2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Multi-layer CNN Features Aggregation for Real-time Visual Tracking

Lijia zhang, Yanmei Dong, and Yuwei Wu
Beijing Laboratory of Intelligent Information Technology, School of Computer Science
Beijing Institute of Technology, Beijing 100081, P.R. China
Email: {zhanglijia, dongyanmei, and wuyuwei}@bit.edu.cn

***Abstract*—In this paper, we propose a novel convolutional neural network (CNN) based tracking framework, which aggregates multiple CNN features from different layers into a robust representation and realizes real-time tracking. We found that some feature maps have interference for effectively representing objects. Instead of using original features, we build an end-to-end feature aggregation network (FAN) which suppresses the noisy feature maps of CNN layers. The feature significantly benefits to represent objects with both coarse semantic information and fine details. The FAN, as a light-weight network, can run at real-time. The highlighted region of feature maps obtained from the FAN is the tracking result. Our method performs at a real-time speed of 24fps while maintaining a promising accuracy compared with state-of-the-art methods on existing tracking benchmarks.**

I. Introduction

Visual tracking has attracted considerable attention in numerous applications, including human-computer interaction, visual navigation, and video surveillance [1][2]. It is challenging due to the cluttered background, rotation, varying illumination, pose variation, etc. Many works resort to robust representations such as histograms [3], sparse features [4], ,scale invariant feature transform (SIFT) [5] and multi-features [6] to design an effective tracker. These hand-crafted features, however, cannot capture rich semantic information. Recently, features extracted from the Convolutional Neural Network (CNN) have achieved state-of-the-art results on visual tracking [7] [8] [9] [10] [11].

Most deep trackers either achieve real-time tracking but lack robustness or obtain high accuracy with low speed. Under the premise tracking in real-time, it is necessary to improve the robustness. In general, a robust representation is a crucial component for visual tracking. With robust features that are invariant to varying illumination, pose variation, etc., a good performance can be realized using a simple visual tracker. To this end, we aim at aggregating CNN features into a robust representation for visual tracking.

We observe that different feature maps contain different information such as background, foreground, and boundary, as shown in Fig. 1. Some feature maps are useless to the object representation and they should not be considered [13]. Based on this observation, we consider the intra-feature relationship and inter-feature relationship to aggregate CNN features. We define the intra-feature as the relationship between feature maps in the same layer, and the inter-feature as the relationship between features of different layers. In the inter-feature relationship, features in higher layers have more semantic information for classification. Features in lower layers have more fine details to locate the object precisely. In the intra-feature relationship, feature maps often contain miscellaneous information, and some of them are noisy for object representation.

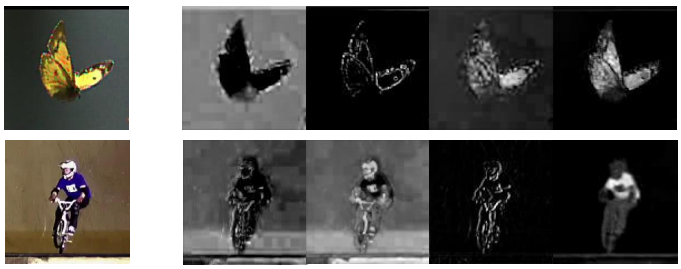

Fig. 1. Motivation of reducing noisy feature maps. (a) the input images. (b) feature maps random selected form pool2 layer in VGG [12]. The brighter part of the feature maps represents higher response value. The larger object response value denotes the better tracking representation. In (b), the latter two feature maps are more useful for tracking.

In this paper, we propose to aggregate CNN features from different layers to generate a robust representation by a top-down pathway with lateral connections. We suppress all noisy feature maps properly by a bottle net architecture to weight feature maps dynamically. We generate the robust representation by a network termed as the feature aggregation network (FAN). The output of FAN contains both semantic information and details using end-to-end training. The highlighted region in the feature map is the location of tracking objects. In addition, the FAN has few parameters and thus our tracker is able to run at a real-time speed with competitive accuracy.

The contributions of this paper are two-fold.

(1) We propose a novel feature aggregation strategy which utilizes convolutional features from multiple levels and reduces noisy feature maps of CNN layers to generate a robust representation for visual tracking.

(2) We construct a light-weight feature aggregation network (FAN) for generating a strong object representation. The FAN combines the coarse semantic features in high-level layers with the fine details in low-level layers.

978-1-5386-3788-3/18/$31.00 ©2018 IEEE

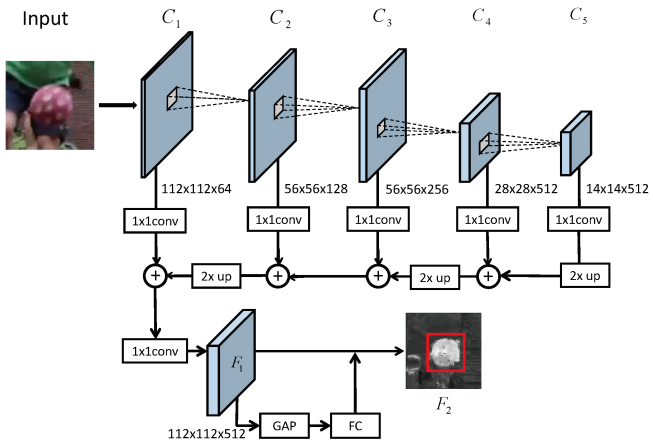

Fig. 2. Architecture of our framework which consists of top-down path with lateral connections. All features C_1, C_2, C_3, C_4, and C_5 pass through a 1×1 convolutional layer to reduce the noisy feature maps. The features from higher-level layers are upsampled by a factor of 2 corresponding to the size of adjacent lower-level layers, except C_3 since it has the same size as C_2. To suppress noisy feature maps further, F_1 weights all feature maps through a global average pooling (GAP) layer followed by a fully connection (FC) layer to generate the final output feature F_2. The highlight region of F_2 is the position of the tracking object.

II. Related Work

CNN based trackers. Several recent works have sought to train CNN for object tracking like [8][14][15]. However, They are difficult to be applied to time-critical systems which retrain network with thousand parameters at online tracking.

To achieve high-speed tracking, many approaches based on the CNN have been proposed. Held *et al.*[16] proposed a deep regression network that can calculate object position directly and it achieved 100fps. Bertinetto *et al.*[17] presented a fully convolutional Siamese Network to calculate the similarity between the search area and the object template. However, the aforementioned methods compared invariance template with the following images. Valmadre *et al.*[7] embedded the kernelized correlation filter into the siamese network to update the object template. The feature in [7] is simple so that it cannot get a robust result in complex situations. Different from these methods mentioned above, we aim at generating a robust representation to represent the object and construct a light-weight network that can update online and realize real-time tracking.

Multiple CNN features aggregation methods. Using multiple CNN features has been proposed for many visual tasks. Existing tracking methods [18][19] have been developed via combining weak trackers based on features from different CNN layers. Wang *et al.*[20] proposed that features from high-level layers contain semantic information, and features from low-level layers contain discriminative information. Long *et al.* [21] predicted object instance segmentation using each feature independently and averaged all of them as the result. Lin *et al.* [22] used top-down framework and regarded the output of multiple layers as a feature pyramid which can detect objects using all of these features. In this work, we utilize a top-down pathway and lateral connection to integrate multiple features into a strong one, and reduce noisy feature maps in each layer.

Feature maps analysis methods. Most existing methods about analyzing feature maps information have been proposed on visual semantic segmentation and object classification. Zhou *et al.* [23] exploited the global average pooling (GAP) layer to describe class activation maps (CAM), and calculated the weights corresponding to multi-classes by a fully connection (FC) layer. The generation of their method is an aggregated class CAM which highlights the class-specific discriminative regions. Hu *et al.* [13] proposed that the global average pooling (GAP) layer followed by a fully connection (FC) layer has the ability to squeeze information of each feature map and counting each channel's weight. In this work, we integrate a bottle net architecture containing GAP and the FC layer into our framework to aggregate CNN features from different layers.

III. Feature Aggregation Network

Our feature aggregation network, the FAN, aims at aggregating multiple CNN features to obtain a robust representation by the top-down pathway with skip connections. The FAN can remove noisy feature maps to enhance the robustness of the object representation.

A. Network Architecture

The network consists of three parts: feature extraction, inter-feature aggregation and intra-feature aggregation, as shown in Fig. 2. We employ the features from five layers of VGG-Net [12] to encode object appearances, including pool1, pool2, conv3-4, conv4-4 and conv5-4, symbolized as C_1, C_2, C_3, C_4 and C_5.

For inter-feature aggregation, the top-down pathway with lateral connection is utilized in the FAN to integrate multiple layer CNN features, including semantic information and details. The top-down pathway enlarges the high-level features to the same size as the former layer features by the nearest neighbor upsampling. These features are then passed through a 1×1 convolutional layer to reduce noisy feature maps. Lateral connections are utilized to merge these higher level features with the lower level ones via element-wise addition. For convenience, we denote the corresponding features generated from each lateral connection as P_1, P_2, P_3, P_4, P_5. To increase the information of the extracted feature, we employ another 1×1 convolutional kernel to increase the channel number of P_1, and the generated feature is symbolized as F_1.

Intra-feature aggregation is realized by a global average pooling layer (GAP) and a fully connected layer (FC). The GAP is a channel descriptor that includes the global distribution of corresponding response for each channel-wise feature. The GAP is an FC layer which governs the weight of each channel. The final feature F_2 is generated by weighing all the channels of F_1.

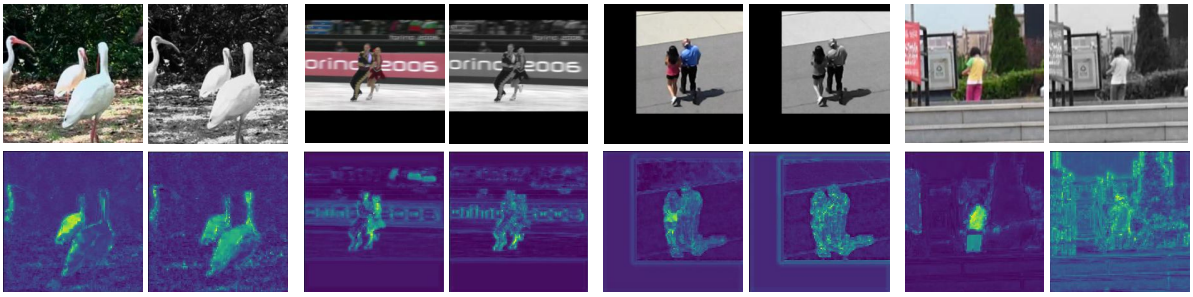

Fig. 3. Comparisons of the features of color videos with grayscale videos.

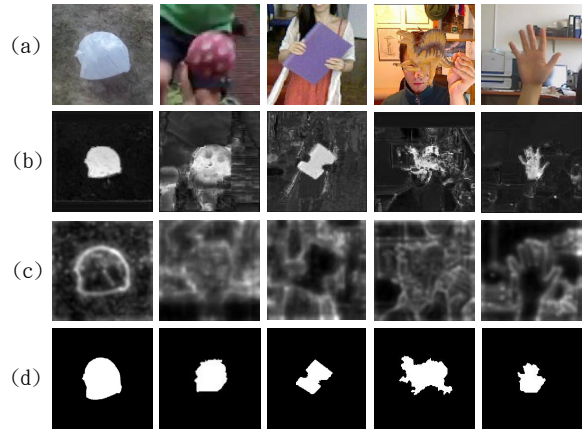

Fig. 4. Visualization of feature maps. (a) The input images with objects at center. (b) The feature maps generated from our network which can highlight the object accurately. (c) The feature weighted average of all features extracted from VGG-Net, which only capture edge information and can not be used to locate object accurately when the background is complex. (d) The soft ground truth.

B. Training Method

We adopt the square loss function during our training. It is designed between the generated feature F_2 and the binary ground truth map $\hat{G}$ shown in Fig. 4 (d), and takes the form as

$$\mathcal{L} = \left\| F_2 - \hat{G} \right\|^2 + \lambda_1 \left\| P \right\|^2, \tag{1}$$

where λ_1 denotes the regularization parameter, and P is the parameter of the FAN. We train the network using stochastic gradient descent (SGD).

We employ the training set of VOT [24] as our training set, noting that we remove the videos which also appear in the object tracking benchmark (OTB) [25]. To improve the generalization ability of our network, the training set contains both the original videos and the corresponding converted grayscale videos.

IV. Online Tracking using FAN

At the beginning of online tracking, we crop an image patch from the initial position to fine-tune the pre-trained FAN. The new position of the target is determined by the position of the largest response in F_2. Since the object appearance changes significantly during tracking, we design two online update methods for the model update, including the periodic update and conditional update. The periodic update is triggered every ten frames and the conditional update is only utilized when the response value is lower than a specific threshold.

We find that the pre-trained FAN can capture more discriminative information from color videos than grayscale videos. To validate this finding, we convert the color videos into grayscale, and compare the learned features. The visualized features are shown in Fig. 3. We found that the FAN feature extracted from the color video can carry more discriminative information than the grayscale one. Therefore, the generated feature from the grayscale videos may not localize object directly. To settle this problem, we combine the kernelized correlation filter (KCF) [26] after generating the FAN feature when the input is grayscale video.

The correlation filter W of KCF is learned by solving the following minimization problem:

$$W^* = \arg\min_W \left\| Wx - y \right\|^2 + \lambda_2 \left\| W \right\|_2^2, \tag{2}$$

where x indicates the output feature of FAN, and y denotes a gaussian shape label with zero mean and standard deviation proportional to the target size [26]. λ_2 ($\lambda_2 \geq 0$) is a regularization parameter and the inner product is calculated by a linear kernel in the Hilbert space. The learned filter can be written as

$$W = \frac{Y \odot \bar{X}}{X \odot \bar{X} + \lambda_2}, \tag{3}$$

in which the operator $\odot$ denotes the Hadamard (element-wise) product. X and Y are the Fourier transformation form of x and y, respectively. $\bar{X}$ is the complex-conjugate form of X. Denoting $Y \odot \bar{X}$ as A, and $X \odot \bar{X}$ as B. The correlation filter is updated through

$$W = \frac{A}{B + \lambda_2}, \tag{4a}$$

$$A_{t+1} = (1 - \eta) A_t + \eta Y \odot \bar{X}, \tag{4b}$$

$$B_{t+1} = (1 - \eta) B_t + \eta X \odot \bar{X}, \tag{4c}$$

where η is the learning rate and is assigned as 0.01. The whole tracking algorithm is presented in Algorithm 1.

Algorithm 1 Online tracking algorithm

Input: Pre-trained parameter P, Initial object location c_1, usekcf=0.

Output: The estimated object location c_t

Crop an image patch from c_1 and generate a soft ground truth map gt_1.
Update P using c_1 and gt_1.
Calculate the loss $\mathcal{L}$ and find the max score s_1.
if $\mathcal{L} > 0.5$ **then**
    usekcf=1
end if
repeat
    Crop object candidate S_t
    Extract the corresponding feature f_t
    if usekcf=1 **then**
        Determine c_t using KCF.
        Update the correlation filter using Eq.(3).
    else
        Find the max response region c_t in F_2, and find the max score s_t
        if $s_t < s_1$ **then**
            Update P using c_t and gt_t
        else if t mod $10 = 0$ **then**
            Update P using c_t and gt_t
        end if
    end if
until end of the sequence

TABLE I
TRACKING EVALUATION OF PRECISIONS AND SPEED. THE **FIRST** AND SECOND BEST RESULTS ARE HIGHLIGHTED IN EACH COLUMN.

	OTB50 [27]		OTB100 [25]	
Trackers	Precision	Speed (fps.)	Precision	Speed (fps.)
TLD [28]	55.9	21.7	59.2	23.3
Struck [29]	61.0	10	63.5	9.84
KCF [26]	61.3	243	69.5	245
MEEM [30]	71.3	9.4	78.0	10.3
CFNet [7]	70.2	76.3	74.8	75
SiameFC [17]	70.1	57.8	77.4	58
FAN(Ours)	**73.7**	25.3	**79.0**	24

V. EXPERIMENTS

In this section, we describe the implementation details and extensive experimental evaluations of our method. The algorithm is evaluated on OTB100 [25] and OTB50 [27] containing 11 tracking challenges, such as illumination changes, camera shake, scale variation, pose variation, partial or full occlusion, and rotation. Besides, we also validate the effectiveness of two parts of our network: inter-feature aggregation and intra-feature aggregation.

A. Implementation Details

We implement our FAN in Python using the TensorFlow toolbox [31] on a desktop with an Intel(R) Core(TM) i7-7800X CPU @ 3.50GHZ and a single NVIDIA GTX 1080 with 8G RAM. Image patches are cropped as 2.5 times the size of the object, and will be resized to 224×224 for the feature extraction network. For CNN features C_1, C_2, C_3, C_4 and C_5, the corresponding kernel sizes of the convolutional layers are $1 \times 1 \times 32, 1 \times 1 \times 64, 1 \times 1 \times 128, 1 \times 1 \times 256$ and $1 \times 1 \times 256$. The size of the convolutional layer followed by F_1 is $1 \times 1 \times 512$. The learning rate of our network is assigned as 0.01 at initial and decay 0.9 for every 5000 iterations, and the total iterations is 200K. The regularization parameter λ_1 of the FAN is assigned as 1. The regularization parameter λ_2 for the correlation filter is assigned as 10^{-4}.

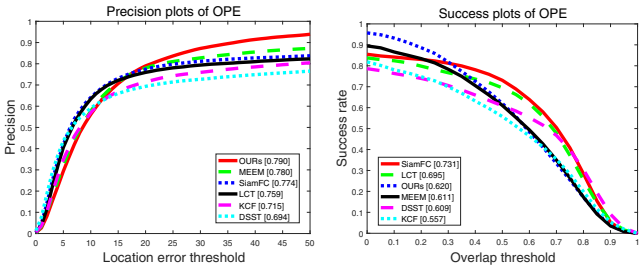

Fig. 5. Evaluation results on OTB100

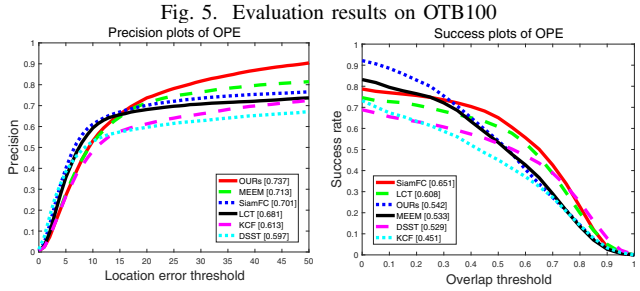

Fig. 6. Evaluation results on OTB50

B. Evaluation

We use the center location error and overlap ratio for evaluation. The trackers are ranked by the area under the curve and distance precision at a threshold of 20 pixels, respectively. The one-pass evaluation (OPE) is used for comparison between our method and several state-of-the-art trackers including (i) real time deep learning trackers, e.g., SiameFC [17] and CFNet [7], (ii) correlation filter trackers, e.g., KCF [26] and DSST [32], (iii) methods with single or multi classifiers, e.g., MEEM [30], Struck [29] and TLD [28].

The evaluation results on OTB100 and OTB50 are shown in Fig. 5 and Fig. 6, respectively. The figures show that the FAN performs better against the state-of-the-art methods of precision, which indicates our method can track the object robustly. For overlap ratio, we also can achieve a comparable result.

For more intuitive comparisons, the tracking performance is presented in Table I, showing precision and speed of all methods. Our tracker achieves the best precision in both OTB50 and OTB100. Although our method is not the fastest one, we can achieve real-time performance. Overall, the FAN performs better against the state-of-the-art tracking methods balancing robust and speed.

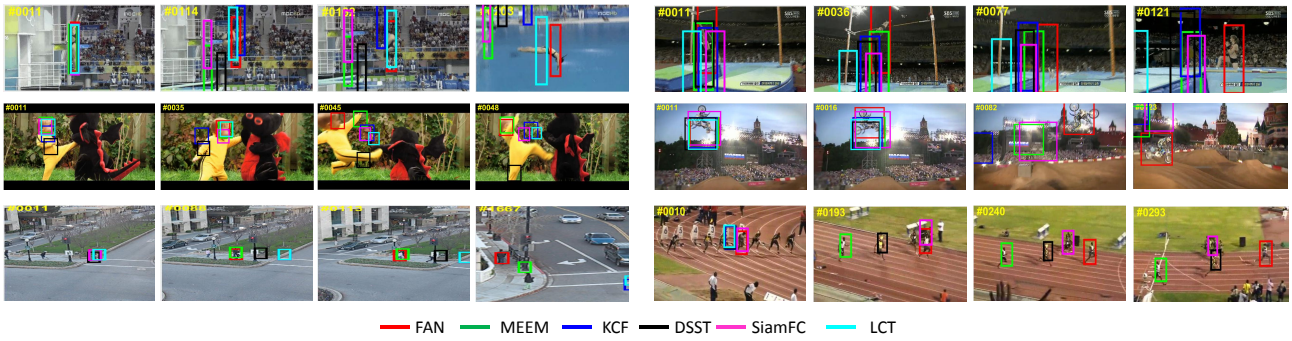

Fig. 7. Tracking results on six challenging sequences with each row for one sequence. Left: *diving,dragoBaby* and *human3* Right: *jump, motorRolling* and *bolt2*.

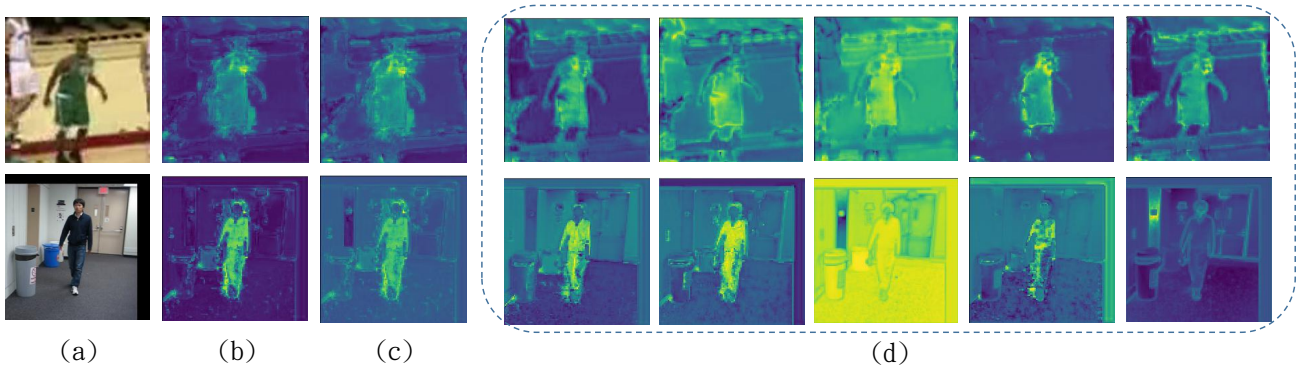

Fig. 8. Visualization of features for two image frames. (a) the input images. (b) F_2, the learned feature from the FAN. (c) F_1, the feature extract from pool1 layer. (d) visualization of feature maps in pool1 which have the largest weights.

Tracking results on some challenging frames among 100 videos are presented in Fig. 7. It is shown that the localization of our method is more precise than the other trackers. Most of these methods fail when the large deformation appears, such as the objects in *diving* and *jump*. Although the SiameFC can learn a good representation with a deep siamese network, it also fails on these challenging sequences. The reason may be that its deep network does not consider the discriminative information of the shallow layers. The MEEM losses the object when there are similar objects appeared, such as that in *bolt2*. This method also fails when occlusion appears, such as in *human3*. On this sequences, our method performs better since the FAN can generate rich semantic and discriminative feature.

C. Intra-feature Aggregation Analysis

Fig. 8 shows the visualization between the F_2, F_1 and some channel feature maps of F_1 with the largest weight. The Fig. 8 shows that F_2 gets the larger discrepancy between the object response value and background than F_1. From the Fig. 8 (d), we found that these feature maps are beneficial to represent the object or suppressing the background response value. Therefore, intra-feature aggregation is conducive for object localization. In Fig. 8 (c), F_1 is also available for locating tracking objects, hence 1×1 convolutional layers in the top-down pathway of FAN have the ability to choose useful feature maps when they reduce channel number.

D. Inter-feature Aggregation Evaluation

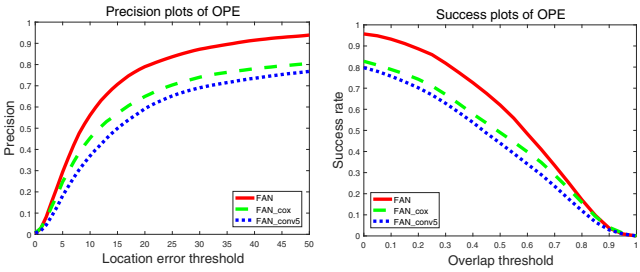

Fig. 9. Comparison between the proposed method and several baselines: FAN-conv5 and FAN-cox

Fig. 4 shows the visualization of the FAN feature F_2 and the feature generated by summing C_1 to C_5. From the visualization, we found that F_2 can represent the object more visible than the original CNN features for object representation. To validate the effectiveness of inter-feature aggregation, we construct two baseline trackers: FAN-conv5 and FAN-cox. FAN-conv5 employs the output of conv5 layer as the object representation for the reason that it has the highest semantic information. Aggregating features from conv4 and conv5, we

obtain the second baseline FAN-cox. The evaluation results are shown in Fig. 9. As the figure shown, our FAN performs better than both baseline trackers, which demonstrates the effectiveness of our method. The performance of FAN-conv5 doesn't perform as well as the FAN-cox, which further validate the effectiveness of the inter-feature aggregation.

VI. Conclusion

We have proposed a novel tracking framework based on the CNN trained in a top-down pathway with lateral connections network which is referred to as FAN. The FAN generates robust representation by aggregating both inter-feature and intra-feature. The FAN feature has coarse semantic information and fine details so that it can locate directly. In addition, the proposed network has simple architecture compared to the other tracking methods using multiple CNN layers. The entire network is pre-trained offline and fine-tuned online. Extensive experimental results show that the FAN is constructed reasonably, and it achieved outstanding performance compared with the state-of-the-art tracking algorithms.

VII. Acknowledgment

This work was supported in part by the Natural Science Foundation of China (NSFC) under Grants No. 61702037, Beijing Municipal Natural Science Foundation under Grant No. L172027, and Beijing Institute of Technology Research Fund Program for Young Scholars.

References

[1] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

[2] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 4, p. 58, 2013.

[3] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer vision and pattern recognition*. IEEE, 2012, pp. 1822–1829.

[4] Y. Wu, B. Ma, M. Yang, J. Zhang, and Y. Jia, "Metric learning based structural appearance model for robust visual tracking." *IEEE Trans. Circuits Syst. Video Techn.*, vol. 24, no. 5, pp. 865–877, 2014.

[5] G. Zhao, L. Chen, J. Song, and G. Chen, "Large head movement tracking using sift-based registration," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 807–810.

[6] Y. Wu, M. Pei, M. Yang, J. Yuan, and Y. Jia, "Robust discriminative tracking via landmark-based label propagation," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1510–1523, 2015.

[7] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," *arXiv preprint arXiv:1704.06036*, 2017.

[8] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.

[9] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *Proceedings of the IEEE Conference on International Conference on Computer Vision*. IEEE, 2017, pp. 2574–2583.

[10] Z. Teng, J. Xing, Q. Wang, C. Lang, S. Feng, and Y. Jin, "Robust object tracking based on temporal and spatial deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1144–1153.

[11] J. Supancic III and D. Ramanan, "Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning," *arXiv preprint arXiv:1707.04991*, 2017.

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

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[14] H. Fan and H. Ling, "Sanet: Structure-aware network for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2017, pp. 2217–2224.

[15] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[16] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proceedings of the IEEE Conference on European Conference on Computer Vision*. Springer, 2016, pp. 749–765.

[17] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the IEEE Conference on European Conference on Computer Vision*. Springer, 2016, pp. 850–865.

[18] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4303–4311.

[19] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3074–3082.

[20] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3119–3127.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv preprint arXiv:1612.03144*, 2016.

[23] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[24] 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*, 2015, pp. 1–23.

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

[26] 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*, vol. 37, no. 3, pp. 583–596, 2015.

[27] 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*, 2013, pp. 2411–2418.

[28] Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 49–56.

[29] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[30] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *Proceedings of the IEEE Conference on European Conference on Computer Vision*. Springer, 2014, pp. 188–203.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[32] 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.