Filomat 34:15 (2020), 5139–5148
https://doi.org/10.2298/FIL2015139Z

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

An Adaptive ECO with Weighted Feature for Visual Tracking

Yan Zhou[a], Hongwei Guo[a], Dongli Wang[a], Chunjiang Liao[a]

[a] *School of Automation and Electronica Information, Xiangtan University, Xiangtan, China 411105*

Abstract. The efficient convolution operator (ECO) have manifested predominant results in visual object tracking. However, in the pursuit of performance improvement, the computational burden of the tracker becomes heavy, and the importance of different feature layers is not considered. In this paper, we propose a self-adaptive mechanism for regulating the training process in the first frame. To overcome the over-fitting in the tracking process, we adopt the fuzzy model update strategy. Moreover, we weight different feature maps to enhance the tracker performance. Comprehensive experiments have conducted on the OTB-2013 dataset. When adopting our ideas to adjust our tracker, the self-adaptive mechanism can avoid unnecessary training iterations, and the fuzzy update strategy reduces onefifth tracking computation compared to the ECO. Within reduced computation, the tracker based our idea incurs less than 1% loss in AUC (area-under-curve).

1. Introduction

Generic visual object tracking is a classical and important computer vision problem, which is to estimate the trajectory about the target from an image sequence with the initial target location. Tracking of objects has numerous applications for traffic control surveillance [1], human-computer interactions [2], and the latest robotic service [3]. Even with significant progress has been made in this field, the capability of tracker still cannot meet the practical needs, for the slower training and tracking speed.

In visual object tracking, there are two main-streams, the deep learning and discriminative correlation filter (DCF). With the means of the DCF, there is the greater progress in the visual tracking accuracy, as shown on the benchmark of OTB [4], VOT [5]. These methods evolved from using the single channel feature representation [6] to multi-channel feature representation. Before the impressive performance of CNN (convolutional neural network) in the image classification, the discriminative correlation filter uses the handcrafted features such as the HOG [7] and the Color Names [8], expressed as some image intensity information or the simple color transformations.

During the past few years, the CNN has shown impressive progress in image classification. These networks take a fixed image as input, through a combination of convolution, local normalization, pooling

2010 *Mathematics Subject Classification*. Primary 93XX; Secondary 68XX
Keywords. object tracking, self-adaptive mechanism,the fuzzy model update strategy, convolutional neural network,weighted feature
Received: 27 September 2018; Revised: 19 November 2018; Accepted: 22 March 2019
Communicated by Shuai Li
Research supported by the National Natural Science Foundation of China (61773330), the Natural Science Foundation of Hunan Province (2017JJ2253), and the Research Project of Department of Education of Hunan Province (19C1740).
Email address: yanzhou@xtu.edu.cn (Yan Zhou)

operation and other components to extract features. Before extracting the features of the image, CNNs require enough training samples, for example, the ImageNet dataset [9], to conclude the coefficients of the network. Then, with the picture as input, the features extracted from the fixed weights network are generic and can be applied to a variety of vision applications.

As discussed earlier, the features extracted from the CNN are of great interest to the image classification. From [10], it is successful to integrate the DCF (discriminative correlation filter) with the features extracted from CNN. Combining the last layer with the intermediate layer is the better idea for the visual tracking [11]. Further-more, [12] integrates hand-crafted features with activation from the CNN. With the CNN feature, the performance of DCF has received a big boost.

In this paper, we propose a self-adaptive mechanism for regulating the training process of efficient convolution operators (ECOs). First, we perform the computation of the Gauss-Newton method [19] under control of some conditions to cease the unnecessary iterations automatically. Second, we exploit the fuzzy model update strategy, performing indefinite but within certain range frames to update the filter. Third, we express the weights for the different feature maps to enhance the adaptive ECO.

The remainder of this paper is organized as follows. Section 2 introduces application of feature representations from CNN, early correlation filter related to the object tracking. Section 3 discusses some ideas about the C-COT [11] and ECO [12]. Section 4 contains our improvement strategies. Experimental evaluations and results are provided in section 5.

2. Related Work

In recent years, the development of the convolutional neural network has improved the performance in object recognition and detection benchmark [13]. This makes the convolutional neural network to be a hot topic in machine learning. The activation from the certain layer of the trained deep network can be applied in some visual recognition tasks such as action recognition and scene classification [14]. [15, 16] shows the improved performance, which is achieved by the incorporation of feature from the intermediate layer and the fully connected layers of CNN. Due to the discrimination of the intermediate layers, the need for the task-specific fine-tuning is not essential. A cross-convolutional layer pooling method is proposed in [15]. The image representation is captured by pooling the extracted features from the successive convolutional layers. An approach based multi-scale convolutional feature is proposed by [16] for object recognition and texture classification. It has shown the superior performance from the activation of the last convolutional layer against other layers for the visual recognition [16].

The tracking approaches based DCF learn a correlation filter to discriminate the target from the background appearance. The observed target and the related background constitute the training data. Initially, Bolme et al. [6] proposed the MOSSE tracker, only using a single feature channel, which is a grayscale image. Later, Hen-riques et al. [7] introduced a variant DCF, which is kernelized, to allow non-linear classification boundaries. Recently, [7, 8, 17]attempted to incorporate the multi-dimensional feature representations and achieved a significant increase in the tracking performance. Despite their success, the DCF trackers are under the periodic assumption induced by circular correlation, which leads to the inaccurate location and a restricted search area. Very recently, Danelljan et al. [18] works out the solution to the issues by introducing SRDCF (the Spatially Regularized DCF). The expansion of the training and searching area are allowed in their formulation, without increasing the effective filter size. This prompts the robustness and discriminative power of the tracker, leading to a significant performance gain.

3. C-COT and ECO

In this section, we describe the C-COT [11] and the ECO [12]. The ECO achieved the top rank in the VOT2016 challenge [5] and OTB-2015 [4] for two main advantages compared to the native correlation filter

trackers. Firstly, the factorized convolution operator is introduced for reducing the quantity of the coefficient in the model. Secondly, with the compact generative model of the sample, the tracking computation is reduced a lot.

3.1. C-COT

Here, we briefly introduce the C-COT formulation, with the same symbol as in [11] for convenience. The C-COT trains a discriminative correlation filter from a collection of M samples $\{x_j\}_1^M \subset \chi$. Each sample x_j can be used to extract the feature maps which have D feature channels, the feature channel $x_j^d \in \mathbb{R}^{N_d}$ can be understood as a function $x_j^d[n]$ indexed by the discrete spatial variable $n \in \{0, \ldots, N_d - 1\}$, N_d denotes the resolution.

To transfer the training problem into a continuous spatial domain, it assumes the spatial support of the feature maps to be the continuous interval $[0, T) \subset \mathbb{R}$. Then, the interpolation operator J_d can be defined as:

$$J_d\{x^d\}(t) = \sum_{n=0}^{N_d-1} x^d[n] b_d(t - \frac{T}{N_d} n). \tag{1}$$

here,b_d is the interpolation function. The filter $f = (f^1, \ldots, f^D) \in L^2(T)^D$ is trained with a set of sample feature maps $\{x^1, x^2, \ldots, x^M\}$ and corresponding label score maps $\{y^1, y^2, \ldots, y^M\}$, by minimizing the objective function,

$$E(f) = \sum_{j=1}^{M} \alpha^j \| S_f\{x^j\} - y^j \|_{l^2}^2 + \sum_{d=1}^{M} \| \omega^d \cdot f_d \|_{l^2}^2. \tag{2}$$

The multi-channel convolution operation

$$S_f\{x\} = \sum_{d=1}^{D} f_d * J_d\{x^d\}. \tag{3}$$

is obtained by fusing the result of all channels, $\alpha^j > 0$ is the weight for the sample x^j,ω^d is to mitigate the drawbacks of the periodic assumption for the extended spatial support [18]. During the tracking,$f = (f^1, \ldots, f^D)$ is used to obtain the tracking scores by $S_f\{x_j\} = \sum_{d=1}^{D} f_d * J_d\{x^d\}$.

3.2. Training the continuous filter

The Fourier coefficients of the interpolated feature maps are given by

$$\widehat{J_d\{x^d\}}[k] = X^d[k]\widehat{b_d}[k] \tag{4}$$

Here, the $X^d[k] = \sum_{n=0}^{N_d-1} x^d[n] e^{-i\frac{2\pi}{N_d}nk}, k \in \mathbb{Z}$ is the discrete Fourier transform (DFT) of x^d. Then,

$$\widehat{S_f\{x\}}[k] = \sum_{d=1}^{D} \widehat{f^d}[k] X^d[k] \widehat{b_d}, k \in \mathbb{Z}. \tag{5}$$

And the Parseval's formula implies the equivalent function,

$$E(f) = \sum_{j=1}^{M} \alpha_j \|\widehat{S_f\{x_j\}} - \widehat{y}_j\|_{l^2}^2 + \sum_{d=1}^{D} \|\widehat{\omega} * \widehat{f^d}\|_{l^2}^2 \tag{6}$$

To simplify the normal expression (6), we adopt the sample matrix $A = [A_1^T \cdots A_M^T]^T$, where $A_j = [A_j^1 \cdots A_j^D]$, and $A_j^d = X^d \widehat{b_d}$, then, the equation (6) can be transformed as

$$(A^H \Gamma A + W^H W)\widehat{f} = A^H \Gamma \widehat{y} \tag{7}$$

W is the block-diagonal matrix $W = W_1 \bigoplus \cdots \bigoplus W_D$, the $\widehat{f} = [\widehat{f^1}^T \cdots \widehat{f^D}^T]^T$, the diagonal weight matrix $\Gamma = \alpha_1 I \bigoplus \cdots \bigoplus \alpha_m I$ and the label vector $\widehat{y} = [\widehat{y_1}^T \cdots \widehat{y_M}^T]^T$, ^{H} denotes the conjugate transpose of matrix. The filter $\widehat{f}$ can be iteratively solved by the expression (7), which is equivalent to the least squares problem.

3.3. The improved ECO

From the C-COT formulation, it is easy to know when the dimension of the filter $\widehat{f}$ is too large. ECO reduces the dimension of the filter, which is same as in the [8]. And the new filter of c channels can be reduced from the matrix-product Pf. The factorized convolution operator can be denoted as,

$$S_{Pf}(x) = Pf * J\{x\} = \sum_{c,d} p_{d,c} f^c * J_d\{x^d\} = f * P^T J\{x\}. \tag{8}$$

The formulation to train the filter f and the matrix P can be expressed as,

$$E(f, P) = \|\widehat{z}^T P \widehat{f} - \widehat{y}\|_{l^2}^2 + \sum_{c=1}^{C} \|\widehat{\omega} * \widehat{f^C}\|_{l^2}^2 + \lambda \|P\|_F^2. \tag{9}$$

Here, ECO adopts the Gauss-Newton method [19] to optimize the non-linear least squares problem (9). The linearizing residuals in (9) is obtained by the first order Taylor series expansion,

$$\widetilde{E}(\widehat{f_{i,\Delta}}, \Delta P) = \|\widehat{Z}^T P_i \widehat{f_{i,\Delta}} + (\widehat{f_i} \bigotimes \widehat{Z})^T vec(\Delta P) - \widehat{y}\|_{l^2}^2 + \sum_{c=1}^{C} \|\widehat{\omega} * \widehat{f_{i,\Delta}^C}\|_{l^2}^2 + \lambda \|P_i + \Delta P\|_F^2. \tag{10}$$

Here, $\widehat{f_{i,\Delta}} = \widehat{f_i} + \Delta \widehat{f}$, the $\bigotimes$ denote the Kronecker product. For simplicity, the (10) can be expressed as,

$$\widetilde{E}(\widehat{f}, \Delta P) = \|A_p \widehat{f} + B_f \Delta P - \widehat{y}\|_{l^2}^2 + \|W \widehat{f}\|_{l^2}^2 + \mu \|P + \Delta P\|_F^2. \tag{11}$$

Then, it is viable to learn the filter f and P jointly.

4. The Adaptive ECO

In the first place, the main problems with the ECO are computation-burden in the first frame. This due to the too much unnecessary iterations of the Gauss-Newton method. Next, we argue that the update strategy in ECO can be further sparse. Finally, the ECO does not emphasize weights of the related features. We set out to deal with these issues respectively, for improving both performance and speed.

4.1. Terminating training process automatically

From the ECO formulation, it needs two steps to train the filter f and the principal component matrix P in the first frame. Firstly, the ECO sets partial derivatives of the equation (11) with respect to f and ΔP to be zero, then, the formula can be expressed as

$$\begin{bmatrix} A_p^H A_P + W^H W & A_p^H B_f \\ B_f^H A_P & B_f^H B_f + \lambda I \end{bmatrix} \begin{bmatrix} \widehat{f} \\ \Delta P \end{bmatrix} = \begin{bmatrix} A_p^H \widehat{y} \\ B_f^H \widehat{y} - \lambda P \end{bmatrix} \tag{12}$$

Obviously, the Conjugate Gradient method can be performed iteratively on (12).

When we apply the flexible preconditioned Conjugate Gradient [20] method to the tracker equation (12), we find the residuals from the CG process converges to near zero without too many iterations (Figure 1). We proposed that there is no need for the further optimization on the coefficient of the filter when the residuals are small enough, and there is no major improvement in the tracking performance that adopts the latter filter f and matrix P (through 150 iterations), which is closed to the previous state (around 100 iterations for some sequences).

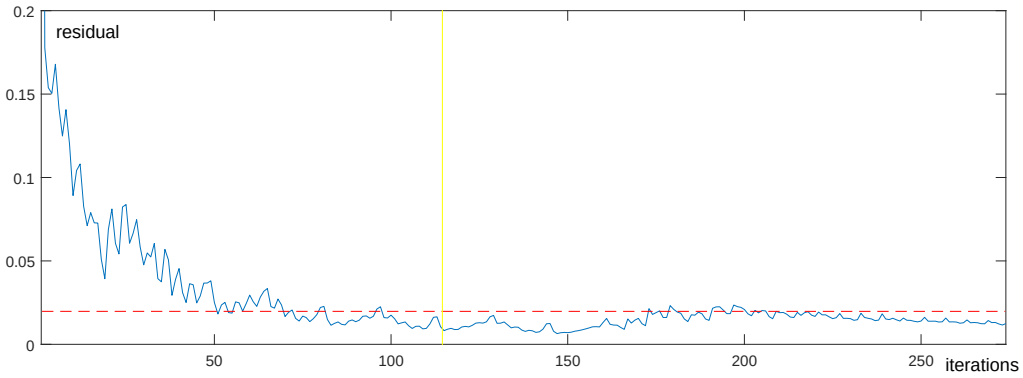

Figure 1: The curve of the residual related to the iteration. After 150 iterations is the tracking process. The red dotted line represents the cease condition, the yellow line represents the actual end of the iteration when adopted the auto-end method. Without our method the training process will not cease until 150 iterations.

To terminate these unnecessary training processes automatically, we set termination condition relating to the inner residual from the iteration of the equation (12), which can be expressed as the average of the successive residuals is smaller than the value associated with the first residual. With this setup, the number of iterations in the first frame decreases a lot, and it is obvious to notice that the training process becomes quicker for omitting some unnecessary iterations, the heavy computation for the training has also been reduced. From the perspective of practical use, it balances the accuracy and speed of the tracker. Most critically, when we need the tracker to start as soon as possible.

4.2. Filter update strategy

Concluded from the observation on the sample sequences and the real-time video (Figure 2), it is easy to know that the adjacent frames are of many similarities, and it is not necessary to update the filter frame by frame, which of redundant features for the sample space and wastes unnecessary time and memory consumption for filter update.

Figure 2: Visualization of some training samples in the part of the Crossing sequence, which are of great similarity around the target, this can lead to overfitting of the tracker, and slow down the tracking speed.

Intuitively, it is better to start an optimization process once sufficient change occurred. Under the formulation (11), we know it is hard and computational to estimate the condition to update the model, and may lead to unwanted complex heuristics. Inspired by the observation, we propose a fuzzy strategy to update the filter, which expressed as updating the filter with the uncertain number frames. This sparse updating scheme is common practice in non-DCF trackers [21, 22]. In our strategy, we choose the $N_{gap} \in [min, max]$ as the training gap to determine how often the filter is updated stochastically. With the fuzzy strategy, we avoid studying the change in the objective and simply update the filter between N_{gap} frames, where $N_{gap} = 1$ corresponds to updating the filter every frame, it is same as the standard DCF methods.

As a result, the times for the CG iterations in the whole sequences are decreased to $N_{tr} = (\sum(N_{seq}/N_{gap}))/(max - min + 1)$, which is also for the computational complexity of the filter adjustment. Delaying the model update a few frames, the loss become large enough, easy to notice the effect for training the filter, instead of only small change between the successive frames, the loss is too small to update the filter obviously. This is also for stabilizing the tracker when a target is affected by the sudden changes between frames, for instance, occlusions, deformation and out-of-plane rotations. In the ECO, a sequence with $N_{seq} = 100$ frames needs 20 update times, while for our method, about 16 times. If we just consider the number of iterations for updating the tracker during the tracking process, it is about one fifth computational saving.

4.3. The weighted feature

When we applied our method to adapt the computational complexity of training and tracking process, the performance of the ECO based our idea has been affected to some extent. We insist that the training process of the filter f and mapping matrix P is adequate, and the extracted features are not utilized properly.

From the structure of the CNN, the deeper the network layers used to extract feature maps, the more abstract the feature maps, and can be used for classification and recognition task. With the feature fusion for different layers, it is obviously to witness the progress made in the image classification, face recognition, and so on. And we also noted in the ECO, the author uses the layer Conv-1 and layer Conv-5 from the convolutional neural network and achieves the better results. But the ECO doesn't underline the importance of different feature maps, just sets feature maps from the two CNN layers and HOG to the equal importance in the target location process.

In-depth study of the CNN for image classification, we know that the feature from the last layer is used for deciding which category this picture belongs to, with its more abstract features. While in the tracking,

given the detail of the formulation of the standard DCF, enough spatial resolution conduces to the target location. In addition, from the [10], the feature maps from the first convolutional layer achieved best tracking performance compared to other single layers. Moreover, it also showed the decreased performance for the deeper layer, but the last layer, which provides a significant performance gain compared to the fourth layer. Because of the better performance from the fifth layer feature, we also conclude that the advanced features encoded by the deepest layers in the network, just as in the image classification.

Under the previous analysis, we assume that the response from the deeper layers of the CNN may need the higher weight, even the decreased spatial resolution in the deeper layers is not suitable for locating the target accurately. Therefore, during the locating process, we set the fifth layer with higher weight, for the higher-level feature expression.

$$S_{Pf}\{x\} = w_{l5} f_{l5} * P_{l5}^T J_{l5}\{x\} + w_{l1} f_{f1} * P_{l1}^T J_{l1}\{x\} + w_h f_h * P_h^T J_h\{x\}. \tag{13}$$

With the condition $w_{l5} + w_{l1} + w_h = t$, the t denotes the number of the feature types.

5. Experiments

We validate the proposed method by performing comprehensive experiments on the OTB-2013 dataset, it is part of the OTB-2015 dataset, with comparisons to some state-of-the-art methods.

5.1. Implementation Details

The experiments are performed on an Intel(R) Core(R) i5-7300HQ 2.50GHz CPU with 16GB RAM, and using the MatConvNet toolbox [23], the computation for extracting the features from the image sequences on CNN is transferred to the GeForce GTX 1060 GPU. The tracker based our idea runs at an average of 9 FPS.

We apply the same feature representation as ECO, the first convolutional layer (layer Conv-1) and the last convolutional layer (layer Conv-5) from the VGG-m Network [24], and HOG [25]. For the factorized convolution presented in section (3.3), we learn the projection matrix P for each feature type correspondingly. In the first frame, we perform the Gauss-Newton method to equation (9) 10 times, while every time we perform the Conjugate Gradient method on the equation (11) 15 iterations. With our idea, the training process is under our auto-end mechanism. After the first frame, the filter is updated in every $N_s \in [5, 7]$ frames with 5 iterations of the Conjugate Gradient applying to the equation (7). The filter $\widehat{f_0}$ is initialized to zero, and we initialized the coefficient matrix P_0 by applying the PCA to the sample feature representation in the first iteration to keep the discriminative property of the tracker.

We set w_{l5}, w_{l1} and w_h for the different convolutional feature maps, which are the layer Conv-1 and layer Conv-5, and the HOG feature representations, to identify the appropriate weight for improving the tracking performance. When we assign the weights to the convolution responses produced by the layer Conv-1, layer Conv-5 and HOG, the spatial size and the dimensionality of feature maps are not considered for simplicity, and we just let the weights influence the locating process. For our experiments, $w_{l1} = 0.7$, $w_{l5} = 1.4$, $w_h = 0.9$. By the weighted feature maps, the performance of adaptive ECO is closed to the original ECO.

5.2. Baseline comparison

When using ECO for tracking, its main computation burden comes from the input of sequences and the iterative operation of Conjugate Gradients. Here, we just show how many iterations needed in the first frame. Indirectly, by the mechanism of terminating the loop automatically, almost the one-third of iterations are saved from the OTB-2013 sequences. More importantly, it is of great significance for practical use.

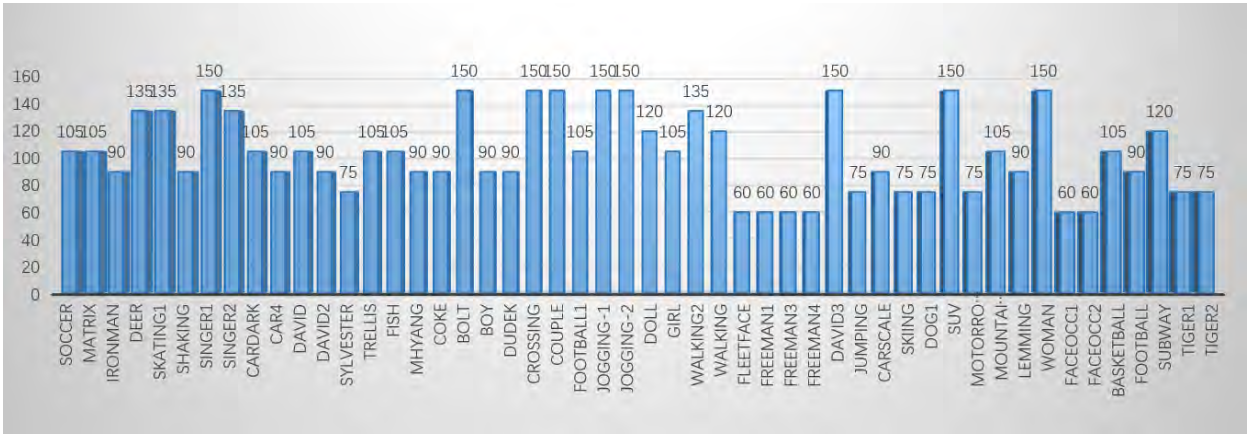

Figure 3: The iterations needed for training in the first frame within our idea.

5.3. State-of-the-art comparison

We conduct the comparison on the OTB-2013 dataset, which contains 50 sequences with 51 target annotations, in terms of AUC (area-under-curve). The AUC is calculated with a range of thresholds for the IoU (intersection-over-union) overlap over the videos.

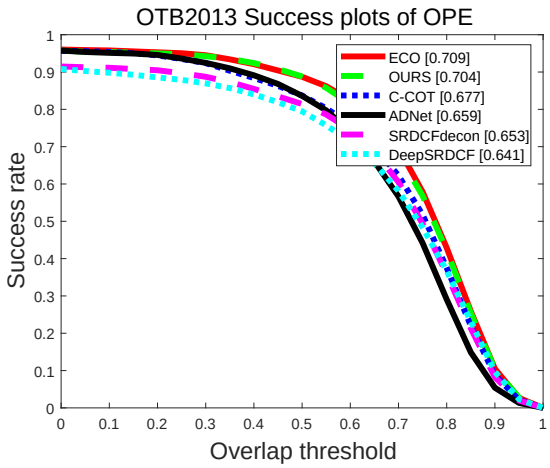

Figure 4: Success plot on the OTB-2013. The AUC score of each tracker is shown in the legend.

We provide a comparison of our method with some state-of-art methods: ECO [12], C-COT [11], DeepSRDCF [10], ADNet [26], SRDCFdecon [17].

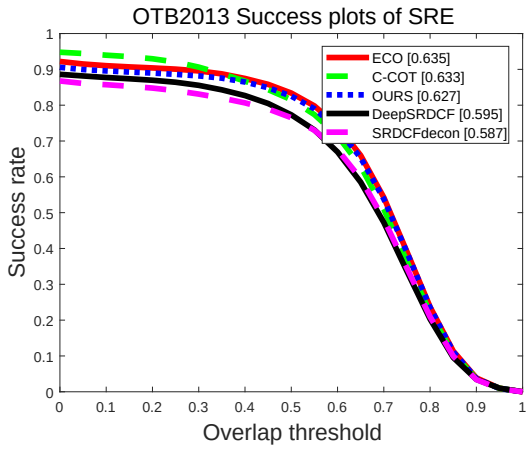

Figure 5: Comparison with respect to the spatial robustness initialization on OTB-2013

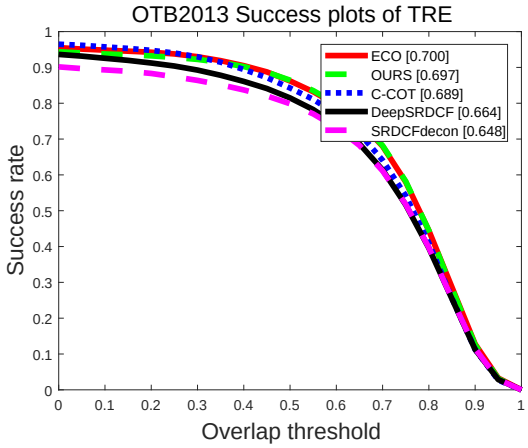

Figure 6: Comparison with respect to the temporal robustness initialization on OTB-2013.

Figure 4 shows the success plot over the OTB-2013 dataset [28]. Our tracker provides the results with an AUC score of 0.704, only loses less than 1% score compared to the ECO. With the compensation of weighted features, there is almost none loss in tracking performance, under a modest amount computation reduction of training and tracking process.

5.4. Robustness to initialization

The robustness evaluation scenarios happen a lot in the real-world application when a tracker is initialized, which is likely to incur initialization with a slightly biased position and different scales. We identify the robustness of our tracker following the protocol proposed in [4]. Two different perturbing initialization criteria, namely, TRE (temporal robustness evaluation) and SRE (spatial robustness evaluation). With the TRE, the tracker is evaluated with 20 frames initialized the ground-truth bounding box of the target, until the end of the sequence, just like one segment of the entire sequence with the ground-truth target. For the SRE criteria, the initialization of the tracking is performing by shifting or scaling the ground-truth bounding box of the target, and this process repeats 12 times for every sequence in the dataset.

From the Figure 5 and Figure 6. It is obvious to notice that the robustness score has not been affected too much. For the SRE, our approach causes the score loss about 0.008 compared to ECO, and for the TRE, the score loss is about 0.003 compared to ECO. Considering the adaptation in the computational complexity of training, the robustness score is almost unchanged by the weighted features, because of the higher weight on the fifth layer feature representation from the CNN.

6. Conclusion

In this paper, the training process and tracking process of the ECO have been adjusted to count the issues of computational complexity and overfitting. An auto-stop mechanism has been introduced for the training phase in the first frame. To reduce the redundant training complexity, we also proposed the fuzzy update strategy for the overfitting in tracking process. Lastly, the weighted feature representation makes the tracker perform well with the reduced computational complexity, by higher weights for the more abstract features in the deeper layer of the CNN. The experimental results have shown the superiority.

7. Acknowledgments

This work was supported by the National Natural Science Foundation of China (61773330), the Natural Science Foundation of Hunan Province (2017JJ2253), and the Research Project of Department of Education of Hunan Province (17B259).

References

[1] Janzen A W, Monroe R M K. Traffic Control Systems and Methods: U.S. Patent Application 15/725,200[P]. 2018-4-5.
[2] Lazar J, Feng J H, Hochheiser H. Research methods in human-computer interaction[M]. Morgan Kaufmann, 2017.
[3] Computer-Assisted and Robotic Endoscopy: Third International Workshop, CARE 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Revised Selected Papers[M]. Springer, 2017.
[4] Wu Y, Lim J, Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.
[5] Kristan M, Leonardis A, Matas J, et al. The Visual Object Tracking VOT2016 Challenge Results[J]. Lecture Notes in Computer Science, 2016, 9914: 777.
[6] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2010: 2544-2550.
[7] Henriques J F, Caseiro R, Martins P, et al. High correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.
[8] Danelljan M, Shahbaz Khan F, Felsberg M, et al. Adaptive color attributes for realtime visual tracking[C]//IEEE Conference on Computer Vision and Pattern Recognition Columbus, Ohio, USA, June 24-27, 2014: 1090-1097.
[9] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248-255.
[10] Danelljan M, Hager G, Shahbaz Khan F, et al. Convolutional features for correlation filter based visual tracking[C]// the IEEE International Conf. Computer Vision. 2015: 58-66.
[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] Danelljan M, Bhat G, Khan F S, et al. Eco: Efficient convolution operators for tracking[C]// IEEE Conf. Computer Vision & Pattern Recognition, Honolulu, USA. 2017: 21-26.
[13] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.
[14] Razavian A, Azizpour H, Sullivan J, et al. CNN features off-the-shelf: an astounding baseline for recognition// IEEE Conf. comp. vision & pattern recog. 2014: 806-813.
[15] Liu L, Shen C, van den Hengel A. Cross- convolutional-layer pooling for image recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 39(11): 2305-2313.
[16] Cimpoi M, Maji S, Vedaldi A. Deep convolutional filter banks for texture recognition and segmentation[J]. arXiv preprint arXiv:1411.6836, 2014.
[17] anelljan M, Häger G, Khan F, et al. Accurate scale estimation for robust visual tracking[C]//British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press, 2014.
[18] Danelljan M, Hager G, Shahbaz Khan F, et al. Learning spatially regularized correlation filters for visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 4310-4318.
[19] Nocedal J, Wright S. Numerical optimization, series in operations research and financial engineering[J]. Springer, New York, USA, 2006, 2006.
[20] Notay Y. Flexible conjugate gradients[J]. SIAM Journal on Scientific Computing, 2000, 22(4): 1444-1460.
[21] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking[C]// IEEE Conf. Computer Vision & Pattern Recognition, 2016: 4293-4302.
[22] Zhang J, Ma S, Sclaroff S. MEEM: robust tracking via multiple experts using entropy minimization[C]//European Conf. Computer Vision. Springer, Cham, 2014: 188-203.
[23] Vedaldi A, Lenc K. Matconvnet: Convolutional neural networks for matlab[C]// the 23rd ACM International Conference on Multimedia. ACM, 2015: 689-692.
[24] Chatfield K, Simonyan K, Vedaldi A, et al. Return of the devil in the details: Delving deep into convolutional nets[J]. arXiv preprint arXiv:1405.3531, 2014.
[25] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2005: 886-893.
[26] Yoo S Y J C Y, Yun K, Choi J Y. Action-decision networks for visual tracking with deep reinforcement learning[J]. 2017.
[27] Danelljan M, Hager G, Shahbaz Khan F, et al. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2016: 1430-1438.
[28] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]// the IEEE conf. computer vision and pattern recognition. 2013: 2411-2418.