VISUAL OBJECT TRACKING VIA GRAPH CONVOLUTIONAL REPRESENTATION

Zhengzheng Tu, Ajian Zhou, Bo Jiang, Bin Luo*

School of Computer Science and Technology
Anhui University

ABSTRACT

CNN-based trackers are easily interfered by insufficient feature learning, causing model drift. In recent years, graph convolutional networks (GCNs) have been widely used for the representation of graph data in the fields of machine learning and computer vision. In our work, we employ a GCN module to learn structural features for visual tracking. First, we utilize a dual path network to extract heterogeneous features. Then, we adopt a GCN module to construct features to have structured information. Finally, we connect all the features and use the attention mechanism to adaptively select features. Extensive experiments on two benchmark datasets validate the effectiveness of our approach.

***Index Terms*—** Visual tracking, Graph convolutional network, Structural representation, Attention mechanism.

1. INTRODUCTION

In recent years, visual tracking has become a hot issue and applied to video surveillance, behavior recognition, etc [1, 2, 3]. Visual tracking is usually conducted via a tracking-by-detection framework, which has two steps, i.e., 1) generate candidate samples around target object, and 2) employ a trained classifier to classify each sample as target or background [4, 5]. Recently, deep neural networks have been shown very powerfully on representing visual object in visual tracking. BranchOut [6] proposed to employ multiple branches of fully connected layers to maintain variable abstraction levels of target appearances. SANet [7] modeled the object structure by recurrent neural network. VITAL [8] utilized the adversarial learning to capture the robust features over a long temporal span.

The robustness of feature learning is a crucial factor for tracking results. However, traditional convolutional neural networks extract target features by local weighting. The features are often local and there are no relationships among the features. Such features are not robust for sequences with similar objects and objects deformation. Recently, Graph Convolutional Networks(GCNs) has been used for data representation in computer vision [9, 10, 11]. In some tasks, data are usually represented as structured graphs because they have irregular data forms. Traditional convolutional neural networks lose their effect when they encounter this situation.

In our work, we utilize the GCN module to construct the relation of features. We take each point on feature map as a node of the graph and use the feature output of network to construct the adjacency matrix of the graph. We change the architecture of the original MDNet [4], using a dual path network for feature enhancement. Then, GCN module are used to learn the structured features. The features learned from GCN module and original network have different properties, we expect to adaptively select features. In order to assign weights to the structured features of the GCN output and ordinary CNN features,we add channel attention mechanism before the classifier. The channel attention mechanism highlights more useful features. Taking advantages of two-steam network, as well as GCN module and attention mechanism, our tracker has achieved better results than many state-of-the-art methods. We summarize the main contributions as follows:

- We propose a novel dual path network which integrates two architectures of network simultaneously to learn heterogeneous feature representation.
- We propose to employ the GCN module to learn structural features, and add the attention mechanism to adaptively select desired important feature channels.
- Extensive experiments on two widely used benchmark datasets demonstrate the better performance over other state-of-the-art tracking methods.

2. RELATED WORK

2.1. Visual Tracking

In tracking-by-detection framework, tracking task is generally regarded as a classification problem, which aims to distinguish the target from background by classifying each sample as target or background. Recently, CNN-based tracking models [4, 12, 7, 8, 1]have attracted extensive attention recently since the traditional low-level hand-crafted features are not robust enough to various challenges.

* Corresponding author. E-mail: jiangbo@ahu.edu.cn

978-1-5386-9214-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMEW.2019.00-81

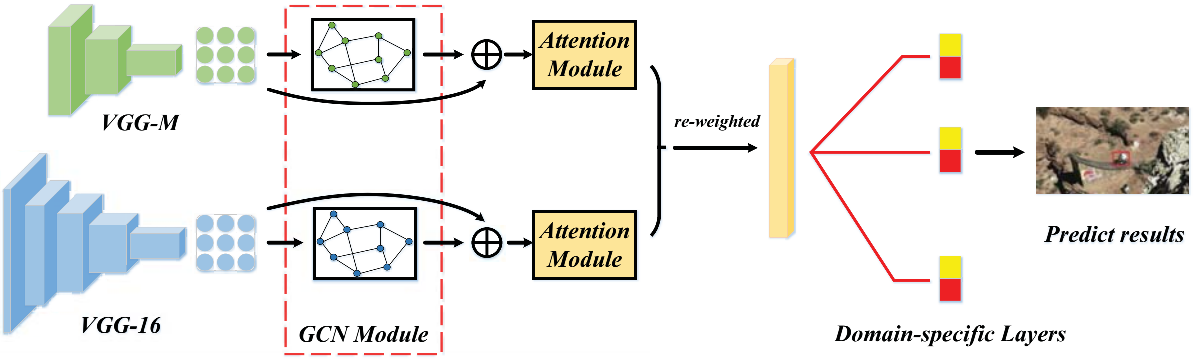

Fig. 1. Flowchart of our tacker architecture. The Figure illustrates the pipeline of two-stream network with GCN module and channel attention module.

Existing CNN-based tracking models generally pretrain CNNs on a large-scale dataset for image classification such as ImageNet [13]. For example, in CNN-SVM [12], the outputs from the last convolutional layer of the CNN are employed as generic feature descriptors for objects, and then can be used to learn discriminative target appearance models using an online SVM. For visual tracking, CNNs can usually be further retrained on a set of annotated video sequences to make CNNs more robust and discriminative. For example, in MDNet [4], the network is based on a multi-domain CNN framework. It can obtain domain-independent information with domain-specific lays and thus learn generic feature representations.

2.2. Graph Convolutional Network

In recent years, the graph convolutional network (GCN) [9, 14] has been widely used in graph data representation and graph node classification. Defferrard et al. [14] introduced the necessary mathematical background and proposed a form of CNN that designs a fast localization convolution filter on the graph with an efficient numerical scheme. Kipf et al. [9] proposed a first-order approximation of spectral filters for semi-supervised learning on graphical structure data based on valid variants of convolutional neural networks operating directly on the graph. It can be seen that GCNs provide a feasible convolution for structured graph data, enabling efficient processing of structured graph data.

GCNs are the extension of CNNs from regular grids to irregular graphs and applied to some computer vision tasks, such as person re-identification [15], skeleton-based action recognition [10, 16], etc. Shen et al. [15] firstly introduced graph neural network to person re-identification task, and proposed a novel person re-identification method with similarity-guided GNN(SGGNN) model, which can get richer feature for helping similarity measurement for pedestrian images. Yan et al. [10] proposed a spatial temporal GCN (ST-GCN) model to improve skeleton-based action recognition. In our work, we focus on visual tracking by exploiting GCN.

3. PROPOSED MODEL

3.1. Overview

Figure 1 shows the overall network module for visual tracking which contains CNN-based feature extraction module, structural graph convolutional network [9] module and attention module.

- **Feature extraction.** We utilize a two-stream CNN network module to extract the appearance features for each object. This two-stream architecture allows the extraction of object features from different levels.
- **Structural representation.** We employ a GCN [9] module to construct structural feature representation for each object. In particular, the aim of GCN module is to obtain a contextual feature representation for each object by exploiting the structural relationship among different object parts.
- **Attention mechanism.** In our tracker, we further use the attention mechanism for adaptive feature selection so that the classifier can take advantage of structured features and conventional convolution features.

We introduce the details of each module in the following.

3.2. Feature extraction via two-stream CNN

Our two-stream CNN feature extraction module involves VGG-M [17] network and VGG-16 [18] network, respectively, as illustrated in Figure 1. In particular, we adopt 3 and 13 convolutional layers in our VGG-M and VGG-16 networks, respectively. The parameters of VGG-16 and VGG-M are pretrained on Imagenet [13].

Both networks receive a RGB bounding box image (candidate object regions) with the size of 107×107 as input. The last convolutional layer of these two networks generate a $512 \times 3 \times 3$ feature map for the input bounding box image. Let $\mathcal{X} \in \mathbb{R}^{3\times3\times512}$ and $\mathcal{X}' \in \mathbb{R}^{3\times3\times512}$ denote the obtained feature maps obtained from VGG-16 and VGG-M, respectively. For simplicity, we can reshape $\mathcal{X}$ and $\mathcal{X}'$ as matrix form $X \in \mathbb{R}^{9\times512}$ and $X' \in \mathbb{R}^{9\times512}$. Then, we input them into GCN module for structural feature representation, as discussed below.

3.3. Structural representation via GCN

We use the features X and X' from the VGG-M and VGG-16 networks respectively as inputs to the GCN. To be specific, we first construct relation graph $G(X, A)$ and $G(X', A')$ for each object. Each row of the feature map X corresponds to a specific part of object image. This motivates us to construct $G(X, A)$ and $G(X', A')$ as follows. That is, each node of graph corresponds to each row of X and each edge A_{ij} represents the relationship between the i-th and j-th rows of X, respectively. The weight of each edge is computed based on Gaussian function as

$$A_{ij} = \exp\left(\frac{-\parallel x_i - x_j \parallel_2^2}{2\sigma^2}\right) \tag{1}$$

where x_i denotes the i-th row of X, $\sigma > 0$ is the parameter to be specified.

Given input $H^0 = X$ and graph A, GCN [9] layer conducts the layer-wise propagation as,

$$H^{(k+1)} = \mathrm{ReLU}(W^{(k)} H^{(k)} A), k = 0, 1 \cdots k - 1. \tag{2}$$

where W is a layer-specific weight matrix needing to be trained. In our tracker, we use a two-layer GCN. The layer-specific weight matrix $\mathcal{W} = \{W_0, W_2 \cdots W_{K-1}\}$ can be learned via minimizing the final cross entropy loss.

3.4. Adaptive feature selection via channel attention

The output features of the VGG-M, VGG-16 and GCN module have different characteristics. We adopt the attention module to assign different weights to all features. We use se-block [19] for this attention module. F is obtained by concatenating feature H and feature X. Formally, a nonlinear transformation is defined to transform F to $\widehat{F}$ that to model the relationships among feature channels,

$$\begin{aligned} &\omega = \delta(\theta_2 \varphi(\theta_1 f_{\mathrm{GAP}}(F))) \\ &\widehat{F} = F \odot \omega \end{aligned} \tag{3}$$

where $\odot$, δ, φ and f_{GAP} denote the channel-wise multiplication, Sigmoid function, ReLU [20] function and Global Average Pooling [21], respectively. θ_1 and θ_2 refer to the weights of two fully connected layers.

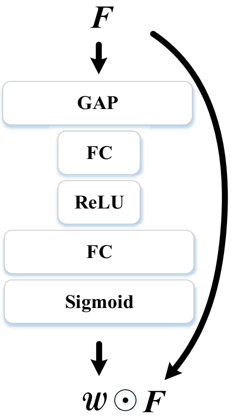

Fig. 2. The pipeline of our channel attention module.

As shown in Figure 2, the attention module returns the weight vector w to re-weight each feature channel, which can selectively emphasize useful features and suppress undesired features, thus to improve the whole tracking network.

3.5. Online tracking

Finally, we connect all features together and integrate them into the next three fully connected layers. During the training stage, the last fully connected layer has K branches (domains) which correspond to K training sequences, respectively. We take all modules and the first two fully connected layers as the shared layer, and the last layer as the domain-specific layer [4].

For the current t-th frame, we generate a set of candidate samples around the previous tracked target. For each candidate sample x, we obtain its positive score $C^+(x;\theta)$ from the network. The optimal target position of the current frame is determined by finding the sample with the maximum position score as

$$x^* = \arg\max_x C^+(x;\theta) \tag{4}$$

where x^* is current frame target state, θ denotes the parameters of fully connected layers.

4. EXPERIMENT

In this section, we analyze the effectiveness of our approach. Then we compare our tracker with some advanced trackers on the benchmark datasets including OTB-2015 [22], TC-128 [23]. For fair comparison, we use the Python version of the MDNet provided by the original author and use the same training method to train our tracker. We use the py-MDNet[1] [4] as our baseline.

[1] https://github.com/HyeonseobNam/py-MDNet

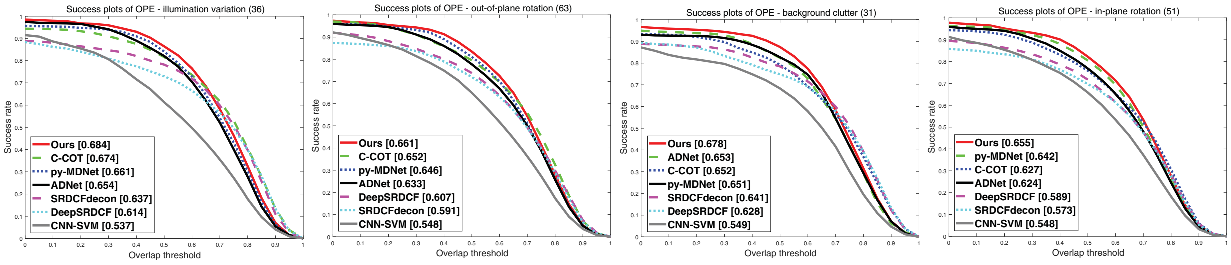

Fig. 3. Success plots for illumination variation, out-of-plane rotation, background clutter and in-plane rotation challenges in OTB-2015 dataset.

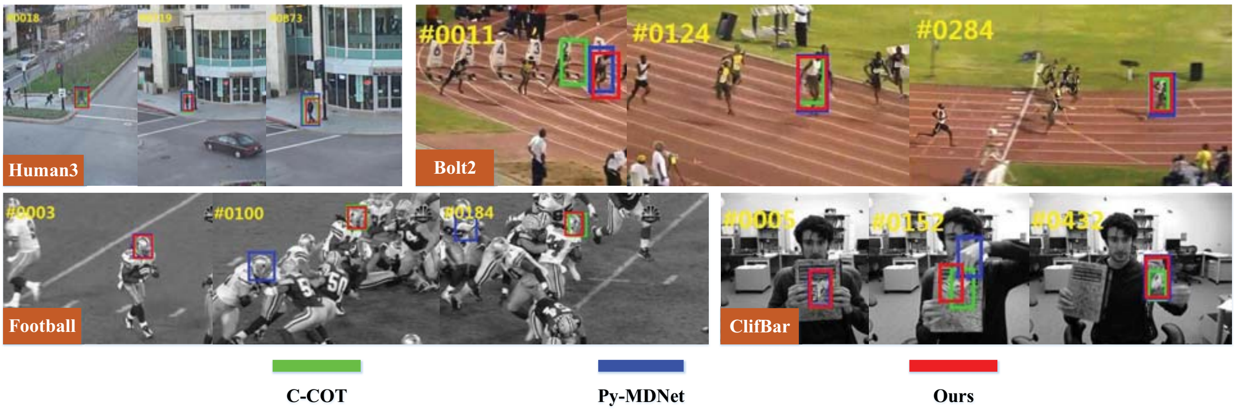

Fig. 4. Qualitative comparisons of the proposed algorithm with several algorithms on some challenging sequences in OTB: Human3, Bolt2, Football and ClifBar.

4.1. Experimental details

First, we extract positive and negative samples from each sequence in the training set and the sum is set to 128, where the positive sample number is 32 and the negative sample number is 96. The overlap rates of the positive and negative samples with the bounding box of ground-truth are $\geq$ 0.7 and ≤ 0.5. Then, we use the K branches of the classifier to distinguish the target from the background, where K is the number of training sequences. All parameters of the fully connected layers and attention module are initialized randomly with zero-mean Gaussian distribution. The parameters of the GCN module are set by sampling from a uniform distribution of intervals. Intervals are decided by the number of input features.

We train our network by using a Stochastic Gradient Descent(SGD) [24] algorithm. For network learning, we train the network through $40K$(K is the number of training sequences) iterations with learning rate 0.0001 for convolutional layers, GCN module and attention module, with learning rate 0.001 for fully-connected layers.

4.2. Evaluation on OTB Dataset

The OTB [22] dataset is a popular tracking benchmark that includes 100 fully annotated videos with various challenges. We employ the precision plots and success plots defined in [22] to evaluate the the robustness of the tracking approaches. To train our network, we use the same training strategy in [4]. We take sequences in VOT2013 [25], VOT2014 [26] and VOT2016 [27] as our training set. We compare our method with most recent state-of-the-art trackers on the OTB-2015 benchmark [22], including MDNet [4], ADNet [28], SRDCFdecon [29], C-COT [30], DeepSRDCF [31], CNN-SVM [12].

Ablation studies. Table 1 shows the success rates and the precision rates on the OTB-2013 dataset [32]. We can observe that the accuracy has dropped when the main architecture of the tracker is replaced with the VGG-16 network. When we use the dual path network as the main architecture of the tracker or add GCN to the dual path network to construct structured information, the accuracies are all greatly improved.

Figure 5 shows that our tracker overall performs well

Table 1. Ablation studies of our methods. The table shows the success rates and precision rates on the OTB-2013 dataset.

Methods	Accuracy
VGG-M(baseline)	0.928/0.689
VGG-16	0.889/0.672
VGG-M+VGG-16	0.927/0.694
VGG-M+VGG-16+GCN	0.936/0.703

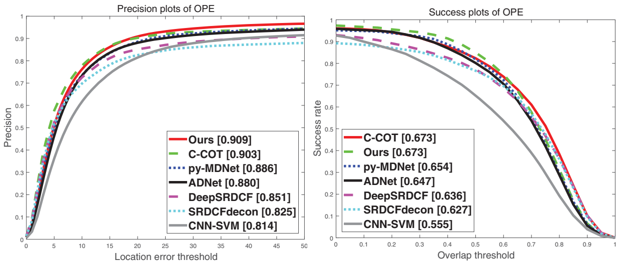

Fig. 5. Our tracker compares with other trackers on the OTB-2015 dataset. These figures show the precision and success plots using the one-pass evaluation.

on the OTB-2015 dataset [22]. Comparing with py-MDNet, our tracker increases the accuracy from $0.886/0.654$ to $0.909/0.673$. The performance of our tracker is comparable to C-COT.

Attributes analysis. Figure 3 compares our tracker with the mentioned above trackers under four video attributes using one-pass evaluation [32]. Comparing with the tracker MDNet, our tracker takes advantages of heterogeneous features for training the classifier to make it more robust. We can observe that our tracker has better performance for illumination variation, in-plane and out-of-plane rotation, and background clutter.

Qualitative comparion. Figure 4 shows qualitative comparion of our method and C-COT [30], MDNet [4] on four challenging sequences. These trackers mostly perform well on these sequences. Obviously, our tracker has better performances on deformation (Bolt2) and occlusion (ClifBar). Our tracker also performs better for long sequences (Human3). Benefiting from structural features, our trackers achieves satisfied effects for sequences with similar targets (Football).

4.3. Evaluation on TC-128

TC-128 [23] contains 128 color image sequences. We use the same metrics in [23] and [22], i.e., precision and success plots, to evaluate the tracking methods. In addition to the trackers tested in the benchmark, we also add some trackers including MDNet [4], C-COT [30], DeepSRDCF [31]. To train our network, we take sequences in VOT2013 [25], VOT2014 [26] and VOT2016 [27] as our training set. Figure 6 illustrates the comparison of our algorithm with other methods in terms of precision and success plots. The experimental results show that our method can deal with various challenges well and is superior to other trackers in the two metrics.

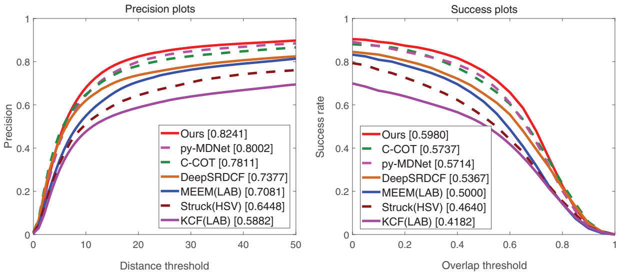

Fig. 6. Precision and success plots on the TC-128 dataset.

5. CONCLUSION

We propose a novel architecture for visual tracking, which employs graph neural network(GCN) to learn structural features. First, we employ a dual path network to learn different heterogeneous features. Then, we utilize the GCN module to construct features with structural information. Finally, we combine all features and use the attention mechanism to adaptively select the features. Extensive experiments on two datasets validate the effectiveness of the proposed method.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No.61602006, 61671018, 61602001); Natural Science Foundation of Anhui Province (1708085QF139); Anhui Key Research and Development Plan under Grant (No.1804a09020101).

6. REFERENCES

[1] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, “Hierarchical convolutional features for visual tracking,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3074–3082.

[2] Chenglong Li, Liang Lin, Wangmeng Zuo, Jin Tang, and Ming-Hsuan Yang, “Visual tracking via dynamic graph learning,” *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[3] Chenglong Li, Chengli Zhu, Jian Zhang, Bin Luo, Xiaohao Wu, and Jin Tang, “Learning local-global multi-graph descriptors for rgb-t object tracking,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[4] Hyeonseob Nam and Bohyung Han, “Learning multi-domain convolutional neural networks for visual tracking,” in *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4293–4302.

[5] Bo Jiang, Yuan Zhang, Jin Tang, Bin Luo, and Chenglong Li, "Robust visual tracking via laplacian regularized random walk ranking," *Neurocomputing*, vol. 339, pp. 139–148, 2019.

[6] Bohyung Han, Jack Sim, and Hartwig Adam, "Branchout: Regularization for online ensemble tracking with convolutional neural networks," in *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 2217–2224.

[7] Heng Fan and Haibin Ling, "Sanet: Structure-aware network for visual tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 2217–2224.

[8] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang, "Vital: Visual tracking via adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8990–8999.

[9] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] Sijie Yan, Yuanjun Xiong, and Dahua Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] Bo Jiang, Zhenli Zhang, Doudou Lin, and Jin Tang, "Graph learning-convolutional networks," *CoRR*, vol. abs/1811.09971, 2018.

[12] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International Conference on Machine Learning*, 2015, pp. 597–606.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, A-ditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[14] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[15] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and X-iaogang Wang, "Person re-identification with deep similarity-guided graph neural network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 486–504.

[16] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," *arXiv preprint arXiv:1902.09130*, 2019.

[17] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

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

[19] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, vol. 7, 2017.

[20] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[21] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[22] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[23] Pengpeng Liang, Erik Blasch, and Haibin Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[24] Léon Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

[25] Matej Kristan, Roman P. Pflugfelder, Ales Leonardis, Jir-i Matas, et al., "The visual object tracking VOT2013 challenge results," in *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, 2013, pp. 98–111.

[26] Matej Kristan, Roman P. Pflugfelder, Ales Leonardis, Jir-i Matas, et al., "The visual object tracking VOT2014 challenge results," in *Computer Vision - ECCV 2014 Workshops - Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*, 2014, pp. 191–217.

[27] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, et al., "The visual object tracking VOT2016 challenge results," in *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, 2016, pp. 777–823.

[28] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2711–2720.

[29] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1430–1438.

[30] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 472–488.

[31] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4310–4318.

[32] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.