A NEW FRAMEWORK FOR MULTIPLE DEEP CORRELATION FILTERS BASED OBJECT TRACKING

Yi Liu[1], *Yanjie Liang*[1,3], *Qiangqiang Wu*[1], *Liming Zhang*[2], *Hanzi Wang*[1,*]

[1]Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, Xiamen, China
[2]University of Macau, Macau, China [3]Peng Cheng Laboratory, Shenzhen, China

ABSTRACT

In recent years, Correlation Filter (CF) based tracking methods using Convolutional Neural Network (CNN) features have achieved the state-of-the-art performance for object tracking. However, how to design an efficient deep CF based tracking method has not been well studied in the literature. To address this issue, we first develop a generic framework, which breaks a deep CF based tracking method into five components, including motion model, CNN feature extractor, CF model, CF updater, and location model. According to this framework, we design each component step by step. Then we propose a novel deep CF based tracking method by combining five effective components together. The proposed method outperforms several state-of-the-art tracking methods on two tracking benchmarks. Then the ablative experiments are conducted to study the influence of each component. The results show that the CF model and the CNN feature extractor play the most important roles in a deep CF based tracking method. Moreover, the CF updater, the location model, and the motion model can also improve the performance substantially.

Index Terms— Correlation Filter, Object Tracking, Convolutional Neural Network

1. INTRODUCTION

Object tracking is a task that estimates the location of a given target with its initial state in the first frame of a video [1, 2]. It is one of the fundamental problems in computer vision and plays a vital role in various applications [3, 4, 5], such as robot services, human motion analysis, surveillance, and human-computer interaction. However, object tracking remains a hard problem due to the challenging factors, such as occlusions, illumination variation, and out-of-view. In recent years, Correlation Filter (CF) based methods using the deep features extracted by the Convolutional Neural Network (CNN) [6, 7, 8] (i.e., deep CF based tracking methods) have shown outstanding results on some object tracking benchmarks.

This work was supported by the National Natural Science Foundation of China under Grant No. 61872307, U21A20514, and by the Science and Technology Development Fund of Macao SAR FDCT0060/2021/A. (* Corresponding author: Hanzi Wang, hanzi_wang@163.com.)

Bolme *et al.* propose the first CF based tracking method (i.e., MOSSE) [9], which uses the grey-scale image and extracts single-channel features for high-speed tracking. Later, Henriques *et al.* [10] introduce the kernel technology into CF, and propose the kernelized correlation filter (KCF). Meanwhile, CNNs [11, 12] have been successfully applied to visual tracking due to the ability of the powerful feature representation. Therefore, deep CF based tracking methods have become popular in the past few years [13, 14]. For example, DeepSRDCF [15] exploits the CNN features for visual tracking by the spatially regularized CF to solve the problem of the boundary effects, and this method achieves the state-of-the-art performance. CCOT [6] proposes to train continuous CFs that can efficiently fuse multi-resolution deep features generated by CNNs. ECO [7] is an improved version of CCOT. However, how to design a deep CF based method for effective object tracking is still an unsolved problem.

Our main contributions are summarized as follows. We propose a new framework that divides a deep CF based tracking method into five components. According to this division framework, we design each effective component independently and propose a novel deep CF based tracking method (named as MDCF). The proposed MDCF achieves the state-of-the-art performance. Then we conduct a detailed analysis through ablation studies. The results enlighten several valuable directions for improving tracking performance.

The rest section of this paper is organized as follows. The proposed framework is introduced in Section 2. The experimental results and the analysis are presented in Section 3, followed by the conclusion in Section 4.

2. THE PROPOSED FRAMEWORK

In this section, we give a detailed description of the proposed framework and MDCF. The framework is shown in Fig. 1, the function of each component is summarized as follows:

- **Motion Model:** The motion model generates a searching region, which is a cyclically shifted sample area and contains both background and target regions.
- **CNN Feature Extractor:** The CNN feature extractor outputs the convolutional features extracted from the input frame by using a deep neural network.

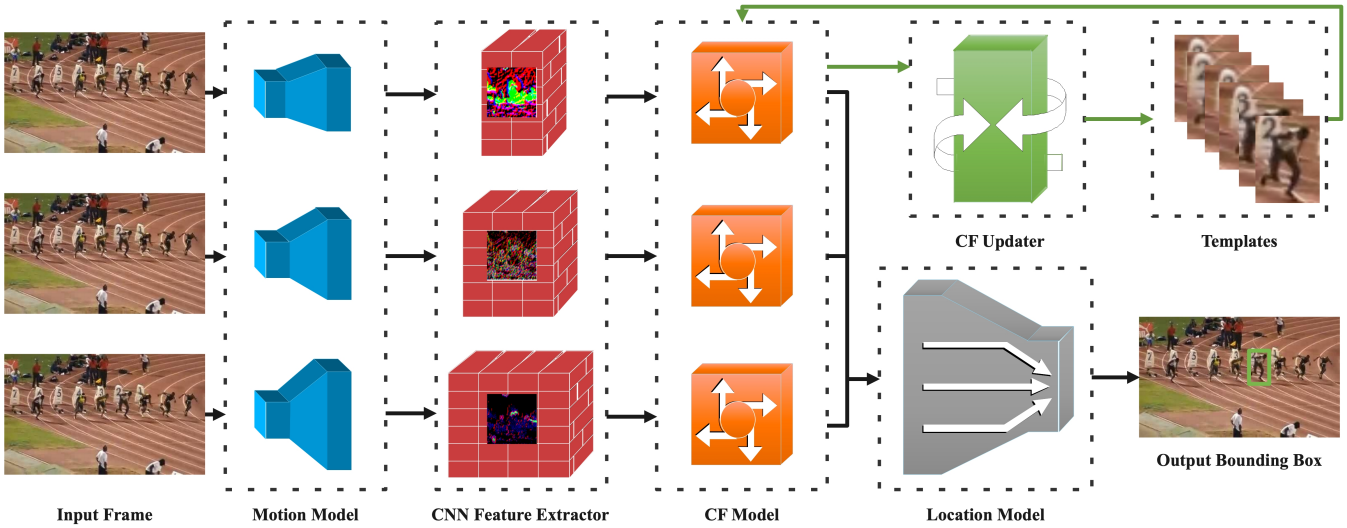

Fig. 1. Pipeline of the framework to design the deep CF based method. The proposed framework consists of five components, including motion model, CNN feature extractor, CF model, CF updater, and location model.

- **CF Model:** The CF model generates the response map corresponding to the input frame by calculating the cross-correlation between the learned CF model and the extracted CNN features of the input frame. There are several different CF models, and each CF model needs to be trained and initialized based on the template cropped from the bounding box in the first frame.
- **CF Updater:** The CF updater controls the strategy of the template updating in the CF model. A good CF updater has to adapt to the appearance changing of the target and remove noisy examples to prevent the tracking method from drifting to the background.
- **Location Model:** When the deep CF based tracking method consists of multiple CFs (i.e., the tracking result consists of multiple response maps), the location model takes all the response maps and uses an ensemble approach to combine them into the final result.

During the tracking process, a typical deep CF based tracking method initializes the CF model in the first frame by using the labeled bounding box. Next, in each of the following frames, the motion model generates a searching region based on the estimated location of the previous frame. Then, the searching region is input to the CNN feature extractor to extract the deep features. Moreover, the CF model generates a response map based on the extracted features. After that, the CF updater determines whether the template set of the CF model needs to be updated or not. Finally, the location of the target is estimated based on the response map by using the location model. Finally, the location model combines all the response maps generated by multiple deep CFs to obtain a more accurate location. Based on the framework, we design each component independently and propose a novel method.

2.1. Motion model

In each frame of a video, the motion model determines the padding size of the deep CF based tracking method. In detail, based on the estimated location of the target in the previous frame, the motion model generates a searching region for the tracking method. When the searching region is too small, the tracking method may lose the target with fast movement. However, the running speed of the tracking method will slow down when the searching region is too large. The existing methods generally set the size of the searching region multiple times (i.e., from 4 to 8) of the target size. To adapt to various scenarios in real-world applications, MDCF uses three different sizes (i.e., 4, 6, 8 times). By using three different searching regions, the CF model in MDCF generates three response maps during tracking. As shown in Fig. 1, the inputs of the followed location model in MDCF are the three response maps.

2.2. CNN Feature Extractor

The CNN feature extractor transforms the video data to compact latent representations. Some deep CF based methods use the features extracted from the fully connected layer. However, these features are redundant for object tracking, which leads to the instability of the tracking methods. Different from these methods, MDCF uses the features extracted from the convolutional layers. In particular, MDCF uses the first three shallow layers of the VGG network [16] to extract deep features. The deep features extracted from the shallow layers contain more shape features, while the deep features extracted from the deep layers contain more semantic features. For fundamental computer vision tasks, shape features are more effective than semantic features.

2.3. CF Model

The CF model is used to generate the response map of a given frame in a video. It is a crucial component of a deep CF based tracking method. MOSSE [9] is the basic CF model, which is initialized by a single frame. This model is a multi-channel CF based on a collection of N training template samples $X = \{x_k\}(k = 1, ..., K)$. In this case, the response map M of the filters $F = \{f_d\}(d = 1, ..., D)$(which consists of D channels) on the sample set X (which consists of N samples) is given by Eq. (1)

$$M_F(X) = X * F, \tag{1}$$

where $*$ denotes the operation of the spatial correlation. MDCF is based on MOOSE, and it benefits of combining BACF [20] and ECO [7]. In detail, MDCF introduces the expandable size of the CF model to MOSSE and initializes it with a large set of negative training samples of several frames. The filters are learned by minimizing the function $S(f)$,

$$\begin{aligned} S(f) = \frac{1}{2}||y - \sum_{d=1}^{D}\sum_{k=1}^{K} f_d * x_k||_{\iota^2}^2 \\ + \frac{\lambda_1}{2}\sum_{d=1}^{D}\sum_{k=1}^{K}||f_d * x_k||_{\iota^2}^2 + \frac{\lambda_2}{2}\sum_{d=1}^{D}||f_d||_{\iota^2}^2, \end{aligned} \tag{2}$$

where f_d and x_k refer to the d-th channel of the filter and the k-th sample of the template, respectively. D and K are the number of channels and the number of samples, respectively. y is the correlation response. λ_1 and λ_2 are the regularization parameters. The optimized CF model in MDCF can solve the problem of the boundary effect efficiently.

2.4. CF Updater

The CF updater determines both the strategy and frequency of template updating. Among deep CF based tracking methods, the CF updater is also a key component to improve the tracking performance. A good CF updater should maintain a tradeoff between collecting the new templates and preventing the tracking method from drifting to the background. The updater in MDCF defines the template as a mixture of Gaussian components, where each component represents a different aspect of the target appearance. Furthermore, the updater in MDCF updates the template at a default learning rate. By using this efficient updater, MDCF can successfully track the target when the appearance of the target significantly changes.

2.5. Location Model

A single deep CF based tracking method can sometimes drift to the background. Especially, the performance of a tracking method can vary considerably even with a small perturbation of parameters. The location model overcomes this limitation by fusing several response maps. We use a factorial hidden Markov model [17] to solve a structured problem and treats the reliability of each response map generated by the CF model as a hidden variable to be inferred. This factorial hidden Markov model makes the tracking performance of the proposed MDCF robust. In MDCF, the location model treats three response maps as inputs for the final location estimation, which improves the tracking performance substantially.

According to the proposed framework, MDCF achieves state-of-the-art tracking performance by putting together the five efficient components mentioned above.

3. EXPERIMENTS

In this section, we introduce the detailed experimental setups and compare the proposed tracking method with several state-of-the-art deep CF based tracking methods on the OTB100 [18] and LaSOT [19] benchmarks for performance evaluation. Furthermore, we conduct the experiments of ablation studies to show how each component of the proposed tracking method affects the tracking performance on the benchmarks.

3.1. Experimental Setups

In this paper, we use two benchmarks for evaluation, i.e., OTB100 and LaSOT. OTB100 is a classic benchmark, which consists of 100 video sequences with annotations. LaSOT is a more challenging benchmark for large-scale single object tracking. This benchmark is one of the biggest datasets. It consists of $1,400$ video sequences with more than $3.5M$ frames in total. The average length of these sequences in LaSOT is more than 2,500 frames, and each sequence involves various challenges deriving from the wild where target objects may disappear and reappear again in the view. Meanwhile, we report the tracking results based on two evaluation metrics (i.e., the distance precision at a threshold of 20 pixels and the area under the curve of overlap success) of one-pass evaluation for performance evaluation. Our experiments are implemented by using MATLAB on a computer with an NVIDIA GTX 1080 GPU and an Intel 6700K 4.0 GHz CPU.

3.2. Experiments

We compare MDCF with the other six state-of-the-art deep CF based tracking methods on the OTB100 [18] benchmark, including ECO [7], DeepSTRCF [22], HSG-DCF[21], DeepSRDCF [13], BACF [20], KCF [10]. The evaluation is based on the precision and success plots on 100 video sequences. As shown in Table 1, MDCF (0.908/0.708) shows better performance than the DeepSTRCF (0.893/0.683), DeepSRDCF (0.825/0.627), BACF (0.801/0.631) and KCF (0.693/0.477). MDCF also shows comparable results to ECO (0.910/0.694) and HSG-DCF(0.881/0.704). Although ECO achieves the highest accuracy in the precision plots, it achieves a lower average overlap score than MDCF in the success plots.

Table 1. The tracking results of seven tracking methods based on different CF models on the OTB100 benchmark. The best results are highlighted by bold.

Method	Precision plots	Success plots
ECO[7]	**0.910**	0.694
DeepSTRCF[22]	0.893	0.683
HSG-DCF[21]	0.881	0.704
DeepSRDCF[13]	0.825	0.627
BACF[20]	0.801	0.631
KCF[10]	0.693	0.477
MDCF (Ours)	0.908	**0.708**

Table 2. The tracking results of seven tracking methods based on different CF models on the LaSOT benchmark. The best results are highlighted by bold.

Method	Precision plots	Success plots
ECO[7]	0.298	0.340
DeepSTRCF[22]	0.292	0.315
HSG-DCF[21]	0.283	0.318
DeepSRDCF[13]	0.227	0.271
BACF[20]	0.239	0.277
KCF[10]	0.184	0.211
MDCF (Ours)	**0.317**	**0.351**

We also compare the proposed MDCF with the above six tracking methods on the LaSOT [19] benchmark. As shown in Table 2, we can see that the proposed MDCF (0.317/0.0.351) achieves the best accuracy among the seven competing state-of-the-art tracking methods on the LaSOT benchmark. MDCF achieves 0.019/0.011 performance improvement over the second-best method in terms of the precision/success plots on the LaSOT benchmark. MDCF achieves high-performance tracking by combining the five efficient components. Overall, the precision and success plots demonstrate that the proposed MDCF performs favorably against the other state-of-the-art deep CF based tracking methods. These results show the effectiveness and efficiency of MDCF.

3.3. Ablation Studies and Detailed Analysis

In this subsection, we evaluate five variants of the proposed tracking method. We take MOSSE as the basic method. Each variant of MDCF is composed of five components, each of which retains one component in MOSSE, and the other four components in our proposed MDCF. The five variants are denoted by MDCF$_{CF}$, MDCF$_{CNN}$, MDCF$_{MM}$, MDCF$_{UP}$, and MDCF$_{LM}$, respectively. The subscript indicates the component used in MOSSE. We report the average center location errors at a threshold of 20 pixels for the precision (P20) and the average overlap score for the success (AOS) on both OTB100 (OTB) and LaSOT (LaS) in Table 3.

Table 3 shows the tracking results of six tracking methods (i.e., MDCF and five variants). As shown in Table 3, we observe that the usage of the basic CF model leads to 0.203/0.187 and 0.114/0.116 performance decline in terms of the P20/AOS score on OTB100 and LaSOT, respectively. Compared with the other components, the CF model plays a more important role in improving the tracking performance. The usage of the HOG features leads to 0.147/0.178 and 0.065/0.066 performance decline in terms of the P20/AOS score on OTB100 and LaSOT, respectively. The results show that a suitable CNN feature extractor brings significant improvement, but not as much as a good CF model. We can also observe that the usage of the basic motion model leads to 0.096/0.171 and 0.029/0.039 performance decline in terms of the P20/AOS score on OTB100 and LaSOT, respectively. The updater in MDCF$_{UP}$ updates the template on every frame. In Table 3, we also observe that the usage of the basic updater leads to 0.067/0.088 and 0.021/0.014 performance decline in terms of the P20/AOS score on OTB100 and LaSOT, respectively. And MDCF$_{LM}$ uses only one response map to locate the target. As shown in Table 3, we observe that the usage of the basic location model leads to 0.031/0.029 and 0.016/0.008 performance decline in terms of the P20/AOS score on OTB100 and LaSOT, respectively. In general, the improvement of the motion model, the CF updater, and the location model will slightly help the overall improvement.

Table 3. The tracking results of six tracking methods on two different datasets. The best results are highlighted by bold.

Method	P20(OTB)	AOS(OTB)	P20(LaS)	AOS(LaS)
MDCF$_{MM}$	0.812	0.537	0.288	0.312
MDCF$_{CNN}$	0.761	0.530	0.252	0.285
MDCF$_{CF}$	0.705	0.521	0.203	0.235
MDCF$_{UP}$	0.841	0.620	0.296	0.337
MDCF$_{LM}$	0.877	0.679	0.301	0.343
MDCF	**0.908**	**0.708**	**0.317**	**0.351**

4. CONCLUSION

In this paper, we develop a new framework that divides a deep CF based tracking method into five components. According to this framework, we design each effective component step by step and propose a novel deep CF based tracking method, namely MDCF. By improving the performance of each component, the overall tracking performance can be improved. MDCF outperforms several other state-of-the-art tracking methods on two tracking benchmarks. Then we perform ablative experiments to study the influence of each component. Based on the analysis of the results, the CF model and the CNN feature extractor play the most important roles in the deep CF based tracking methods, while the motion model, the CF updater, and the location model also slightly help the overall improvement. This paper enlightens several valuable directions for further study, including developing effective features, CF updater, and location model.

5. REFERENCES

[1] R. Pauwels, E. Tsiligianni, N. Deligiannis, "HCGM-Net: A deep unfolding network for financial index tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 3910–3914

[2] B. I. Ahmad, P. M. Langdonn, S. J. Godsill, "A bayesian framework for intent prediction in object tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 8439–8443

[3] G. Y. Gopal, M. A. Amer, "Dynamic channel pruning for correlation filter based object tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 5700–5704.

[4] H. Z. Chen, X. F. Xing, X. M. Xu, "Deep regression tracking with shrinkage loss," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 2153-2157.

[5] J. L. Yang, X. P. Chen, H. Y. Hen, J. J. Liu, "Adaptive visual target tracking based on label consistent k-svd sparse coding and kernel particle filter," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 1633-1637.

[6] M. Danelljan, A. Robinson, F. S. Khan, M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, 2016, pp. 472–488.

[7] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg, "ECO: Efficient convolution operators for tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6931–6939.

[8] X. Zhu, X. Wu, T. Xu, Z. Feng, J. Kittler, "Robust visual object tracking via adaptive attribute-aware discriminative correlation filters," *IEEE Transactions on Multimedia*, Early Access Article, 2021.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550.

[10] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, 2015, pp. 583–596.

[11] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5000–5008.

[12] S. Feng, K. Hu, E. Fan, L. Zhao, C. Wu, "Kalman filter for spatial-temporal regularized correlation filters," *IEEE Transactions on Image Processing*, vol. 30, no. 1, 2021, pp. 3263–3278.

[13] M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4310–4318.

[14] R. Liu, Q. Chen, Y. Yao, X. Fan, Z. Luo, "Location-aware and regularization-adaptive correlation filters for robust visual tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, 2021, pp. 2430–2442.

[15] M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *IEEE International Conference on Computer Vision Workshops*, 2015, pp. 621–629.

[16] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[17] N. Wang, D. Y. Yeung, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *International Conference on Machine Learning*, 2014, pp. 1107–1115.

[18] Y. Wu, J. Lim, M. H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, 2015, pp. 1834–1848.

[19] H. Fan, L. Lin et al., "LaSOT: A high-quality benchmark for large-scale single object tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5369–5378.

[20] H. K. Galoogahi, A. Fagg, S. Lucey, "Learning background-aware correlation filters for visual tracking," in *IEEE International Conference on Computer Vision*, 2017, pp. 1144–1152.

[21] S. Javed, A. Mahmood, J. Dias, L. Seneviratne, N. Werghi, "Hierarchical spatiotemporal graph regularized discriminative correlation filter for visual object tracking," *IEEE Transactions on Cybernetics*, Early Access Article, 2021.

[22] F. Li, C. Tian, W. Zuo, L. Zhang, M. H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4904–4913.