



End-to-end deep metric network for visual tracking

Shengjing Tian¹ · Shuwei Shen¹ · Guoqiang Tian¹ · Xiuping Liu¹ · Baocai Yin²

Published online: 24 July 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

In this paper, we propose an end-to-end deep metric network (DMN) for visual tracking, where any target can be accurately tracked given only a bounding box of the first frame. Our main motivation is to make the network learn to learn a deep distance metric by following the philosophy of one-shot learning. Instead of utilizing a hand-crafted distance metric like Euclidean distance, our DMN focuses on providing a learnable metric, which is more robust to appearance variations. Furthermore, we are the first to properly combine mean square errors and contrastive loss into a joint loss function for back-propagation. During online tracking, DMN firstly applies our instance initialization for obtaining sequence-specific information and then straightforwardly tracks the target without the help of box refinement, occlusion detection and online updating. The final tracking score considers both our DMN scalar output and the constrain of motion smoothness. Ablation analyses are carried out to validate the effectiveness of our proposed method. And experiments on the prevalent benchmarks show that our method can achieve a competitive performance when compared with some representative trackers, especially those existing metric learning-based algorithms.

Keywords Metric learning · Visual tracking · Deep neural networks · One-shot learning

1 Introduction

Visual tracking has been extensively studied on account of its wide applications, such as human–computer interaction [25,28], augmented reality [18] and video surveillance [37]. Even though a large fraction of tracking methods have been developed, there are still some challenging factors like scale variation, occlusions, deformation, cluttered backgrounds and illuminations to be overcome [45,46].

The core of many existing trackers is to measure similarities between the template patch and candidate patches. For the moment current visual tracking methods that employ similarity measurement are mainly based on matching function or metric learning. The former one aims to construct an appearance model with more powerful feature represen-

tations and utilize the pre-defined distance metrics, such as normalized cross-correlation [2], Bhattacharyya coefficient [4], cosine distance [1], Euclidean distance [26,34] and Kullback–Leibler [7] to track the target. However, due to large appearance and scale variations, pre-defined metrics are not precise enough to estimate the target distribution in the feature space. Different from aforementioned methods, the latter one puts a lot more attention on finding a discriminative linear or nonlinear metric, which makes positive pairs close and negative pairs remote. As we all know, metric learning has achieved some satisfactory results in visual tracking [6,16,22,35,43,47]. For instance, Mahalanobis distance-based algorithms [22,47] were proposed to learn a linear metric for visual tracking. Subsequently, Hu et al. [15] and Lu et al. [23] introduced a nonlinear metric learning methods under the particle filter framework. Nevertheless, although a linear or nonlinear transformation f has been learnt, previous metric learning-based methods eventually have to recognize the tracked object by the isotropic Euclidean distance $d_f(x_i, x_j) = \|f(x_i) - f(x_j)\|_2$.

Sung et al. [33] initially proposed to learn a deep distance metric in a meta-learning way for few-shot learning. And they clearly illustrated the feasibility of the learnt deep metric via a 2D example. Inspired by this [33], we aim to enable the

✉ Shengjing Tian
tye@mail.dlut.edu.cn

Baocai Yin
ybc@dlut.edu.cn

¹ School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, People's Republic of China

² Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China

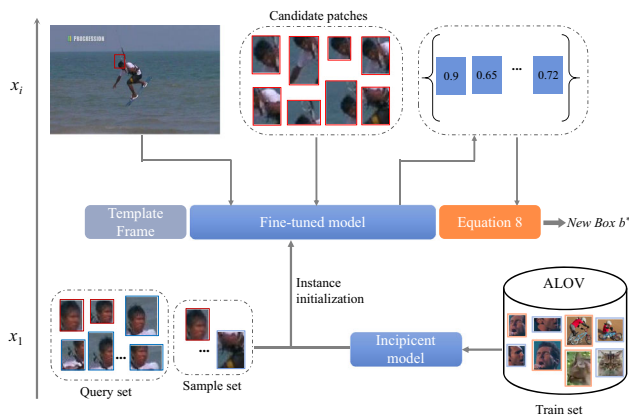


Fig. 1 Illustration of tracking procedure. DMN firstly learns transferrable knowledge from training set. Then, the incipient model is fine-tuned with sample and query sets in the first frame. Based on this model, we locate the target in remaining unseen clips. The network output is a scalar value that evaluates whether the sampled patch is similar to the template patch or not. According to Eq. (8), we determine the best box that will guide sampling in the next frame x_{i+1} . The top row represents the coming frame, candidate patches and similar scores from left to right

network to learn what distance metric a specific sequence should apply during online tracking phase, which can be seen as providing a learnable rather than hand-crafted metric.

Therefore, we propose a novel end-to-end deep metric network for visual tracking. A radically new intention is that online tracking can be seen as a one-shot learning problem, and deep feature embedding and coupled nonlinear metric can be trained in a joint manner to replace conventional hand-crafted metrics. Concretely, a joint loss function, which favorably combines mean square errors and contrastive loss, is put forward purposively to reduce regression errors of metric module and enhance discernment of feature module. By jointly learning feature embedding and deep nonlinear metric, our network can directly output a scalar value instead of manually choosing a hand-crafted metric for target tracking. We hold that the learnt metric can better distinguish between matching and mismatching pairs suffering from various challenges. During training phase, the deep metric network trained with external video datasets can capture common properties, such as robustness to illumination changes and motion blur. For online tracking phase, we use sample set and query set to fine-tune the incipient model in a one-shot learning manner so as to learn sequence-specific information. The whole tracking procedure, as shown in Fig. 1, is straightforwardly conducted without online updating, occlusion detection, data augmentation and box refinement. Experiments on the prevalent benchmarks demonstrate that our method can obtain competitive results compared with some state-of-the-art methods and its speed is over three times than existing metric learning-based trackers.

Our main contributions are three-fold:

- An end-to-end deep metric network framework without additional tedious step, such as box refinement, is proposed, which directly outputs scalar similarity and can reach 16 FPS. And the instance initialization using one-shot learning manner is adopted for a specific sequence tracking.
- A radically new idea of jointly learning deep feature embedding module and nonlinear metric module is presented in contrast to using pre-defined metric like Euclidean distance to recognize the tracked object. And ablation experiments indicate that two modules can well couple with each other.
- An efficient joint loss function that favorably blends discriminative feature learning and powerful metric learning is purposively put forward.

2 Related work

In this section, we briefly review some related works from the following three aspects: matching function-based methods, deep learning-based methods, as well as metric learning-based methods.

2.1 Matching function-based trackers

Numerous matching function-based trackers have been proposed in the past few years. By means of similarity score with respect to the template, these trackers can find the best candidate from interest regions in the coming frame. For instance, starting from the point of image patch retrieval, [2] tracks the target using normalized cross-correlation. Based on mean shift iterations, MST [4] finds the most probable target position by Bhattacharyya coefficient. IVT [29] utilizes incremental principal component analysis to obtain the object appearance feature. SPT [42] builds an appearance model based on superpixels and obtain the most likely target location with maximum a posteriori estimates. Many sparse representation methods [26,38,49,50] have achieved memorable performance with particle filters. Specially, RSST [50] obtains appreciable improvement by capturing the underlying relationships among all local patches and outliers. SINT [34] and SiamFC [1] learn a generic matching function by Siamese network from video dataset. Although these methods obtained a powerful matching function and satisfied results, they put more attention on appearance modeling approach rather than suitable and effective metric.

2.2 Deep learning-based trackers

With the remarkable success of deep learning, many methods using convolution neural network (CNN) have achieved state-of-the-art performance. DLT [40] initially brings stacked denoising autoencoder network into visual tracking. SO-DLT [39] presents a structured output CNN instead of treating tracking as classification task. SINT [34] trains a Siamese network to learn the matching mechanism. SINT++ [44], the improved version of SINT, concentrates on data augmentation using variational autoencoder and deep reinforcement learning. Based on the rich hierarchical features of CNN, HCFT* [24] utilizes correlation filter framework to locate the target in a coarse-to-fine manner. MDNet [27] firstly learns general feature representation from multiple annotated video sequences and then captures domain-specific information through online learning. GOTURN [12] learns a generic relationship between appearance and motion, and directly predicts the target location. SiamFC [1] addresses a more general similarity learning problem by fully convolutional Siamese network. Subsequently, in order to obtain a suitable template, CFNet [36] integrates correlation filter into SiamFC, and RASNet [41] proposes three attention mechanisms. SA-Saim [11] introduces semantic branch, which is trained in the classification task, to enhance the robustness of SiamFC. SiamRPN [21] combines Siamese network and region proposal network [8] to predict the label and location of the anchor. Recently, by incorporating discriminative feature fine-tuning, adaDDCF [9] introduces an adaptive deep correlation filter. Its main differences with ours lie in that 1) deep correlation filter aims to represent the target appearance using convolution kernel and generate the correlation response map of each frame, whereas DMN focuses on learning a reasonable metric which can precisely estimate the target distribution in the feature space and measure similarity between the template patch and candidate patch; 2) adaDDCF has no offline training and directly uses fisher discriminative analysis layer to fine-tune the pre-trained VGGNet model [31] online, while DMN elaborately trains discriminative features and nonlinear metric from scratch.

2.3 Metric learning-based trackers

To address various challenging factors, based on advanced metric learning approaches, many algorithms have been proposed to learn a powerful similarity measure. For example, ITML [6] uses an information-theoretic approach to learn a Mahalanobis distance function. Jiang et al. [16] proposed a sparsity-regularized metric learning method. Li et al. [22] introduced an online reservoir metric learning method for appearance-based visual tracking. Wu et al. [47] presented a metric learning-based structural appearance model (MLSAM) for structure object representation and matching.

With the particle filter framework, Hu et al. [15] introduced a deep metric learning (DML) approach, which learns metric by multiple fully connected layers. Meanwhile, NML [23] trains a set of hierarchical nonlinear transformations to match the most similar candidate box. Nevertheless, during online tracking phase, all of above methods still resort to pre-defined metrics to calculate the similarity. Different from those metric learning-based methods, our deep metric network, which jointly learns deep feature embedding and nonlinear metric, directly outputs a scalar value during both offline and online phase.

3 Proposed method

Conventional metric learning-based methods for visual tracking learn either a Mahalanobis distance metric [22,47] or a nonlinear transformation [15,23], which still possesses the potentiality of performance improvement. As shown in Fig. 2, our DMN tracker consists of learning a function map φ from image space Λ to feature space Ω and seeking an appropriate metric ϕ corresponding to the above feature space. These two modules are modeled with deep convolution network in an end-to-end manner and can be well coupled with each other. We firstly train the deep metric network offline with annotated video dataset. Then, given one target shot in the first frame, sample and query sets can be generated for the instance initialization. Finally the fine-tuned model can be utilized to track new object. The network outputs a scalar value for each candidate patch, which mainly determines the optimal location selection.

3.1 Problem formulation

Let x_i be a single frame and $b_s = [x_s^{lt}, y_s^{lt}, x_s^{rb}, y_s^{rb}]$ be a rectangular bounding box in the frame x_i , where (x_s^{lt}, y_s^{lt}) is the left-top position and (x_s^{rb}, y_s^{rb}) is the right-bottom position. The image patch p_s cropped within the bounding box b_s is called region of interest (ROI). The next section will detailedly describe sample strategy of box pairs for network input. Given a pair of frames (x_i, x_j) and their corresponding box pair (b_s, b_t) , DMN jointly learns a function map $\varphi : \Lambda \rightarrow \Omega$ and a tightly coupled nonlinear metric $\phi : \Omega \times \Omega \rightarrow \mathcal{R}$ to measure similarity between cropped patches p_s and p_t , as illustrated in Fig. 2.

To begin with, two frames x_i and x_j are fed into feature embedding module φ , which produces feature maps $\varphi(x_i)$ and $\varphi(x_j)$. Here, we consider different feature blocks for more precise location. Secondly, their ROI feature maps can be obtained by ROI pooling operation f_P [8], which generates the same size feature map for any different size ROI. Because outputs of conv4 and conv5 (Fig. 4) are utilized as intermediate feature, we design an operation f_C to fuse dif-

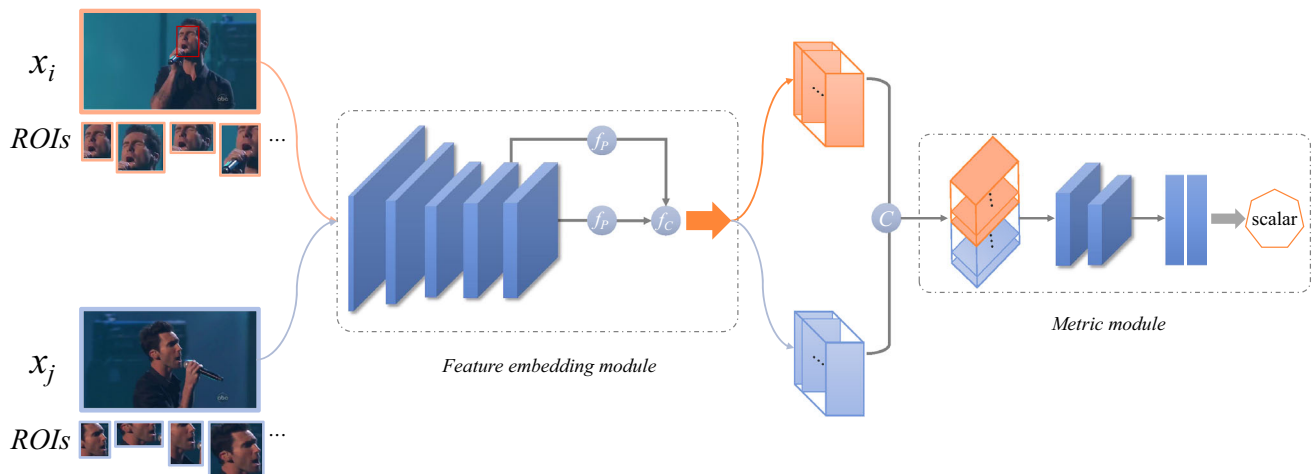


Fig. 2 Framework of our deep metric network. The network is composed of feature embedding module and metric module. Firstly, frames and ROIs are fed into feature embedding module for feature extraction, where f_P is ROI pooling operation and f_C is fusion operation. Then,

the paired feature maps are concatenated by operation C . Taking this concatenated feature as input, metric module outputs a scalar value. Note that C is only channel concatenation when f_C consists of channel concatenation and 1×1 convolution transformation

ferent layer features. Specifically, the fusion operation f_C consists of channel concatenation and 1×1 convolution, and the fused features can be formulated as

$$\begin{aligned} F[b_s] &= f_C(f_P(\varphi(x_i), b_s)), \\ F[b_t] &= f_C(f_P(\varphi(x_j), b_t)). \end{aligned} \quad (1)$$

Subsequently, the feature maps $F[b_s]$ and $F[b_t]$ are further combined with operator C , and subnetwork g_ϕ is responsible to predict their similarity. In this work, we define operator C as concatenation of feature maps in channel. Finally, the learnable metric ϕ can be represented as

$$\phi(F[b_s], F[b_t]) = g_\phi(C(F[b_s], F[b_t])), \quad (2)$$

which is a scalar value ranged from 0 to 1.

In Fig. 3, three confidence maps of the sequence *KieSurf* are shown to validate how the proposed deep metric network works. We scanned the whole frame using a bounding box with the same size as regions of interest and passed them through DMN. The resulting scalar value of each corresponding ROI can be aggregated to form confidence map. Figure 3 clearly demonstrates that the learnt metric network outputs a high similarity score for target and a low one for background. Furthermore, the second column and third column are results without and with the instance initialization, respectively. Because the network obtains generic feature representation by offline training from coverall dataset, pixels of the person have a more higher output score in Fig. 3b. After the instance initialization, DMN already knows that the head of this person is what it is supposed to track. Therefore, in Fig. 3c, the head has a high output score when others reduce to a small value.

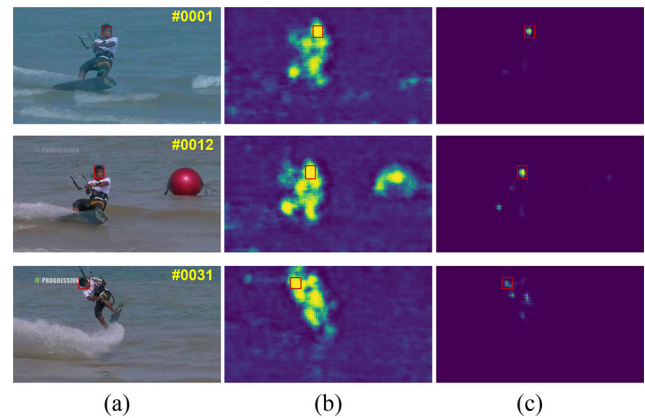


Fig. 3 Visualization results of similarity score. The heat map is generated based on the network outputs. **a** Is frame to be tracked. **b, c** Are results without and with the instance initialization, respectively

3.2 Network input

Network inputs include frame pairs and corresponding box pairs. Since expecting the network to be robust to many types of object appearance variations, we randomly choose some frame pairs (x_i, x_j) during training phase. And these frame pairs from the same video sequence do not need to be adjacent with each other. Then, a set of box pairs (b_s, b_t) can be generated in the following way. One element b_s in such box pair is ground-truth bounding box in frame x_i . The other element b_t is a box sampled around the corresponding ground truth in frame x_j . Thus, we can obtain a large quantity of triples $(b_s, b_t, o_{s,t})$, where $o_{s,t}$ is a label value determined by intersection-over-union (IoU) overlap between sampled box b_t and its corre-

sponding ground truth G . The formula is defined as the following

$$o_{s,t} = \begin{cases} 1 & \text{if } IoU(b_t, G) > \rho^+, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where the threshold ρ^+ is used to decide whether a sample is positive or not.

3.3 Loss function

The top of the network outputs a scalar value which represents similarity of pairs. We simply regress this predicted value to ground truth using mean square errors (MSE) for making positive pairs close and negative pairs remote. This loss can be written as

$$L_S(b_s, b_t, o_{s,t}) = \frac{1}{m} \sum \|\phi(F[b_s], F[b_t]) - o_{s,t}\|_2^2, \quad (4)$$

where m is batch size and $o_{s,t} \in \{0, 1\}$. The binary label $o_{s,t}$ is determined by Eq. (3).

In addition, considering that more discriminative feature embeddings could facilitate the learning of a proper metric, we also adopt contrastive loss for feature embedding module. This loss aims to minimize the distance of positive pairs and enable the distance of negative pairs to be large than a margin α . Following [3], the large margin contrastive loss function can be written as

$$L_F(b_s, b_t, o_{s,t}) = \frac{1}{m} \sum o_{s,t} D_{s,t}^2 + (1 - o_{s,t})[\alpha - D_{s,t}]_+^2, \quad (5)$$

where $D_{s,t} = \|F[b_s] - F[b_t]\|_2$, operation $[\cdot]_+$ indicates the hinge function $\max(0, \cdot)$. Bearing these two aspects in mind, we train the network with the joint loss function

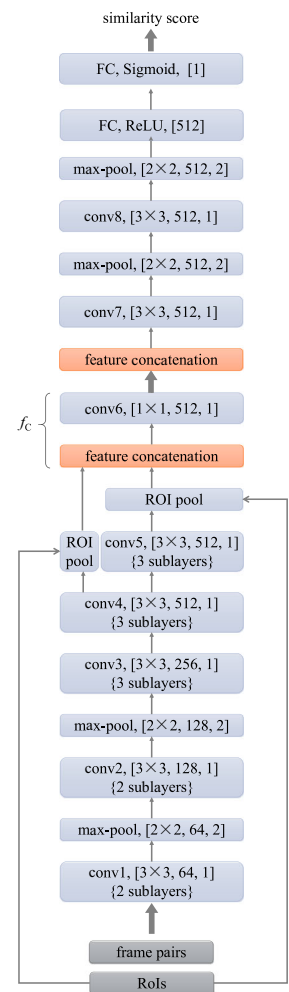
$$L(b_s, b_t, o_{s,t}) = L_S(b_s, b_t, o_{s,t}) + \lambda L_F(b_s, b_t, o_{s,t}), \quad (6)$$

where λ is a balance factor. To illustrate the necessity of the joint loss function, we separately use Eq. (4) and Eq. (6) to train the network. As can be seen from Fig. 6, the joint loss function has an obvious improvement when compared with mean square errors.

3.4 Network architecture

Many visual tracking network models utilize popular classification network like VGGNet [31] and AlexNet [20] as backbone with subtle modification, most of which present a standout performance. In fact, compared to AlexNet, VGGNet has a more stronger feature representation and shows the superiority of applying to visual tracking [34]. In

Fig. 4 The architecture detail of our proposed network. The feature concatenation is the operator C . The ‘conv,’ ‘ROI pool’ and ‘max-pool’ are convolution, ROI pooling operation and max pooling operation, respectively. The ‘FC,’ ‘ReLU’ and ‘Sigmoid’ are fully connected layer, ReLU activation function and Sigmoid function, respectively. Contents in braces are how many sublayers are included in current convolution block. Contents in square brackets are kernel size, kernel number and stride, aside from the last two FCs where content in square brackets is neuron’s number



this work, we use VGGNet as our backbone to learn jointly feature embedding and nonlinear metric. The visual architecture is shown in Fig. 4.

As for feature embedding module, its architecture is composed of five convolution blocks. The kernel size and activation function of each convolution layer are 3×3 and ReLU, respectively. As ablation studies suggest in SINT [34], max pooling layer deteriorates tracking accuracy and causes poor localization, due to the reduction in the feature maps resolution. So, considering the susceptiveness of visual tracking to rough discretizations, we adopt the SINT’s strategy that the first two blocks contain a 2×2 max pooling, while the rest do not. In addition, we use outputs from the last two blocks as the intermediate feature embedding.

Regarding metric learning module, the fused feature maps are fed into the subnetwork which is composed of two convolution blocks and two fully connection layers. Each of convolution block is a 3×3 convolution with 512 filters followed by ReLU activation function and 2×2 max pooling. The two fully connection layers have 512 and 1 units. The first fully connection layer which has 512 units is followed by ReLU. The last one which has 1 unit ends in Sigmoid function for generating a scalar value ranged from 0 to 1.

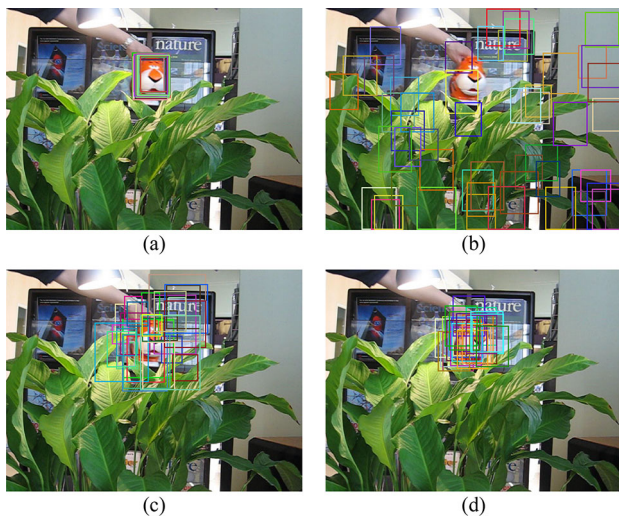


Fig. 5 Illustration of sampling strategy. During instance initialization, **a** is Gaussian sampling for positives on the *Tiger* sequence; **b**, **c** are global sampling and uniform sampling for negatives, respectively. During tracking phase, **d** shows results of Gaussian sampling in the next frame, and these candidate boxes serve the purpose of the decision of new location

4 Online tracking

4.1 Instance initialization in tracking

This section mainly introduces how to track the target when given a specified sequence. Once we completed the training procedure, the instance initialization using one-shot learning is employed to make the whole network possess video-specific knowledge, without any further online adapting. In each training iteration, we treat the instance initialization as 2-way 1-shot problem. Firstly, N_{neg} negative samples and N_{pos} positive samples are generated around the target box. Then, we randomly select one positive and negative sample as sample set, and the rest of positive samples and negative samples are pushed into query set. Exploiting these two sets, we can further fine-tune the network in a one-shot learning way. Here the sampling strategy is implemented in three ways. Assuming that object motion abides by Gaussian distribution, we apply Gaussian random sampling [27] to obtain positive samples $\{\tilde{b}_i\}_{i=1}^{N_{pos}}$ (Fig. 5a). For negative samples $\{\tilde{b}_i\}_{i=1}^{N_{neg}}$, uniform sampling and global sampling (Fig. 5b, c) are adopted to cover as much backgrounds as possible. When initializing the network, we select different learning rate for two modules. This is because feature embedding module has generic representation ability and the metric module has video-specific ability for a given video instance. Therefore, it is natural that the learning rate of feature embedding module (0.0001) should be smaller than the latter (0.001). In Fig. 3b, c, it is worth noting that a more higher-quality confidence map can be obtained after instance initialization.

4.2 Candidate sampling and decision of location

In the coming frame, we employ Gaussian random sampling around the predicted box of previous frame to generate N_{det} target candidates $\{\tilde{b}_i\}_{i=1}^{N_{det}}$ (Fig. 5d). Since what we can only trust is the bounding box \tilde{b}_{tep} in the first frame, we regard it as the template patch and compare all candidate boxes with it in each frame. Using deep metric network fine-tuned with the instance initialization, we can eventually obtain their similarity scores $\phi(F[\tilde{b}_i], F[\tilde{b}_{tep}])$. During online tracking phase, there is no need to compute the template feature repeatedly in every frame. To be efficient, we just calculate and store it after the instance initialization.

In addition, it is pretty reasonable that most objects in the real world tend to move smoothly through space [12]. In other words, most situations ought to be that the target in the next frame should be near to the location where it is observed on the previous frame. So, taking into account motion smoothness, we utilize Gaussian kernel function to control space relationship of two sequential frames

$$\kappa(c_i, c_{pre}) = e^{-\|c_i - c_{pre}\|^2 / \sigma^2}, \quad (7)$$

where $c_i = (c_i^x, c_i^y)$ is the center of the candidate box \tilde{b}_i , c_{pre} is the center of the previous tracked box, and σ is bandwidth. The optimal location is determined by finding the candidate box with the maximum similarity score

$$b^* = \arg \max_{\tilde{b}_i} \phi(F[\tilde{b}_i], F[\tilde{b}_{tep}]) + \beta \kappa(c_i, c_{pre}), \quad (8)$$

where β is a balance factor.

5 Experiments

This program was implemented on the following configuration: Python using Tensorflow1.3.0, 4.00GHz Intel Core i7-4790K CPU with 8 cores, 32GB RAM, Nvidia GeForce GTX 1080Ti GPU. The speed of our method runs at approximately 16 frames per second. To evaluate the performance of our proposed DMN, experiments were conducted on two prevalent benchmarks: OTB [45,46] and VOT2017 [19].

OTB includes two dataset: OTB2013 [45] with 51 video sequences, OTB2015 [46] with 100 video sequences. These sequences are labeled with ground-truth bounding box and cover various challenging situations in visual tracking, such as illumination variation, background clutter, deformation, occlusion and so on. The tracking performances were evaluated by conducting one-pass evaluation (OPE) based on two metrics: center location error and overlap ratio. The overlap ratio measures IoU between the predicted bounding box and the ground-truth bounding box, i.e., $(area(B_T \cap$

$B_G)/area(B_T \cup B_G)$), where B_T is the predicted bounding box and B_G is the ground-truth bounding box. The center location error measures Euclidean distance between the center of B_G and B_T . Based on these two metrics, success ratio plot and precision ratio plot on the whole benchmark can be drawn with a series of thresholds. Generally, the area under curve (AUC) of each success plot and the precision ratio at the specific threshold (Prec.@20) are calculated to rank different trackers.

VOT2017 [19] contains 60 challenging video clips. And the tracking performance is measured by expected average overlap (EAO), accuracy (A) and robustness (R). Accuracy measures the average overlap between the predicted and ground-truth bounding boxes. Robustness measures how many times the tracker fails. The primary one, EAO, evaluates the average overlap that a tracker is expected to perform on plenty of sequences with the same attribute.

We first introduced implementation details of our experiment and then analyze DMN with ablation experiments to validate the effectiveness of each part. Afterward, we compared our DMN tracker with metric learning-based trackers. Finally, we performed comparison experiments with some representative trackers.

5.1 Implementation details

5.1.1 Network training

For offline training of DMN, we used the Amsterdam Library of Ordinary Videos (ALOV) [32] as it covers diverse circumstances, such as illuminations, specularity, confusion with similar objects, occlusion, zoom, severe shape changes, motion patterns and so on. ALOV consists of 314 video sequences whose total frames are more than 89000. Before training, we removed twelve video sequences that are also included in OTB [45,46] in case of overfitting. For each video, we randomly chose dozens of frame pairs and resized its resolution to 512×512 . Then, we drawn a number of box pairs for each frame pair without any data augmentations. One element in a box pair was the ground-truth bounding box in one frame, and the other element was a box sampled around the ground-truth box of the other frame. The box pair was regarded as positive training data if the IoU ratio of sampled box and its corresponding ground-truth box was larger than 0.7 and negative training data if the IoU ratio was smaller than 0.5. The rest of the box pairs was discarded.

During training procedure, we totally had more than 9000 frame pairs coming from ALOV and each frame pair included 128 box pairs. And we applied stochastic gradient descent (SGD) with momentum of 0.9 to train DMN and set the weight decay to 0.0001. The learning rate of feature embedding module and metric module was 0.0001 and 0.001, respectively. The parameter λ aims to balance the magnitude

of two losses L_S and L_F , and is smaller than 0.01 practically. Here, we set it as 0.001. The parameter ρ^+ , which is usually larger than 0.65, determines whether a sample is positive or not. Here, we set ρ^+ as 0.7. The parameter α makes the feature distance between positive and negative samples large than a margin. In this work, α were experimentally set as 1.2.

5.1.2 Online tracking

In the first frame, we firstly drawn $N_{pos} = 33$ positive samples and $N_{neg} = 96$ negative samples roughly according to the ratio of 1:3, and then fine-tuned the network with 100 iterations. After the instance initialization, DMN began to track the target in the coming frame, without updating one by one. In order to generate candidate boxes in the coming frame, we drawn $N_{det} = 200$ samples from the Gaussian distribution in three dimensions: x -axis c_i^x , y -axis c_i^y and scale s_i . The mean of Gaussian distribution was the center of the previous target box b^* , and covariance was a diagonal matrix $diag[(0.5r)^2, (0.5r)^2, 0.5^2]$, where r was the mean of the width and height of the b^* . The scale of each candidate bounding box was computed by multiplying 1.05^{s_i} to the initial target scale. Finally, the candidate box with the highest similarity was determined as the target location. In this phase, the parameter σ , bandwidth of Eq. (7), was empirically set as 10. The parameter β aims to balance the weight of similarity score and motion smoothness, and its value is generally between 0 and 1. Here, it was empirically set as 0.15.

5.2 Analyses of DMN

5.2.1 Self-comparison

To validate the effectiveness of our proposed deep metric network (DMN), we designed three ablation experiments conducted on OTB2013 and OTB2015 benchmarks. Firstly, DMN-MSE, which was trained with the mean square errors (Eq. 4), was designed to verify the feasibility of jointly learning feature embedding module and deep metric module. And its parameter settings kept the same with DMN. Secondly, DMN-Cosin replaced the metric module of DMN with the specific cosin metric, aiming to evaluate whether the learnt metric in a joint manner is better than hand-crafted metric or not. Lastly, in order to validate the significance of the fusion operation f_C , DMN-NoFusion removed the fusion operation f_C and only utilized feature maps of the last block to train the network, while its configurations were also consistent with DMN.

As shown in Fig. 6, DMN completely outperformed all the variants according to the precision and success plots on OTB2013 and OTB2015. Here, we tersely introduced

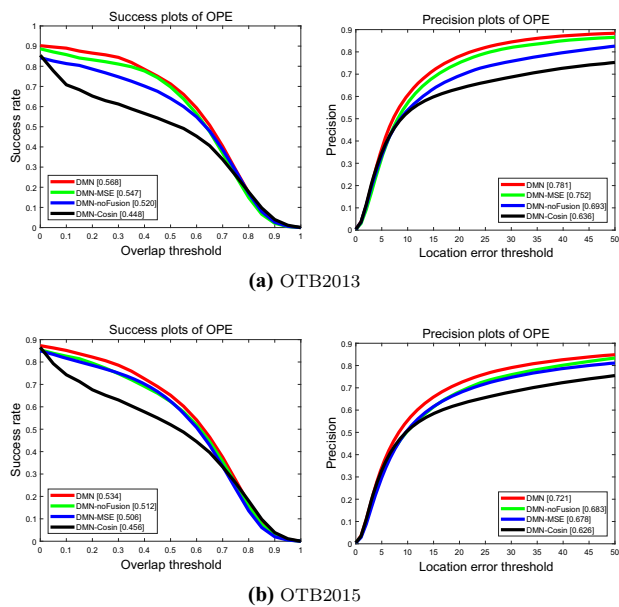


Fig. 6 The success plots and precision plots of our variants on OTB2013 and OTB2015. The different variants in the left and right figures are ranked according to AUC and Prec.@20, respectively

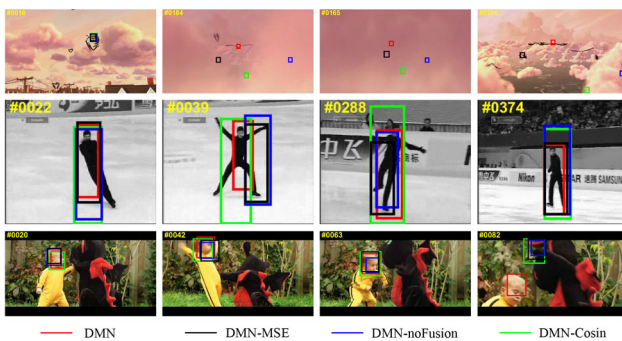


Fig. 7 Tracking results of our variants on the *Bird1*, *Dragonbaby* and *Skaterd2* sequences coming from OTB benchmark. Frame number is shown in the left-top of each image

the performances on OTB2015 benchmark. Compared with DMN, DMN-MSE decreased by 4.3% and 2.8% in precision ratio and AUC on OTB2015, which favorably proved that the network was benefit from our joint loss. The results that DMN gained 2.2% and 3.8% additional improvement than DMN-NoFusion in precision ratio and AUC on OTB2015 provided compelling evidences for the necessity of the fusion operation f_c in feature embedding module. In addition, DMN-Cosin only obtained 62.6% and 45.6% performance, while DMN achieved 72.1% and 53.4% in precision ratio and AUC, which clearly demonstrated that the feature embedding and the learnt metric can be well coupled with each other. Visualization results of some video sequences further illustrated that every part of our design could play an appreciable role in DMN, as can be seen from Fig. 7. Concretely, in the *Bird1* sequence, we can clearly observed that DMN captured

Table 1 Ablation studies of the instance initialization. We fix the weights of feature module and metric module, respectively, to show their impacts. The area under curve (AUC) of success plots on OTB benchmark is reported

Variants	OTB2013	OTB2015
DMN	0.568	0.534
DMN-updateMetric	0.555	0.528
DMN-updateFeat	0.518	0.482

the target, while three variants lost it over time. In the *Dragonbaby* and *Skaterd2* sequences, it was apparent that DMN predicted more tighter boxes than three variants to cover the target.

Moreover, in order to explore how two modules impact the final results during the instance initialization, we frozen feature module and metric module, respectively, and test their performance on OTB benchmark. In Table 1, DMN-updateFeat only fine-tunes feature module, and DMN-updateMetric only fine-tunes metric module. As shown in Table 1, DMN-updateMetric is superior to DMN-updateFeat. Thus, the instance initialization process focuses on adjusting the metric module, which enables the network to learn what metric a specific sequence should apply during tracking phase.

5.2.2 Comparison with existing metric learning-based methods

Based on the conventional Mahalanobis distance, MLSAM [47] and ITML [6] were designed to learn a linear metric for visual tracking. In contrast to MLSAM and ITML, DML [15] brought the idea of deep metric learning into single object tracking, which aimed to learn a set of hierarchical nonlinear transformations using fully connected network architecture. It projected the template patch and candidate patches into a latent feature subspace and eventually determines the best bounding box in the next frame by Euclidean distance. However, we introduced an end-to-end deep metric network with jointly learning deep feature embedding and deep nonlinear metric, which directly outputted a scalar value.

The comparison results in Table 2 demonstrated that our DMN was superior to DML, MLSAM and ITML in terms of AUC and precision ratio on OTB2013 benchmark and the speed of DMN achieved considerable improvement. As shown in Table 2, we observed that our method obtained approximately 10–20% and 18–28% improvement in precision ratio and AUC when compared with three other metric learning-based methods. Moreover, the speed of our method can reach 16 frames per second, which was thrice as fast as DML.

Table 2 Performances of different metric learning-based methods on OTB2013. Frames per second (FPS) evaluate the speed of tracker. Our results are bold

Tracker	AUC	Prec.@20	FPS
DMN	0.568	0.781	16
DML [15]	0.466	0.603	5
MLSAM [47]	0.380	0.526	–
ITML [6]	0.365	0.501	–

Table 3 Comparison results with state-of-the-art methods on OTB2013. RSST-Color, RSST-HOG and RSST-Deep are three variants of RSST with different features. Our results are bold

Tracker	AUC	Prec.@20	FPS
DMN	0.568	0.781	16
RSST-Color [50]	0.520	0.691	4
RSST-HOG [50]	0.543	0.726	–
RSST-Deep [50]	0.590	0.789	1
HCPT* [24]	0.638	0.923	7
adaDDCF [9]	0.643	0.882	9

5.3 Comparison with others

5.3.1 Quantitative evaluation

OTB benchmark We compared our DMN with state-of-the-art methods including HCPT* [24], RSST [50], and adaDDCF [9] on OTB2013. Particularly, we fully reported the results of RSST with three different features: gray color (RSST-color), Histogram of Oriented Gradient (RSST-HOG) and VGGNet (RSST-deep). For through evaluations, we also compared the proposed method with 11 representative trackers on OTB2013 and OTB2015. These trackers include three correlation filter-based methods (CSK [14], KCF [13], DSST [5]), two matching function-based methods (IVT [29], MTT [49]), three deep learning-based methods (DLT [40], SINT¹ [34], SiamFC [1]), and three other track-by-detection methods (Struck [10], MIL [17], CT [48]). All results were obtained fairly using the OTB toolkit [45].

As illustrated in Table 3, our DMN, which can be seen as the extension of DML [15], achieved comparable performance against five state-of-the-art trackers. Concretely, while DMN was slightly inferior to RSST-Deep in AUC and precision ratio, it was superior to both RSST-Color and RSST-HOG. This indicated that RSST greatly benefited from a powerful feature. In addition, both HCPT* and adaDDCF adopted the VGGNet deep features, which lays a solid foundation for accurate tracking results. And although HCPT*

¹ SINT is a version without optical flow, and its results were obtained on our own PC using the pre-trained Caffe model.

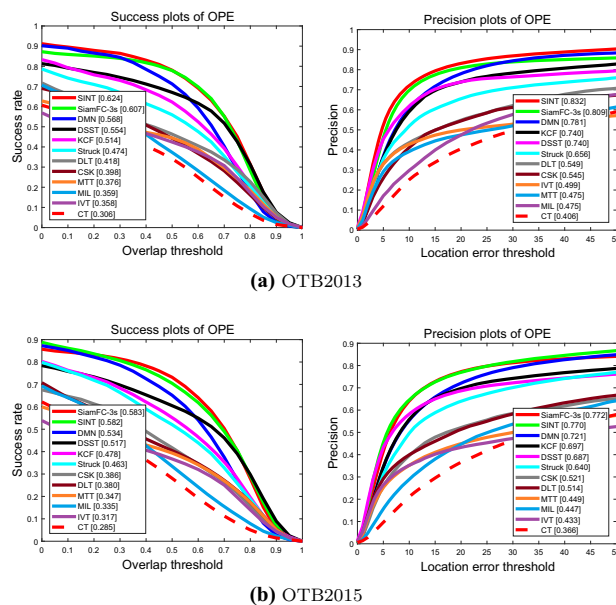


Fig. 8 The success plots and precision plots of different trackers on OTB2013 and OTB2015. The trackers in the left and right figures are ranked according to AUC and Prec.@20, respectively. The results of SiamFC-3s were obtained using three scale estimation

and adaDDCF obtained better performance, their speeds were significantly slower than DMN.

Figure 8 further shows the comparison experiments between the proposed DMN and another 11 representative trackers on OTB2013 and OTB2015 benchmarks. Without box refinement, occlusion detection and online updating, our method virtually surpassed all trackers in terms of AUC and precision ratio on two benchmarks, except for two deep learning-based methods: SiamFC, SINT. Specially, DMN significantly increased by 15.4% and 20.7% against DLT according to the AUC and precision on OTB2015. The reasons were that the proposed joint loss facilitated the learning of features, and DMN measured the underlying relation between two samples by the learnt metric when DLT scantily predicted the classification score of one sample by sigmoid layer. Moreover, due to the effective feature embedding and coupled nonlinear metric, DMN considerably achieved 9.4% and 12.5% improvements against Struck in terms of the AUC and precision on OTB2013. On the other hand, the AUC and precision ratio of SiamFC was slightly superior to that of DMN because 1) DMN paid attention to the learnable metric, using a single branch network rather than Siamese network; 2) SiamFC used an ample enough dataset (ILSVRC [30]) to train the fully convolution Siamese network; and 3) SiamFC densely convolved the search region with the template. Furthermore, it was worth to mention that SINT ranked first, but it ran at 3 FPS. And our DMN was over five times faster than SINT.

Table 4 The tracking performances on VOT2017 benchmark are shown partly. EAO, A and R are expected average overlap, accuracy and robustness, respectively [19]. Our result is bold

	DMN	LSART	ECOhc	MEEM	SiamFC	Staple	KCF	SRDCF	MIL	DSST	IVT
EAO \uparrow	0.159	0.323	0.238	0.192	0.188	0.169	0.135	0.119	0.118	0.079	0.076
A \uparrow	0.451	0.493	0.494	0.463	0.502	0.530	0.447	0.490	0.393	0.395	0.400
R \downarrow	0.492	0.218	0.435	0.534	0.585	0.688	0.773	0.974	1.011	1.452	1.639

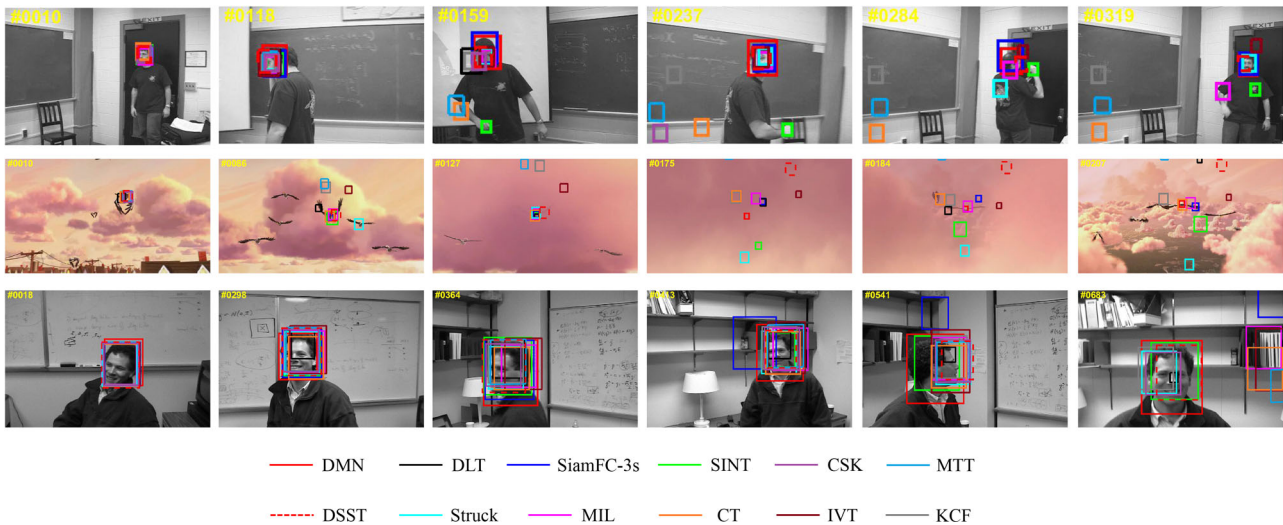


Fig. 9 Visual tracking results on three sequences with out-of-view, out-of-plane rotation and in-plane rotation. Frame number is shown on the left-top of each image. The sequences from top to bottom are *Freeman1*, *Bird1* and *FleetFace*

VOT2017 benchmark We also compared our DMN with some state-of-the-art trackers on VOT2017 baseline sub-challenge. As shown in Table 4, DMN obtained satisfying performance under the measurement of expected average overlap (EAO \uparrow), accuracy (A \uparrow) and robustness (R \downarrow). Specially, according to EAO, A and R, DMN was completely superior to KCF, SRDCF, MIL, DSST and IVT. And the robustness of DMN (0.492) was better than MEEM (0.534), SiamFC (0.585) and Staple (0.688) though the EAO of DMN was lightly inferior to theirs. The reason is that DMN captured some common properties, such as robustness to illumination and blur, and could properly handle variations in each individual sequence by the instance initialization. It is worth mentioning that the accuracy of DMN achieved comparable performance (0.451) against LSART (0.493) which is the top-ranked trackers on VOT2017 benchmark. In the future box regression may possess the potential to improve the accuracy of DMN.

5.3.2 Qualitative evaluation

From Figs. 9, 10 and 11, many representative sequences with different challenging factors were tested for comprehensive comparison. Generally, our DMN could obtain satisfying

visualization tracking results when confronted with various challenges.

Specially, Fig. 9 shows the qualitative tracking results on several sequences, where out-of-view, out-of-plane rotation and in-plane-rotation occurred. DMN obtained considerably stable results on the *Freeman1* and *Bird1* sequences. Unfortunately, SINT and DLT, which had no joint loss function, gradually lost the target. And DMN had a better performance than SiamFC on the *FleetFace* sequence, as the hand-crafted metric, inner product, was not discriminative enough for abrupt appearance variation.

Figure 10 presents the visualization results of the *Coke*, *FaceOcc1*, *Tiger1* and *David3* sequences. With the instance initialization and video-specific metric, the proposed DMN performed well on the *Coke* and *Tiger1* sequences, whereas some discriminative models (Struck, MIL, DSST, KCF, CSK) drifted to the background due to blur, occlusion and fast motion. It was also noted that SiamFC lost the target of the *David3* sequence as a result of the occlusion of a tree.

Furthermore, in Fig. 11, the visualization tracking results from several representative frames are illustrated, where the targets from top to bottom suffer from scale variation, illumination variation and background clutters. As we can see, due to the instance initialization that gives the network sequence-specific information, DMN could favorably adapt the scale

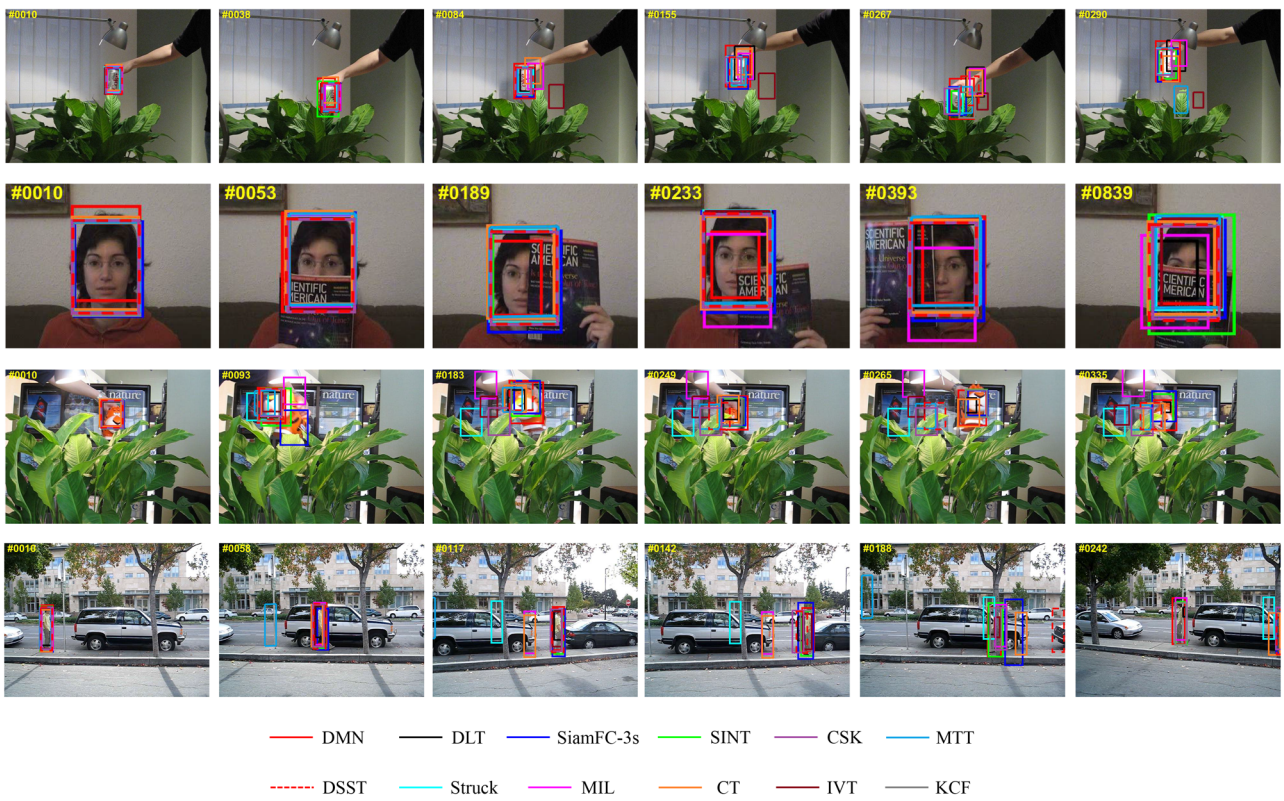


Fig. 10 Visual tracking results on four sequences with motion blur, occlusion and fast motion. Frame number is shown on the left-top of each image. The sequences from top to bottom are *Coke*, *FaceOcc1*, *Tiger1* and *David3*

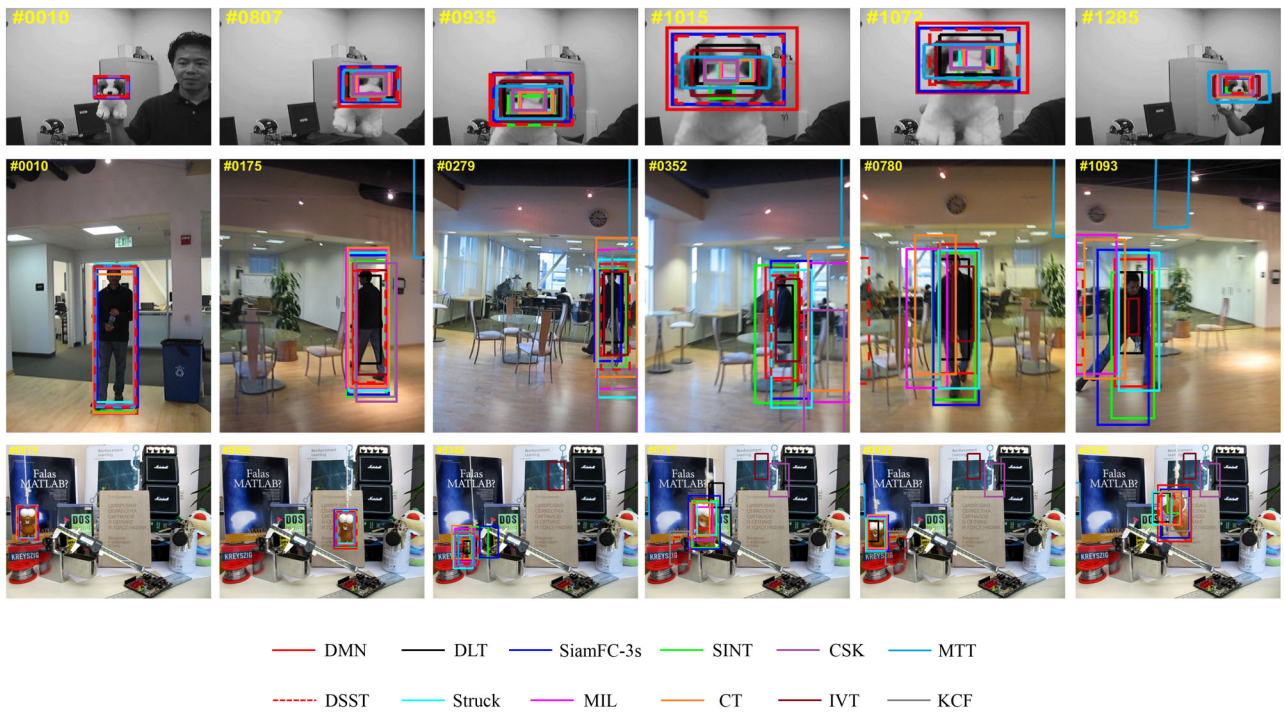


Fig. 11 Visual tracking results on three sequences with background clutters, scale variation, illumination variation. Frame number is shown on the left-top of each image. The sequences from top to bottom are *Dog1*, *Human2* and *Lemming*

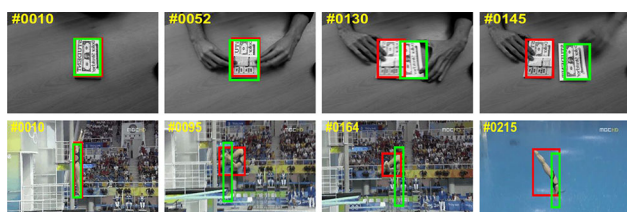


Fig. 12 Failure cases of our DMN. The sequences from top to bottom are *Coupon* and *Diving*. Green boxes are the results of our method. And red boxes are the ground-truth bounding boxes. Both sequences fail to track their target

change on the *Dog1* and *Human2* sequences. However, limited by the fixed metric, some matching function-based methods (MTT and IVT) cannot successfully track these two sequences. Note that, as a result of the three sampling strategies used in the instance initialization, DMN could adequately suppress distractors in the first frame. Our tracker therefore obtained satisfactory tracking results on the *Lemming* sequence.

5.4 Failure case

Figure 12 shows a few failure cases of our method. Firstly, DMN failed to track the target on the *Coupon* sequence. There were three indispensable reasons: (1) the coupon appearance was changed abruptly due to the fold; (2) our method had no online updating; and (3) the background was extremely similar to the template. Secondly, with the drastic deformations of the athlete, DMN cannot also generate accurate bounding boxes on the *Diving* sequence.

6 Conclusion

In this paper, we present a simple yet efficient tracking framework DMN, which differs from existing metric learning-based trackers. The proposed DMN can jointly learn feature embedding and coupled nonlinear metric, and directly output a scalar to determine the best candidate box. Moreover, we design three ablation experiments to illustrate the validity of each part of DMN. Compared with existing metric learning-based methods, the proposed tracker expectably learns a coupled nonlinear metric to locate the target and achieves a superior performance. Furthermore, experiments on the OTB and VOT benchmarks show that our DMN tracker obtains competitive results against other representative trackers. However, there still remain several failure scenarios. In the future, it ought to be elaborately considered to improve the performance by means of online updating, an ample train-

ing dataset and data augmentation strategies of hard positive and negative samples.

Funding This work was funded by the National Natural Science Foundation of China (Grant Number U1811463).

Compliance with Ethical Standards

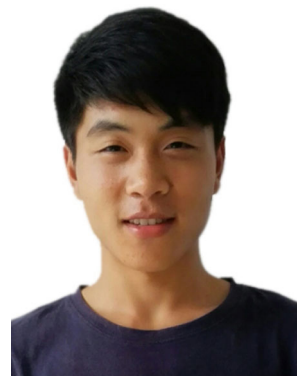
Conflict of interest The authors declare that they have no conflict of interest.

References

- Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision (ECCV), pp. 850–865 (2016)
- Briechele, K., Hanebeck, U.D.: Template matching using fast normalized cross correlation. In: Proceeding of SPIE on Optical Pattern Recognition XII (2001)
- Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 539–546 (2005)
- Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2000)
- Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference (BMVC) (2014)
- Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: International Conference on Machine Learning (ICML) (2007)
- Elgammal, A., Duraiswami, R., Davis, L.S.: Probabilistic tracking in joint feature-spatial spaces. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2003)
- Girshick, R.B.: Fast r-cnn. In: IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015)
- Han, Z., Wang, P., Ye, Q.: Adaptive discriminative deep correlation filter for visual object tracking. *IEEE Transactions on Circuits and Systems for Video Technology* (Early Access) (2018)
- Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.S.: Struck: structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(10), 2096–2109 (2016)
- He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
- Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: European Conference on Computer Vision (ECCV) (2016)
- Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 583–596 (2015)
- Henriques, J.F., Caseiro, R., Martins, P., Batista, J.P.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision (ECCV) (2012)
- Hu, J., Lu, J., Tan, Y.P.: Deep metric learning for visual tracking. *IEEE Trans. Circuits Syst. Video Technol.* **26**(11), 2056–2068 (2016)
- Jiang, N., Liu, W., Wu, Y.: Order determination and sparsity-regularized metric learning adaptive visual tracking. In: IEEE

- Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1956–1963 (2012)
17. Kehl, R., Bray, M., Gool, L.V.: Full body tracking from multiple views using stochastic sampling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
 18. Kim, K., Lepetit, V., Woo, W.: Scalable real-time planar targets tracking for digilog books. *Vis. Comput.* **26**(6–8), 1145–1154 (2010)
 19. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Puggfelder, R., et al.: The visual object tracking vot2017 challenge results. In: IEEE International Conference on Computer Vision Workshop (ICCVW) (2017)
 20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105 (2012)
 21. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
 22. Li, X., Shen, C., Shi, Q., Dick, A.R., van den Hengel, A.: Non-sparse linear representations for visual tracking with online reservoir metric learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 1760–1767 (2012)
 23. Lu, J., Hu, J., Tan, Y.P.: Nonlinear metric learning for visual tracking. In: *IEEE International Conference on Multimedia and Expo (ICME)* (2016)
 24. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Robust visual tracking via hierarchical convolutional features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Early Access)* (2018)
 25. Ma, Z., Wu, E.: Real-time and robust hand tracking with a single depth camera. *Vis. Comput.* **30**(10), 1133–1144 (2014)
 26. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 1436–1443 (2009)
 27. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4293–4302 (2016)
 28. Rechy-Ramirez, E.J., Marin-Hernandez, A., Rios-Figueroa, H.V.: A human–computer interface for wrist rehabilitation: a pilot study using commercial sensors to detect wrist movements. *Vis. Comput.* **35**(1), 41–55 (2019)
 29. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **77**(1–3), 125–141 (2008)
 30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
 31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint: arXiv:1409.1556* (2014)
 32. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: an experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7), 1442–1468 (2014)
 33. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
 34. Tao, R., Gavves, E., Smeulders, A.W.M.: Siamese instance search for tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1420–1429 (2016)
 35. Tsagkatakis, G., Savakis, A.E.: Online distance metric learning for object tracking. *IEEE Trans. Circuits Syst. Video Technol.* **21**(12), 1810–1821 (2011)
 36. Valmadre, J., Bertinetto, L., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5000–5008 (2017)
 37. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. *Vis. Comput.* **29**(10), 983–1009 (2013)
 38. Wang, D., Lu, H., Yang, M.H.: Least soft-threshold squares tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2371–2378 (2013)
 39. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring rich feature hierarchies for robust visual tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015)
 40. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 809–817 (2013)
 41. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.: Learning attentions: residual attentional siamese network for high performance online visual tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
 42. Wang, S., Lu, H., Yang, F., Yang, M.H.: Superpixel tracking. In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 1323–1330 (2011)
 43. Wang, X., Hua, G., Han, T.X.: Discriminative tracking by metric learning. In: *European Conference on Computer Vision (ECCV)* (2010)
 44. Wang, X., Li, C., Luo, B., Tang, J.: Sint++: Robust visual tracking via adversarial positive instance generation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018)
 45. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013)
 46. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1834–1848 (2015)
 47. Wu, Y., Ma, B., Yang, M., Zhang, J., Jia, Y.: Metric learning based structural appearance model for robust visual tracking. *IEEE Trans. Circuits Syst. Video Technol.* **24**(5), 865–877 (2014)
 48. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: *European Conference on Computer Vision (ECCV)* (2012)
 49. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via structured multi-task sparse learning. *Int. J. Comput. Vis.* **101**(2), 367–383 (2013)
 50. Zhang, T., Xu, C., Yang, M.H.: Robust structural sparse tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(2), 473–486 (2019)

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



Shengjing Tian is currently pursuing his Ph.D. degree in the School of Mathematical Sciences, Dalian University of Technology (DUT). His current research interest is visual tracking and reconstruction.



Shuwei Shen is currently a post-graduate student in the School of Mathematical Sciences, DUT. Her current research interest is visual tracking.



Xiuping Liu is currently a Professor in the School of Mathematical Sciences, DUT. Her research interests are computer vision, computational geometry and computer graphics.



Guoqiang Tian is currently a post-graduate student in the School of Mathematical Sciences, DUT. His current research interest is visual tracking.



Baocai Yin is currently a Professor in the Faculty of Electronic Information and Electrical Engineering, DUT. His research interests include digital multimedia technology, virtual reality and graphics technology.