2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Joint Identification-Verification Model for Visual Tracking

Min WU Yufei ZHA* Yuanqiang ZHANG Tao KU Lichao ZHANG Bin CHEN
(Aeronautics and Astronautics Engineering College, Air Force Engineering University)
e-mail address：kj123123213@163.com

Abstract—Similarity algorithms determine the location of the target by the similarity between the template and the candidate, the most similar candidate to the template is considered as the target in visual tracking. Similarity algorithms search the most similar candidate to the template as the current estimation for visual object. In practice, most trackers only take usage of the intra-class similarity, yet the inter-class semantic separability is ignored. In this paper, a joint identification-verification model is proposed to learn the similarity with the category attribute for visual tracking. This approach constructs the cost function both on the inter-class semantic separability and intra-class similarity, firstly. Then, the training dataset is fed into the network. To the end, the discriminative features are learned in the embedding space. During tracking phase, the template and candidates are fed into the network simultaneously. Thereforce, the target will be located correctly by the similarity metric between the template and candidates in the learned embedding space. We evaluate the proposed approach on the open benchmark: OTB50 and UAV123 dataset. A large number of experimental results show that the inter-class semantic separability can increase the discrimination for the similar distractors effectively, and bootstrap the tracking performances of the trackers based on the similarity learning.

***Keywords*—visual tracking, similarity learning, identification-verification, inter-class separability**

Ⅰ Introduction

Visual tracking is a basic problem in the community of computer vision. The target's position and scale in the next frame are located quickly and accurately by giving the location and scale of the target in the initial frame. However, visual tracking is easy to fail because the target's appearance is varied due to external factors such as scale, rotation and deformation, as well as internal factors such as similar targets, complex backgrounds and lighting [1, 2].

Similarity tracking can learn the matching relationship between patch pairs directly, and improve the ability of the target representation. Hence, similarity tracking has been received academic attention in recent years [3, 4]. YCNN [5] is the early algorithm to use Siamese network for tracking, it trains a network to map an exemplar and a search region into a response map. SINT [6] uses deep learning to track targets, and trains the Siamese network to identify the position of candidates that match the initial target. Different from SINT, Siamese-fc [4] uses a full convolutional structure in the search image, so it's tracking speed can be achieved in real time. So far, most similarity trackers extract the embedding features of the target by Siamese neural network, which are learned by offline end-to-end training. In the tracking phase, the best match is attained by regression or sorting in the search area. On the one hand, similarity learning can achieve features learning and similarity measures [6] simultaneously. On the other hand, they have powerful characterization ability and good real-time in tracking [7, 8].

Figure 1: Tracking result of different algorithm. We select 36th, 75th, 83rd and 96th from similar target video sequence. The red box is the tracking results of our algorithm, and the tracking results of Siamese-fc is blue box.

As shown in Figure 1, the red box is the tracking results of our algorithm, and the tracking results of Siamese-fc is blue box. At 36-th and 75-th, both our algorithm and the Siamese-fc algorithm can track the target well. But at 83-th and 96-th, a distractor appears around the target. Because the distractor has great similarity with the target, the distractor's match score may be bigger than the true target. To the end, the tracker will fail. The main reason for tracking failure is that the deep embedding features supervised by the weak similar label are not discriminative to distinguish the target from the background.

To address the above problem, a joint identification-verification model is proposed in this paper. This method exploits the target category attribute as stronger supervised signal to enlarge the inter-class margin. .To the end, it pulls the samples that belong to same category of the target, yet pushes the samples that belong to different category fo the target. Moreover, the learned embedding features can enlarge the margin between different category samples. That is to say, the features learned by joint loss are intra-class similarity and inter-class semantic separability. The contributions of this work can be summed up as:

- **Joint Loss:** Verification loss focuses on the similarity of the target, yet is invalid for the semantic of the target. On the contrary, identification loss can discriminate the different category samples, yet ignore the intra-class variations. The proposed method can complement each other effectively.

- **Rich experiments:** This article has conducted a large number of experiments on the weights between different losses and various scenes. Extensive experiments fully demonstrate the effectiveness of the proposed algorithm.

Ⅱ Related Work

A. *Siamese CNN For Tracking*

In recent years, deep learning has many successful applications in the field of target tracking, and gradually surpasses traditional methods in performances. However, it is difficult to train a deep model from the beginning to track the current target.

Recently, the Siamese networks are applied for visual tracking task, which learns similarity and features jointly suitable for the Y-shaped network structure. All proposed methods are needed to train hierarchical convolutional features[] of arbitrary training targets with verification information[] offline fed into the end-to-end network by using back-propagation[]. On the one hand, Siamese networks can learn the generic similarity of new targets, even the target category not present in the training datasets. On the other hand, the approach can learn feature and similarity jointly and be understood better for its simplicity.

B. *Metric Learning*

The metric learning aims at learning a mapping to an embedding space, where distance is in correspondence with a predefined notion of similarity. On the one hand, the softmax function [9] is used to learn similarity with richer identity - related information to deeply features. Otherwise, the normalized feature is modified the softmax to promote the performance. In essence, when the feature and parameter are normalized simultaneously, the formulation of the softmax can be consistency with the distance-based metric learning.

What is more, many methods learn the similarity directly through the distances. Contrastive loss [10] aims at learning model to make the distance of the samples belonged to same label smaller than the margin, yet the samples with different label need to bigger than the margin, which is used as dimensionality reduction in the original work. In order to take full advantage of the training batches, the sample pairs are designed to replace the original random selection.

V. OUR APPROACH

A. Siamese Learning for Tracking Revisited

The training data including similar samples and dissimilar samples are fed into the Siamese network to learn deep model, which is utilized to identify the target among the candidates. Specially, only the initial target is used in most Siamese trackers. Given the labels between the target o and the candidates u, the similarities function h() can learned by the loss:

$$L = \sum_j h(r(o,u_j), y_j), \tag{1}$$

where $y_i \in \{+1,-1\}$ is the label of the candidate u_j, and $r()$ is the respond between the template and candidate.

During tracking procedure, the template is labeled in the initial frame. Then, the candidates in the following frame are sampled and are fed into the network with the template. In the embedding features space, the best matched candidate is considered as the tracked result:

$$\hat{x}_t = \arg\max_{j=1,2,\ldots N} r(x_0, x_j), \tag{2}$$

where x_0 is the template, and x_j is the j-th candidate in the current frame. Eq. 2 shows that the tracked result is obtained by looking for the maximum respond between the template and the candidate.

In summary, target tracking algorithm based on Siamese network uses deep neural networks to extract similarity features between the template and the candidate. Specially, this method can achieve ultra-real-time frame rate for no-updating, yet most CNN target trackers are very slow. However, the similar label is week supervised signal, which only indicates the sample pair is similar or not. As a result, this will reduce the discrimination of the learned embedding features, which are the key issues in the robust target tracking. Namely, this approach only focused on intra-class similarity, yet the inter-class separability is ignored.

B. *Identification-Verification Model*

In our approach, in order to achieve the discriminative feature, we take the identification and verification of the samples into account simultaneously. That is to say, we expect that the learned embedding features are inter-class separability and intra-class similarity.

The samples are divided into positives and negatives according to the overlaps between the samples and ground truth. The identification attribute can distinguish the samples with different labels in the learned high-dimensional space. What is more, the joint identification-verification loss guides the network to learn the more discriminative features, which can effectively distinguish the target from the background. The main idea of our method is shown in Figure.2. When the inputs are similar targets, their features extracted by neural network are very similar (Figure. 2(b)). Based on the abovethe similarity constraint is increased to the network, which can reduce intra-class distance and obtain the deep compact feature (Figure. 2(c)). In this case, the compact feature can't distinguish similar targets, this is not conducive to tracking task. To solve this problem, we increase the category to restrict network according to the target category can distinguish similar targets. To the end it achieves inter-class separability and intra-class similarity, and gains the discriminative features shown in Figure. 2(d).

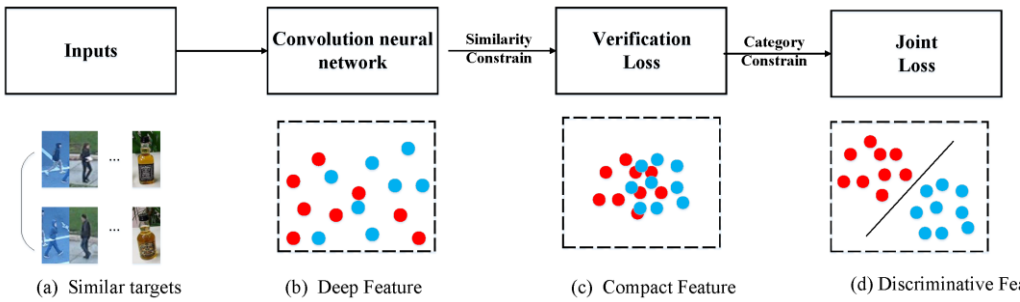

Figure 2: Discriminative feature learning by the joint loss. The Figure 2(a) shows that the similar targets are selected as the inputs, which are fed into the convolutional neural network to extract the deep feature shown in Figure 2(b). Then the similarity constrain is used to construct the verification loss to learn compact feature shown in Figure 2(c), where the similar samples are clustered, yet the different clusters have an overlap. In order to get the discriminative feature, the category attribution is employed to build the joint loss that encourages the intra-class compact and inter-class separability.

targets, this is not conducive to tracking task. To solve this problem, according to the target category can distinguish similar targets, we increase the category to restrict network, achieve inter-class separability and intra-class similarity, and gain the discriminative features shown in Figure 2(d).

Identification Model: Softmax is most common function in most CNN neural network, and is proved as an effective approach to separate the different categories. The outputs are probabilities belonged to the classes. After the samples $\{x_i | i=1, 2\ldots N\}$ are fed into the network, the learned d dimensional features can be denoted as $z \in R^d$. Then the softmax function can be written as:

$$\hat{p}_i = \frac{e^{\mathbf{w}_{y_i}{}^T f(\mathbf{x}_i)}}{\sum_{j=1}^{K} e^{\mathbf{w}_j{}^T f(\mathbf{x}_i)}}, \quad (3)$$

where $w \in R^{d \times k}$ is the filter, K is the number of the classes, x_i belong to the y_ith class. The output p_i represents that the probability belonged to the i-th category. The cross-entropy loss is denoted as:

$$L(f,t) = \sum_{i=1}^{N} -p_i \log(\hat{p}_i), \quad (4)$$

Verification Model: In this work, Siamese network is employed to learn the similarity between the pairs. Contrastive loss and triplet loss are the common functions to pursuit the embedding features, where the Euclidean distance is acted as a metric measure. However, not only semantics but also accuracy location is a key issue in visual tracking.

In this work, the verification model is based on the expected and real response, which is obtained by the Siamese network. Different with the general approach, we do not compare the distance between the embedding features directly, and add convolution layer to calculate the response between the template and the search region. In practice, the response is normalized, then Euclidean distance can be equal to inner product between the responses and the similar labels. Therefore, the verification model is written as hinge loss:

$$L_V = \max(0, -yv + \alpha), \quad (5)$$

where v and y are the response and the label, respectively. The parameter α is the super-parameter, which is the predefined margin.

Unlike previous Euclidean based losses (e.g. contrastive loss) that set a hard invariable margin α to separate positive pairs from negative pairs, the proposed soft margin loss employs the logistic function, which inherits the soft margin without any hyper parameters, as well as the margin is adaptive implicitly in terms of the hinge loss.

$$L_V = log(1 + e^{-yv}), \quad (6)$$

The function $log(1+e^x)$ owns an upper bound of the hinge loss $\max(0, \alpha + x)$ asymptotically, where α is a nonnegative value. It is shown that soft margin approximates to the hinge loss, but it decays exponentially instead of having a hard cut-off. To this end, the new loss formulate the different distances with logistical function, where the margin is encoded implicitly, yet not defined in advance. Moreover, the loss can be implemented stably, and it can achieve the same error in 5 ~ 10 times fewer iterations.

Joint Model: Identification model utilizes the stronger supervised category label to distinguish the different targets. In essence, the model is pursuit the common features in the same categorical target. However, the aim of visual tracking is to locate the special target, not the targets with the same category. On the contrary, verification model expects that the similar samples have small distances than the dissimilar samples, yet the weak labels lead to that the deep model is not discriminative. To address the above problems, a joint identification-verification model is proposed in this work, which can learn more discriminative features to distinguish the target from the background. Namely, the learned features are intra-class similarity and inter-class separability.

The joint identification-verification model can be written as:

$$L = \underbrace{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|D|}\sum_{u \in D}\log(1+e^{-y(u)v(u)})}_{verification} + \lambda \underbrace{\sum_{i=1}^{N} -p_i \log(\hat{p}_i)}_{identification}, \quad (7)$$

where D is the position set of the candidates, λ is the Lagrange constant. $y(u)$ is a expected label in the location u. The first term of Eq. 7 is the verification loss, yet the identification loss is shown in the second part. Therefore, the joint model has the intra-class similarity and the inter-class separability, respectively.

In order to obtain the category labels, we calculate the distance $d = \|u - c\|^2$ between the template and the candidates,

where u and c are the centers of the target and candidates, respectively. If the distance is less than the predefined threshold, then this sample can be considered positive or negative.

$$y(u)=\begin{cases}+1, & if\ \ d<\theta\\ -1, & others\end{cases}, \tag{8}$$

C Target tracking

1) Offline training of shared network

We establish identification-verification model to train deep model with the network. Figure.4 shows the proposed joint identification-verification model, which is composed of an identification loss, a verification loss and the shared network. The input of the network is sample pairs, which include a template and a search image. The templates are derived from the ground truth in one frame of the sequence. While the search images are the extended region of the ground truth from other frames. In the training, the template and search image must be not from the same frame. Specially, the shared network is the full convolutional network, which is used to extract the deep feature for both the identification loss and verification loss. Moreover, the category label is employed to minimize the identification loss, yet the regression value is used to optimize the verification loss.

The shared network is trained by the identification loss and the verification loss simultaneously. In our method, the *softmax* loss is employed as the identification loss, which is formulated by Eq.4. According to the identification loss, it is easy to get the predicted label of the template, which is compared with the category label to get the loss value. The loss value is used to update the network parameters:

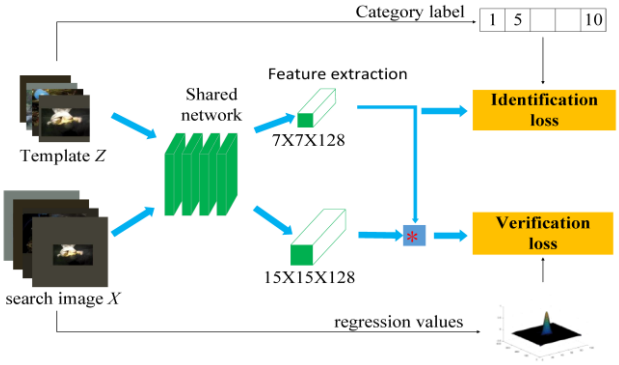

Figure 3: Network architecture of the joint loss model. The ground truth in one frame is selected as template Z, which is fed into the shared network and obtains the deep feature that is 7×7×128. The search image X is the extended region of the ground truth from other frames, these feature is 15×15×128 extracted by shared network. The identification loss is trained by category label; the verification is optimized by regression values.

$$\Delta w_{sh1}=\sum_{i=1}^{N}(w_i^T-\frac{\sum_{j=1}^{N}w_j^T e^{w_j^T f(x)}}{\sum_{j=1}^{N}e^{w_j^T f(x)}})*\frac{\partial f(x)}{\partial w_{sh}}, \tag{9}$$

where Δw_{sh1} is the updated value of the network parameter obtained by the identification loss. z_{n_i} is the n-th convolution result of the i-th template.

Verification loss updates the network through the template, the search image and the corresponding label. The candidate is extracted from the search image, and the candidate closed to the template is defined as the positive sample, the candidate fared away from the template is defined as the negative sample. Then, the cross-correlation is used to calculate the response of the template and the candidate. Guiding w in the verification of Eq. 7:

$$\Delta w_{sh2}=\frac{1}{k}\sum_{i=1}^{k}\frac{1}{|D|}\sum_{u\in D}\frac{-y(u)e^{-y(u)v(u)}}{1+e^{-y(u)v(u)}}*(\frac{\partial v(u)}{\partial w_{sh}}), \tag{10}$$

where Δw_{sh2} is the updated value of the shared network parameter obtained from the verification loss function. x_{n_i} is the n-th convolution result of the i-th candidate, w_n is the convolution kernel of the n-th convolution layer, $x(u)$ is a candidate in the search area.

$$w_{sh}'=w_{sh}+\Delta w_{sh1}+\lambda\Delta w_{sh2}, \tag{11}$$

where w_{sh} is the parameter of the shared network before updating, w_{sh}' is the parameter of the shared network after updating.

2) Online tracking

The offline training process is used to learn the matching mechanism with the network parameters; the online tracking process uses the trained network to extract the features of the template and the search image, which is shown in Figure.4. The ground truth of the first frame is defined as the template x, the search image z is extended region of the ground truth of the current frame. The template and the search image are fed into a well-trained shared network, and obtain the deep embedded features. Then, the function $f(x,z)$ is responsible for comparing the similarity of the deep embedded features between the template and candidate. In order to locate the position of the target in the current frame, the cross-correlation is used to calculate the response of the template and the search image. The candidate with the highest response from all candidates is selected as the tracking result:

$$o_t=\arg\max_{u\in D}(f(z_{t=0},x_t(u))),$$

$$f(z_{t=0},x_t(u))=z_{t=0}*x_t(u), \tag{12}$$

where $z_{t=0}$ is the template, $x_t(u)$ is the candidate in t-th frame. o_t is tracked result in the t-th frame.

Ⅳ Experiments

Our algorithm is implemented in MATLAB2016a using MatConvNet toolbox[], and the evaluated with the state-of-the-art trackers on two public benchmark data sets: OTB50[] and UAV123[]. Our algorithm runs at 35fps on Nvidia

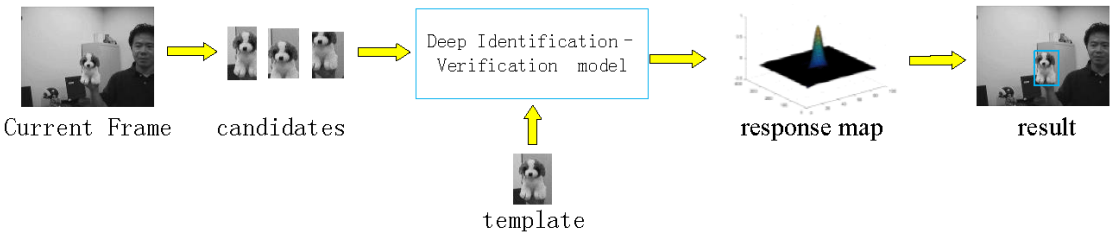

Figure 4: The framework of online tracking. The input of tracking is the current frame, the candidate is obtained from the search image of the current frame by sampled. Then, the template and the candidates are fed into deep identification-verification model to obtain the response. The candidate with the highest response from all candidates is selected as the tracking result.

GeForce GTX 1080Ti GPU and Intel Core i7-6850K with 3.6GHz

A Experiment Setups

Shared Network Architecture: The architecture that we adopt for the embedding function resembles the convolutional stage of the network of Luca Bertinetto [4]. There are five convolutional layers. Max-pooling is employed after the first two convolutional layers. ReLU non-linear active layer is followed by every convolutional layer except for conv5. During training, batch normalization is inserted immediately after every linear layer. The stride of the final representation is eight. An important aspect of the design is that no padding is introduced within the network.

Training data setup: The training data is from ILSVRC database, and there are 4500 videos in the ILSVRC database. However, some videos only appear on a certain part of the body and often appear on the edge of the image, some videos are too large, too small or too close to the boundary, these factors are not conducive to model training, so we removed these videos and give each target in the rest of the database a different label. When training, we select the ground truth of one frame in the video as the template, choose extended region of the ground truth of the other frame as the search image. We have cataloged the database and have 1751 categories in all videos.

B Algorithm validation

We evaluate the proposed algorithm with comparisons to the algorithm based on similarity and the algorithm based on category. To this aim, we select partial data from the ILSVRC database to train the network model. In addition, we choose the similarity algorithm, the category algorithm and our algorithm, and verify these algorithms from both qualitative and quantitative perspectives.

1) Qualitative analysis

Similarity constraint focuses on the intra-class similarity, and ignores the inter-class differently separability. Our algorithm increases the target's category and enhances the model's ability to determine similar goals. Therefore, our method can learn the intra-class similarity and the inter-class separability. In order to prove the effectiveness of our method, we select six sequences with similar goals from the OTB2013 database for verification.

As can be seen from the results of Figure.5, our algorithm can achieve robust tracking in sequences with 4 attributes: similar targets, background interference, occlusion and fast moving. 'Liquor' and 'Matrix' are sequences with similar goals and fast moving. In 708-th frame at the Liquor, similarity algorithm, category algorithm and our algorithm can track the target accurately, but in the 1111-th frame, the similarity algorithm will track similar targets when similar targets appear around the target, our algorithm can still track the target accurately. In 68-th frame and 90-th frame at the 'Matrix', our algorithm can still effectively track when similar targets appear around the target.

'Subway' and 'Soccer' are sequences with similar targets and occlusion. In 35-th frame at the 'Subway', similarity algorithm, category algorithm and our algorithm can track the target accurately, but in the 39-th frame, similar targets occlude the tracking target, the similarity algorithm tracks drift at this time. Our algorithm increases the constraints on the target, so that extracted features reflect more accurately the true target and ensure effective tracking. In the 'Soccer', when the target occlusion occurred, our algorithm can still achieve better tracking results compared with the similarity algorithm and the category algorithm.

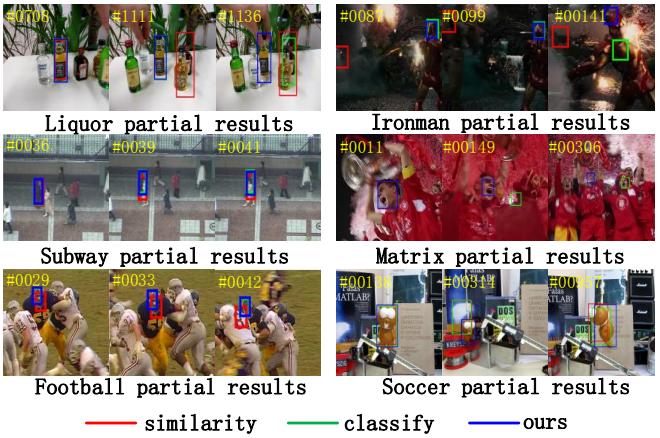

Figure 5: Diagram of the tracking results. The red box is results of the similarity algorithm, the green box is the results of the category algorithm, the blue box is the results of our algorithm. Our algorithm is better than the other two algorithms.

'Football' and 'Ironman' are sequences with similar targets and the complicated background. Complex backgrounds and interference from similar targets in video make target tracking difficult. However, our algorithm improves the model's ability to judge the target by using the similarity and the category. It is easy to see that our algorithm can achieve better tracking than the similarity algorithm.

2) Quantitative analysis

In order to prove the superiority of our algorithm, we select some of the data in the ILSVRC database to train the network and experiment in OTB100 with the network trained by different constraints. λ is the weight of different constraints. When λ is 0, networks are trained using only similarity of the samples. If λ is ∞, networks are trained using only category of the samples.

Figure 6 shows the results under one-pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) using the distance precision rate and overlap success rate. As is show of the experiment results,

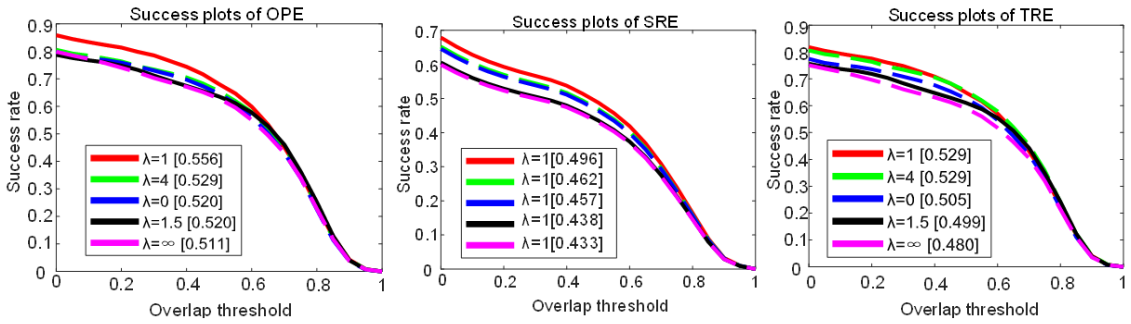

Figure 6: Comparison of success rates for different parameters. Distance precision and overlap success plots over 50 sequences using one-pass evaluation (OPE), temporal robustness evaluation (SRE) and spatial robustness evaluation (TRE). The legend of distance precision contains threshold scores at 20 pixels while the legend of overlap success contains area under-the-curve score for each tracker

both the similarity algorithm and the category algorithm can't achieve very good results. However, when λ is 1, we get the best tracking results. As is show of the results, when we train network using the similarity and the category simultaneously, the track effect has greatly improved.

C OTB50 database evaluation

The OTB 50 database contains a total of 50 sequences, these video sequences involve 11 attribute changes such as scale changes, deformations and occlusions. We compare our tracker with 7 state-of-the-art methods: C-COT [10], HDT [11], FCNT [6], SINT [5], RPT [1], Siamese-FC [4], DSST [12], and CNT [7] As is shown in Figure 7, the precision of our algorithm is 89.2%, the success rate is 80.7%, that is better than similarity algorithms such as Siamese-fc[4] and SINT[5]. The results prove that it can improve the characterization ability of the target that we add the category based on the similarity. Compared with the latest tracking algorithm, the tracking performance of our algorithm is the second, only lower than the C-COT[10]. However, the tracking speed of our algorithm is higher than C-COT[10], this indicates that our algorithm has better tracking performance. The speed of each algorithm is showed in Table 2.

Algorithm	*C-COT [10]*	*SINT [5]*	*Siamese -fc [4]*	*FCNT [6]*	*HDT [11]*	*RPT[1]*	*DSST [12]*	*CNT [7]*	*Ours*
Time(fps)	3	4	58	1	0.14	0.13	0.01	1.5	35

Table 2: The speed of each algorithm. The algorithm in this paper is faster than SINT, and only slower than Siamese-fc.

D UAV123 database

The UAV123 database is a relatively large tracking database of the UAV target tracking, which contains 123 sequences and has 12 attribute changes. Compared to the OTB 50 database, the UAV 123 database mainly deals with the UAV's shooting of target at low altitude. In addition, UAV target's the angle and height are inconsistencies, which will cause target deformation and scale changes. Therefore, the UAV 123 database is even more challenging than the OTB 50

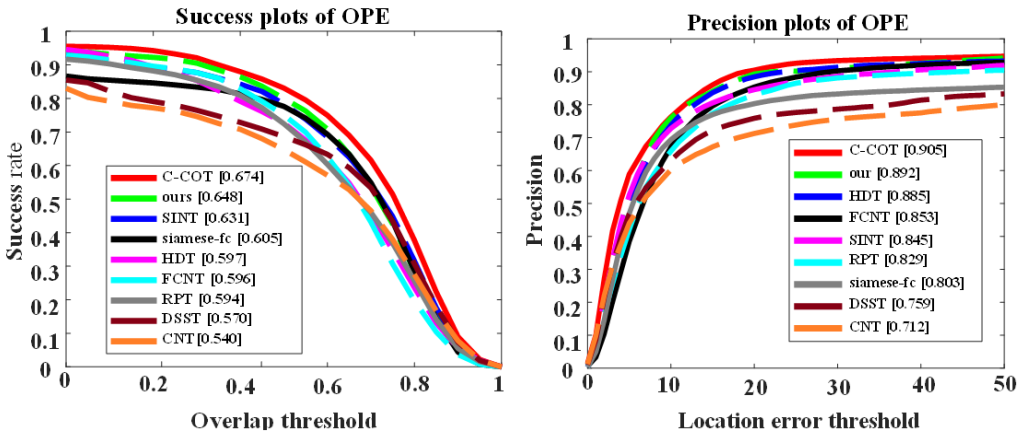

Figure 7: The precision and the success rate in the OTB50. Our algorithm achieves the state-of-art performance with a precision score of 0.892 and a success score of 0.807, which is second only to C-COT.

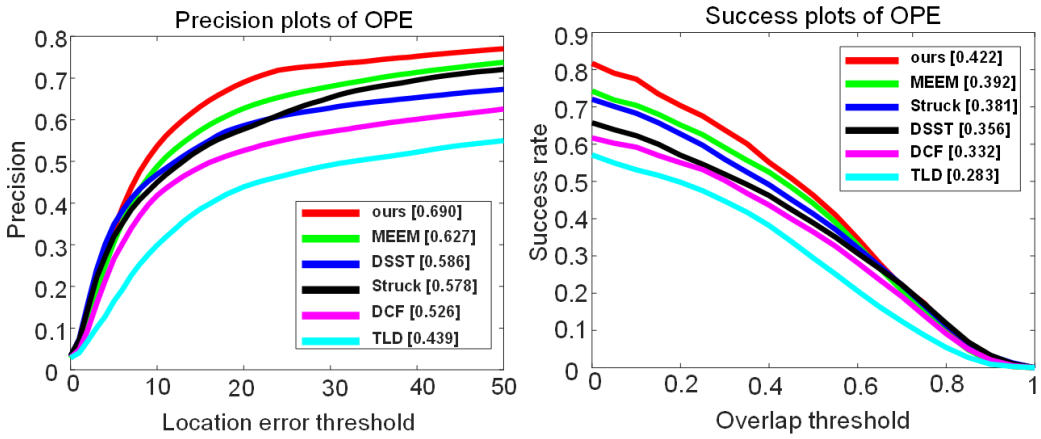

Figure 8: The precision and the success in the UAV123. Our algorithm has achieved the best tracking results.

database. We compare our tracker with 5 state-of-the-art methods: MEEM [13], DSST [12], Struck [14], DCF [15], and TLD [16].

The UAV 123 database is a large tracking database and contains a variety of scene changes. As is showed in the Figure 8, the precision and the success rates in the UAV123database were low compared with the OTB20 database, this shows that tracking in the UAV database is even more challenging. In theUAV123 database, the precision of our algorithm is 69%, the success rate is 46.3%, and our algorithm achieves the best tracking results.

Ⅴ Conclusion

The similarity and category of the target is complementary to each other, which is explored to establish joint identification-verification model. Specially, the learned discriminative features achieve a better characterization of the target. On the one hand, the similarity can reduce the intra-class distance; on the other hand, the category can distinguish different targets effectively. As a result, the proposed tracker can learn the intra-class similarity and the inter-class separability simultaneously. As is showed of the experiments, the precision and the success rate of our tracker surpasses most of the contrast tracker in OTB2013 and UAV123 databases and achieves the accurate tracking of the target.References.

VI Acknowledgments

This research has been supported by National Natural Science Foundation of China (No.61701524, No.61773397, No.61472442, No.61473309, No.61703423).

[1] Li Y, Zhu J, Hoi S C H. Reliable Patch Trackers: Robust visual tracking by exploiting reliable patches[C]// Computer Vision and Pattern Recognition. IEEE, 2015:353-361.

[2] Zhang K, Liu Q, Wu Y, et al. Robust Visual Tracking via Convolutional Networks Without Training[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2016, 25(4):1779-1792.

[3] Jack Valmadre, Luca Bertinetto, Joao Henriques, et al. End-to-End Representation Learning for Correlation Filter Based Tracking[C] IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2017:5000-5008.

[4] Luca Bertinetto, Jack Valmadre, João F. Henriques, et al. Fully-Convolutional Siamese Networks for Object Tracking [J]. 2016:850-865.

[5] Chen K, Tao W. Once for All: a Two-flow Convolutional Neural Network for Visual Tracking[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2016, PP(99):1-1.

[6] Tao R, Gavves E, Smeulders A W M. Siamese Instance Search for Tracking[C] Computer Vision and Pattern Recognition. IEEE, 2016:1420-1429.

[7] Wang L, Ouyang W, Wang X, et al. Visual Tracking with Fully Convolutional Networks[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:3119-3127.

[8] Huang D. Enable Scale and Aspect Ratio Adaptability in Visual Tracking with Detection Proposals[C] British Machine Vision Conference. 2015.

[9] Dash T, Nayak T, Chattopadhyay S. Handwritten Signature Verification (Offline) using Neural Network Approaches: A Comparative Study[J]. International Journal of Computer Applications, 2012, 57(7):33-41.

[10] Uchimura K, Hu Z. Face Recognition Based on Dominant Frequency Features and Multiresolution Metric[C]// International Conference on Innovative Computing, Information and Control. IEEE, 2007:9-9.

[11] Danelljan M, Robinson A, Khan F S, et al. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking[C]// European Conference on Computer Vision. Springer, Cham, 2016:472-488.

[12] Qi Y, Zhang S, Qin L, et al. Hedged Deep Tracking[C]// Computer Vision and Pattern Recognition. IEEE, 2016:4303-4311.

[13] Danelljan M, Häger G, Khan F S, et al. Accurate Scale Estimation for Robust Visual Tracking[C]// British Machine Vision Conference. 2014:65.1-65.11.

[14] Zhang J, Ma S, Sclaroff S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization[C]// European Conference on Computer Vision. Springer, Cham, 2014:188-203.

[15] Hare S, Saffari A, Torr P H S. Struck: Structured output tracking with kernels[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 38(10):2096-2109.

[16] Henriques J F, Rui C, Martins P, et al. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(3):583-596.

[17] Kalal Z, Mikolajczyk K, Matas J. Tracking-Learning-Detection.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2012, 34(7):1409.