Generic Pixel Level Object Tracker Using Bi-Channel Fully Convolutional Network

Zijing Chen[1,3(✉)], Jun Li[1], Zhe Chen[2], and Xinge You[3]

[1] Faculty of Engineering and Information Technology, Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia
z.j.chen219@gmail.com
[2] School of Information Technology, UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Darlington, NSW 2006, Australia
[3] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

Abstract. As most of the object tracking algorithms predict bounding boxes to cover the target, pixel-level tracking methods provide a better description of the target. However, it remains challenging for a tracker to precisely identify detailed foreground areas of the target. In this work, we propose a novel bi-channel fully convolutional neural network to tackle the generic pixel-level object tracking problem. By capturing and fusing both low-level and high-level temporal information, our network is able to produce pixel-level foreground mask of the target accurately. In particular, our model neither updates parameters to fit the tracked target nor requires prior knowledge about the category of the target. Experimental results show that the proposed network achieves compelling performance on challenging videos in comparison with competitive tracking algorithms.

Keywords: Visual tracking · Segmentation · Convolutional neural network

1 Introduction

Practical object tracking in videos is often formulated as updating the location and size of a bounding box upon observing each new frame in the video, where the target is specified by the bounding box in the previous frame. Using bounding box in tracking follows the conventional usage of a rectangular region of interest (ROI). A rectangle is a minimalistic (only 4 numbers) and practical representation of a target and has been ubiquitously used in many machine vision tasks, including object detection [1] and action recognition [2]. On the other hand, pixel-level analytics has long been considered desirable as it provides richer details and naturally accommodates complicated cases such as multi-target detection/tracking, especially when dealing with occlusion and shape variance. Unfortunately, pixel-level processing of images and videos entails the formidable task of capturing fine structures in the visual signals.

D. Liu et al. (Eds.): ICONIP 2017, Part I, LNCS 10634, pp. 666–676, 2017.
https://doi.org/10.1007/978-3-319-70087-8_69

A breakthrough has been made recently with the impressive development of deep convolutional neural networks [3,4]. Given sufficient data and with the cost of an expensive training session, when deployed those models are able to make quick and accurate predictions at the similar resolution of the input signal [5]. Thus a wide range of machine vision tasks, such as object identification and semantic scene understanding, have advanced their granularity of analysis to pixel-level. The work we present in this paper aims to harvest the benefit of the analytic tools based on neural networks and achieve finer and more accurate object tracking.

In particular, we propose a bi-channel fully convolutional neural network to tackle pixel-level tracking problem. The proposed model accepts two video frames as well as the tracking result of the previous frame as input. It introduces two branch of sub-networks which can capture and analyse low-level motion variance and high-level semantic variance respectively. The low level branch focuses on the movements of local parts of the target by extracting and operating optical flow data, while the high-level semantic branch outputs the prediction of to and fro alternation between background and target for each pixel in the current frame. Both branches employ fully convolutional neural networks for processing. Combining these two, the foreground target area is obtained and can be calculated to carry on the tracking operation for new frames.

It is important to differentiate this work from existing attempts to neural network based object tracking and video segmentation. The two most noteworthy innovations proposed in this paper are (i) pixel-level object tracking and (ii) category independent, generic object tracker. Instead of fitting the network to the appearance of any specific object class given at *training time*, we train the network to identify objects given at *runtime*. Our aim is the *temporal relation* between consecutive observations of a target belonging to *any object class*. Therefore, the rationale behind the design above is to let the network acquires the *behavior* of "following the appearance represented by previous target mask", instead of the appearance itself. Unlike many learning based segmentation or tracker, the parameters of the proposed tracker network are fixed by training and need no update when deployed. The trait makes our technique desirable in mass production scenarios such as embedding the tracker to low-cost mobile devices with limited computational resources. Experiment results show that our method exhibits excellent performance when compared with state-of-art trackers.

2 Related Work

Tracking methods aim at learning the latest appearance of the target which changes throughout time [6]. L1APG [7] employs multiple images patches cropped around the target in recently tracked frames for building the appearance model. In a more abstract way, CSK [8] uses translation filters to encode the state of the target. Then DSST [9] adds scale filter, which is independent of the translation filter, into the scheme. It provides more accurate scale estimation to

exclude corruptions from the background. STRUCK [10] transfer tracking into a classification task. The appearance model is updated with an on-line learning classifier. The convolutional neural network (CNN) is an ideal model for tracking task. Since the spatial resolution is different among convolutional layers, it naturally encodes the low-level visual features with high-level semantic information to build a robust appearance model for the target. Thus CF2 [11] combines CNN with KCF filter [12] to boost the accuracy of tracking. Siamese Tracker [13] matches the initial path of the target in the first frame with candidates in new frames and return the result by matching algorithm. With the help of deep learning, siamese-fc [14] trained on millions of images (ImageNet) can generate the result with only one forward operation. However, the tracking result is depicted in bounding boxes which only provide location and scale information of the target. It lacks semantic information and inevitably contains corruptions from background areas.

Algorithms based on video segmentation illuminate us about how to acquire a more accurate representation of the target, since these methods output the specific shape of target together with its location. To acquire more robust performance, most video segmentation methods take both visual and temporal information as input. Compared with single image segmentation, the temporal information is key for capturing the latest stage of a target. For instance, [15] use unsupervised motion-based segmentation on videos to obtain segments and FusionSeg [16] adapts optical flow as temporal hint. Different from above, Osvos [17] do not use any temporal information and process each frame independently as they are uncorrelated. Thus the performance of Osvos is strongly depended on the pre-trained models developed upon millions of images. However, the performance of these segmentation methods is restricted by lacking densely labeled training data. Thus [18] generate artificial masks by deforming the annotated mask via affine transformation as well as non-rigid deformation via thin-plate splines. [16] gets hypothesized foreground regions from bounding boxes to generate training samples. However, a single object may display multiple motions simultaneously. To learn the rich signals in unconstrained images, sufficient training data is necessary for video segmentation methods.

Our method is different from one-shot learning based trackers. These trackers employ a quick tuning upon observing the target object, which often dubbed as one-shot learning or appearance model [7,17]. Our work is also different from zero-shot learning method [21]. Zero-shot needs an intermediate description to extrapolate to novel classes, which is not applicable to tracking.

3 Generic Pixel Level Tracker

Our aim is to build a category-independent model to track targets given at run time. In particular, we capture the low-level motion variance to provide an intuitive estimation of the movement of each local part of the target, and represent the overall change of the distribution of foreground pixels by introducing high-level target-specific semantic variance. Thus we introduce a bi-channel neural

network to process both of the variances for producing a pixel-level tracking result. In particular, the network consists of two processing branches: one for robust prediction of low-level optical flow and the other for tracking high-level semantic objects. Both branches employ the deep fully convolutional network (FCN) architecture [3]. Figure 1(a) shows the structure of the network. The low- and high-level branches share the input of a pair of consecutive video frames, with the high-level branch additionally taking the target object mask in the previous frame. Then after a series of convolutional and de-convolutional feed-forward operations, the high-level semantic branch outputs the predicted target object mask in the new frame. The prediction is enhanced by fusing information from the low-level branch, which outputs predicted optical flow summarised in super-pixels by clustering.

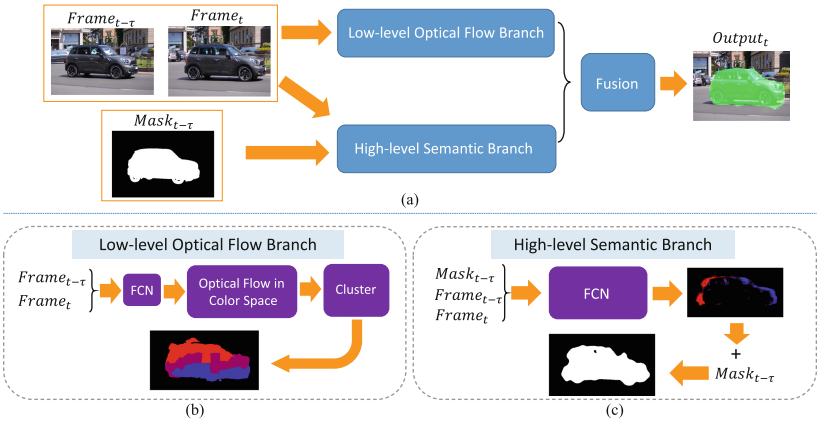

Fig. 1. The processing flow of the bi-channel fully convolution neural network. (a) Based on the input information, low-level and high-level temporal information are extracted and analysed in corresponding branches. By fusing the results of two branch, foreground area of the target can be identified. (b) In low-level branch, the optical flow data is extracted by a fully convolutional neural network with clustering operation afterwards, so that foreground and background areas can be separated. (c) High-level branch adopts the fully convolutional neural network to predict the decrease and increase (red and blue) of foreground mask of the target. By adding the predictions to the previous foreground mask, an initial estimation of the target can be obtained. (Color figure online)

3.1 Low-Level Optical Flow Branch

We define that the low-level motion variance represents the displacements of the same pixel in two adjacent frames. Particularly, optical flow is an ideal description for such variance since a flow of light and colors directly indicates the low-level visual changes of a moving target in the video. However, the raw flow data cannot be directly used to predict the mask of a tracking target, due to

following limits. First, the raw flow contains noise from the background and would be scattered when corruptions like occlusion appear on the image. Second, when one object moves in diversified speeds and directions, the raw flow will present different features and may confuse the judgment of the algorithm. Third, different parts of a single object may present utterly different optical flow features.

Considering above, we design the low level branch to extract and manage the optical flow for getting an output where the foreground and background areas are distinguished from each other. To accomplish this, we first refer to a deep convolutional neural network based on FCN to extract the optical flow considering the high speed and accuracy. The network has a similar structure with FlowNetC and FlowNetS provided by Flownet [20]. The number of channels is reduced to make a trade for better time efficiency. After that, this branch would process the obtained optical flow using the following steps. Step1: the flow data represented in angle and amplitude are mapped into color images. Step2: optical flows with different attributes (angle, amplitude) are clustered into superpixels, so that the underlying correspondence between flow data and the target can be revealed. Step3: optical flows clustered by the frames at different time intervals are combined, to reduce the impact of variance in moving speed. Figure 1(b) illustrates this process of generating the optical flow summarized in groups by clustering.

3.2 High-Level Semantic Branch

In high-level branch, we introduce the fully convolutional neural network to update the parsing of object/scene semantics in each new frame regardless of its category. We call this responsible sub-network as "semantic branch".

Mathematically, suppose $M_{t-\tau}$ and M_t are foreground areas at time $t-\tau$ and t respectively. For a pixel located at (x, y), the related semantic variance during time interval τ is marked as $d_{x,y,\tau}$. Then the relationship of $d_{x,y,\tau}$ and M_t can then be written as:

$$M_{x,y,t} = f(M_{x,y,t-\tau} + d_{x,y,\tau}) \tag{1}$$

where f is the operation that constrains the values of the changed foreground pixels to lie in $[0, 1]$.

In this branch, we introduce a deep convolutional neural network to directly capture the difference between M_t and $M_{t-\tau}$. Unlike segmentation based algorithms which need prior knowledge as a reference of the foreground area, the proposed network does not need fine-tuning on the first frame to learn the target's appearance from zero. The detailed design of this branch is shown in Fig. 1(c). The inputs include consecutive video frames and tracking results on previous frames. The former contains rich difference information while the latter gives a reference for the location of the target. Three kinds of pixel-level labels (0, 1, and 2) are designed for the network to reflect what happens between input images (colored in red, black and blue in Fig. 1(c)). If the target mask covers one pixel in

the former image but excludes the pixel at the same location in the latter image, label 0 is assigned to the pixel to represent target vanish on it. On the contrary, label 2 will be assigned to such a pixel which is newly added to the target mask in latter images. Label 1 covers the rest situations: the attribution of the pixel does not change during the interval between images. It remains to the target or background during the time-slot. The basic architecture of the neural network is based on FCN [3] except that batch normalization is introduced to stabilize the training procedure. In addition, to capture more details about the variance, the feature maps are upsampled to the input image size. Furthermore, multiple image pairs of different time intervals are loaded to better capture the change. The branch generates a foreground probability map at last.

3.3 Fusion

Based on the observation that the outputs two branches share locations on the image, the output of high-level semantic branch can be directly enriched with flow data at the same location given by the low-level optical flow branch. By fusing the outputs of two branches, we obtain the appropriate tracking results.

The detailed algorithm can be summarized using a four-stage procedure. In the first stage, we perform a voting scheme on the optical flow in groups according to the foreground probabilities at the shared location. In the second stage, we distinguish out foreground clusters and background clusters based on a threshold, with an appearance descriptor constructed for each group. In this work, the appearance descriptor is the average value of the attribute of corresponding optical flow. Then the third stage discards the foreground areas predicted in stage 1 if its appearance descriptor is close to the appearance descriptor of background clusters. In the last stage, the overall tracking result is generated by merging the identified foreground clusters together and being smoothed among temporal and spatial axis.

4 Experiment

4.1 Implementation Details

The convolutional network of semantic branch has been modified from that of FCN, and the architecture is illustrated in Fig. 2. We introduce batch normalization after every convolution layer of the network. Also, we employ five upsampling operations to make the final output the same shape with the input image. When training the network, we additionally introduce an auxiliary loss function on the top of the fifth convolution layer to make the training more stable. For the convolutional network used by the optical flow branch, we use the pretrained network parameters instead of fine-tuning the net on DAVIS [19]. We use the thin models which have $\frac{3}{8}$ of the channels corresponds to FlowNetS and FlowNetC.

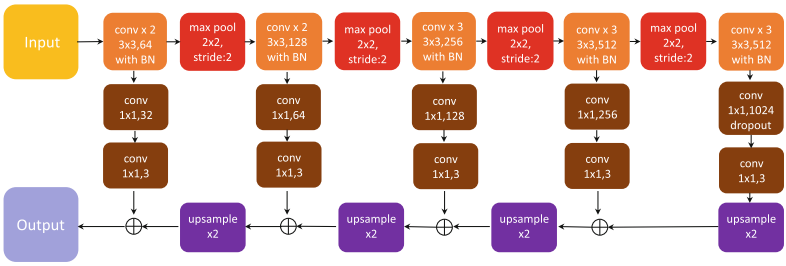

Fig. 2. Architecture of CNN of semantic branch. We add batch normalization to the five convolutional layer adapted from FCN. Five up-sampling operations are applied to make the final output the same shape with the input image.

The source code of this work will be accessible to on[1]. Please refer to our project page to see all the experiment results[2].

4.2 Data and Evaluation

In this work, we evaluate the proposed tracking method, along with several state-of-the-art trackers on the densely annotated dataset for video trackers [19] (DAVIS dataset). The video contains challenges such as fast motion, shape complexity, and deformation. Besides, the pixel-accurate annotations are ideal for our requirements. Using the DAVIS, we have 30 video clips of training, which include 2079 images. To illustrate the detailed performance of each method on different kinds of tracking conditions, we randomly pick out another 15 video sequences from the remaining set of DAVIS as our evaluation set. The target in our evaluation set can be a single object like a dancing girl. It can also be multiple objects that connected with each other, for example, the *soapbox* video. Since our method is based on bi-channel FCN, we call it FCN^2 tracker.

We refer to the pixel-level ROC curve as the basic evaluation metric. The ROC curve refers to receiver operating characteristic curve, where true positive rates are plotted against false positive rates at various threshold settings, which correspond to y- and x-axis respectively. In particular, our model gives pixel-by-pixel predictions of class probability, ROC is calculated by varying the classification threshold θ, (i.e. $I_{i,j}$ is predicted as target if $P(I_{i,j} = target) > \theta$. For trackers representing target using bounding boxes, say, a tracker predicting a box B^*, we generate a series of boxes, centred at the centre of B^*, with varying sizes $\{B_1, B_2, \ldots\}$. ROC curve for the tracker is calculated by predicting target as pixels within B_1, B_2, respectively.

Our performance is compared with state-of-art trackers: siamese-fc [13], CF2 [11], CSK [8], STRUCK [10], DSST [9], and L1APG [7]. Figure 3 presents the results of the compared trackers in bounding boxes and the proposed method in

[1] https://github.com/chenzj2017/TBD_tracker.
[2] https://sites.google.com/site/tbdtracker2017/.

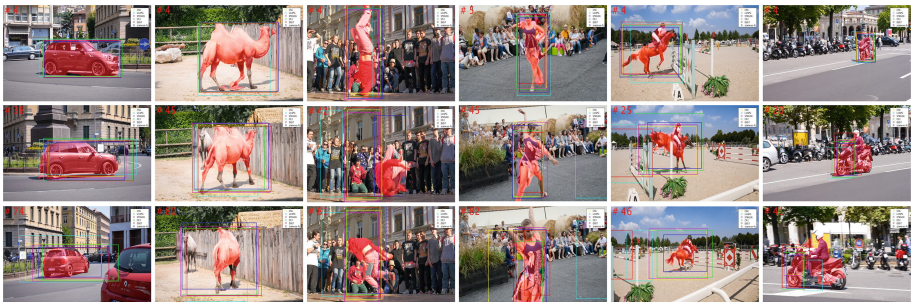

Fig. 3. Qualitative comparison among trackers. Our output is marked in red shadow. The result of other trackers are shown by bounding boxes. (Color figure online)

probability map. The presented frames come from 6 challenging video sequences which include in-plane rotation, large-scale deformation, ambiguous edge and so on. The illustrated results demonstrate that our method is robust to a various challenging transformation of the target while other trackers become

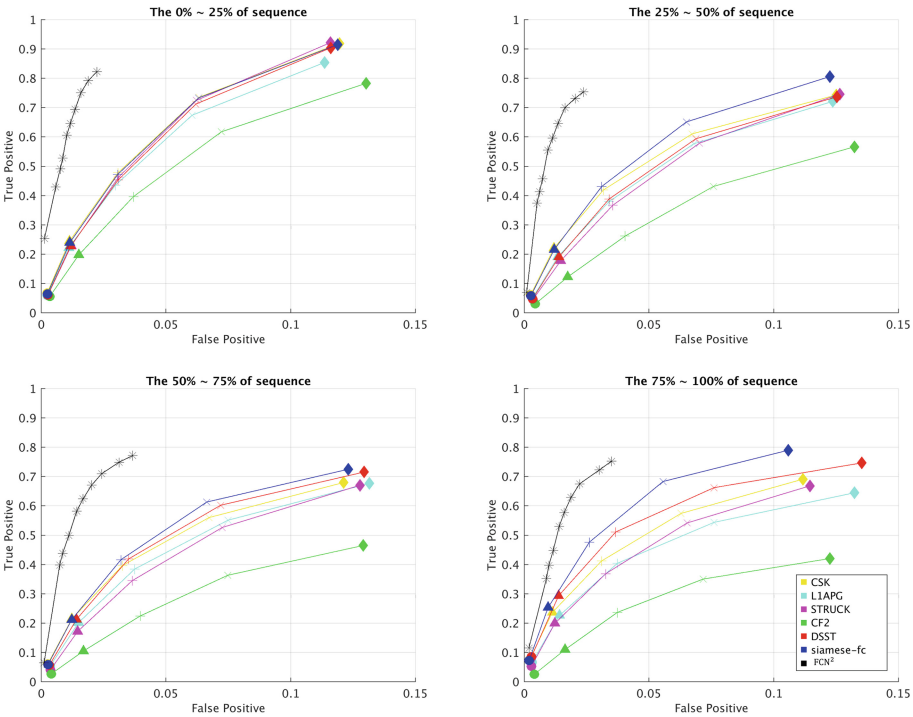

Fig. 4. ROC. Our output is shown by blacklines marked with stars. The rest of other trackers are shown by curves in color. (Color figure online)

Table 1. IoU (in pixels). **Blod** fonts indicate the best performance. Abbreviation of video name is used to save space. Please refer to our project page for the full names.

Methods	bkswan	bdance	camel	car-rd	car-sh	cows	dance-t	drift-s	horsej	kite-s	libby	motoc	para-l	scoot	soapb	*Ave*
Osvos	0.34	0.14	0.66	0.69	**0.91**	0.56	0.27	**0.53**	0.48	**0.64**	**0.57**	**0.74**	**0.61**	0.39	**0.62**	0.54
FusionSeg	0.32	0.48	0.54	0.73	0.67	0.34	0.55	0.37	0.57	0.22	0.19	0.47	0.39	**0.55**	0.56	0.46
FCN^2	**0.59**	**0.58**	**0.69**	**0.80**	0.72	**0.77**	**0.55**	0.43	**0.62**	0.43	0.50	0.39	0.51	**0.67**	0.42	**0.58**

quite vulnerable. For example, when tracking the dancers, many trackers cannot tightly cover the target due to significant deformations. Instead, the proposed method can still predict precise foreground layout for the target.

Figure 4 shows the ROC of our algorithm and compared trackers. Each frame has its own ROC. However, we only report the average value of ROC to present a statistic result. To illustrate the performance for evaluated methods during different periods of the video sequence, we divide each video sequence into four separate parts according to arrival orders. The results presented in the figure supports that the proposed algorithm achieved superior tracking performance, which is consistent with the intuitive assessment shown in Fig. 2. In specific, with tracking through more frames, the ROCs of all trackers deteriorate due to drifting and failures. Nevertheless, FCN^2 tracker remains superior to rival methods.

In addition, we also compare out method to competing segmentation techniques, including Osvos [17] and FusionSeg [16]. The Osvos model used to compare is first pre-trained on ImageNet and then trained on DAVIS training set. The results of FusionSeg are generated by *Ours-M* model. The average Jaccard scores, which computes the intersection over union (IoU) between the predicted pixels and ground-truth, are shown in Table 1. Different from the compared algorithms, our method does not rely on a large-scale dataset for training, and the presented statistics show that we can still achieve the highest score in more than half of the videos, demonstrating the effectiveness of the proposed method.

5 Conclusion

We present a new approach for visual object tracking based on bi-channel FCN that (1) produce finer tracking result and (2) works for the generic object without fitting the network to the appearance of any specific object class which needs a large scale of training data. Our model can extract the temporal relationship between two observations of a target which works together with optical flow information to produce a robust tracking result. In future work, we plan to explore extensions that could encode more change of semantic information of the tracking target.

Acknowledgments. This work is supported by Big Massive Open Online Course (MOOC) Data Retrieval and Classification Based on Cognitive Style.

References

1. 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)
2. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: IEEE International Conference on Computer Vision, pp. 4489–4497. IEEE Press, New York (2015)
3. 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. IEEE Press, New York (2015)
4. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 534–549. Springer, Cham (2016). doi:10.1007/978-3-319-46466-4_32
5. Levi, D., Garnett, N., Fetaya, E., Herzlyia, I.: StixelNet: a deep convolutional network for obstacle detection and road segmentation. In: British Machine Vision Conference, p. 109-1. BMVC Press (2015)
6. Shen, S.-C., Zheng, W.-L., Lu, B.-L.: Online object tracking based on depth image with sparse coding. In: Loo, C.K., Yap, K.S., Wong, K.W., Beng Jin, A.T., Huang, K. (eds.) ICONIP 2014. LNCS, vol. 8836, pp. 234–241. Springer, Cham (2014). doi:10.1007/978-3-319-12643-2_29
7. Mei, X., Ling, H., Wu, Y., Blasch, E.P., Bai, L.: Efficient minimum error bounded particle resampling L1 tracker with occlusion detection. IEEE Trans. Image Process. **22**, 2661–2675 (2013)
8. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33765-9_50
9. Danelljan, M., Hger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference. BMVC Press (2014)
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. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3074–3082. IEEE Press, New York (2015)
12. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
13. 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. IEEE Press, New York (2016)
14. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 850–865. Springer, Cham (2016). doi:10.1007/978-3-319-48881-3_56
15. Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, New York (2017)

16. Jain, S., Xiong, B., Grauman, K.: FusionSeg: learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, New York (2017)
17. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, New York (2017)
18. Khoreva, A., Perazzi, F., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. arXiv preprint arXiv:1612.02646 (2016)
19. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 724–732. IEEE Press, New York (2016)
20. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., et al.: Flownet: learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision, pp. 2758–2766. IEEE Press, New York (2015)
21. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4166–4174. IEEE Press, New York (2015)