Correlation Filters with Weighted Convolution Responses

Zhiqun He[1*], Yingruo Fan[1*], Junfei Zhuang[1*], Yuan Dong[1,2], HongLiang Bai[2]
[1]Beijing University of Posts and Telecommunications, [2] Beijing FaceAll Co.
{he010103, evelyn}@bupt.edu.cn {junfei.zhuang, yuan.dong, hongliang.bai}@faceall.cn

Abstract

In recent years, discriminative correlation filters based trackers have shown dominant results for visual object tracking. Combining the online learning efficiency of the correlation filters with the discriminative power of CNN features has aroused great attention. In this paper, we derive a continuous convolution operator based tracker which fully exploits the discriminative power in the CNN feature representations. In our work, we normalize each individual feature extracted from different layers of the deep pretrained CNN first, and after that, the weighted convolution responses from each feature block are summed to produce the final confidence score. By this weighted sum operation, the empirical evaluations demonstrate clear improvements by our proposed tracker based on the Efficient Convolution Operators Tracker (ECO). On the other hand, we find the 10-layers design is optimal for continuous scale estimation, which contribute most to the performance. Finally, our tracker ranks top among the state-of-the-art trackers on VOT2016 dataset and outperforms the ECO tracker on VOT2017 dataset.

1. Introduction

Visual object tracking has several applications such as robotic services, traffic control, surveillance, human-computer interactions and so on. Even though significant progress has been made in this research area, tracking is still a challenging problem due to fast motions, occlusions, deformations, and illumination variations.

In recent years, the progress in deep learning spreads to tracking field remarkably. In the meantime, discriminant correlation filters (DCF) based trackers [8, 14, 17, 18] achieve the desired effect between accuracy and speed by solving a ridge regression problem efficiently in Fourier frequency domain. Combining the correlation filters with CNN features has been done in several works [11, 16, 27, 23], which have shown that pretrained deep CNNs and adaptive CFs are complementary and achieved state-of-the-art results on many object tracking benchmarks [28, 20, 21].

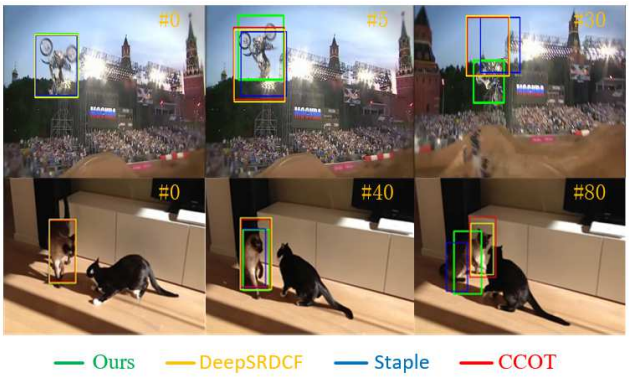

Figure 1: Comparison of our tracker (in green) with the state-of-art trackers , *e.g*. DeepSRDCF (in yellow), Staple (in blue) and CCOT (in red). Example frames are shown from the motocross1 (top row) and fernando (bottom row) sequences. Our tracker is able to handle the scale changes and avoid overfitting in these sequences, thereby increasing both the accuracy and robustness of the tracker.

The recent advancement in DCF based tracking performance is driven by the use of color information [14], robust scale estimation [10], reducing boundary effects [12], non-linear kernel [18] and multi-dimensional features [13, 7]. Different features would capture different channels of target information and result in a better performance. Nevertheless, learning deep convolutional features for correlation filters based trackers is still not completely explored.

The tracker proposed in this paper is built upon a correlation filters based tracker popularly known as the Efficient Convolution Operators Tracker (ECO) [7], an improved version of the tracker C-COT [13]. C-COT has achieved impressive results on the visual tracking benchmark [28, 20, 19] and ranked first in the VOT 2016 challenge [19]. On the other hand, ECO has addressed the problems of computational complexity and over-fitting in the C-COT framework, with the aim of simultaneously improving

*contributed equally

both speed and performance.

Our main contributions are three folds and summarized as follows:

- We exploit the great power in the CNN feature representations and revise the feature extraction for ECO tracker, which takes advantages of the multi-resolution deep feature maps without any hand-crafted features such as HOG [6] and Color Names [9].
- We propose the sum operation of the weighted convolution responses from each feature block. In our work, we normalize each individual feature extracted from different layers of the deep pretrained CNN first, after that, the weighted convolution responses from each feature block are summed to produce the final confidence score. Based on the feature normalization and the weighted sum operation, the Expected Average Overlap (EAO) [5] would be improved compared to the original ECO [7] tracker.
- We find the 10-layers design is optimal for continuous scale estimation, which is developed to determine the scale of target object and improves the baseline by 2%.

Finally, our tracker achieved a very appealing performance both in accuracy and robustness against the state-of-the-art trackers on VOT2016 dataset [19] and outperforms the ECO [7] tracker on VOT2017.

2. Related Work

For visual object tracking, correlation filters based methods have shown dominant and impressive results on many object tracking benchmarks [28, 19, 20, 21]. Discriminative Correlation Filters (DCF) is one of the CF frameworks that has achieved state-of-art results in Visual Object Tracking Challenge (VOT). Generally, it uses cyclic shifted samples [18] to train the correlation filters to discriminate between the target and background appearance.

Bolme *et al*. [4] proposed the MOSSE tracker, which used the grey-scale image to extract single channel feature with high speed. Later, Henriques *et al*. [18] utilized the kernelized correlation filter (KCF) by introducing the kernel trick into ridge regression [24]. KCF *et al*. [18] could solve for the filter taps very efficiently by utilizing the circulant structure of the underlying kernel matrix. Since then, using correlation filters for tracking has attracted more attention in visual object tracking. Due to the great power in the CNN feature representations, Martin *et al*. [13, 7] utilized deep convolutional features to learn more discriminative information.

The DCF based tracker C-COT [13] allows an integration of multi-resolution feature maps and hand-crafted features, which is the best tracker in VOT 2016 challenge [19]. Based on the C-COT [13] framework, Martin *et al*. [13, 7] proposed the ECO [7] tracker by using PCA to reduce the dimensionality and introducing efficient generative sample space model to boost the overall performance. Therefore, the ECO [7] tracker could achieve better performance with faster speed than the C-COT [13] tracker.

2.1. Based Framework

A theoretical framework for learning continuous convolution operators is proposed in C-COT [13]. They introduce an implicit interpolation model of the training samples. Each sample x_j contains D feature channels x_j^1, x_j^2, ..., x_j^D, extracted from the same image patch. In the formulation, let N_d denote the number of spatial samples in x_j^d, which indexed by the feature channel variable $d \in \{0, 1, 2...\}$. The feature channel $x_j^d \in \mathbb{R}^{N_d}$ is viewed as a function $x_j^d[n]$ indexed by the discrete spatial variable $n \in \{0, ..., N_d - 1\}$. The spatial support of the feature map is assumed to be the continuous interval $[0, T) \subset \mathbb{R}$. The interpolation operator J_d is constructed as:

$$J_d\left\{x^d\right\}(t) = \sum_{n=0}^{N_d-1} x^d[n]\, b_d\left(t - \frac{T}{N_d} n\right), \qquad (1)$$

where $b_d \in L^2(T)$, the Hilbert space, represents the interpolation function. As discussed, the confidences are defined on a continuous spatial domain. Therefore, the convolution operator maps a sample $x \in \chi$ to $s(t) = S_f\{x\}(t)$. $t \in [0, T)$ denotes the location in the image. In the continuous formulation, a set of convolution filters $f = \left(f^1, ..., f^D\right) \in L^2(T)^D$ is estimated to construct the convolution operator. The convolution operator is given by:

$$S_f\{x\} = \sum_{d-1}^{D} f^d * J_d\left\{x^d\right\}, x \in \chi. \qquad (2)$$

Here, $*$ is the circular convolution operation : $L^2(T) \times L^2(T) \rightarrow L^2(T)$. The interpolated sample $J_d\left\{x^d\right\}(t)$ is convolved with its corresponding filter. Then, the convolution responses from all filters are summed to produce the final confidence function $S_f\{x\}$.

In the continuous learning framework, each training sample $x_j \in \chi$ is labeled by confidence functions $y_j \in L^2(T)$, the desired output of the convolution operator $S_f\{x_j\}$. Therefore, the correlation filter cost function is the proposed formulation:

$$E(f) = \sum_{j=1}^{m} \alpha_j \left\|S_f(x_j) - y_j\right\|^2 + \sum_{d=1}^{D} \left\|w f^d\right\|^2. \qquad (3)$$

In the above relation, α_j represents the importance of each training sample, and the penalty function $w \in L^2(T)$ is a spatial regularization term. Note that $\left\|w f^d\right\| < \infty$ is

required because w has many non-zero Fourier coefficients $\hat{w}[k]$. By minimizing the function Eq.(3), the procedure is derived to train the continuous filters $f = \left(f^1, ..., f^D\right) \in L^2\left(T\right)^D$.

In the object tracking framework, the minimization of the functional is equivalent to solving the least squares problem, using the Conjugate Gradient method iteratively.

3. Method

In this section, we first present the integration of CNN features used in this tracker, which help our tracker becom lightweight but efficient. Then we discuss weighted sum operation and model update strategy. Finally, we intruduce our scale adaptation scheme, which contributes most to the performance.

3.1. Integration of Convolutional Features

Hand-crafted features such as HOG [6] and CN [9] have been used to represent the outline information of the object and color information, which gain excellent performance in visual tracking in the recent years. Due to the advances of CNN, the quality of image recognition and object detection has been progressing at a dramatic pace. CNN features have been proved to be efficient and robust in visual tracking. Therefore, we use popular VGG-M [26] network pretrained on the ILSVRC [1] dataset to extract multi-resolution convolutional features. Our model well incorporates both the deep but highly semantic features and the shallow but high-resolution features of the image. We ignore the fully-connected layers that contain little spatial resolution, *i.e.*, 1×1, which are not efficient to locate the target.

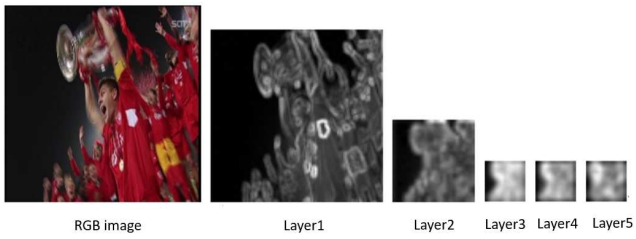

Figure 2: The visualization of the grey-scale feature maps extracted from different convolutional layers of the employed network. The RGB image denotes the input , after being resized with certain proportion.

We visualize the grey-scale feature maps of the first five convolutional layers. As shown in Figure 2, the first convolutional layer have much information about the outline of the object, while the last convolutional layer is efficient to discriminate the targets from the complicated background. When merging the features extracted from different layers of CNN, one must be careful to normalize each individual feature to make the combined features work well. As the larger features dominate the smaller ones, simply concatenating features may lead to poor performance. At this point, the weighted convolution responses from each layer are summed to produce the final confidence score. Finally, we choose the conv1 layer and the conv5 layer as the multi-resolution deep feature maps without any handcrafted features.

3.2. Weighted Sum Operation

As mentioned in Section 3.1, the spatial size of the convolutional features extracted from the VGG-M network are respectively 109×109, 26×26, 13×13, 13×13, 13×13, while the input RGB image is 224×224 after the necessary preprocessing steps. The size of conv5 feature map is 13×13, whose area is approximately 0.12 as that of the conv1 layer. There is supposed to be some imformation loss when we simply regard the conv5 feature map is as important as that of the conv1. Therefore, we assign larger weight to the feature map of the conv5 layer.

The convolution responses from all filters are summed to produce the final confidence function,

$$\mathrm{S}_f(x) = \mathrm{W}_1 \sum_{a=1}^{D_{conv1}} f^a * J_a\{\mathrm{x}^a\} + \mathrm{W}_2 \sum_{b=1}^{D_{conv5}} f^b * J_b\{\mathrm{x}^b\}, \quad (4)$$

Here, the feature maps of the conv1 layer and the conv5 layer are first interpolated using Eq. 1, as presented in Section 2.1 and then convolved with its corresponding filter. Note that D_{conv1} and D_{conv5} represent the dimensionality of the convolutional features extracted from the employed network. W_1 and W_2, the coefficients of each convolution response, denote the significance of each convolutional layer.

Based on Eq. 4, the sum of the weighted convolution responses from each feature block can be obtained to decide the final location of the target. Similar to the C-COT [13] method, the final confidence score is defined on a continuous spatial domain and the convolution operator maps a sample $x \in \chi$ to $s\left(t\right) = S_f\left\{x\right\}\left(t\right)$. Here, $t \in [0, T)$ denotes the location in the image. By applying Eq. 1 and Eq. 4, we obtain

$$E(f) = \sum_{j=1}^{m} \alpha_j \left\| W_1 \sum_{a=1}^{D_{conv1}} f^a * J_a\left\{x_j^a\right\} + W_2 \sum_{b=1}^{D_{conv5}} f^b * J_b\left\{x_j^b\right\} - y_j \right\|^2 + \sum_{d=1}^{D} \left\| w f^d \right\|^2 . \quad (5)$$

Hence, W_1 and W_2 are applied to the original loss, as expressed in Eq. 5. α_j represents the importance of each training sample and the penalty function $w \in L^2(T)$ is

a spatial regularization term. Note that $D = D_{conv1} + D_{conv5}$. By minimizing the functional Eq. 5, the model update procedure is derived to train the continuous filters $f = \left(f^1, ..., f^D\right) \in L^2\left(T\right)^D$.

3.3. Model Update Strategy

Ns	4	5	6	7
EAO	0.271	0.303	0.275	0.281
Formula	PR		FR	
EAO	0.303		0.283	

Table 1: Two different model update strategy experiments on VOT2017 dataset. The top row demonstrates the effect of training gap Ns on Expected Average Overlap (EAO), we can dramatically increase EAO with a suitable Ns. Two conjugate iterative algorithms are Fletcher-Reeves (FR) and Polak-Ribiere (PR) respectively in the bottom row. Obviously, PR formula provide a better convergence rates.

Most existing tracking approaches [4, 12, 18] are to update their models in each frame, assuming that the model is always credible. However, this strategy fails when our tracker meets some complicated situations such as occlusion, illumination variation, abrupt motion and deformation. Moreover, tracking models suffer from heavy computational load by updating the model in each frame.

A sparser updating scheme should replace the model update strategy that updates the model in a continuous fashion every frame. In the ECO [7] method, the approach is simply updating the filter by starting the optimization process in a fixed number of frames. A moderately infrequent update of the model generally not only leads to improved tracking results but also has a substantial effect on the overall computational complexity of the learning. By postponing the model update a few frames, the loss is updated by adding a new mini-batch to the training samples, instead of only a single one which helps to reduce over-fitting to the recent training samples. We find that this strategy is simple and brash. Furthermore, it neglects the relationship between the convergence speed of the optimization and model update frequency. Intuitively, a sparser updating scheme leads to a low convergence speed. Hence, we should increase the number of Conjugate Gradient iterations. More than that, to improve convergence rates , we should also choose a suitable momentum factor by using Fletcher-Reeves formula [15] or the Polak-Ribiere formula [25].

To find the best training gap strategy, a series of experiments were carried out to increase the updating frame number from 4 to 7 by a fixed number of CG iterations . According to Table 1, it can be concluded that the number of 5 is the best updating frame number. And it can be informed from Table 1, Polak-Ribiere formula bring us to achieve a faster convergence. Table 1 provides us the relationship between the convergence speed of the optimization and model update frequency, we can conclude that the most suitable and are 5 and 5 separately.

3.4. Scale Adaptation Scheme

In the detection step, an ideal scale estimation approach should be robust to scale changes while being computationally efficient. We find a new scale adaptation scheme, significantly improving the performance of correlation filter based trackers. Similar to the exhaustive scale space tracking framework [10], we first construct a feature pyramid in a rectangular area around the given target location. Let $P \times R$ denote the target size in the current frame and S be the size of the scale dimension. For each $n \in \left\{\left\lfloor -\frac{S-1}{2} \right\rfloor, ..., \left\lfloor \frac{S-1}{2} \right\rfloor\right\}$, we extract an image patch J_n of size $a^n P \times a^n R$ centered around the target. Here, a denotes the scale factor between feature layers. The scale layers S is the most important parameter. We evaluate its effects on the tracking performance when its value varies as S = 10, with corresponding $a = 1.01, 1.02, 1.03$ respectively. Different from previous parameter selection, we find the 10-layers design may be better for model training and scale change capture, thus promotes the performance. The special structure of response weights training samples diversely and permits our algorithm to estimate the position and scale simultaneously. Considering the reasonable scale changes, the min scale factor is given by

$$\delta_{min} = a^{\lceil \log_a (5/I) \rceil}, \tag{6}$$

where I is the supported size of the input image of our network. a denotes the scale factor between feature layers. The max scale factor can also be expressed as

$$\delta_{max} = a^{\lfloor \log_a \lceil min(W,H)/P \times R \rceil \rfloor}, \tag{7}$$

where W and H are the width and height of the original image respectively. The optimal object scale in current frame δ_{cur} would be decided under the restriction of δ_{min} and δ_{max}.

4. Experiments

We conduct two experiments to evaluate the efficiency and accuracy of our proposed tracker. First, we compare our tracker against state-of-art trackers that participated in the VOT 2016 challenge [19]. Secondly, we evaluate our tracker using the VOT 2017 toolkit on a set of 60 challenging videos.

4.1. Experimental Setup

Our tracker was implemented in MATLAB. The experiments were performed on an Intel(R) Xeon(R) 2.60GHz CPU and a GeForce GTX 1080 GPU. Our tracker runs at an average of 4fps on GPU and 1.4fps on CPU.

4.2. Implementation Details

In our experiment, the search image is cropped centered at the target object with the size of 4 times larger than the target in a restricted area [200,250], introduced in Section 3.4, after that, it would be resized to 224×224 as the input of the VGG-M network. We have also tried various input size, *e.g.*, 336×336, 168×168. The results can be seen in Table 2. For the sample space model in Section 3.3, the maximum number of stored training samples is set to L=100 and the learning rate is set to $\gamma = 0.012$. We update the filter in every $N_s = 6$ frame with 5 Conjugate Gradient iterations, whlie the initiate CG iterations is 200 in the first frame. As discussed in Section 3.1, we use the feature maps of the first normalization layer (norm1) and the feature maps of relu layer after the last convolution layer (conv5) as the features without PCA. Then we use the L2-norm method to normalize the features to boost the performance.

	168×168	224×224	336×336
EAO	0.244	0.303	0.289

Table 2: Analysis of the effect by changing the search image size on the VOT2017 dataset. We show the performance in Expected Average Overlap (EAO). It can be concluded that 224×224 is the best resized image size.

4.3. Features Comparison

conv layer	1	5	1,3	1,5	1,3,5
EAO	0.231	0.212	0.234	0.303	0.281

Table 3: A baseline comparison when using different combinations of convolutional layers in our object tracking framework. We show the results of expected average overlap (EAO) on VOT2017 dataset. The best results are obtained when combining conv1 and conv5 in our framework. The results shows the performance will not always be better when more features are used.

To our knowledge, grayscale intensity, Histogram of Oriented Grandients (HOG) [6] and Color Names (CN) [9] are useful hand-crafted features in visual tracking. However, when merging the CNN features with hand-crafted features, the performance is not so good as we expected. One reason is that the CNN features are so powerful that the hand-crafted features may sometimes weaken the representation. To exploit the great power in the CNN feature representations , we evaluate the combination of different convolutional layers of the VGG-M network without any handcrafted feature in the feature comparison experiment.

The evaluation is performed on all 60 videos in the VOT2017 dataset and the results are presented in terms of Expected Average Overlap (EAO). From Table 3, the performance will not always be better when more CNN features are used. As discussed in Section 3.1, the first convolutional layer has much information about the outline of the object, while the last convolutional layer is efficient to discriminate the targets from the complicated background. For the tracking problem, better spatial resolution alleviates the task of accurately locating the target. We conjecture that the fifth convolutional layer provides a significant performance gain compared to the fourth layer. In summary, our results suggest that the combination of the initial convolutional layer and the fifth convolutional layer provides the best performance for visual tracking.

4.4. Weights in Convolution Responses

When the weights are assigned to the convolution responses produced by different convolutional layers, the spatial size and dimensionality of the convolutional features extracted from the VGG-M network must be considered. Let $\sigma = W_2/W_1$ denotes the relative weight. Here, W1 and W2 are the coefficients of the convolution response of the conv1 layer and the conv5 layer, represent its corresponding significance. Figure 3 shows that the performance then degrades if the relative weight σ is not selected properly, such as 4, 3 and 1.5. As we have supposed, the Expected Average Overlap (EAO) on VOT2017 dataset could be improved if the significance of different convolutional layers considered. The final convolutional layer of VGG-M network (conv5), which has recently been successfully applied in image classification, provides a large amount of invariance while still discriminative. Therefore, we conjecture that the feature map extracted from the conv5 layer accounts for larger proportion. This is likely due to the high level features encoded by the deepest layers in the network.

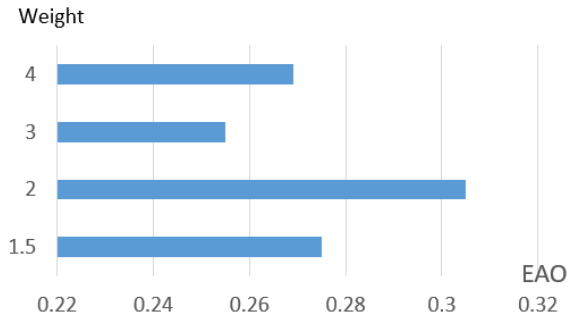

Figure 3: Comparison of the Expected Average Overlap (EAO) on VOT2017 with different weight proportion (conv5/conv1) in convolution responses.

4.5. Scale Parameter Analysis

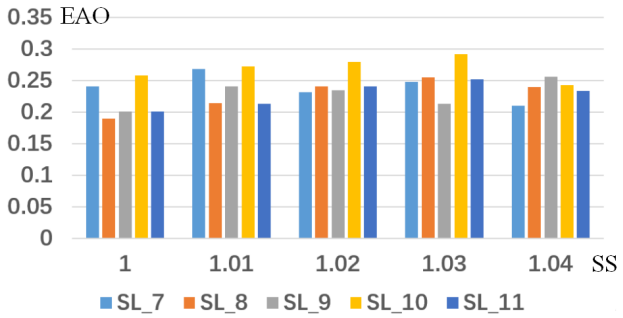

Figure 4: Analysis of the effect on Expected Average Overlap (EAO) on VOT2017 dataset by changing the number of scale layers (SL) and the scale step (SS) respectively.

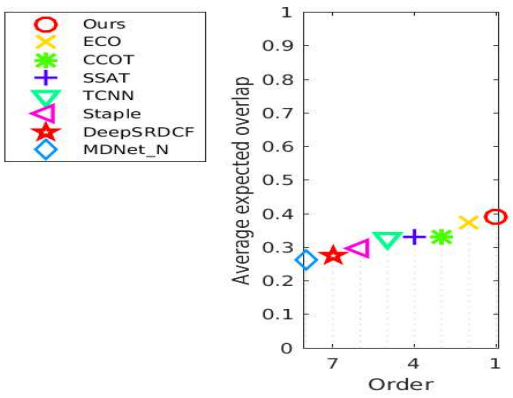

Figure 5: Ranking plots for the baseline experiments in the VOT2016 dataset. The order and EAO are plotted along the vertical and horizontal axis respectively. Our approach (denoted by the red circle) achieves superior results in the baseline experiments.

We use our CNN based tracker with the 7-layers scale design as a baseline. The scale layers (SL) is the most important parameter. We evaluate its effects on the tracking performance when its value varies as SL = 7, 8, 9, 10 with corresponding scale step SS= 1.00, 1.01, 1.02, 1.03, 1.04. The ECO [7] tracker uses the 7-layers design, which is thought enough for continuous scale estimation. Figure 4 shows the 10-layers design with corresponding a = 1.03 has improved the baseline by 2% in EAO on VOT2017 dataset. Meanwhile, the 11-layers design may be too meticulous for scale change capture during the model training, thus degrades the performance. Based on these analysis, we therefore choose SL = 10 and SS = 1.03 in all the following experiments.

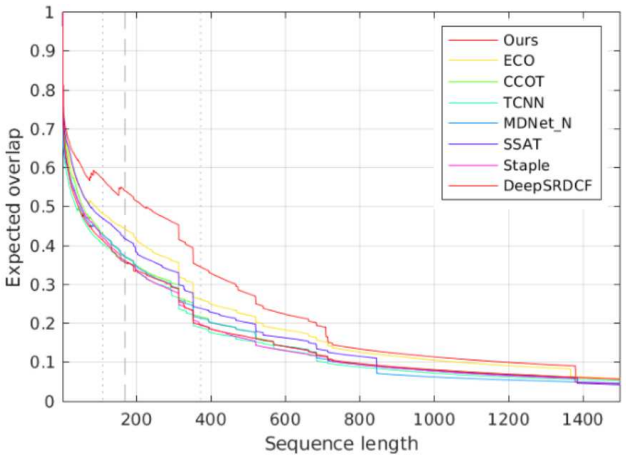

Figure 6: Expected overlap curves for baseline (size_change).

4.6. State-of-the-art Comparison on VOT2016

Tracker	EAO	Acc. Raw	Fail. Raw
Ours	0.3905	0.58	0.81
ECO	0.3742	0.55	0.87
CCOT	0.3310	0.54	0.89
SSAT	0.3292	0.56	0.77
TCNN	0.3266	0.53	0.90
Staple	0.2952	0.54	0.96
DeepSRDCF	0.2763	0.54	1.42
MDNet_N	0.2610	0.52	1.23

Table 4: VOT2016 performance results.

The Visual Object Tracking (VOT) Challenge is a competition among short-term, model-free visual object tracking algorithms. For clarity, the proposed algorithm is compared with the 6 state-of-art trackers including CCOT [12], TCNN [22], Staple [2], DeepSRDCF [11], SSAT [3] and ECO [7]. It is clear from Table 4 that our tracker outperforms all the trackers in VOT2016 challenge with an expected average overlap (EAO) of 39.05%, which achieves a relative gain of 17.98% compared to C-COT [13], the top-ranked tracker with an expected average overlap(EAO) of 33.1%. Moreover, as shown by Figure 6, the proposed tracker performs greatly in expected overlap curves when the scale size changes.

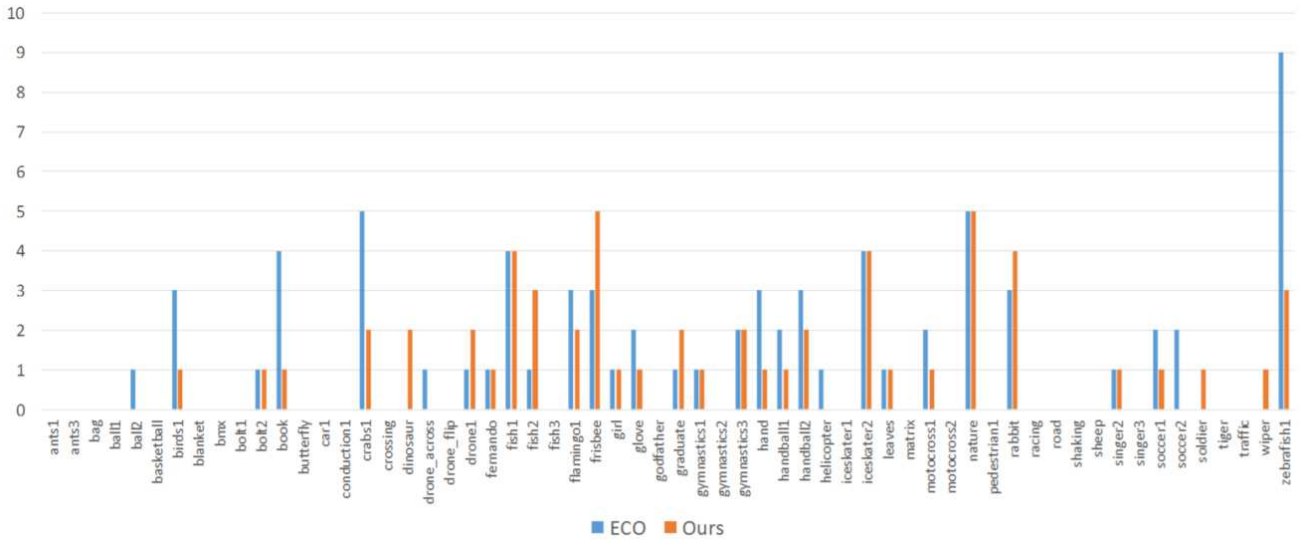

Figure 7: Failures on VOT2017 dataset for 60 videos, comparing our proposed method and ECO.

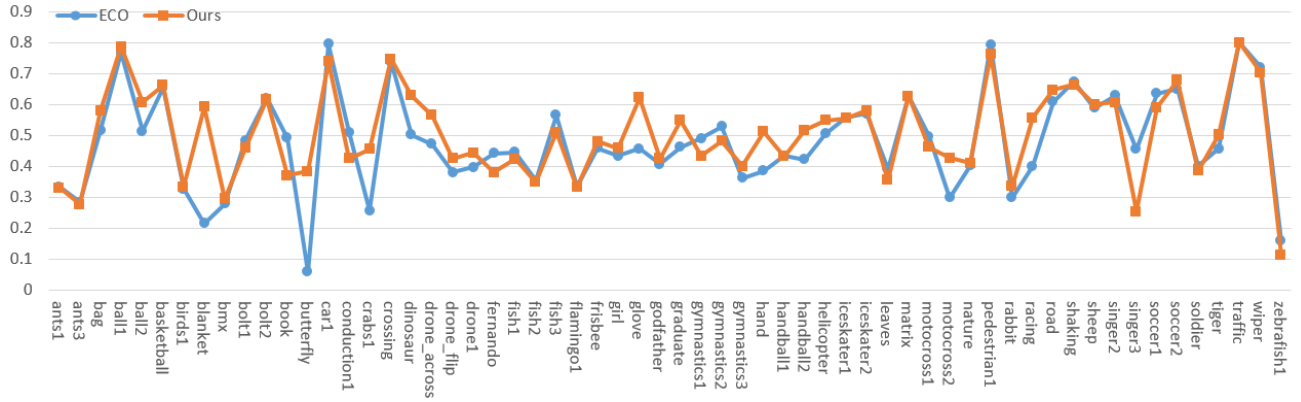

Figure 8: Accuracy results on VOT2017 dataset for 60 videos, comparing our proposed method and ECO.

4.7. Experiment on VOT2017

Finally, we compare our tracker with the ECO [7] tracker on the most recent visual tracking benchmark, VOT2017. In terms of scale variation, the proposed method performs better than the ECO [7] tracker on some challenging sequences. To validate better robustness of our proposed tracker, we analyse the results of challenging sequences in different conditions, such as, camera motion, motion change, illumination change, occlusion and so on. According to Figure 7, compared with the ECO [7] tracker, our tracker performs better when both significant scale and rotation occur (birds1, book) due to the optimal scale estimation. Our trained detector effectively re-detects target objects in the case of tracking failure, *e.g.*, with the heavy occlusion or out-of-view conditions(book, tiger). Besides, our approach follows the target undergoing significant deformation and fast motion (soccer1, soccer2) well. The ECO [7] tracker has 9 tracking failures in sequence zebrafish1 while our tracker only has 3 tracking failures, which demostrates that our tracker effectively alleviates the scale drifting problem.

In addition, Figure 8 demonstrates a detailed per-video comparison, we show the accuracy of our method and the ECO [7] tracker on all the 60 videos contained in VOT2017. It is obvious that our tracker outperforms ECO [7] in the majority of videos.

5. Conclusions

In this paper, we eliminate the hand-crafted features and investigate the impact of multi-resolution deep features for visual tracking. We propose to use convolutional features in the C-COT [13] based framework for visual tracking. We reformulate the final confidence score function by adding the weighted sum operation. When the weights are as-

signed to the convolution responses produced by different feature block, the spatial size and dimensionality of the feature maps extracted from the VGG-M network must be considered. Our results suggest that the combination of the initial convolutional layer and the fifth convolutional layer provides the best performance for visual tracking. Moreover, we use a similar model update strategy approach as in previous methods [7, 13], but we find the 10-layers design is optimal for continuous scale estimation. This subtle difference makes the tracker more robust to gradual scale changes. To validate our proposed tracker, we perform comprehensive experiments on two public benchmarks: VOT2016 and VOT2017. Our experimental results show that our tracker ranks top among the state-of-the-art trackers on VOT2016 and outperforms the ECO [7] tracker on VOT2017.

References

[1] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge (ilsvrc), 2010. *URL http://www. image-net. org/challenges/LSVRC*, 2010. 3

[2] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1401–1409, 2016. 6

[3] A. Bibi, M. Mueller, and B. Ghanem. Target response adaptation for correlation filter tracking. In *European Conference on Computer Vision*, pages 419–433. Springer, 2016. 6

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE, 2010. 2, 4

[5] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *IEEE Transactions on Image Processing*, 25(3):1261–1274, 2016. 2

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 2, 3, 5

[7] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. *arXiv preprint arXiv:1611.09224*, 2016. 1, 2, 4, 6, 7, 8

[8] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014. 1

[9] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Coloring channel representations for visual tracking. In *Scandinavian Conference on Image Analysis*, pages 117–129. Springer, 2015. 2, 3, 5

[10] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1, 4

[11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 58–66, 2015. 1, 6

[12] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015. 1, 4, 6

[13] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016. 1, 2, 3, 6, 7, 8

[14] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, 2014. 1

[15] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964. 4

[16] E. Gundogdu and A. A. Alatan. Good features to correlate for visual tracking. *arXiv preprint arXiv:1704.06326*, 2017. 1

[17] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. *Computer Vision–ECCV 2012*, pages 702–715, 2012. 1

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015. 1, 2, 4

[19] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, and R. Pflugfelder. Luka cehovin, tomas vojir, gustav häger, alan lukežic, and gustavo fernandez. the visual object tracking vot2016 challenge results. In *Proceedings of European Conference on Computer Vision Workshops*, pages 777–823, 2016. 1, 2, 4

[20] 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 *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015. 1, 2

[21] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay,T. Vojir. The visual object tracking vot2014 challenge results. 1, 2

[22] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016. 6

[23] J. Ren, Z. Yu, J. Liu, R. Zhang, W. Sun, J. Pang, X. Chen, and Q. Yan. Robust tracking using region proposal networks. *arXiv preprint arXiv:1705.10447*, 2017. 1

[24] R. Rifkin, G. Yeo, T. Poggio, et al. Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, 190:131–154, 2003. 2

[25] J. R. Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994. 4

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[27] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017. 1

[28] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 1, 2