DEEP TRACKING WITH OBJECTNESS

Xinyu Wang[1], *Hanxi Li*[1*], *Yi Li*[2], *Fatih Porikli*[3], *Mingwen Wang*[1]

Jiangxi Normal University, China[1]
Toyota Research Institute of North America, USA[2]
Australian National University, Australia[3]

ABSTRACT

Visual tracking is a fundamental problem in computer vision. However, due to the (sometimes) ambiguous target information given at the first frame, it has also been criticized as less well-posed compared with other tasks with clearly-defined targets, such as object detection and semantic segmentation. In this paper, we try to evaluate the importance of object category in visual tracking by tracking objects with known object types. The proposed algorithm, termed Deep-Track with Objectness (DTO), naturally combines the state-of-the-art deep-learning-based detectors and trackers, which essentially share a large part of the network. In DTO, a deep tracker, which is scale-fixed and sensitive to small translations tracks the object in a relative short lifespan. A deep detector, which is scale-changeable and robust to pose or illumination changes guides the deep tracker in a longer lifespan. As the deep tracker and detector share the main part of their networks, no much extra computation is imposed while the performance gain is significant. We test the proposed algorithm on two well-accepted benchmarks and on both of them, the proposed method increases the tracking accuracies remarkably compared with state-of-the-art visual trackers.

Index Terms— visual tracking, deep learning, object detection

1. INTRODUCTION

Visual tracking is one of the long standing computer vision tasks. During the last decade, as the surge of deep learning, more and more tracking algorithms benefit from deep neural networks, e.g. Convolutional Neural Networks [1, 2] and Recurrent Neural Networks [3, 4]. Despite the commonly-admitted success, visual tracking is still criticized as less well-posed compared with other tasks with clearly-defined targets, such as object detection and semantic segmentation. In visual tracking, the only reliable target information is given at the first frame while the information could be ambiguous or misleading in many circumstances. For example, in Figure 1, a car is to be tracked in the sequence. From the viewing angle at the first frame, only the car back can be observed so it is defined as the "target" by the blue bounding box. Nonetheless, this simple target definition usually leads to an ambiguity: when the target pose changes significantly, it is hard to evaluate tracking results. In specific, as shown in Figure 1, either the yellow box or the blue box can be considered as a "perfect" tracking, depending on what exactly the tracking target is, the car back or the whole car.

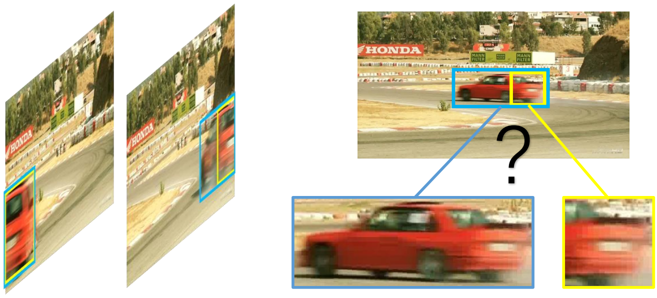

Fig. 1. The commonly existing ambiguity in visual tracking. From left to right, the car back is labeled as the tracking target at the first frame, as the viewing angle changes, the car back and the visible part of the car become more and more different. Finally, when the pose changes significantly, as shown in the right column, it is hard to judge which bounding box (among blue and yellow ones) is the better tracking result.

Unfortunately, a clearly-defined tracking target is usually absent in visual tracking due to the very limited information (a bounding box) given at the first frame. However, by examining the famous tracking datasets [5, 6] further, we can find an underlying assumption of defining tracking target: *the tracking target is usually defined as a whole object, rather than a side of it.*

In this work, we try to partially address the ill-posed problem by imposing the objectness in visual tracking tasks. In other words, the tracker tracks the object given the target's bounding box at the first frame as well as the category of the target. There already exists some visual tracking methods employing the objectness for higher tracking performance. For instance, [1] and [7] learn the object features in conventional deep-learning style and then the network is updated in the specific video sequence; [8] designs a heuristic object proposal algorithm for eliminating the non-object tracking candidates. While these methods mainly focusing on the generic objectness, we pay more attention to some specific object categories. The proposed method, termed DeepTrack with Objectness (DTO), is designed based on the HCF [9] tracking algorithm and assumes that the tracker is aware of the object category of the tracking target. This leads to a significant boost in tracking performance over the ordinary HCF tracker in two well-adopted tracking benchmarks, as we show in the experiment part.

2. DEEPTRACK WITH OBJECTNESS

2.1. Tracking with two types of results

The proposed DTO algorithm is built on the HCF tracker [9]. Besides the target bounding-box, the object category is also given at the first frame. This assumption is similar to the original DeepTrack algorithm [10] while we exploit the object information in a more natural way.

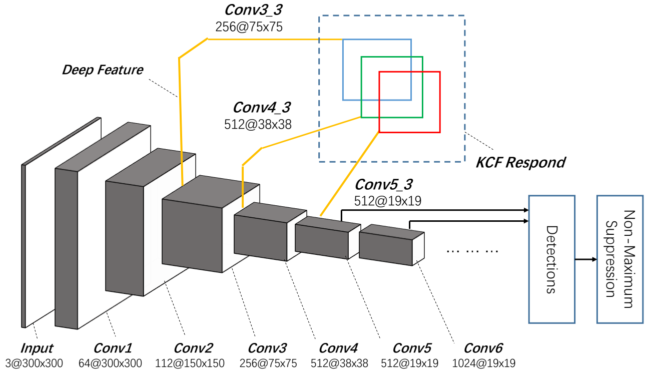

Fig. 2. The CNN structure of the proposed DTO tracker. It is built on a VGG-16 network and the features extracted from $conv3_3$, $conv4_3$ and $conv5_3$ are used for KCF tracking, the features from other 6 layers are used by SSD [11] for regressing objects with different scales.

Recall that in HCF, a VGG-19 CNN model [12] is used for extracting deep features for visual tracking. And the VGG-19 network is learned for image classification on the ILSVRC dataset [6]. In this work, instead of using the image classification network, a CNN model for object detection is used for extracting deep features. When tracking, the KCF trackers are performed on the deep features and the tracking results are inferred following the strategy of HCF. Meanwhile, the detection results (only for the current object category) are also obtained based on the deep detector. In specific, the original VGG-19 network is replaced by the CNN model of Single Shot MultiBox Detector (SSD), a state-of-the-art detection algorithm [11].

The CNN structure of the proposed DTO method is shown in Figure 2. One can see that this CNN model is essentially VGG-16 expect that some auxiliary CNN branches are added for regressing the object bounding box in different scales [11]. Note that VGG-16 is less complex than VGG-19, thus even with the extra detection layers, the proposed DTO is only slightly slower than its prototype, the HCF tracker.

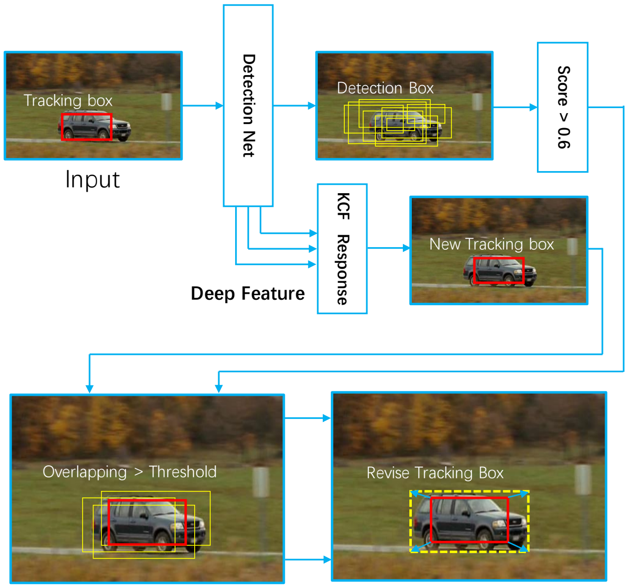

Fig. 3. The flowchart of the detection-guided tracking process. Top: the tracking box (shown in red) is obtained following the same strategy as HCF. Meanwhile, some detection bounding boxes are also generated by SSD [11]. Bottom: after removing the unqualified detection bounding boxes, the average scale and aspect ratio of the detections are used to correct the current tracking box. Better view in color.

2.2. A simple yet effective guidance from detector

Given the tracking bounding-box and detection bounding-boxes, DTO merges the results in a simple yet effective way. Figure 3 demonstrates the merging process. Specifically, let us assume the tracking bounding-box (red bounding-box obtained in the same way as the ordinary HCF tracker) is represented as a 4-D vector $\mathbf{B}_t = [x_t, y_t, w_t, h_t] \in \mathbb{R}^{4\times 1}$ where x_t, y_t, w_t and h_t are the x-axis coordinate of the box center, the y-axis coordinate of the box center, the width and the height of the tracking box, respectively. The SSD detector [11] generates multiple detection bonding-boxes stored in the set $\mathbb{B}_d = \{\mathbf{B}_d^1, \mathbf{B}_d^2, \dots \mathbf{B}_d^N\}$ with the SSD scores $\{s_d^1, s_d^2, \dots, s_d^N\}$. As shown in Figure 3, we firstly remove some unqualified detection boxes that are far away from the tracking box or with low scores. Normally, the qualified detection box set is selected as

$$\mathbb{B}'_d = \{\forall \mathbf{B}_d^i \mid \mathrm{IoU}(\mathbf{B}_d^i, \mathbf{B}_t) > 0.5 \ \& \ s_d^i > 0.6\} \qquad (1)$$

We use $a_t = \sqrt{w_t \cdot h_t}$ and $r_t = w_t/h_t$ to represent the scale and aspect ratio of the tracking box. Suppose the number of qualified detection boxes is N_q, we calculate the average scale and aspect ratio for the qualified detection boxes as

$$\bar{a}_d = \tfrac{1}{N_q} \sum_{\mathbf{B}_d^i \in \mathbb{B}'_d} a_d^i \qquad (2)$$

$$\bar{r}_d = \tfrac{1}{N_q} \sum_{\mathbf{B}_d^i \in \mathbb{B}'_d} r_d^i \qquad (3)$$

Then the scale and aspect ratio of the final prediction, *i.e.*, a_t^* and r_t^* are given by

$$\begin{aligned} a_t^* = \ & \left(1 - \tfrac{1}{1+\exp(-\lambda(s_d^* - s_0))}\right) \cdot a_t + \qquad (4) \\ & \tfrac{1}{1+\exp(-\lambda(s_d^* - s_0))} \cdot \bar{a}_d \\ r_t^* = \ & \left(1 - \tfrac{1}{1+\exp(-\lambda(s_d^* - s_0))}\right) \cdot r_t + \qquad (5) \\ & \tfrac{1}{1+\exp(-\lambda(s_d^* - s_0))} \cdot \bar{r}_d \end{aligned}$$

where $s_d^* = \max([s_d^1, s_d^2, \ldots, s_d^{N_q}])$, *i.e.*, the max scores over the qualified detection boxes. The hyper-parameters λ and s_0 are set to 10 and 0.6 in practice.

Finally, the predicted bounding-box of DTO writes

$$\mathbf{B}_t^* = \left[x_t, y_t, \frac{w_t \cdot a_t^*}{a_t}, \frac{w_t \cdot a_t^*}{a_t \cdot r_t^*}\right]. \qquad (6)$$

From 6 and Figure 3 one can see the original HCF tracking box is corrected by the detection boxes. We found the correction is usually beneficial thanks to the more clear definition of the target category and the well-learned detector.

3. EXPERIMENT

In this section, we report the results of a series of experiment involving the proposed tracker and some state-of-the-art approaches. Our DTO tracker is compared with some well-performing shallow visual trackers including the Struck [13], MIL [14], TLD [15] CT [16] and CSK [17]. Also, some recently proposed deep trackers including MD-net [18], HCF [9], GOTURN [19] and the Siamese tracker [20] are also compared. All the experiment is implemented in MATLAB with matcaffe [21] deep learning interface, on a computer equipped with a Intel i7 4770K CPU, a NVIDIA GTX1070 graphic card and 32G RAM.

Recall that one condition of using DTO is that it can only track the object that the detector recognizes while the ordinary SSD is learned for predicting 20-class objects in VOC dataset [22]. In visual tracking datasets, on the other hand, the most common categories include cars, pedestrians and human faces [5, 6]. However, there is no face category in VOC and the “person” category in VOC is defined very differently from the concept “pedestrian” in visual tracking[1]. We thus only test the involved trackers on the car-subset of the original tracking benchmarks. We claim this is sufficient for illustrating the importance of target category in visual tracking. A DTO-like face tracker can also be built based on a well-learned face detector while this is out of the scope of this paper.

3.1. Results on OTB-100-Car

Similar to its prototype [23], the Object Tracking Benchmark 100 (OTB-100) [5] consists 100 video sequences and involves 51 tracking tasks. It is one of the most popular tracking benchmarks since the year 2013, The evaluation is based on two metrics: center location error and bounding box overlap ratio. To evaluate the proposed method, we select all the video sequences targeting on cars from OTB-100, the totally 12 video sequences contain almost all of the tracking challenges such as scale variation, illumination variation, occlusion and motion blur. The one-pass evaluation (OPE) is employed to compare our algorithm with the HCF [9], GOTURN [19], the Siamese tracker [20] and the afore mentioned shallow trackers. The result curves are shown in Figure 4

According to Figure 4 we can see that the proposed DTO ranks the first on location accuracy while ranks the second with the overlapping metric. It achieves significantly better results than its prototype, *i.e.*, the HCF tracker. The performance DTO is also very close to the best-performing MD-Net. Siamese tracker is also comparable to DTO and MD-Net while GOTURN performs worse than the other deep trackers. On the other hand, the shallow methods perform consistently worse than the DTO, MD-Net and the Siamese tracker.

3.2. Results on ILSVRC2016-VID-Car

The ILSVRC(Large Scale Visual Recognition Challenge) [6] is one of the largest visual recognition datasets. The object detection from video is a new detection task in recent years, and there are 30 basic-level categories for this task, which is a subset of the 200 basic-level categories of the object detection task. We selected 58 videos that contains car from this dataset and compare the location accuracy and overlap score over the selected deep trackers [2]. As Siamese tracker and the GOTURN algorithm learned their models in this dataset, we remove them from comparison. The success plots and the precision plots are shown in Figure 5

From the figure we can see that DTO still achieves comparable accuracies as the best-performing MD-Net algorithm. The remarkable gap between the DTO plots and the HCF plots proofs the validity of the introduction of the object categories.

[1] The former one includes any part of a person while the latter one usually stands for the whole human body

[2] We do not involve shallow trackers in this experiment as they usually perform worse than the deep ones and the results of the shallow trackers are not directly available

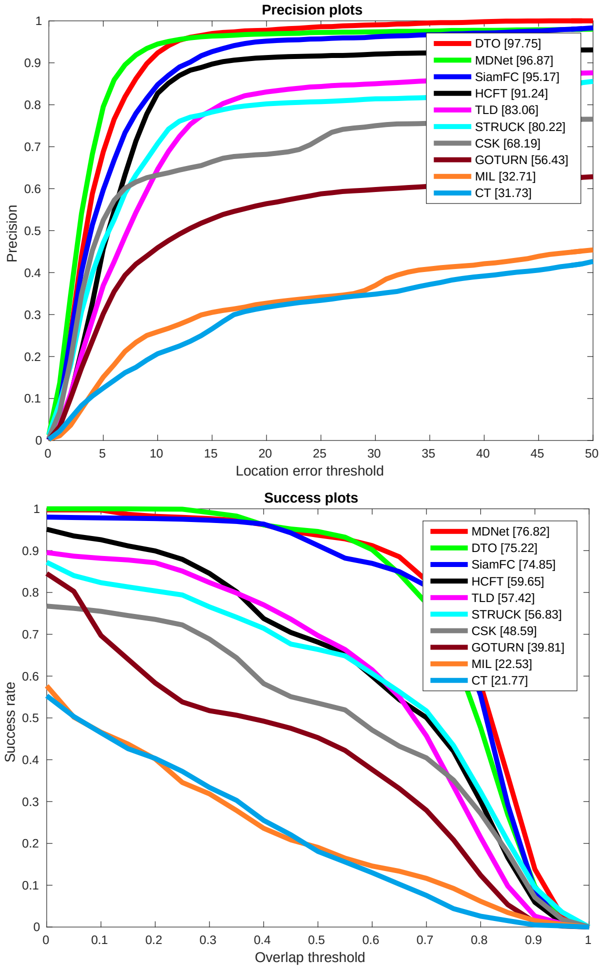

Fig. 4. The location error plots and the overlapping accuracy plots tested on the "car subset" of OTB-100. The comparing methods including MD-Net [18], HCF [9], the Siamese Tracker [20], GOTURN [19], DTO (this paper) and the shallow trackers.

4. CONCLUSION

In this paper, we propose a very simple yet effective way to guide the visual tracking by the detection results. The proposed DTO tracker can be considered as a fusion of the state-of-the-art deep tracker and deep detector. As they share most part of the network structure, no much extra computation is required. On the other hand, we can see a dramatic performance improvement in DTO, compared with its prototype, the HCF tracker. This improvement implies the absence of the target object could lead to poor tracking performance while to address this absence in a more sophisticated way could yield much better deep trackers in the future.

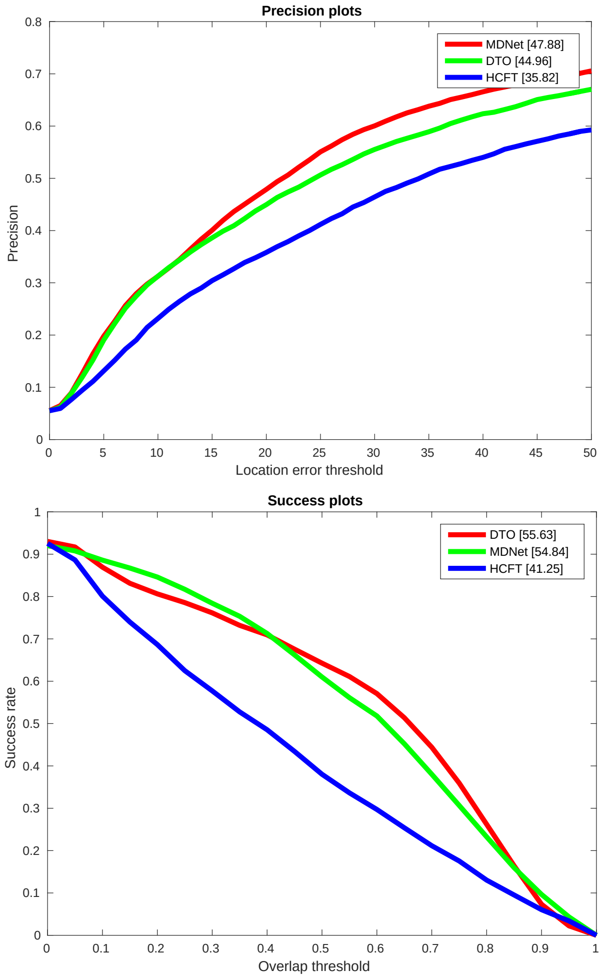

Fig. 5. The location error plots and the overlapping accuracy plots tested on the "car subset" of the ILSVRC2016-VID dataset. The comparing methods including HCF [9], DTO (this paper), MD-Net [18] and the shallow ones. We do not involve the Siamese Tracker [20] and GOTURN [19] as they are trained on this dataset.

5. REFERENCES

[1] Naiyan Wang and Dit-Yan Yeung, "Learning a deep compact image representation for visual tracking," in *NIPS*, pp. 809–817. 2013.

[2] Hanxi Li, Yi Li, and Fatih Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 4, pp. 1834–1848, 2016.

[3] Anton Milan, Seyed Hamid Rezatofighi, Anthony Dick, Konrad Schindler, and Ian Reid, "Online multi-target tracking using recurrent neural networks," *arXiv preprint arXiv:1604.03635*, 2016.

[4] Guanghan Ning, Zhi Zhang, Chen Huang, Zhihai He, Xiaobo Ren, and Haohong Wang, “Spatially supervised recurrent convolutional neural networks for visual object tracking,” *arXiv preprint arXiv:1607.05781*, 2016.

[5] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Object tracking benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 9, pp. 1834–1848, 2015.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[7] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” in *ICML*, 2015, pp. 597–606.

[8] Gao Zhu, Fatih Porikli, and Hongdong Li, “Beyond local search: Tracking objects everywhere with instance-specific proposals,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 943–951.

[9] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, “Hierarchical convolutional features for visual tracking,” in *ICCV*, 2015, pp. 3074–3082.

[10] Hanxi Li, Yi Li, and Fatih Porikli, “Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking,” *BMVC*, 2014.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott Reed, “Ssd: Single shot multibox detector,” *arXiv preprint arXiv:1512.02325*, 2015.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[13] Sam Hare, Amir Saffari, and Philip HS Torr, “Struck: Structured output tracking with kernels,” in *ICCV*, 2011, pp. 263–270.

[14] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, “Visual tracking with online multiple instance learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, pp. 1619–1632, 2011.

[15] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk, “Pn learning: Bootstrapping binary classifiers by structural constraints,” in *CVPR*, 2010, pp. 49–56.

[16] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang, “Real-time compressive tracking,” in *European Conference on Computer Vision*. Springer, 2012, pp. 864–877.

[17] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *European conference on computer vision*. Springer, 2012, pp. 702–715.

[18] Hyeonseob Nam and Bohyung Han, “Learning multi-domain convolutional neural networks for visual tracking,” *arXiv preprint arXiv:1510.07945*, 2015.

[19] David Held, Sebastian Thrun, and Silvio Savarese, “Learning to track at 100 fps with deep regression networks,” *arXiv preprint arXiv:1604.01802*, 2016.

[20] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr, “Fully-convolutional siamese networks for object tracking,” in *ECCV*, 2016, pp. 850–865.

[21] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *ACM MM*, 2014, pp. 675–678.

[22] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, “The pascal visual object classes (voc) challenge,” *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[23] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Online object tracking: A benchmark,” in *CVPR*, 2013, pp. 2411–2418.