Object-Adaptive LSTM Network for Visual Tracking

Yihan Du[1], Yan Yan[1*], Si Chen[2], Yang Hua[3], Hanzi Wang[1]
[1]School of Information Science and Engineering, Xiamen University, China
[2]School of Computer and Information Engineering, Xiamen University of Technology, China
[3]School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK
Email: yihandu@stu.xmu.edu.cn, {yanyan, hanzi.wang}@xmu.edu.cn, chensi@xmut.edu.cn, y.hua@qub.ac.uk

***Abstract*—Convolutional Neural Networks (CNNs) have shown outstanding performance in visual object tracking. However, most of classification-based tracking methods using CNNs are time-consuming due to expensive computation of complex online fine-tuning and massive feature extractions. Besides, these methods suffer from the problem of over-fitting since the training and testing stages of CNN models are based on the videos from the same domain. Recently, matching-based tracking methods (such as Siamese networks) have shown remarkable speed superiority, while they cannot well address target appearance variations and complex scenes for inherent lack of online adaptability and background information. In this paper, we propose a novel object-adaptive LSTM network, which can effectively exploit sequence dependencies and dynamically adapt to the temporal object variations via constructing an intrinsic model for object appearance and motion. In addition, we develop an efficient strategy for proposal selection, where the densely sampled proposals are firstly pre-evaluated using the fast matching-based method and then the well-selected high-quality proposals are fed to the sequence-specific learning LSTM network. This strategy enables our method to adaptively track an arbitrary object and operate faster than conventional CNN-based classification tracking methods. To the best of our knowledge, this is the first work to apply an LSTM network for classification in visual object tracking. Experimental results on OTB and TC-128 benchmarks show that the proposed method achieves state-of-the-art performance, which exhibits great potentials of recurrent structures for visual object tracking.**

I. Introduction

Visual object tracking is one of the most important and challenging problems in computer vision, which has a variety of applications including video surveillance, driverless cars, *etc.* Only given the annotation in the first frame of a video, tracking algorithms operate to locate the object in the subsequent frames, probably confronting various appearance and motion changes caused by deformation, rotation, background clutter and so forth.

In recent years, Convolutional Neural Networks (CNNs) have been widely used in visual object tracking [1, 2, 3, 4]. These CNN-based tracking methods can be roughly divided into two categories: classification-based tracking methods and matching-based tracking methods. Classification-based tracking methods [1, 2, 5] treat tracking as a binary classification problem, where the classifiers are trained to distinguish the object from the background. These methods achieve excellent performance at the cost of high computational complexity due to massive proposal evaluation and sophisticated online fine-tuning. Besides, some high-accuracy trackers (*e.g.*, MDNet [1] and SANet [5]) use videos from the same domain or two intersecting datasets (*e.g.*, OTB [6] and VOT [7]) to train and test their models, which leads to the problem of over-fitting.

Matching-based tracking methods [3, 4] match the candidate patches with the target template and do not involve any updating procedures. Thus, they can operate at real-time speeds [3, 4]. However, the lack of online adaptability and background information causes these methods prone to drift or failure, especially when the target appearance significantly changes in some challenging scenes.

Most of existing CNN-based trackers [1, 2] perform object detection in each frame independently and thus they ignore the temporal dependencies among successive frames in a video sequence. Recently, Recurrent Neural Networks (RNNs) have drawn extensive attention in computer vision [8, 9] owing to their powerful capability of modeling sequential data and capturing time dependencies. However, few RNN-based object tracking works have demonstrated state-of-the-art performance on the canonical benchmarks. We presume that this is mainly due to the absence of an effective recurrent network well suited for the tracking task.

Motivated by the above analysis, in this paper we propose a novel object-adaptive LSTM network (OA-LSTM) for visual object tracking. Different from conventional deep models, we advocate the LSTM (Long Short-Term Memory) recurrence to characterize long-range sequential information, while maintaining an intrinsic object representation model (via memorizing target appearance variations and ignoring confusing distractors). Due to its intrinsic recurrent structure, our network is able to dynamically update the internal state during forward passes. Moreover, to lighten the computational burden and enable our network to track an arbitrary object, we adopt an efficient strategy for proposal selection. Specifically, the densely sampled proposals are firstly pre-evaluated using the fast matching-based method and then the well-selected high-quality proposals are classified based on the online learning LSTM network. Possessing the above properties, our method can effectively track an arbitrary object with adaptability to target appearance variations and sequence-specific circumstances. Fig. 1 illustrates the pipeline of the

*Corresponding author.

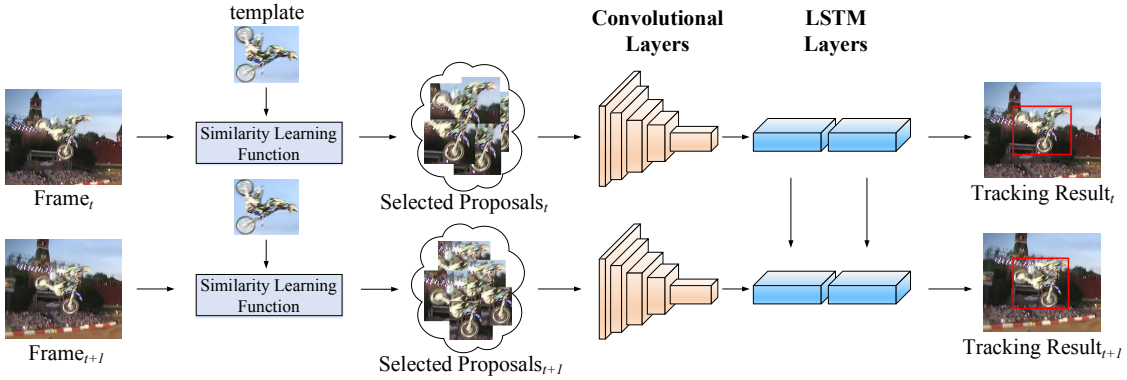


Fig. 1. Pipeline of the proposed method for visual object tracking. The arrows from the LSTM layers of frame$_t$ to those of frame$_{t+1}$ denote the forward propagation in time of memory information.

proposed method for visual object tracking.

In this paper, we make an important step towards the promising application of recurrent structure for visual object tracking. Our main contributions are summarized as follows:

- We propose a novel object-adaptive LSTM network (OA-LSTM), which not only effectively exploits long-range temporal dependencies in a video sequence, but also dynamically adapts to the object appearance variations via constructing an intrinsic model for object appearance and motion during online tracking. To the best of our knowledge, this is the first work to apply an LSTM network for classification in visual object tracking with both state-of-the-art performance and a moderate speed at a practical level.
- We present an efficient strategy for proposal selection. Specifically, the densely sampled proposals are firstly pre-evaluated using the fast matching-based method and then the well-selected high-quality proposals are provided to the sequence-specific learning LSTM network. This strategy enables our method to track an arbitrary object and operate faster than existing CNN-based classification tracking methods.
- Our method achieves state-of-the-art performance on public tracking benchmarks (OTB [6] and TC-128 [10]), at speeds that exceed existing representative CNN-based classification tracking methods, which demonstrates that the proposed recurrent structure is well suited for visual object tracking.

II. Related Work

Visual object tracking has been actively studied for decades. In the following, we mainly discuss three types of deep learning based tracking methods related to ours.

CNN-based tracking-by-detection: CNN-based tracking-by-detection methods train convolutional classification networks to distinguish the object from the background. MDNet [1], the winner of the VOT2015 challenge [7], firstly trains the CNN with respect to each video offline and then learns a per-object classifier online. Although this method has achieved outstanding accuracy, massive feature extractions and sophisticated online fine-tuning techniques limit its efficiency. To deal with this problem, we leverage the fast matching-based method to select high-quality proposals so that the heavy computational burden for irrelevant proposal generation and evaluation is avoided. Besides, we adopt simple online updating techniques for computational efficiency (since the LSTM network can dynamically update the recurrent parameters during forward passes). Furthermore, different from MDNet [1] which performs training and testing on two intersecting datasets (*i.e.*, VOT [7] and OTB [6]), we learn an object-adaptive LSTM network online to track the object from arbitrary video domains.

Siamese network based tracking: Siamese network based methods [3, 4] learn generic matching models offline based on image pairs. For example, SiamFC [3] adopts a novel fully-convolutional Siamese network and it can operate in real-time. However, it neither utilizes background information nor captures target appearance variations during online tracking. DSiam [11] uses transformation learning to provide online adaptation ability for Siamese network, exhibiting state-of-the-art performance. In contrast to the above Siamese network based methods, we propose an object-adaptive LSTM network for classification, which effectively utilizes background information and highly adapts to the changes of target appearance.

Recurrent network based tracking: Milan *et al.* [12] present an RNN-based method for multi-target tracking, where the RNN is used to predict target motion and determine target initiation and termination. Gan *et al.* [8] and Kahou *et al.* [9] apply attention-based recurrent neural networks to object tracking. However, these methods have only been shown to work on simple datasets (such as MNIST digits) instead of natural videos. SANet [5] fuses CNN and RNN feature maps to model the object structure, but the heavy computational burden constrains its speed ($<$ 1 fps). Gordon *et al.* [13] propose real-time recurrent regression networks (Re3) to learn the changes of target appearance and motion offline. Despite its speed superiority, Re3 is prone to drift in complex scenes for lack of online adaptability. In contrast, we propose to adopt an object-adaptive LSTM network, which exploits sequence-specific information more sufficiently and possesses better object adaptability.

III. LSTM Network for Object-Adaptive Modeling

RNNs have shown the powerfulness in handling sequential data owing to the capability of storing the representations of recent inputs using feedback connections. As an alternative RNN, the LSTM recurrence can not only properly capture long-range temporal dependencies, but also ignore distracting information. Therefore, we take advantage of LSTM for visual tracking in this paper.

During online tracking, new target positions need to be located successively when new frames come one after another. To fit in with the tracking task, we use the LSTM network to evaluate the proposals and identify the optimal target state in each frame. More specifically, we calculate the forward pass of the LSTM network with Eqs. (1) to (9). The superscript t represents the frame index. The subscripts ι, ν and ω respectively refer to the input gate, forget gate and output gate of the LSTM memory block. The subscript c refers to the memory cells. U, V, and W are the weight matrices for the input, recurrent, and peephole connections, respectively. b is the bias vector. ϕ and σ are the non-linear activation functions. $\odot$ is point-wise multiplication. x^t and h^{t-1} are the input and previous output vectors, respectively. The output vector h^t is used to classify the proposal. The cell state c holds memory information of the object appearance and motion. Both h^t and c^t are produced by the forward pass in the t^{th} frame and fed to the following forward pass in the next frame, which allows the object information to propagate forward in time.

Input Gates

$$z_\iota^t = U_\iota x^t + V_\iota h^{t-1} + W_\iota c^{t-1} + b_\iota \tag{1}$$
$$i^t = \sigma(z_\iota^t) \tag{2}$$

Forget Gates

$$z_\nu^t = U_\nu x^t + V_\nu h^{t-1} + W_\nu c^{t-1} + b_\nu \tag{3}$$
$$f^t = \sigma(z_\nu^t) \tag{4}$$

Cells

$$z_c^t = U_c x^t + V_c h^{t-1} + b_c \tag{5}$$
$$c^t = f^t \odot c^{t-1} + i^t \odot \phi(z_c^t) \tag{6}$$

Output Gates

$$z_\omega^t = U_\omega x^t + V_\omega h^{t-1} + W_\omega c^t + b_\omega \tag{7}$$
$$o^t = \sigma(z_\omega^t) \tag{8}$$

Cell Outputs

$$h^t = o^t \odot \phi(c^t) \tag{9}$$

In order to effectively utilize sequence-specific information and obtain per-object adaptability, we train the LSTM network online using training data from the video itself. In this paper, instead of training the network with a sequence of inputs, we use the previous network state and the samples from the current frame to train this network online. In this manner, the loss directly comes from the current classification results, allowing the parameters to be updated according to the hidden state and training data. Based on this training scheme, the recurrent network quickly converges since the loss does not need to propagate through noisy intermediate timesteps.

Specifically, in the first frame, we initialize the network state with the original target appearance. Then, we use this initialized network state and training samples from the first frame to train the LSTM network. In the subsequent frames, we update the network using the previous network state and training samples from the current frames to obtain better online adaptability. The backward pass during training can be calculated with Eqs. (10) to (16). $\mathcal{L}$ is the softmax cross-entropy loss function used for training. ϵ and δ represent the derivatives defined as Eq. (10). k denotes the subsequent layer of the cell outputs. The subscript j refers to the gates of the LSTM memory block, *i.e.*, $j \in \{\iota, \nu, \omega\}$.

$$\epsilon_h^t \stackrel{def}{=} \frac{\partial \mathcal{L}}{\partial h^t} \qquad \epsilon_c^t \stackrel{def}{=} \frac{\partial \mathcal{L}}{\partial c^t} \qquad \delta_j^t \stackrel{def}{=} \frac{\partial \mathcal{L}}{\partial z_j^t} \tag{10}$$

Cell Outputs

$$\epsilon_h^t = \sum_k w_{hk} \delta_k^t \tag{11}$$

Output Gates

$$\delta_\omega^t = \sigma'(z_\omega^t)\ \phi(c^t)\ \epsilon_h^t \tag{12}$$

Cells

$$\epsilon_c^t = W_\omega \delta_\omega^t + o^t\ \phi'(c^t)\ \epsilon_h^t \tag{13}$$
$$\delta_c^t = i^t\ \phi'(z_c^t)\ \epsilon_c^t \tag{14}$$

Forget Gates

$$\delta_\nu^t = \sigma'(z_\nu^t)\ c^{t-1}\ \epsilon_c^t \tag{15}$$

Input Gates

$$\delta_\iota^t = \sigma'(z_\iota^t)\ \phi(z_\iota^t)\ \epsilon_c^t \tag{16}$$

IV. Proposed Tracking Algorithm

A. Overview

Our tracking pipeline is depicted in Fig. 1. A search region centered at the previously estimated target position in the current frame is firstly taken as the input. Then, a similarity learning model is utilized to select high-quality proposals by comparing the target template with the search region, which significantly reduces redundant computation for irrelevant proposals (see Section IV-B). Next, the features of these high-quality proposals extracted from the convolutional layers are fed to the online learning LSTM network (see Section III), together with the previously estimated target state. Finally, the LSTM network is able to identify the optimal target state from these proposals according to the memorized target appearance and learned sequence-specific information (see Section IV-C). Note that the internal state of the LSTM network can be simultaneously updated when a forward pass is performed during online tracking so that sophisticated fine-tuning techniques used in conventional CNN-based tracking methods [1, 5] are

unnecessary. Therefore, we can perform simple and effective online updating to enhance the adaptability of our network.

B. Fast Proposal Selection

In conventional tracking frameworks, massive proposals are generated via dense sampling [14]. However, the majority of these proposals are typically irrelevant and trivial, leading to expensive and redundant computation for proposal evaluation. In order to optimize the efficiency of our tracking method, we propose to leverage the fast matching-based method to pre-evaluate the densely sampled proposals in the search region and the high-quality proposals are selected to feed to the subsequent classification network.

Recently, SiamFC [3] reaches a good trade-off between speed and accuracy, and it also offers an explicit confidence map for our proposal selection purpose. Therefore, inspired by the success of [3], we employ a similarity learning function (as in [3]) to compare the target template from the first frame with the search region centered at the previously estimated target position in each frame. Based on the similarity learning function, we can firstly obtain a confidence map, which corresponds to all the translated sub-regions in the search region. Then, the sub-regions with high confidence scores are selected and translated from the final confidence map to the original frame. Finally, we feed the subsequent LSTM network with these well-selected proposals for high efficiency, which significantly improves the speed of classical tracking-by-detection framework. Note that the proposal selection strategy is mainly used to filter out irrelevant proposals for algorithm acceleration purpose, while the LSTM network is designed to effectively determine the tracked object from all the selected proposals. Both components are tightly coupled to boost the performance of tracking in both speed and accuracy, even in challenging scenes.

C. Tracking with OA-LSTM

In the proposed LSTM network, for the purpose of supplying the network with rich target appearance information, we adopt five convolutional layers pre-trained on the ILSVRC15 [15] dataset to extract high-level target features. These convolutional layers are kept fixed during the online learning stage since they have learned the capability of generic feature extraction. The fully-connected feature vectors can be directly addressed by the subsequent LSTM layers.

Initially, given the annotation of the target in the first frame, we feed the target features to the LSTM network to obtain the original network state which stores the initial target appearance. Then we use training data sampled from the first frame, together with the original network state, to perform the online training procedure, as discussed in Section III. In the subsequent frames, the well-selected proposals via the matching-based method are evaluated by the LSTM network according to the past target appearance, which is modeled into the LSTM memory cells. The network outputs correspond to the positive scores and negative scores of the proposals, denoting their probabilities belonging to the target class and background class, respectively. The proposal with the highest positive score is selected as the tracked result and its appearance is stored by the network, which supplies the latest target appearance information for the next proposal evaluation. During forward passes, the recurrent parameters can be dynamically updated as the target appearance changes. By taking advantage of its intrinsic recurrent structure, the LSTM network is able to capture the temporal dependencies of the video sequence and memorize the changes of target appearance and motion during online tracking.

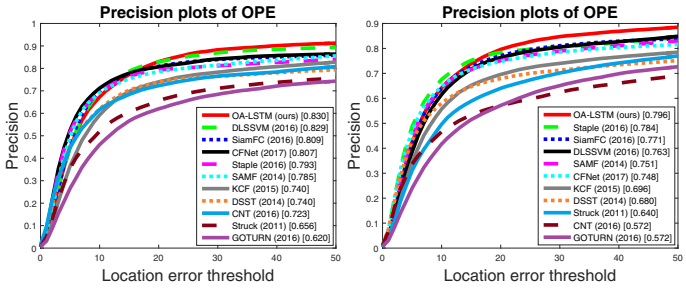
In addition, in order to obtain great adaptability to complex scenes, we perform online updating using an efficient hard negative mining approach. More specifically, we directly use the confidence map from the proposal selection step to select hard negative examples. Therefore, unlike some sophisticated trackers [1, 5] that require numerous iterations to identify the hard negative examples, our hard negative mining approach does not require extra computational cost for example evaluation. As the online updating proceeds, the network efficiently learns the variations of both the target appearance and background with hard negative examples, thus becoming more discriminative and robust. Algorithm 1 summarizes the main steps of the proposed tracking algorithm.

Algorithm 1 The proposed tracking algorithm

Input: Initial target position $\mathbf{x}_1$, similarity learning function $\mathcal{F}$, threshold θ
Output: Estimated target position $\hat{\mathbf{x}}_t$
Initialize the LSTM network using $\mathbf{x}_1$;
Sample training data S_1 from the 1st frame;
Train the LSTM network using S_1;
repeat
  Apply the similarity learning function $\mathcal{F}$ to obtain a confidence map $\mathcal{M}$;
  Select N high-score proposals $\{\mathbf{x}_t^i\}_{i=1}^N$ from $\mathcal{M}$;
  Evaluate $\{\mathbf{x}_t^i\}_{i=1}^N$ with the previous LSTM state $\hat{\mathbf{c}}_{t-1}$ to obtain their positive scores $\{p(\mathbf{x}_t^i)\}_{i=1}^N$;
  Find the tracked result by $\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t^i} p(\mathbf{x}_t^i)$;
  Set the optimal LSTM state $\hat{\mathbf{c}}_t$ corresponding to $\hat{\mathbf{x}}_t$;
  if $p(\hat{\mathbf{x}}_t) > \theta$ **then**
    Sample training data S_t with hard negative mining;
    Update the LSTM network using S_t;
  end if
until end of sequence

V. Experiments

We evaluate the proposed tracking method on two challenging object tracking benchmarks, *i.e.*, OTB [6] and TC-128 [10]. The proposed method is implemented in Python using TensorFlow [16] and runs at an average speed of 11.5 fps with 2.00GHz Intel Xeon E5-2660 and an NVIDIA GTX TITAN X GPU. During implementation, we set the hidden units of both LSTM layers to 2048 and select high-quality proposals from the dense sub-regions at a percentage of 2%. We perform the pre-training process for the convolutional layers on the



(a) OTB-2013 (b) OTB-2015

Fig. 2. Precision plots on the (a) OTB-2013 and (b) OTB-2015 datasets.

ILSVRC15 [15] dataset and learn the per-object LSTM layers online without over-fitting. In addition, we perform all of the experiments with the same parameter settings to guarantee the reliability of our results and conclusions.

A. Evaluation on OTB

OTB (Object Tracking benchmark) [6] is a popular tracking benchmark consisting of 100 fully annotated videos with substantial variations. For the performance evaluation criteria, the success plots heavily penalize trackers that do not track across scale, even if the target position is perfectly tracked. Since our method does not focus on handling scale variations, we employ the precision plots defined in [6] to evaluate how accurate the target position is tracked by the trackers. In addition to the results on the entire videos in [6] (OTB-2015), we also present the results on its earlier version consisting of 50 videos [17] (OTB-2013). We compare our method with ten state-of-the-art trackers including Struck [14], SAMF [18], DSST [19], KCF [20], DLSSVM [21], Staple [22], SiamFC [3], GOTURN [4], CNT [23] and CFNet [24]. The precision plots obtained by all the competing trackers are shown in Fig. 2. As can be seen, our tracker, denoted by OA-LSTM, is superior to the other state-of-the-art competing trackers on both benchmark versions. Among the competing trackers, SiamFC, GOTURN, CNT and CFNet also employ deep networks to learn the object feature representations and achieve good performance. In contrast to their feed-forward neural networks, our recurrent network structure utilizes the temporal dependencies of sequential data and maintains an internal object appearance representation. Based on these properties, our OA-LSTM possesses better online adaptability to target appearance variations. Besides, our tracker obtains better performance than the state-of-the-art matching-based trackers (*i.e.*, SiamFC, CFNet and GOTURN), which clearly validates that the proposed LSTM network can accurately identify the object from the proposals selected using the matching-based method.

B. Evaluation on TC-128

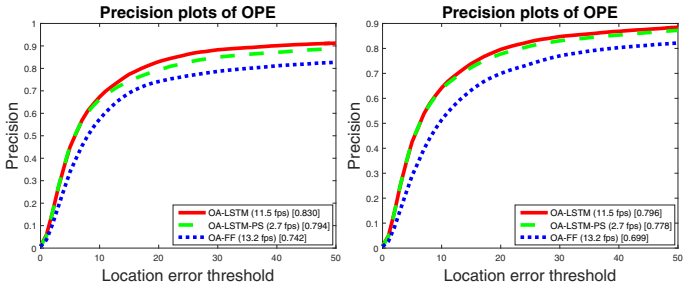
The TC-128 (Temple-Color) [10] dataset contains 128 fully annotated color image sequences. Table I compares the precision scores, the AUC (Area Under the Curve) scores and speeds (fps) of our OA-LSTM and the state-of-the-art trackers: MDNet [1], DNT [25], HDT [26], STCT [2], SRDCF [27], KCF [20], SCM [28], DLT [29] and CNT [23]. Our OA-LSTM achieves the second highest precision and AUC scores and runs the fastest among the top performers. Although OA-LSTM achieves lower accuracy than MDNet, it is 10 times faster than MDNet owing to efficient proposal selection and simple online updating process. In addition, different from MDNet which adopts a biased training strategy, we perform the pre-training on the ILSVRC15 [15] dataset without over-fitting. Our tracker is able to adaptively track an arbitrary object in various scenes. Compared with the fast trackers (*e.g.*, KCF and SRDCF) using hand-drafted features, OA-LSTM is significantly more accurate in both precision and AUC measures. For the recent state-of-the-art trackers (*e.g.*, HDT, DNT and STCT) using deep features, they are slower and less accurate than OA-LSTM. CNT employs a lightweight convolutional neural network to learn object representations and achieves a CPU speed of 5 fps. However, its accuracy still has a big gap to the top deep-based performers. As illustrated in Table I, OA-LSTM achieves outstanding accuracy-speed trade-off among all the state-of-the-art competing trackers.

TABLE I. The precision score, the AUC (Area Under the Curve) score and speed (fps, * indicates GPU speed, otherwise CPU speed) on the TC-128 dataset.

Tracker	Precision	AUC	Speed
MDNet [1]	79.8	58.6	1*
OA-LSTM (ours)	**70.8**	**49.5**	**11.5***
DNT [25]	69.0	48.5	5*
HDT [26]	67.6	47.4	10*
STCT [2]	65.5	48.5	2.5*
SRDCF [27]	64.6	47.8	5
KCF [20]	55.2	38.2	172
SCM [28]	45.3	34.3	<1
DLT [29]	44.0	30.8	15*
CNT [23]	43.7	32.5	5

C. Ablation Study

To verify the effectiveness of each component in our method, we investigate two variants of our method—the feed-forward method without LSTM layers (OA-FF) and OA-LSTM without fast proposal selection (OA-LSTM–PS). Specifically, OA-FF uses two fully-connected layers as the proposal classifier, and OA-LSTM–PS applies the traditional dense sampling strategy. The performance and speed comparison on the OTB dataset is illustrated in Fig. 3. We can see that the performances obtained by both variants are worse than the proposed OA-LSTM method. In other words, each component contributes to improving the performance. Specifically, OA-LSTM achieves significant improvements (the precision scores are increased by 8.8% and 9.7% on the OTB-2013 and OTB-2015 datasets, respectively) over OA-FF, which validates the superiority of the proposed LSTM network. From the perspective of speed, OA-LSTM without fast proposal selection (OA-LSTM–PS) is much slower than OA-LSTM, which demonstrates the efficiency of the proposal selection strategy.



(a) OTB-2013 (b) OTB-2015

Fig. 3. Precision plots showing a comparison of OA-LSTM with two variants on the (a) OTB-2013 and (b) OTB-2015 datasets. The speeds are presented in the legend.

VI. Conclusion

In this paper, we have proposed a novel object-adaptive LSTM network for visual object tracking. By taking advantage of its intrinsic recurrent structure, our network can capture long-range temporal dependencies and dynamically adapt to the object variations via memorizing the changes of object appearance and motion during online tracking. Different from conventional tracking-by-detection frameworks that generate massive proposals via dense sampling, we advocate an efficient strategy for fast proposal selection using the matching-based method, which enables our method to operate faster than conventional CNN-based classification tracking methods. Moreover, with a pre-trained convolutional feature extractor, the LSTM recurrence is learned online to sufficiently utilize sequence-specific information. Thus, our method is able to track an arbitrary object. Experimental results on public tracking benchmarks have shown that our method achieves state-of-the-art performance with a satisfactory speed, demonstrating the successful application of recurrent structure to visual object tracking.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2017YFB1302400, by the National Natural Science Foundation of China under Grants 61571379, 61503315, U1605252, and 61472334, by the Natural Science Foundation of Fujian Province of China under Grants 2017J01127 and 2018J01576, and by the Fundamental Research Funds for the Central Universities under Grant 20720170045.

References

[1] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.
[2] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *CVPR*, 2016.
[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV Workshop*, 2016.
[4] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *ECCV*, 2016.
[5] H. Fan and H. Ling, "SANet: Structure-aware network for visual tracking," in *CVPR Workshop*, 2017.
[6] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *TPAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.
[7] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking VOT2015 challenge results," in *ICCV Workshop*, 2015.
[8] Q. Gan, Q. Guo, Z. Zhang, and K. Cho, "First step toward model-free, anonymous object tracking with recurrent neural networks," *CoRR*, vol. abs/1511.06425, 2015.
[9] S. E. Kahou, V. Michalski, and R. Memisevic, "RATM: recurrent attentive tracking model," *CoRR*, vol. abs/1510.08660, 2015.
[10] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *TIP*, vol. 24, no. 12, pp. 5630–5644, 2015.
[11] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*, 2017.
[12] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks." in *AAAI*, 2017.
[13] D. Gordon, A. Farhadi, and D. Fox, "Re3 : Real-time recurrent regression networks for object tracking," *CoRR*, vol. abs/1705.06368, 2017.
[14] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *ICCV*, 2012.
[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems," *CoRR*, vol. abs/1603.04467, 2016.
[17] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013.
[18] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *ECCV Workshop*, 2014.
[19] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *BMVC*, 2014.
[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.
[21] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *CVPR*, 2016.
[22] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *CVPR*, 2016.
[23] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *TIP*, vol. 25, no. 4, pp. 1779–1792, 2016.
[24] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *CVPR*, 2017.
[25] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking," *TIP*, vol. 26, no. 4, pp. 2005–2015, 2017.
[26] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *CVPR*, 2016.
[27] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *ICCV*, 2015.
[28] W. Zhong, "Robust object tracking via sparsity-based collaborative model," in *CVPR*, 2012.
[29] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *NIPS*, 2013.