.536/2	**0.576/1**	0.393/5	0.355/8	0.307/10	0.329/9	0.454/4	0.374/7	0.379/6
BC	**0.605/1**	0.455/4	0.428/6	0.456/3	0.421/10	0.421/9	0.425/7	0.446/5	0.424/8	0.585/2
LR	0.449/3	0.309/7	0.349/6	0.171/9	0.163/10	0.278/8	0.460/2	0.376/5	**0.510/1**	0.411/4

TABLE 2. The performance on success rate of 10 tracking algorithms.

	CAF(Ours)	DLT	TLD	MIL	Frag	IVT	L1APG	OAB	MTT	CSK
Overall	**0.536/1**	0.499/3	0.521/2	0.373/10	0.399/9	0.424/8	0.440/6	0.427/7	0.445/4	0.443/5
IV	0.469/2	**0.472/1**	0.460/3	0.292/10	0.298/9	0.351/5	0.300/8	0.326/7	0.337/6	0.388/4
OPR	**0.509/1**	0.464/3	0.497/2	0.357/10	0.376/9	0.381/8	0.416/6	0.406/7	0.423/5	0.439/4
SV	**0.577/1**	0.547/2	0.494/3	0.335/9	0.313/10	0.388/7	0.407/5	0.412/4	0.398/6	0.352/8
OCC	**0.510/1**	0.502/2	0.468/3	0.361/10	0.423/5	0.391/9	0.425/4	0.414/7	0.422/6	0.404/8
DEF	0.370/7	0.389/5	**0.456/1**	0.407/3	0.413/2	0.314/9	0.384/6	0.401/4	0.334/10	0.370/8
MB	0.363/2	0.321/6	**0.482/1**	0.247/9	0.283/8	0.213/10	0.362/4	0.363/2	0.288/7	0.336/5
FM	0.455/2	0.418/4	**0.473/1**	0.338/8	0.319/9	0.225/10	0.359/7	0.420/3	0.385/5	0.380/6
IPR	**0.510/1**	0.439/6	0.476/2	0.331/9	0.330/10	0.389/8	0.442/5	0.391/7	0.463/3	0.457/4
OV	0.523/2	**0.552/1**	0.516/3	0.416/5	0.373/8	0.319/10	0.341/9	0.492/4	0.392/7	0.410/6
BC	0.444/2	0.398/7	0.388/8	0.414/3	0.370/9	0.344/10	0.404/6	0.410/5	0.411/4	**0.491/1**
LR	0.430/3	0.297/7	0.327/6	0.157/10	0.170/9	0.287/8	0.458/2	0.366/5	**0.506/1**	0.397/4

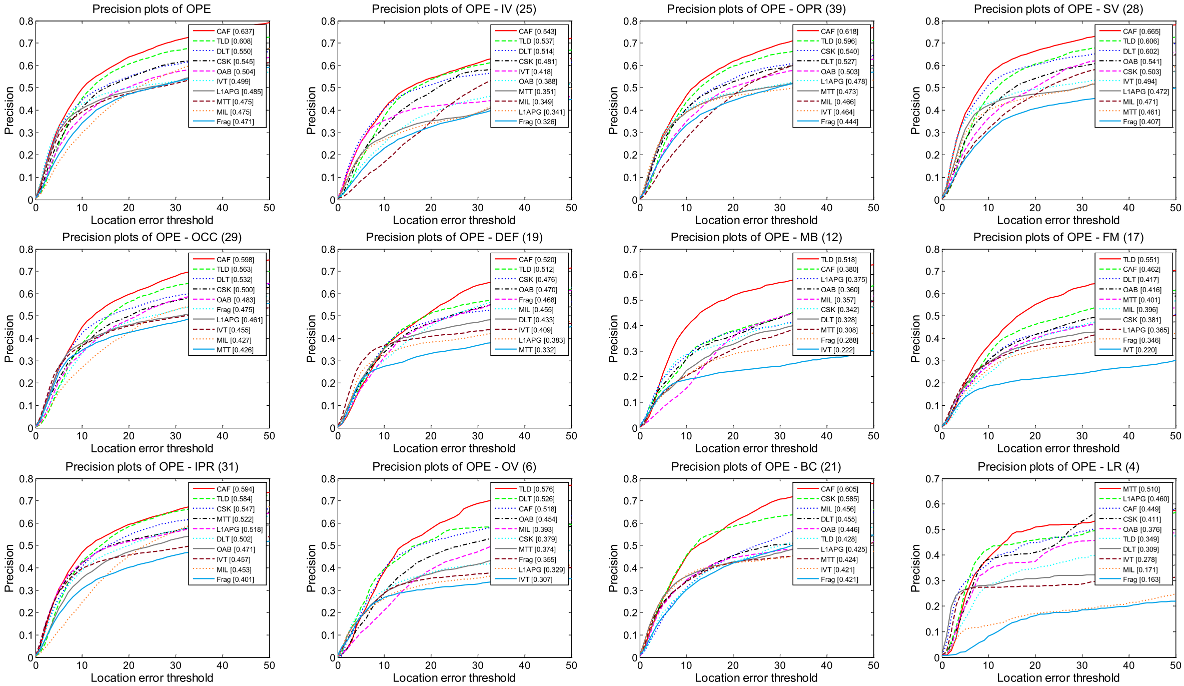


FIGURE 4. The overall and 11 attribute-based performances on precision of 10 tracking algorithms.

In attribute-based performance, CAF also achieves superior tracking results. In precision plots, the IV, OPR, SV, OCC, DEF, IPR and BC of CAF rank first; MB and FM rank second, just below the first TLD 0.089 and 0.138, respectively; OV and LR rank third, but CAF is better than the first MTT when location error threshold of OV and LR are lower than 10 and 7, respectively. In the success plots, OPR, SV, OCC and IPR of CAF rank first; IV, MB, FM, OV and BC of CAF rank second, only less than the first DLT 0.003, TLD 0.119, TLD 0.018, DLT 0.029, and CSK 0.047 respectively; LR of CAF ranks third, but when the overlap threshold is less than 0.1, CAF is superior to MTT; DEF of CAF rank seventh, but when the overlap threshold is less than 0.25, CAF is better than that of the first TLD.

Our algorithm uses gray-scale image and gradient image for training deep learning model to implement multiple model information complementary. The features extracted by gray-scale image are robust to rotation, deformation and occlusion, but they are sensitive to illumination variation. The features extracted by gradient image can capture the shape information of object and sensitive to motion. In addition, the proposed classifier in this paper is a discriminative model, which can distinguish object from background effectively. Therefore, the algorithm is superior to existing algorithms in-plane rotation, out-of-plane rotation, deformation, occlusion, illumination variation, and background clutters. Taking training time of deep learning models into account, our algorithm uses low resolution training images, so it is biased under low resolution.

So, it can be seen that our CAF tracker is comparable to the 9 state-of-art tracking algorithm in both overall and attribute-based performance under most challenging factors.

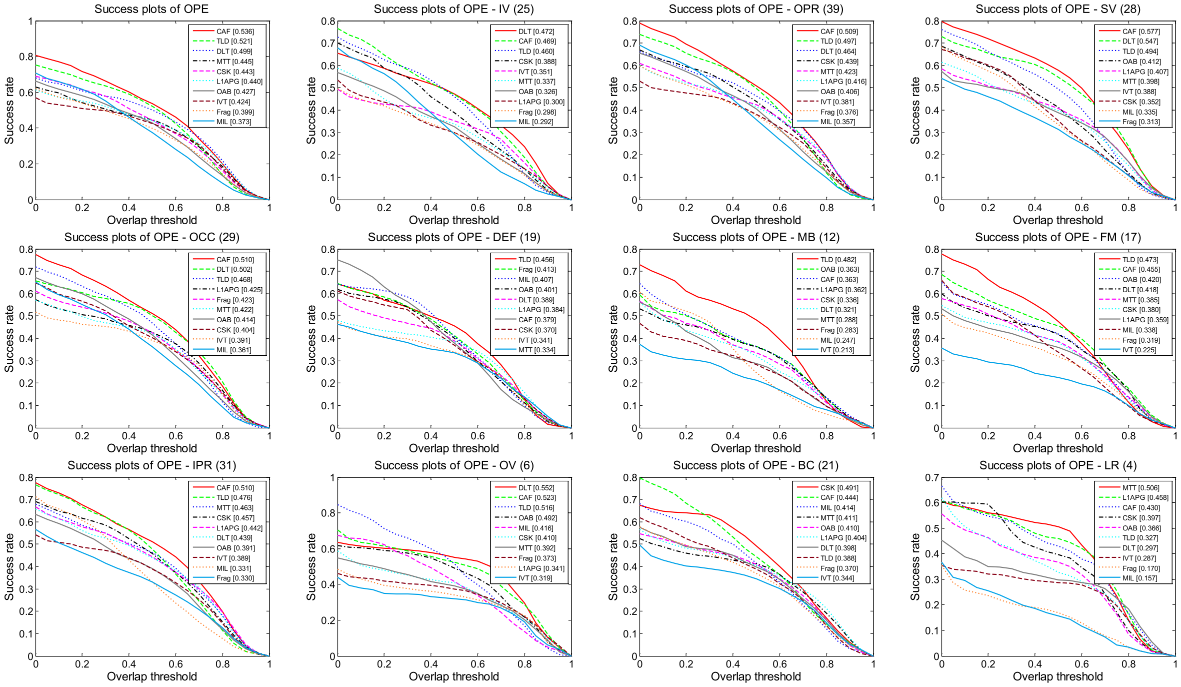


FIGURE 5. The overall and 11 attribute-based performances on success rate of 10 tracking algorithms.

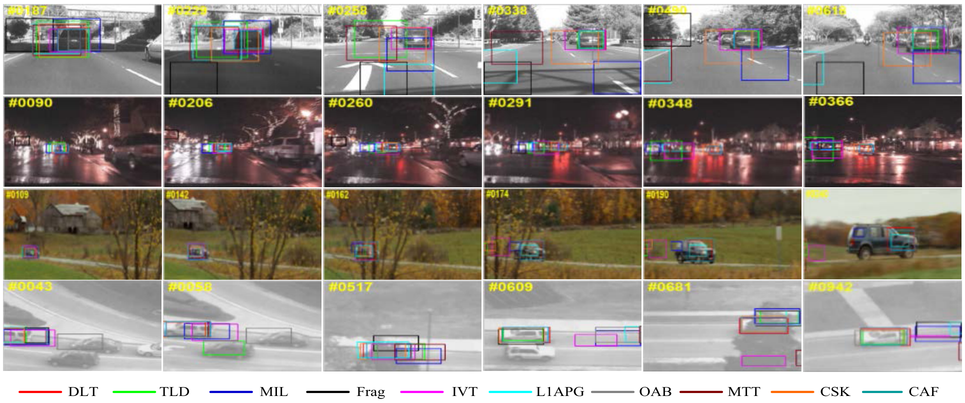


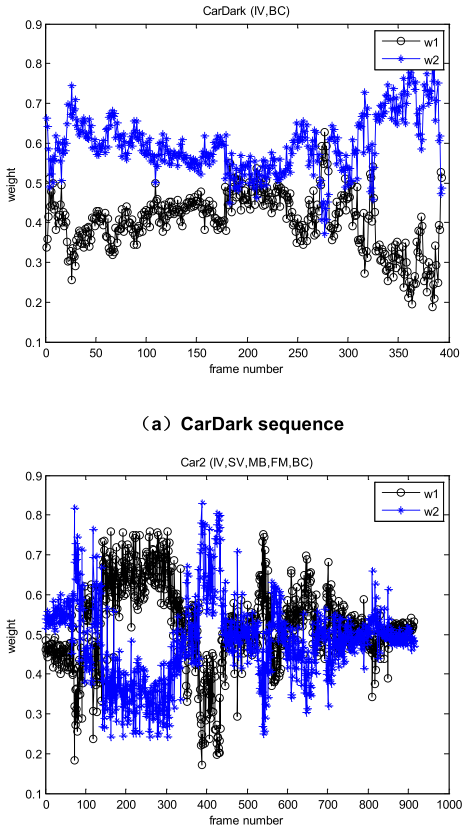
FIGURE 6. Sampled tracking results of 10 tracking algorithms on 4 sequences.

C. QUALITATIVE EVALUATION

We employ the benchmark tracking data sets with 50 fully annotated sequences for quantitative evaluation, and we use 4 vehicle sequences in it for qualitative evaluation, including Car4, CarDark, CarScale, Suv. The four representative sequences include outdoor vehicle tracking multiple challenge scenarios, covering most of the challenging attributes of outdoor vehicle tracking such as illumination various (IV), occlusion (OCC), background clutter (BC), scale various (SV) and so on. The partial tracking results of 10 tracking algorithms on 4 sequences are sampled as shown in Fig. 6.

In #187, #229 of Car4 and #90, #206, #260, #291, #348, #368 of CarDark, the illumination changes obviously, only DLT and CAF can track object accurately. In #258, #338, #490, #618 of Car4 and #190, #240 of CarScale, scale varies significantly, comparing with other tracking algorithm, CAF still has a superior tracking performance though all tracking algorithms can not track the object accurately. In #90, #206, #260, #291, #348, #368 of CarDark and #609 of Suv, background clutter, only DLT, OAB, CSK, and CAF are able to track the object. In #142, #162, #174 of CarScale, occlusion is not serious, only DLT, OAB and CAF can track object accurately. In #517, #681 of Suv, occlusion serious, only DLT, CSK and CAF can accurately track the object. In #109, #142, #162, #174, #190, #240 of CarScale and #942 of Suv, object moved fastly, DLT, OAB and CAF can still track object steady. In #43, #58 of Suv, object out of view, only DLT, L1APG, MTT, CSK, and CAF can keep track object.

Therefore, the proposed tracking algorithm can realize robust tracking of outdoor vehicle in complex environment.



（a）CarDark sequence

（b）Car2 sequence

FIGURE 7. Changing weight of each deep learning model for CarDark and Car2 sequences.

D. FUSION WEIGHT VARIATION ANALYSIS OF THE CAF TRACKER

In order to further verify the validity of the CAF tracker proposed in this paper. Taking the CarDark and Car2 sequences as an example, the variation curve of the two models fusion weights in the tracking as shown in Fig. 7. The black line represents the weight change of the model trained by the grayscale image, and the blue line represents the weight change of the model trained by the gradient image. Fig. 7 (a) represents the variation curve of two models fusion weights in the CarDark sequence. Fig. 7 (b) shows the variation curve of the two models fusion weights in the Car2 sequence.

In the CarDark sequence, the illumination changes are obvious and persist throughout the sequence, while the features extracted by grayscale image are sensitive to illumination changes. As can be seen from Fig. 7 (a), compared with grayscale image, the weight of deep learning model trained by gradient image has been large in the CarDark sequence, which can make up for the shortcomings that features of grayscale images are sensitive to illumination changes. In the Car2 sequence, there is a significant illumination change during the movement between #380 and #442. As can be seen from Fig. 7 (b), the fusion weight of the deep learning model by the gradient image training becomes larger; Near the #500, the object suddenly appears fast motion, while the features of grayscale images is robust to non-rigid changes. It can be seen from Fig. 7 (b) that the fusion weight of the deep learning model by the grayscale images training becomes larger at this time, which can make up for the lack that features of gradient images are sensitive to the motion.

In summary, it can be seen from Fig. 7 that the proposed algorithm CAF can adaptively adjust the fusion weight of two deep learning models in the presence of outdoor challenging factors, so that it can be used in a complex and varying outdoor environment to achieve more robust tracking.

V. CONCLUSION

In this paper, a novel object tracking algorithm based on multi-deep learning model adaptive fusion under the particle filter framework for outdoor vehicle tracking is proposed. Among them, the fusion weight of each deep learning model can be automatically calculated according to the distribution of the particles represented by themselves. Several comparative tracking experiments are conducted on the VTB platform to evaluate quantitatively and qualitatively the tracking performance. The experimental results show that the proposed tracking algorithm can achieve superior tracking results in the most challenging factors of outdoor compared with 9 state-of-the-art tracking algorithms. At the same time, the analysis of the fusion weight curve shows that the proposed algorithm can adjust the fusion weight of each deep learning model in time according to the change of the object appearance model to achieve more robustness tracking.

In the future, we will further improve the tracking performance of our tracking algorithm by improving the feature representation capability. Some possible directions include: employing the advanced deep learning structure, using the large scale tracking data set for model training and utilizing the other powerful feature representation method.

REFERENCES

[1] J. L. Chang, L. F. Wang, G. F. Meng, S. Xiang, and C. Pan, "Vision-based occlusion handling and vehicle classification for traffic surveillance systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 80–92, Feb. 2018.

[2] Z. Li, Y. Chen, Z. Yin, "Vehicle tracking fusing the prior information of Kalman filter under occlusion conditions," *SN Appl. Sci.*, vol. 1, no. 8, p. 822, Aug. 2019.

[3] H. Guan, X.-Y. Xue, and Z.-Y. An, "Advances on application of deep learning for video object tracking," *Acta Automatica Sinica*, vol. 42, no. 6, pp. 834–847, Jun. 2016.

[4] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. IEEE NIPS*, Lake Tahoe, NV, USA, Dec. 2013, pp. 809–817.

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[6] Z. Wang, G. Liu, and G. Tian, "A parameter efficient human pose estimation method based on densely connected convolutional module," *IEEE Access*, vol. 6, pp. 58056–58063, 2018.

[7] Z. Li and F. Zhou, "FSSD: Feature fusion single shot Multibox detector," Dec. 2017, *arXiv:1712.00960*. [Online]. Available: https://arxiv.org/abs/1712.00960

[8] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, no. 1, pp. 333–338, Aug. 2012.

[9] M. N. Stolar, M. Lech, and I. S. Burnett, "Optimized multi-channel deep neural network with 2D graphical representation of acoustic speech features for emotion recognition," in *Proc. IEEE ICSPCS*, Gold Coast, QLD, Australia, Dec. 2014, pp. 1–6.

[10] S. Zhang, H. Yang, and Z. Yin, "Multiple deep convolutional neural networks averaging for face alignment," *Proc. SPIE*, vol. 24, no. 3, May 2015, Art. no. 033013.

[11] Y. Yu, H. Lin, Q. Yu, J. Meng, Z. Zhao, Y. Li, and L. Zuo, "Modality classification for medical images using multiple deep convolutional neural networks," *J. Comput. Inf. Syst.*, vol. 11, no. 15, pp. 5403–5413, Aug. 2015.

[12] Y. Peng, Y. Zhao, and J. Zhang, "Two-stream collaborative learning with spatial-temporal attention for video classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 773–786, Mar. 2018.

[13] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Proc. Adv. Neural Inf. Process. Syst.*, Feb. 2013, pp. 1493–1501.

[14] J. Kuen, K. M. Lim, and C. P. Lee, "Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle," *Pattern Recognit.*, vol. 48, no. 10, pp. 2964–2982, Oct. 2015.

[15] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 4293–4302.

[16] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 1420–1429.

[17] J. Xin, X. Du, and J. Zhang, "Deep learning for robust outdoor vehicle visual tracking," in *Proc. IEEE ICME*, Hong Kong, China, Jul. 2017, pp. 613–618.

[18] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, "An ensemble of deep neural networks for object tracking," in *Proc. IEEE ICIP*, Paris, France, Oct. 2015, pp. 843–847.

[19] L. Dai, Y. Zhu, G. Luo, and C. He, "A low-complexity visual tracking approach with single hidden layer neural networks," in *Proc. IEEE ICARCV*, Singapore, Dec. 2015, pp. 810–814.

[20] N.Y.Wang, S.Y.Li, Gupta, "Transferring rich feature hierarchies for robust visual tracking," Jan. 2015, *arXiv:1501.04587*. [Online]. Available: https://arxiv.org/abs/1501.04587

[21] W. Hua, D. Mu, D. Guo, and H. Liu, "Visual tracking based on stacked Denoising Autoencoder network with genetic algorithm optimization," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4253–4257, Feb. 2018. doi: 10.1007/s11042-017-4702-1.

[22] H. Li, Y. Li, and F. Porikli, "Deep track: Learning discriminative feature representations by convolutional neural networks for visual tracking," in *Proc. BMVC*, Nottingham, U.K., Sep. 2014, pp. 1–12.

[23] H. Y. Li Li and F. Porikli, "Robust online visual tracking with a single convolutional neural network," in *Proc. ACCV*, Singapore, Apr. 2015, pp. 194–209.

[24] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations Online for robust visual tracking," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1834–1848, Apr. 2016.

[25] X. Zhou, L. Xie, P. Zhang, and Y. Zhang, "Online object tracking based on CNN with metropolis-hasting re-sampling," in *Proc. ACM Multimedia*, Brisbane, QLD, Australia, Oct. 2015, pp. 1163–1166.

[26] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.

[27] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1424–1435, Apr. 2015.

[28] J. Hu, J. Lu, and Y.-P. Tan, "Deep metric learning for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2056–2068, Nov. 2016.

[29] H. Li, Y. Li, and F. Porikli, "Convolutional neural net bagging for Online visual tracking," *Comput. Vis. Image Understand.*, vol. 153, pp. 120–129, Dec. 2016.

[30] X. Kalal, J. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[31] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," in *Proc. IEEE CVPR*, Providence, RI, USA, Jun. 2012, pp. 1830–1837.

[32] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, May 2008.

[33] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with Online multiple instance learning," in *Proc. IEEE CVPR*, Miami, FL, USA, Jun. 2009, pp. 983–990.

[34] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. IEEE BMVC*, Edinburgh, U.K., Sep. 2006, pp. 47–56.

[35] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE CVPR*, New York, NY, USA, Jun. 2006, pp. 798–805.

[36] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE CVPR*, Providence, RI, USA, Jun. 2012, pp. 2042–2049.

[37] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*, Florence, Italy, 2012, pp. 702–715.

[38] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE CVPR*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.

JING XIN received the B.S., M.S., and Ph.D. degrees from the Xi'an University of Technology, Xi'an, China, in 1997, 2003, and 2007, respectively, where she is currently a Professor with the Key Laboratory of Shaanxi Province for Complex System Control and Intelligent Information Processing.

From 2010 to 2011, she was a Visiting Scholar with the Australian Centre for Field Robotics, University of Sydney, for one year. In 2012 and 2016, she was a Visiting Professor with the Advanced Analysis Institute and Global Big Data Technologies Centre (GBDTC), University of Technology, Sydney, for three months. Her current research interests include manipulator robot visual servoing, mobile robot visual navigation, and robust object tracking.

Dr. Xin is currently a Senior Member of the Chinese Association of Automation (CAA). She is a member of the Construction Robot Professional Committee, CAA. She is also a member of the Navigation Guidance and Control Professional Committee, CAA. She is an Associate Editor of the *International Journal of Advanced Robotic Systems* (IJARS).

XING DU is currently pursuing the M.S. degree in control theory and control engineering with the Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an, China. Her research interests include deep learning and its applications in object tracking.

YAQIAN SHI is currently pursuing the M.S. degree in control theory and control engineering with the Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an University of Technology, Xi'an, China. Her research interests include deep reinforcement learning and its applications in object tracking.