End-to-End Feature Integration for Correlation Filter Tracking With Channel Attention

Dongdong Li, Gongjian Wen, *Member, IEEE*, Yangliu Kuai, and Fatih Porikli, *Fellow, IEEE*

***Abstract*—Recently, the performance advancement of discriminative correlation filter (DCF) based trackers is predominantly driven by the use of deep convolutional features. As convolutional features from multiple layers capture different target information, existing works integrate hierarchical convolutional features to enhance target representation. However, these works separate feature integration from DCF learning and hardly benefit from end-to-end training. In this letter, we incorporates feature integration and DCF learning in a unified convolutional neural network. This network reformulates feature integration as a differential module that concatenates features from the shallow and deep layers. A channel attention mechanism is introduced to adaptively impose channel-wise weight on the integrated features. Experimental results on OTB100 and UAV123 demonstrate that our method achieves significant performance improvement while running in real-time.**

***Index Terms*—Correlation filters, feature integration, channel attention.**

I. Introduction

VISUAL tracking plays a critical role in many computer vision applications such as video surveillance [1], augmented reality [2] and human-computer interaction [3]. It's the task of estimating the spatial trajectory of a specified target given its initial state (generally an axis-aligned rectangle) in a video sequence. Despite significant progress in recent years, robust tracking under complicated scenarios is still challenging due to illumination change, self-deformation, partial occlusion, fast motion, background clutter and so on.

Recently, Discriminative Correlation Filters (DCF) have achieved enormous popularity in the tracking community due to high computational efficiency and fair robustness. Exploiting the circular structure, DCF transform computationally consuming spatial correlation into efficient element-wise operation in the Fourier domain and achieve extremely high tracking speed. The periodic assumption enables efficient training and detection with the Fast Fourier Transform (FFT). Based on the standard DCF formulation, different variants of correlation filters have been proposed to boost tracking performance using multi-dimensional features [4], robust scale estimation [5], nonlinear kernels [6], context learning [7], long-term memory components [8], complementary cues [9], target adaptation [10] and spatial regularization [11].

To achieve further performance improvement, an emerging trend is to use deep features for their strong discriminative power. Danelljan *et al.* [12] first introduce shallow convolutional features into the SRDCF [11] tracking framework. These shallow convolutional features capture discriminative spatial details for precise target localization but are not robust to significant target appearance variation. Later, HCF [13] and CCOT [14] exploit hierarchical convolutional features to enhance target representation. Different from shallow convolutional features, convolutional features from the deep layers capture abstract semantics which are robust to target deformation, illumination change and background clutter. With both shallow and deep convolutional features, HCF and CCOT make a good balance between localization accuracy and robustness.

While the above trackers with hierarchical features work well, they hardly benefit from end-to-end training due to the following two reasons. First, the exploited convolutional features are extracted from convolutional neural networks trained for a different task, such as image classification. Second, the feature integration strategy are intuitively designed and need careful hyper-parameter tunning which if not performed correctly can lead to poor tracking performance.

Recently, the tracking community leads a fashion of end-to-end training for visual tracking. Following this trend, we propose to incorporate feature extraction, feature integration and DCF learning into a unified convolutional neural network. The overall network architecture consists of a feature extraction sub-network, a feature integration module and a correlation filter layer. The feature extraction sub-network generates hierarchical convolutional features tightly coupled to correlation tracking. The feature integration module is fully differential and concatenates convolutional features extracted from the shallow and deep layers in the feature extraction sub-network. The correlation filter layer computes a standard correlation filter template from the integrated features generated from the feature integration module.

Overall, our main contributions are as follows:

1) We develop an end-to-end feature integration module for visual tracking. This module is very flexible and can be

Manuscript received August 16, 2018; accepted October 9, 2018. Date of publication October 22, 2018; date of current version October 29, 2018. This work was supported in part by the National Natural Science Foundation of China under project 41601487 and in part by the Australian Research Council's Discovery Projects funding scheme under project DP150104645. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joao Paulo Paulo Papa. *(Corresponding author: Dongdong Li.)*

D. Li, G. Wen, and Y. Kuai are with the College of Electronic Science, National University of Defense Technology, Changsha 410073, China (e-mail: moqimubai@sina.cn; wengongjian@sina.com; kuaiyangliunudt@163.com).

F. Porikli is with the Research School of Engineering, Australian National University, Canberra ACT 0200, Australia (e-mail: 745699482@qq.com).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2018.2877008

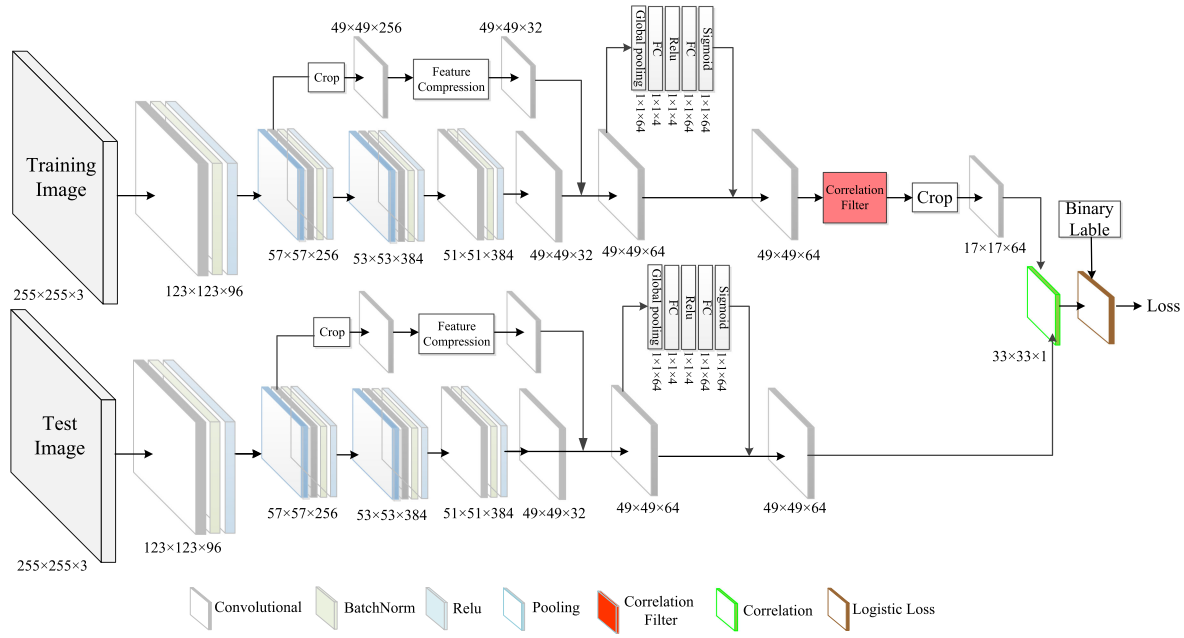

Fig. 1. The overall network architecture of our approach.

integrated into other tracking frameworks to enhance feature representation.

2) To reflect the channel-wise quality of the integrated features, we adopt a channel attention mechanism to impose different weight on each channel of the integrated features.
3) Experiments on the OTB100 [15] and UAV123 [16] datasets demonstrate that our method achieves a remarkable performance improvement while running in real-time.

II. Our Approach

In this section, we give a detailed description of our Feature Integrated Correlation Filter Network (FICFNet).

A. Overall Network Architecture

The overall network architecture of FICFNet is shown in Fig. 1. Our FICFNet follows the two-branch parallel-connected network architecture as in CFNet [17]. Each branch includes a feature extraction sub-network with five group of convolutional layers. Following the feature extraction sub-network is a feature integration module which concatenates convolutional features from the second convolutional layer with those from the five convolutional layer. The integrated convolutional features generated in the training branch are fed into the correlation filter layer to derive a correlation template. This template and the integrated convolutional features in the test branch are then fed to the spatial cross correlation layer for correlation analysis.

B. Feature Extraction Sub-Network

The base architecture that we adopt for the feature extraction sub-network resembles the convolutional stage of the network of AlexNet [18]. The dimensions of the parameters and activations of the feature extraction sub-network are given in Table I. The stride of the final representation is four.

TABLE I
Architecture of the Feature Extraction Sub-Network

Layer	Support	Chan.map	Stride	feature map.
input				255×255×3
conv1	11× 11	96×3	2	123×123× 96
pool1	3×3		2	123×123× 96
conv2	5× 5	256×48	1	57×57× 256
pool2	3×3		1	55×55× 256
conv3	3× 3	384×128	1	53×53× 384
conv4	3× 3	384×192	1	51×51× 384
conv5	3× 3	32×192	1	49×49× 32

C. Feature Integration Module

1) Feature Concatenating: To achieve feature integration, we concatenate the activations from the conv2 and conv5 layers. Fig. 2 shows the activations from these two layers. As shown in Fig. 2, the shallow convolutional features from the conv2 layer capture spatial details (e.g., edge, corner) of the target appearance, which is suitable for precise localization. On contrast, the convolutional features from the conv5 layer capture abstract semantics which are robust against significant appearance change. As shown in Fig. 1, the shallow and deep convolutional features are different in both size and channel. To ensure the same feature size and comparable channels for feature concatenating, shallow convolutional features are fed into a crop layer and a feature compression layer. The crop layer crops the margin of the shallow convolutional features and reduces the feature map size from

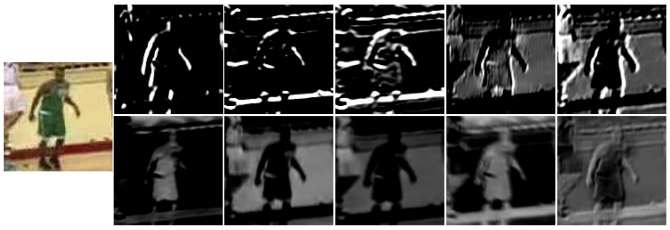

Fig. 2. Visualization of the shallow (first row) and deep (second row) convolutional features extracted from a sample patch (left column) taken from the **Basketball** sequence in the OTB100 dataset.

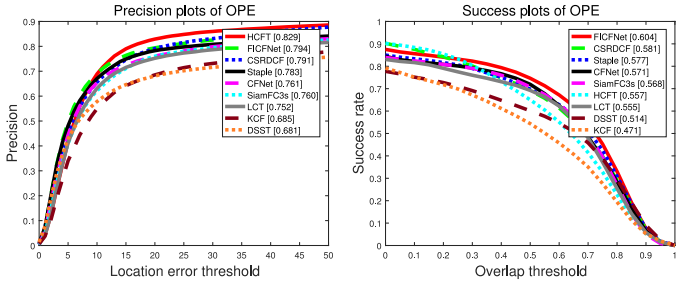

Fig. 3. Precision plots and Success plots for trackers in comparison on OTB100.

$57 \times 57 \times 256$ to $49 \times 49 \times 256$. However, the cropped shallow convolution features still have a much higher channel than the deep convolution features. To prevent the shallow convolutional features from dominating the feature integration, a feature compression layer is designed to perform feature dimensionality reduction. The feature compression layer is a convolution layer ($1 \times 1 \times 256 \times 32$) which reduces the channel of the shallow convolutional features from 256 to 32.

2) Channel Attention Mechanism: While tracking a target from visual input, most of the time the target moves smoothly and its appearance changes little or slowly. Shallow convolutional features usually work fine and contribute more to target localization for such easy cases [12]. By contrast, hard cases typically require to be handled by deep convolutional features to achieve robustness against significant appearance variation [13]. In this sense, we introduce a channel attention mechanism to adaptively adjust the contribution of each channel in the shallow and deep convolutional features. Channel attention indicates the different weights imposed on different feature channels. Our channel attention mechanism adaptively re-calibrates the channel importance with channel-wise weights generated from the input integrated features. As shown in Fig. 1, the integrated features are first passed through a global pooling layer to produce a channel-wise descriptor. To fully capture channel-wise dependencies, a simple gating mechanism with a sigmoid activation is employed with two fully connected (FC) layers around it to limit model complexity and aid generalization. The first FC layer is a dimensionality-reduction layer with a reduction factor 16 while the second FC layer is a dimensionality-increasing layer. It's worth noting that our channel attention mechanism is similar to the squeeze-and-excitation (SE) architecture in [19]. The activations of the SE block act as channel weights for the integrated features. In this regard, our channel attention mechanism intrinsically introduces dynamics conditioned on the input, helping to improve feature integration.

D. Offline Training

As shown in Fig. 1, the input images of our network have a size of $255 \times 255 \times 3$. These input images are fed into each branch of the network to generate the integrated convolutional features which are of size $49 \times 49 \times 64$. Our network is end-to-end trained from scratch on the video dataset from ILSVRC ImageNet Video dataset [20]. We apply the stochastic gradient descent (SGD) solver using mini-batches of 8 during offline training over 50 epochs.

E. Online Tracking

After offline training, the trained network is used to perform online tracking. The exemplar image and search image are cropped around the target and fed into the network to generate the score map. With the labor of GPU, the score map can be efficiently computed by fast CNN forward propagation. The target location is estimated by finding the maxima on the score map. Scale variation is estimated by processing the search image at several scales with a fixed aspect ratio. To achieve robust online tracking, the correlation filter template derived in the exemplar branch is updated using a rolling average with a fixed learning rate.

III. Experiments

All the experiments are run on two challenging tracking datasets: OTB100 [15] and UAV123 [16]. OTB100 is a popular tracking dataset containing 100 fully annotated challenging videos. UAV123 is a recently introduced aerial video benchmark for low altitude UAV target tracking, which contains 123 aerial videos.

A. Implementation Details

FICFNet is implemented in Matlab using Matconvnet [21] on a single NVIDIA GeForce GTX Titan X and an Intel Core i7 CPU at 4.0 GHz. During online tracking, we search for the target over three scales $1.0575^{\{-1,0,1\}}$ and update the correlation filter template by linear interpolation with a factor of 0.005.

B. Experiments on OTB100

1) Overall Quantitative Performance: We provide a comparison of FICFNet with 8 trackers from the literature: Staple [9], CSRDCF [22], LCT [8], DSST [23] and KCF [6], SiamFC3s [24], CFNet [17] and HCFT [13]. Following the protocol in [15], we show the precision and success plots in one-pass evaluation of all trackers. As shown in Fig. 3, our FICFNet achieves the second rank in the precision plots and the first rank in the success plots. Compared with CFNet, FICFNet achieves an absolute performance improvement of 3.3% in both the precision and success plots. It's worth noting that our FICFNet achieves a real-time frame-rate of 28 fps on GPU on the OTB100 dataset.

2) Attribute Based Performance: All the videos in OTB100 are annotated with 11 different attributes, namely: illumination variation (IV), scale variation (SV), occlusion (OCC),

TABLE II
SUCCESS RATES OF THE TRACKERS IN COMPARISON ON 11 ATTRIBUTES OF THE OTB100 DATASET. THE FIRST AND SECOND BEST METHODS ARE SHOWN IN COLOR

Attribute	**FICFNet**	Staple	CSRDCF	LCT	DSST	KCF	SiamFC3s	CFNet	HCFT
LR	0.513	0.399	0.430	0.399	0.382	0.290	0.677	0.525	0.388
IPR	0.583	0.547	0.538	0.545	0.499	0.458	0.564	0.559	0.550
OPR	0.575	0.550	0.542	0.537	0.489	0.457	0.545	0.532	0.524
SV	0.558	0.520	0.522	0.494	0.477	0.394	0.556	0.550	0.483
OCC	0.572	0.562	0.516	0.503	0.471	0.448	0.527	0.509	0.511
DEF	0.584	0.577	0.557	0.507	0.443	0.455	0.510	0.500	0.525
BC	0.624	0.554	0.549	0.536	0.517	0.490	0.520	0.554	0.575
IV	0.594	0.576	0.582	0.538	0.530	0.463	0.566	0.519	0.528
MB	0.543	0.534	0.562	0.519	0.465	0.454	0.544	0.555	0.575
FM	0.547	0.550	0.576	0.560	0.468	0.465	0.570	0.576	0.578
OV	0.509	0.468	0.474	0.429	0.383	0.387	0.500	0.483	0.461

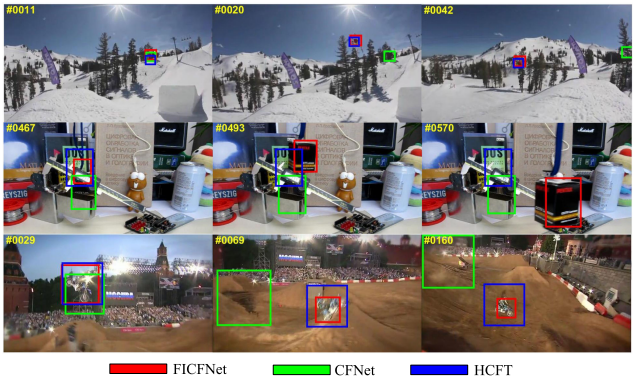

Fig. 4. Tracking screenshots of AdaCFNet, CFNet and HCFT on **Skiing**, **Box** and **MotorRolling** from the OTB100 dataset. Best viewed in color.

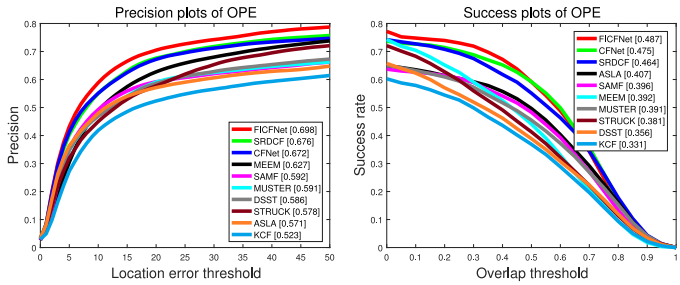

Fig. 5. Precision plots and success plots for trackers in comparison on UAV123.

deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC) and low resolution (LR). Here, we perform an attribute based analysis on the OTB100 dataset. As shown in Table II, our FICFNet achieves the best performance on 8 out of 11 attributes. This can be attributed to the feature integration module in FICFNet, which increases the tracking robustness under challenging scenarios.

3) Qualitative Performance: Due to page limitation, we select three challenging sequences, *Skiing*, *Box* and *MotorRolling*, from OTB100 to visually demonstrate the superiority of our FICFNet against CFNet and HCFT. Among two compared trackers, CFNet is the baseline tracker of FICFNet without feature integration. HCFT employs hierarchical convolutional features but these features are not trained in an end-to-end fashion. As shown in Fig. 4, the targets undergo fast motion in *Skiing*, heavy occlusion in *Box* and severe deformation in *MotorRolling*. Our FICFNet tracker is able to persistently track the target over all the sequences, which demonstrates the effectiveness of end-to-end feature integration to enhance target representation.

C. Experiments on UAV123

We also evaluate our FICFNet on the UAV123 dataset. Different from OTB100, UAV123 contains both realistic and simulated sequences from an aerial viewpoint. Fig. 5 shows the comparative results achieved by FICFNet, CFNet and 8 state-of-the-art trackers included in [16]. Our FICFNet achieves the best performance in both the precision plot (69.8%) and success plot (48.7%).

IV. CONCLUSION

In this letter, we propose an end-to-end framework for feature integration in correlation filter tracking. Convolutional features from the shallow and deep layers are concatenated to enhance the feature representation. We further introduce a channel attention mechanism to adaptively impose channel-wise weight on the integrated features. Experiments on OTB100 and UAV123 show that our method achieves favorable tracking performance while running in real-time. Our future work will focus on substituting the feature extraction sub-network with lightweight architectures (e.g., SqueezeNet [25] and ShuffleNet [26]) for higher frame-rates.

REFERENCES

[1] K. Lee and J. Hwang, "On-road pedestrian tracking across multiple driving recorders," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1429–1438, Sep. 2015.

[2] T. Guan and C. Wang, "Registration based on scene recognition and natural features tracking techniques for wide-area augmented reality systems," *IEEE Trans. Multimedia*, vol. 11, no. 8, pp. 1393–1406, Dec. 2009.

[3] G. Wu and W. Kang, "Vision-based fingertip tracking utilizing curvature points clustering and hash model representation," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1730–1741, Aug. 2017.

[4] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2016.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[7] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1387–1395.

[8] C. Ma, X. Yang, C. Zhang, and M. H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5388–5396.

[9] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.

[10] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 11–14, 2016, pp. 419–433.

[11] M. Danelljan, G. Hger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4310–4318.

[12] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Santiago, Chile, Dec. 7–13, 2015, pp. 621–629.

[13] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.

[14] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 11–14, 2016, pp. 472–488.

[15] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[16] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 11–14, 2016, pp. 445–461.

[17] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5000–5008.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7132–7141.

[20] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[21] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia.*, 2015, pp. 689–692.

[22] A. Lukezic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4847–4856.

[23] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Brit. Mach. Vis. Conf.*, 2014.

[24] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 8–10 and 15/16, 2016, pp. 850–865.

[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," 2016, arXiv:1602.07360.

[26] N. Ma, X. Zhang, H. Zheng, and J. Sun, "Shufflenet V2: Practical guidelines for efficient CNN architecture design," in *ECCV*, 2018, pp. 122–138.