

Deep Scale Feature for Visual Tracking

Wenyi Tang, Bin Liu^(✉), and Nenghai Yu

CAS Key Laboratory of Electromagnetic Space Information,
University of Science and Technology of China, Hefei, China
flowice@ustc.edu.cn

Abstract. Recently, deep learning methods have been introduced to the field of visual tracking and gain promising results due to the property of complicated feature. However existing deep learning trackers use pre-trained convolution layers which is discriminative to specific object. Such layers would easily make trackers over-fitted and insensitive to object deformation, which makes tracker a good locator but not a good scale estimator. In this paper, we propose deep scale feature and an algorithm for robust visual tracking. In our method, object scale estimator is made from lower layers from deep neural network and we add a specially trained mask after convolution layers, which filters out potential noise in this tracking sequence. Then, the scale estimator is integrated into a tracking framework combined with locator made from powerful deep learning classifier. Furthermore, inspired by correlation filter trackers, we propose an online update algorithm to make our tracker consistent with changing object in tracking video. Experimental results on various public challenging tracking sequences show that our proposed framework is effective and produce state-of-art tracking performance.

1 Introduction

Visual tracking plays one of the most fundamental role in the field of computer vision, due to it has wide range of applications, such as safety surveillance, intelligent city system and vision-based self-driving cars. Visual tracking is model-free, which means given a bounding box of target in the first frame, the tracker would estimate its position and scale in the next frames of video with none prior knowledge related to this sequence. It lacks training samples and every sequence is of great difference. Although visual tracking has been researched for years, it still has a lot of challenging problems need solve, including occlusion, scale variation, illumination change and object deformation [27].

Object tracking algorithms are of two categories: generative and discriminative. Generative algorithms learn a high-dimension feature space to describe target and locate the target by minimizing the reconstruction error in thousands of potential regions. Discriminative algorithm builds a metric to minimize the distance between target in consequence frames and target in the first frame and maximize the distance between target and background. These approaches have been developed to gain better performance. However, classical methods

utilize artificial features, such as Histogram of oriented gradients (HoG) [4], Local binary patterns (LBP) [18] to describe texture information, or appearance model. These features cannot represent complicated structure and show deeper information of object and background.

Convolutional Neural Network (CNN), which could learn sophisticated features from original image data, has been adopted in tracking as well. Inspired by transfer learning [19] used in other computer vision field, for example, object detection [8, 21] and semantic segmentation [17], they transfer convolutional layers pre-trained at ILSVRC2012 ImageNet [16] classification dataset. These pre-trained layers have excellent ability of generalization and relieve the lack of training sample in tracking partly.

Those aforementioned trackers, which use highly-deep convolution layers, simply ignore the lack of training samples in model-free tracking. Since the output of deep convolution layers is quite sparse and overfitted to some parts of target, these features might not suit to scale estimate job. It is quite straightforward for us to get features from shallow layers to estimate the current target scale. But, this raises another problem, that shallower layers mean more noise from background and such noise would interfere the scale estimator. So we need learn a mask after the shallow network and filter out noise from background.

In this paper, we propose a tracking framework based on deep scale feature (DSF), which consists of two parts. One based on deeper CNN decides the center of current target at current time. The other one based on shallow CNN learns the target appearance and generates estimation to object size. Recent deep learning trackers usually has only one network and ignore the immanent contradiction between two different tasks, locator and scale estimator. Because, the scale estimation task requires the net sensitive to appearance change, while locator demands invariance feature. Different from these approaches, the locator of our method is relieved from the task of estimating the scale variance and built from deeper networks. On the other hand, the scale estimator is not that deep.

The contribution of our method can be summarised as:

- (1) We propose a self-learnt mask algorithm and deep scale feature to describe the appearance model of target.
- (2) We propose a visual tracking framework consisting of two neural networks, which has state-of-art performance.

The rest of the paper is organised as follows. In Sect. 2, we first review related work. The details of the proposed method are illustrated in Sect. 3. In Sect. 4, we would presents and discuss the experimental results on a tracking benchmark. Section 5 provides conclusions.

2 Related Work

Widely used tracking-by-detection [15] framework consists of two models: appearance model to tell the target shape and motion model to tell the center of target. There are two kinds of appearance model, generative and discriminative.

Generative model mainly focuses on reconstruction error of target candidates. These methods utilize raw pixel information [1] or sparse subspace [13] to describe the appearance model. Discriminative model finds the most discriminative feature to distinguish object from background. Online learning framework based on structured SVM [10], multiple instance learning [2] and correlation filters [5, 7] are adopted and they perform better than generation models. DCF-based tracker initially uses low level feature, such as HOG feature [5]. Recent DCF-based trackers [6] utilize CNN as robust feature extractor. [5] proposes an algorithm to estimate the scale changes based on the Gaussian model and [24] reimplements it by deep features.

As the development of hardware, plenty of algorithms of computer vision have been invented based on neural network. So do they in model-free tracking. In [9], two-stream structure is proposed to build a classifier-based tracker. Pre-trained on auxiliary images, [26] presents an auto-encoder tracker. To reduce over-fitting, [24] uses a complicated sequential ensemble learning strategy. [20] tries to use multi-level feature from stacks of convolutional networks (Fig. 1).

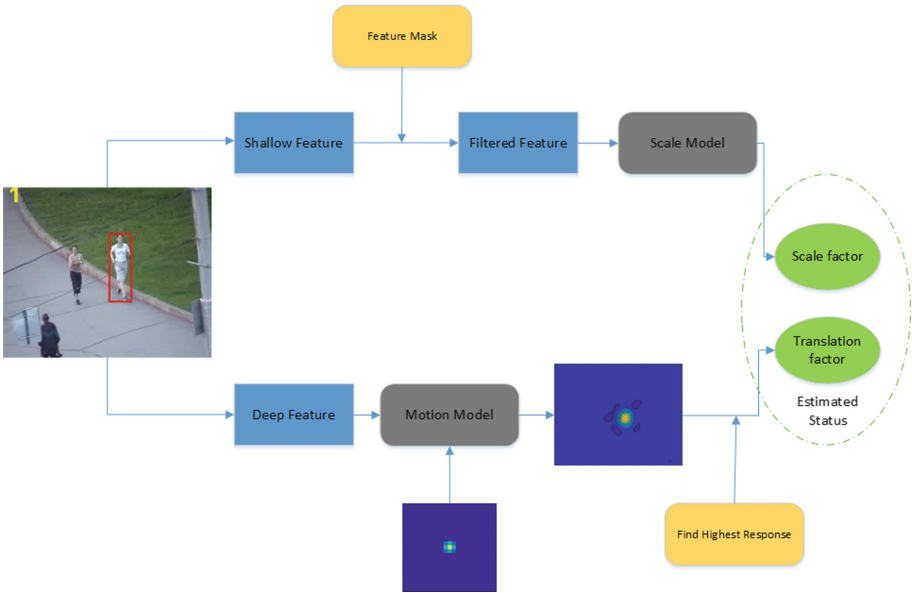


Fig. 1. The structure of proposed algorithm. Our algorithm consists of three main steps: (1) extract robust features from convolutional layers; (2) detect the object center; (3) estimate the scale change

3 Proposed Method

3.1 Deep Network Output When Tracking

Before describing the details of proposed deep scale feature tracker, we first analyse output of deep convolution layers in the field of tracking. When a deep

convolution network, like VGG [22] or ResNet [11], tries to classify an object, it first uses its convolution kernels to slide across the object and produces heat maps indicating what kernels response to this structure. After the last convolutional layer, VGG would connect all its heat map unit to all the unit of fully connection layer to learn every unit’s contribution to the final decision. However, as described in [25], if we transfer these layers to tracking jobs, the most neurons of last convolution layer are nearly zero. These neurons are highly sparse and discriminative to specific object. Since max pooling layer is partly shift invariance, these sparse activated neurons might not change much when object varies in scale. Therefore, we should remove some max pooling layers, and choose feature extractor not quite deep. Thus, we choose layers before conv4.3 of VGG as base feature extractor.

On the other side, deeper convolution layer has more semantic information in object categories, the shallower layer has more structure information in texture. These inactivated or dead neurons of shallow layer however, might become activated when object is occluded by background or new object, which has similar texture structure, appears in the receptive field. These unexpected activated neurons would interfere scale estimator. We should learn a mask to shut down these potential noisy neurons.

3.2 Deep Feature Mask

As mentioned before, we should find those potential noisy deep feature from 512 channels of conv4.3 layer. Simply speaking, the self-learnt mask should disable those neurons which output similar pattern activations between object and background. Because we would append several layers after conv4.3 to estimate current scale. As a result, we should take the discriminative ability of newly appended layers. So the simplest way of disabling those sensitive to everything neurons is not our choice. This is because, these neurons might output easy-to-distinguish pattern between object and background. Inspired by [25], the proposed deep mask method is based on a target heat map regression model. This model is conducted on conv4.3 layers of VGG and consists of a convolutional layer without any nonlinear activation layer. It takes the feature maps of conv4.3 to be masked as input to predict the target heat map g , which is a compact 2-D centered target of ground truth as used in [5]. The model is trained by minimizing the following loss function:

$$L = \|G(F_t) - g\|^2 + \lambda \Sigma \|w\|^2 \quad (1)$$

G function is the newly added layers. If fed with the feature maps F of conv4.3 of whole frame at the time, it would produce a 2-D heat map. The parameter λ balances importance of L2 loss and regularization term.

After back-propagation converges, we select the feature maps according to their output at the location of object and background. If f_i represents the i -th feature maps F of conv4.3, the heat map difference can be computed by masked

out the i -th feature map \hat{G}_i then minus g . Then we define the importance I_i of the element \mathbf{f}_i as its difference with target map and can be computed as follows:

$$I_i = \|\hat{G}_i - g\|^2 \quad (2)$$

All the 512 feature maps are sorted in the descending order by their importance. The K selected feature maps have the top- K importance others are masked out. In our experiments, we choose 300 as K and only do mask learning at the first frame and tracker performs quite well.

3.3 Deep Scale Feature

The proposed deep scale feature is based on conv4_3 from VGG with deep feature mask, which simultaneously sense more low-level information and avoid possible noisy from similar object categories. Next, we would describe the scale estimator and how it works.

The scale estimator is constructed on top of conv4.3 layers with learnt mask and consists of a fully-connected layer with one neuron, which produces the scale variance factor. At the first frame, after mask has been built, we crop the object rectangle in different sizes with a step of 1.02 as suggested in [5]. Then, the scale estimator is trained with these feature map from different size with stochastic gradient descent (SGD) algorithm.

When tracking, the tracker firstly uses locator to determine current center and crop rectangle area around object with size of last frame. Secondly, the cropped image patch is interfered by scale estimator and update the scale coefficient accordingly. Since target could change a lot in the same sequence, the tracker would update scale estimator periodically.

3.4 Locator Construction

According to foremost analysis, the locator should use such features as discriminative as possible. Therefore, we choose one of the best CNN classifiers as feature extractor. The locator is based on the Res4a layers of ResNet-51, then add two convolutional layers and one rectifier activation between them.

We follow the approach of discriminative correlation filter to train our model, which is trained by minimizing the following loss function:

$$L = \|F(I) - g\|^2 + \lambda \Sigma \|w\|^2 \quad (3)$$

F function is the position-CNN. If fed with image patch I , it would produce a 2-D heat map. g is the target heat map which has a compact 2-D Gaussian shaped peak centered at the center. The parameter λ balances importance of L2 loss and regularization term. At the first frame, in order to learn context information around the object, we crop rectangle two times larger than ground truth bounding box and modify target heat map g accordingly.

During tracking, image patch around last position is input into locator and we get the current object center which has the largest confidence in the heat map. The locator parameters are updated in the same way as scale estimator.

3.5 Tracking Algorithm

Algorithm 1. Deep scale feature tracking algorithm

Input: The initial bounding box $bbox_1$
Output: current bounding box $bbox_t$

- 1: Initialize mask with Eq. (1) and Eq. (2)
- 2: Initialize locator and scale estimator
- 3: **repeat**
- 4: Crop region I_t centered at last location and two time larger in size
- 5: Use *locator* to estimate current center $center_t$ of target.
- 6: **if** confidence higher than threshold **then**
- 7: Crop region I'_t centered at $center_t$ with the same size as I_t . Pass I'_t through *scale estimator*, produce current scale factor
- 8: Update object location
- 9: **else**
- 10: current location equals last location
- 11: **end if**
- 12: **if** location changes and tracker has process 10 frames **then**
- 13: Update locator and scale estimator
- 14: **end if**
- 15: **until** The sequence ends

Overview. The overall tracking procedure is presented in Algorithm 1.

Tracking and Update. We use locator to get the center of object, while use scale estimator to track scale variance. Assuming heat map of locator is current possibility distribution and the confidence of each candidates equals the value of heat map. And we find the maximum confidence and convert relative position to pixel position.

To balance periodic update and poorly samples, we propose one update criteria: high locator confidence. The criteria is measured by ways as in [25], we treat the maximum heat map value as current confidence. If it is less than a threshold 0.1, we would stop parameter update. Once scale or size needs update, they vary by one step (1.02 for scale and ratio).

4 Experiment

4.1 Experiment Setup

The proposed framework is implemented in Caffe [14] with MATLAB R2016a and runs at 0.5 frames per second. Our tracker runs on a PC with 3.0GHz i7-X5960 CPU and TITAN X GPU. All of networks are trained with SGD solver at learning rate $1e-7$ with momentum of 0.9.

Our tracker is evaluated on 12 public challenging video sequences, containing plenty sort of challenging factors, such as fast motion, scale and illumination change, background clutter and object occlusion. We compare our tracker

result with 10 state-of-art trackers, consisting of 4 deep learning based trackers, including FCNT [25], SiamFC [3], SINT [23], STCT [24], 3 DCF-based trackers, including SRDCF [7], DeepSRDCF [6], HDT [20] and 3 classic trackers, including Struck [10], MEEM [28], MUSTer [12]. For the fairness, we adopt the source codes or result files provided by the authors.

4.2 Experiment Result

Two common used metrics are applied for quantitative evaluation: Central Location Error (CLE) and Overlap Rate (OR). CLE is defined as the Euclidean distance between center of $Bbox_G$ and $Bbox_T$, where $Bbox_G$ is the ground-truth bounding box and $Bbox_T$ is the bounding box produced by trackers. OR is defined as $OR = \frac{Bbox_T \cap Bbox_G}{Bbox_T \cup Bbox_G}$.

Quantitative Evaluation. We use the precision plot and the success plot, as shown in Fig. 2, to evaluate average performance of trackers on every sequence. The precision plot demonstrates the percentage of frames where the distance between the predicted target location and the ground truth location is within a given threshold. Whereas the success plot illustrates the percentage of frames where OR between the predicted bounding box and the ground truth bounding box is higher than a threshold. The area under curve (AUC) is used to rank the tracking algorithms in each plot. As shown in Fig. 2 and Tables 1 and 2, our method achieves the superior performance in terms of both evaluation metrics compared to state-of-art trackers. Especially, STCT tracker utilizes similar structure as proposed algorithm, but STCT does not consider deep feature mask at scale and uses highly sophisticated framework compared to proposed method. If we exam those not-well-performed sequences, we would some common properties, like out-of-plane rotation in Coke, shape deformation in Basketball and similar object in Car4 and Deer. These factors undermine the power of CNN feature to estimate scale, which is complicated 2-D coarse-grained feature and cannot handle 3-D and fine-grained changes well.

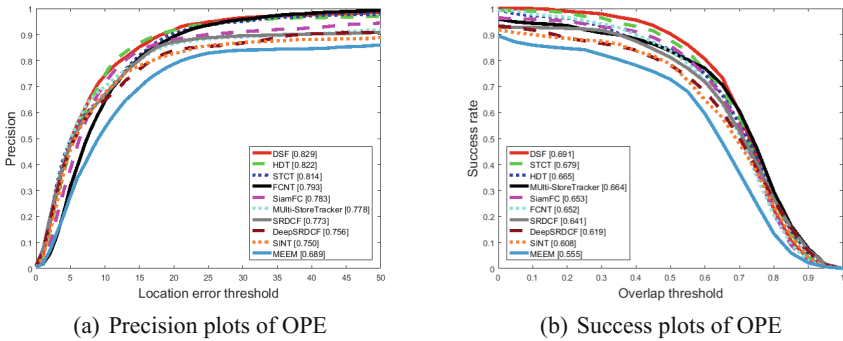


Fig. 2. Result curve

Table 1. Comparison of 11 trackers on 12 video sequences in terms of central location error (in pixels). Red fonts indicate the best performance while the blue fonts indicate the second best.

	FCNT	SRDCF	SiamFC	MEEM	SINT	Struck	HDT	MUSTer	DeepSRDCF	STCT	DSF
Deer	7.7	4.0	5.5	4.5	5.3	5.3	5.1	7.2	4.6	4.3	4.8
Basketball	10.3	10.8	22.9	8.5	19.1	118.3	3.3	4.8	19.8	3.9	5.3
Car4	4.7	1.7	5.6	18.0	3.3	8.7	7.4	1.9	1.8	2.7	5.3
Coke	11.0	18.9	11.3	11.0	7.5	12.1	10.2	15.1	14.9	15.1	13.4
FaceOcc1	22.3	14.8	11.7	16.4	13.8	18.8	17.4	14.3	12.9	18.3	19.2
Football	8.6	5.7	6.3	88.9	210.9	17.3	5.8	14.8	6.2	7.4	6.8
Jogging-2	15.3	3.7	6.1	8.2	4.1	107.7	2.9	4.8	5.7	6.4	2.8
Woman	9.5	4.8	14.9	4.1	11.1	4.2	10.9	9.4	3.7	8.6	3.5
CarScale	10.6	19.7	15.4	30.1	31.5	36.4	29.9	18.7	25.2	15.2	15.3
Subway	5.1	2.6	3.3	4.1	3.6	4.5	2.6	2.2	3.0	3.0	2.4
MountainBike	6.4	9.0	6.1	13.2	10.3	8.6	8.4	8.1	9.7	12.6	7.2
MotorRolling	12.5	247.1	87.9	170.9	24.0	145.7	14.1	110.4	200.8	18.2	17.1
Avg	10.3	28.6	16.4	31.5	28.7	40.6	9.8	17.6	25.7	9.6	8.6

Table 2. Average overlap rate. Red fonts indicate the best performance while the blue fonts indicate the second best.

	FCNT	SRDCF	SiamFC	MEEM	SINT	Struck	HDT	MUSTer	DeepSRDCF	STCT	DSF
Deer	0.71	0.81	0.73	0.75	0.71	0.74	0.75	0.75	0.77	0.76	0.75
Basketball	0.67	0.53	0.57	0.66	0.64	0.20	0.85	0.75	0.39	0.78	0.71
Car4	0.82	0.87	0.78	0.46	0.78	0.49	0.49	0.90	0.88	0.84	0.75
Coke	0.66	0.51	0.58	0.65	0.63	0.67	0.65	0.52	0.52	0.56	0.54
FaceOcc1	0.64	0.76	0.76	0.75	0.75	0.73	0.74	0.76	0.79	0.71	0.77
Football	0.56	0.66	0.70	0.34	0.14	0.53	0.68	0.54	0.65	0.61	0.69
Jogging-2	0.67	0.71	0.70	0.63	0.75	0.20	0.79	0.75	0.60	0.74	0.78
Woman	0.69	0.67	0.54	0.71	0.62	0.73	0.74	0.69	0.72	0.67	0.64
CarScale	0.52	0.73	0.69	0.41	0.54	0.41	0.41	0.68	0.69	0.72	0.74
Subway	0.63	0.76	0.75	0.68	0.68	0.65	0.76	0.72	0.73	0.75	0.79
MountainBike	0.77	0.69	0.76	0.60	0.63	0.71	0.70	0.73	0.69	0.61	0.78
MotorRolling	0.58	0.09	0.37	0.10	0.52	0.15	0.52	0.29	0.10	0.51	0.56
Avg	0.66	0.65	0.66	0.56	0.62	0.52	0.67	0.67	0.63	0.69	0.70

5 Conclusion

In this paper, we have proposed a robust tracking framework based on deep scale feature. To make the tracker sensitive to scale variance and robust against noises, a type of mask is learnt from the first frame and is used to filter out potential noisy feature maps. To estimate current scale factor, we train a fully-connected layer with one neuron right after masked feature map. Last but not least, a periodic update scheme is proposed to trade off between poorly tracking result and object changes. We have tested out method on 12 different challenging sequences and experiment results show the superiority of proposed algorithm compared to 10 state-of-art trackers.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Grant No. 61371192), the Key Laboratory Foundation of the Chinese Academy of Sciences (CXJJ-17S044) and the Fundamental Research Funds for the Central Universities (WK2100330002).

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: 2006 IEEE Computer Society Conference on Computer vision and pattern recognition, vol. 1, pp. 798–805. IEEE (2006)
2. Babenko, B., Yang, M.H., Sivic, J.: Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 983–990. IEEE (2009)
3. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional Siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_56
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
5. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, 1–5 September 2014. BMVA Press (2014)
6. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 58–66 (2015)
7. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)
8. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV), December 2015
9. Gladh, S., Danelljan, M., Khan, F.S., Felsberg, M.: Deep motion features for visual tracking. arXiv preprint [arXiv:1612.06615](https://arxiv.org/abs/1612.06615) (2016)
10. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
12. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 749–758 (2015)
13. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: 2012 IEEE Conference on Computer vision and pattern recognition (CVPR), pp. 1822–1829. IEEE (2012)
14. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Sermanet, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint [arXiv:1408.5093](https://arxiv.org/abs/1408.5093) (2014)

15. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409–1422 (2012)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440 (2015)
18. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
19. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014
20. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged deep tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311 (2016)
21. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: *Advances in neural Information Processing Systems*, pp. 91–99 (2015)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
23. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1420–1429 (2016)
24. Wang, L., Ouyang, W., Wang, X., Lu, H.: STCT: sequentially training convolutional networks for visual tracking. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1373–1381, June 2016
25. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3119–3127 (2015)
26. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587* (2015)
27. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013)
28. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: *Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 188–203. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_13*