ROBUST VISUAL TRACKING VIA DEEP DISCRIMINATIVE MODEL

Heng Fan[†], *Jinhai Xiang*[‡,*], *Guoliang Li*[‡] *and Fuchuan Ni*[‡]

[†]Computer & Information Sciences Department, Temple University, Philadelphia 19122, USA
[‡]College of Informatics, Huazhong Agricultural University, Wuhan, 430070, China

ABSTRACT

In this paper, we exploit deep convolutional features for object appearance modeling and propose a simple while effective deep discriminative model (DDM) for visual tracking. The proposed DDM takes as input the deep features and outputs an object-background confidence map. Considering that both spatial information from lower convolutional layers and semantic information from higher layers benefit object tracking, we construct multiple deep discriminative models (DDMs) for each layer and combine these confidence maps from each layer to obtain the final object-background confidence map. To reduce the risk of model drift, we propose to adopt a saliency method to generate object candidates. Object tracking is then achieved by finding the candidate with the largest confidence value. Experiments on a large-scale tracking benchmark demonstrate that the propose method performs favorably against state-of-the-art trackers.

Index Terms— Visual tracking, deep features, saliency proposal, convolutional neural networks (CNNs)

1. INTRODUCTION

Object tracking is one of the most important components in computer vision and has a variety of applications, such as robotic, surveillance and so forth [1, 2]. Despite great progress in recent years, visual tracking remains a challenging task due to significant appearance changes caused by illumination variations, occlusion, deformation and so on. To deal with these problems, numerous approaches have been proposed. In general, these trackers can be categorized into two families: generative-based methods [3, 4, 5, 6, 7, 8, 9, 10, 11] and discriminative-based methods [12, 13, 14, 15, 16, 17, 18].

The generative methods, which are based on either subspace or template models, formulate tracking problem as searching for regions most similar to object. To alleviate drift issue caused by appearance changes, object appearance model is dynamically updated. [8] proposes an incremental tracking method by learning and updating a low dimension PCA subspace representation for target. In [3], a sparse representation based tracker is proposed by solving a ℓ_1 minimization problem. To improve the efficiency of sparse tracker in [3], [5] proposes to speedup the tracker using accelerated proximal gradient approach. In [7], a local structural sparse appearance model is proposed for tracking via exploiting the inner structure of target. [9] further analyzes the structure of object appearance and proposes a structural sparse tracker.

On the contrary, discriminative methods treat tracking as a classification problem which aims to distinguish the target from fast-varying background. In [18], a compressive tracker is proposed by representing the object with feature from compressed domain. [13] proposes a visual tracker based on transfer learning using gaussian process regression. [17] proposes a real-time tracker with kernelized correlation filters. Nevertheless, these discriminative methods are sensitive to deformation and occlusion. To handle these problems, [16] represents object with superpixels and proposes a discriminative tracker by distinguishing object superpixels from background superpixels. However, it still easily results in drift in presence of similar distractors. To deal with this issue, [14] proposes a discriminative model based tracker which takes similar distractors into account. Nevertheless, this tracker is sensitive to illuminative variations and occlusion because it only uses color features for object appearance modeling which is vulnerable to both illumination changes and occlusion.

Recently, convolutional neural networks (CNNs) [19] have drawn extensive interests in computer vision such as image classification [20, 21] and recognition [22, 23], owing to their powerfulness in feature extraction. The deep features extracted from CNNs are robust under different situations such as illumination changes, and thus suitable to represent object appearance for tracking task [24, 25, 26].

In this paper, we explore deep convolutional features for target appearance modeling and propose a simple while effective deep discriminative model (DDM) for visual tracking. Our DDM consists of two sub-models, i.e., deep object-background model (DOBM) and deep object-distractor model (DODM). The DOBM is to discriminate the object pixels from background pixels, while the DODM is to distinguish the target pixels from distractor (or similar object) pixels. Combining these two sub-models together, we can obtain an object-background confidence map. Besides, taking into account that deep features from different layers play different roles in tracking (e.g., deep features from lower layers contain more spatial information and are more useful for distinguish-

*Corresponding author.

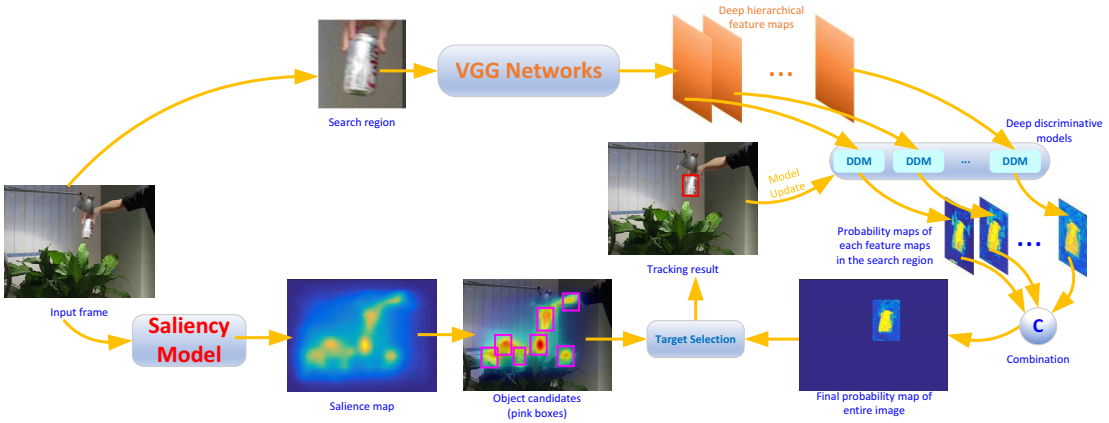

Fig. 1. Illustration of the proposed tracking method.

ing similar objects, while features from higher layers contain more semantic information and thus are more effective to discriminate objects of different classes.), we construct multiple deep discriminative models (DDMs) for each layer and combine the outputs of these different models to obtain the final object-background confidence map. Different from other methods which generate object candidates by simple dense sampling, we propose to use a saliency method to get candidates. The advantage of this strategy is that the candidates are generated from the entire image instead of a local range. In this way, we can reduce the risk of model drift. Finally, visual tracking is achieved by finding the candidate with the largest confidence value. Figure 1 illustrates the proposed method. Experiments on tracking benchmark [1] with 50 sequences evidence the effectiveness of our method.

In summary, our contributions are three-fold: (1) We propose a novel DDM for object appearance modeling and use multiple DDMs to exploit different deep features in different layers for visual tracking. (2) To reduce the risk of model drift, we propose to adopt a saliency method to generate object candidates in the entire image. (3) Extensive experiments on tracking benchmark [1] demonstrate that our tracker performs favorably against state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 describes the proposed approach in details. Section 3 presents experimental results, followed by conclusion in Section 4.

2. THE PROPOSED TRACKING ALGORITHM

2.1. Deep Discriminative Model (DDM)

To distinguish object pixels $\mathbf{x} \in \mathcal{O}$ from surrounding background pixels, we construct the DOBM via a Bayes classifier as in [14]. Different from [14], our classifier is based on deep features extracted from CNNs. Assume that the object region, background region and search region are denoted by O, B and R respectively. Let F_{Ψ}^{R} be the histogram of deep feature extracted over $\Psi \in R$, $F_{\Psi}^{R}(b)$ the b^{th} bin of F, and $b_{\mathbf{x}}$ the b^{th} bin assigned to $R(\mathbf{x})$. Thus, we are able to obtain the object likelihood at location $\mathbf{x}$ with

$$P(\mathbf{x} \in \mathcal{O}|O, B, b_{\mathbf{x}}) \approx \frac{P(b_{\mathbf{x}}|\mathbf{x} \in O)P(\mathbf{x} \in O)}{\sum_{\Psi \in \{O,B\}} P(b_{\mathbf{x}}|\mathbf{x} \in \Psi)P(\mathbf{x} \in \Psi)} \quad (1)$$

In particular, we can compute the likelihood terms by

$$P(b_{\mathbf{x}}|\mathbf{x} \in O) \approx \frac{F_{O}^{R}(b_{\mathbf{x}})}{|O|}, \; P(b_{\mathbf{x}}|\mathbf{x} \in B) \approx \frac{F_{B}^{R}(b_{\mathbf{x}})}{|B|} \quad (2)$$

where $|\cdot|$ is cardinality. Likewise, we can derive the prior probabilities $P(\mathbf{x} \in O)$ and $P(\mathbf{x} \in B)$ as follows

$$P(\mathbf{x} \in O) \approx \frac{|O|}{|O| + |B|}, \; P(\mathbf{x} \in B) \approx \frac{|B|}{|O| + |B|} \quad (3)$$

Substituting Eq. (2) and (3) into Eq. (1), we can simplify $P(\mathbf{x} \in \mathcal{O}|O, B, b_{\mathbf{x}})$ as follows

$$P(\mathbf{x} \in \mathcal{O}|O, B, b_{\mathbf{x}}) = \frac{F_{O}^{R}(b_{\mathbf{x}})}{F_{O}^{R}(b_{\mathbf{x}}) + F_{B}^{R}(b_{\mathbf{x}})}, \mathbf{x} \in (O \cup B) \quad (4)$$

For unseen pixels, i.e., $\mathbf{x} \notin (O \cup B)$, their object likelihoods are set to 0.5.

For visual tracking, distractors (or similar objects) are one of the most common factor that causes drift. To alleviate this issue, we use the same strategy as in [14] and build a DODM. The DODM is similar to DOBM except that the background region is replaced with a set of distracting regions D. Thus, similar to Eq. (4), the DODM is defined with

$$P(\mathbf{x} \in \mathcal{O}|O, D, b_{\mathbf{x}}) = \frac{F_{O}^{R}(b_{\mathbf{x}})}{F_{O}^{R}(b_{\mathbf{x}}) + F_{D}^{R}(b_{\mathbf{x}})}, \mathbf{x} \in (O \cup D) \quad (5)$$

For unseen pixels, i.e., $\mathbf{x} \notin (O \cup D)$, their object likelihoods are set to 0.5 as in DOBM.

Combining DOBM and DODM, we can get the DDM as follows

$$P(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}}) = \alpha P(\mathbf{x} \in \mathcal{O}|O, B, b_{\mathbf{x}}) + (1-\alpha)P(\mathbf{x} \in \mathcal{O}|O, D, b_{\mathbf{x}}) \quad (6)$$

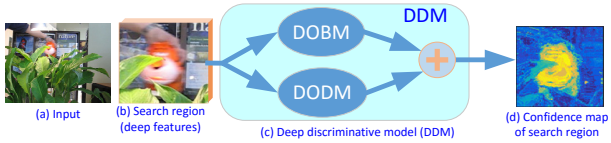

Fig. 2. Illustration of the DDM.

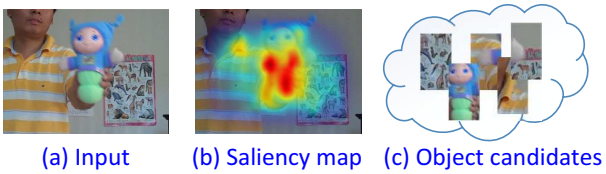

Fig. 3. Illustration of generating object candidates.

where α is a pre-defined parameter. Using DDM, we can obtain an object-background confidence map (see Figure 2).

Considering that both spatial information from lower layers and spatial information from higher layers benefit tracking, we construct multiple DDMs for each layer in CNNs. In specific, we use VGG-16 network [22] to extract deep features, and construct L DDMs for L layers. The final object-background confidence map P_{final} is obtained by summating the outputs of all DMMs as follows

$$P_{final} = \sum_{l=1}^{L} w^l P^l(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}}) \tag{7}$$

where $P^l(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}})$ is the DDM of layer l, and w^l is its weight. Outside the search region, we set the confidence values of pixels to zeros (see Figure 1).

Note that although our DDM is similar to the model in [14], there are still some significant different aspects between these two methods. First, [14] only uses color features for object appearance modeling, while we use deep features to model object appearance which are more robust than color features when appearance changes. Second, we utilize multiple DDMs to exploit different deep features in different layers. In this way, both stability and robustness of the obtained appearance model are improved. Finally, to alleviate the problem of model drift, we propose to use saliency proposal to generate object candidates in the entire image (see Section 2.2) while [14] uses a simple dense sampling method to generate object candidate in a local range.

2.2. Generating Object Candidates via Saliency Proposal

Most previous methods generate object candidates by simple dense sampling in a local range [14, 5, 3, 16]. However, this strategy is prone to result in drift and even failure when inaccurate tracking happens in one frame because this will lead to inaccurate object candidates in next frame. To solve this issue, we adopt an efficient saliency method [27] to generate candidates in entire image. In addition, using saliency method can also help the tracker to accurately estimate the scale of object. Figure 3 illustrates generating candidates.

In specific, after obtaining the saliency map of input, we first take the object region in last frame as the first candidate, and then set the saliency values of pixels in the candidate region to zeros. After that, we select another region with maximum saliency value as the second candidate, and set the saliency values of pixels in this region to zeros. Iteratively, we select object candidates until the maximum saliency value is smaller than a pre-defined threshold θ.

2.3. Tracking and Update

When a new frame t arrives, we first compute its confidence map P^t_{final} using Eq. (7) and then generate object candidates with saliency proposal in Section 2.2. Assume that the target candidate set in frame t is denoted as $C^t = \{c^t_1, c^t_2, \cdots, c^t_N\}$, where N is the number of candidates. Thus, we can compute the confidence value $V(c^t_k)$ of candidate c^t_k as follows

$$V(c^t_k) = \sum\nolimits_{(i,j)\in c^t_k} P^t_{final}(i,j) \tag{8}$$

where $P^t_{final}(i,j)$ denotes the confidence value of pixel at (i,j) in frame t, and the tracking result Φ^t is candidate with the maximum confidence value and can be determined by

$$\Phi^t = \arg\max_{c^t_k} V(c^t_k) \tag{9}$$

To adapt our tracker to changing appearance, we update L DDMs on a regular basis using linear interpolation as follows

$$P^l_{1:t}(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}}) = \beta P^l_t(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}}) + (1-\beta) P^l_{1:t-1}(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}}) \tag{10}$$

where $P^l_t(\mathbf{x} \in \mathcal{O}|b_{\mathbf{x}})$ is the DDM of the l^{th} ($l = 1, 2, \cdots, L$) layer using tracking result Φ^t in frame t, and β denotes the learning rate.

3. EXPERIMENTS

Setting up: Our tracker is implemented in MATLAB on a 3.7 GHz Intel i7 Core PC, and runs at 6.5 frames per second. We use the deep feature maps from five layers ($L = 5$), i.e., Conv1-2, Conv2-2, Conv3-3, Conv4-3 and Conv5-3, in

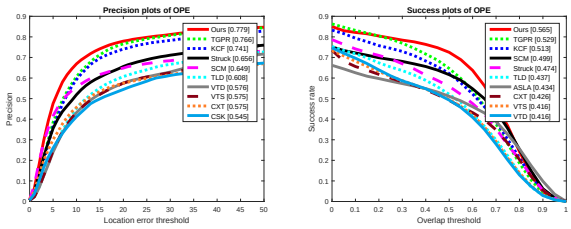

Fig. 4. Comparisons of precision and success plots.

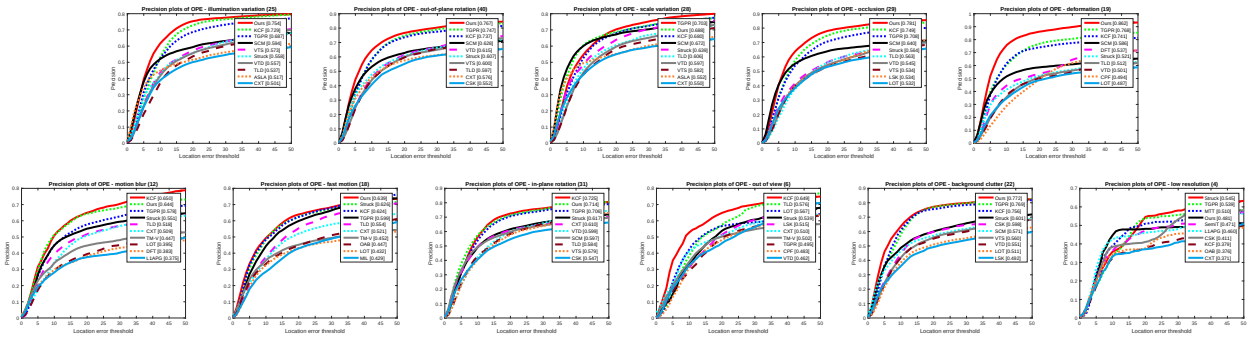

Fig. 5. Comparisons of precision plots of different attributes. Our method outperforms other state-of-the-art trackers.

VGG-16 network [22]. The α in Eq. (6) is set tp 0.7. The w^l in Eq. (7) is set to $1/L$. The saliency threshold θ for generating candidates is set to 0.3. The learning rate β in Eq. (10) is set to 0.15

Dataset and evaluation metric: We evaluate the proposed algorithm on the OTB13 benchmark [1] with comparisons to 31 trackers including 29 trackers and other two recently published state-of-the-art trackers: TGPR [13] and KCF [17]. For better evaluation and analysis of our algorithms, the sequences are categorized according to 11 attributes, including scale variation, occlusion, deformation and so on. We employ the precision and success plots defined in [1] evaluate the robustness of the tracking algorithms.

Overall performance: Figure 4 shows the precision and success plots of our tracker and other methods. To make it clear, only the top 10 trackers are displayed. As shown in Figure 4, our method ranks the first and achieves the best performance in both precision and success ranking plots. In specific, the proposed tracker achieves 0.779 ranking score in precision plots and 0.565 ranking score in success plots, and outperforms the state-of-the-art TGPR tracker [13] with 0.766 precision ranking score and 0.529 success ranking score.

Attribute-based evaluation: The sequences in the benchmark dataset are annotated with 11 attributes to describe the different challenges in tracking problem. These attributes are helpful for analyzing the performance of trackers in different situation. We report the performance of our tracker for these eleven challenging attributes in Figure 5. From Figure 5, we can see that our method achieves favorable results in nine attributes (within top 2), i.e., illumination variation, out-of-plane rotation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation and background clutter.

Qualitative evaluation: We compare our tracker with four state-of-the-art methods: KCF [17], TGPR [13], Struck [28] and SCM [4]. The qualitative results are shown in Figure 6. From Figure 6, we can see that the propose tracker performs well in illumination variations (*CarDark* and *Singer1*), occlusion (*Lemming*, *Basketball*), scale changes (*Dog1* and *Doll*), deformation (*David3* and *Bolt*), while other trackers can only handle some situations and degrade in other cases.

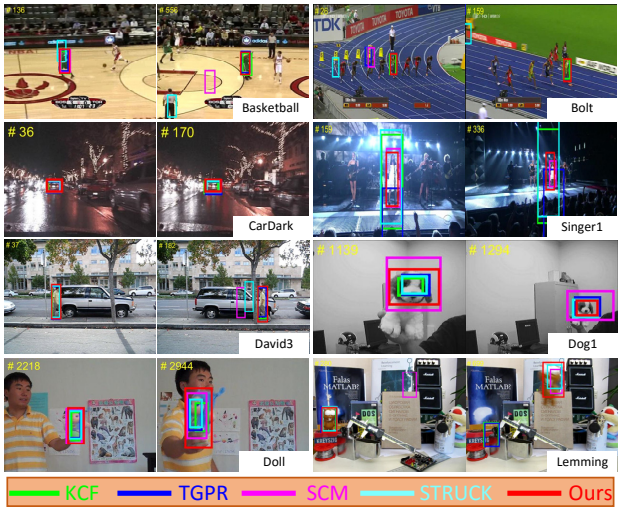

Fig. 6. Qualitative results of five trackers on eight sequences.

4. CONCLUSION

This paper propose a novel deep discriminative model for visual tracking by exploiting deep convolutional features for object appearance modeling. By using deep features, the appearance model obtained is robust to different situations such as illumination changes and motion blur. Besides, to improve the stability and robustness of our model, we construct multiple deep discriminative models for different features of different layers in CNNs, and combine them for visual tracking. In addition, to reduce the risk of model drift, we adopt a saliency method to generate object candidates. Object tracking is then achieved by finding the candidate with the largest confidence value. Experiments on a large-scale tracking benchmark with 50 sequences evidence the effectiveness of our method.

Acknowledgements: This work was primarily supported by Foundation Research Funds for the Central Universities (Program No. 2662016PY008 and Program No. 2662014PY052).

5. REFERENCES

[1] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[2] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM TIST*, vol. 4, no. 4, pp. 58, 2013.

[3] Xue Mei and Haibin Ling, "Robust visual tracking and vehicle classification via sparse representation," *TPAMI*, vol. 33, no. 11, pp. 2259–2272, 2011.

[4] Wei Zhong, Huchuan Lu, and Ming-Hsuan Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 1838–1845.

[5] Chenglong Bao, Yi Wu, Haibin Ling, and Hui Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *CVPR*, 2012, pp. 1830–1837.

[6] Heng Fan, Jinhai Xiang, Honghong Liao, and Xiaoping Du, "Robust tracking based on local structural cell graph," *JVCIR*, vol. 31, pp. 54–63, 2015.

[7] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*. IEEE, 2012, pp. 1822–1829.

[8] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[9] Heng Fan and Jinhai Xiang, "Robust visual tracking with multitask joint dictionary learning," *TCSVT*, 2016.

[10] Junseok Kwon and Kyoung Mu Lee, "Visual tracking decomposition," in *CVPR*. IEEE, 2010, pp. 1269–1276.

[11] Heng Fan, Jinhai Xiang, and Liang Zhao, "Robust visual tracking via bag of superpixels," *Multimedia Tools and Applications*, pp. 8781–8798, 2016.

[12] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-learning-detection," *TPAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[13] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing, "Transfer learning based visual tracking with gaussian processes regression," in *ECCV*, 2014, pp. 188–203.

[14] Horst Possegger, Thomas Mauthner, and Horst Bischof, "In defense of color-based model-free tracking," in *CVPR*, 2015, pp. 2113–2120.

[15] Heng Fan, Jinhai Xiang, and Zhongmin Chen, "Visual tracking by local superpixel matching with markov random field," in *PCM*, 2016, pp. 1–10.

[16] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang, "Robust superpixel tracking," *TIP*, vol. 23, no. 4, pp. 1639–1651, 2014.

[17] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2015.

[18] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang, "Fast compressive tracking," *TPAMI*, vol. 36, no. 10, pp. 2002–2015, 2014.

[19] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *CVPR*, 2015, pp. 1026–1034.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[22] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[24] Hanxi Li, Yi Li, and Fatih Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *TIP*, vol. 25, no. 4, pp. 1834–1848, 2016.

[25] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015, pp. 3074–3082.

[26] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu, "Visual tracking with fully convolutional networks," in *ICCV*, 2015, pp. 3119–3127.

[27] Jonathan Harel, Christof Koch, and Pietro Perona, "Graph-based visual saliency," in *NIPS*, 2006, pp. 545–552.

[28] Sam Hare, Amir Saffari, and Philip HS Torr, "Struck: Structured output tracking with kernels," in *ICCV*, 2011, pp. 263–270.