Convolutional LSTM Based Video Object Detection

Xiao Wang[1,2,3], Xiaohua Xie[1,2,3](✉), and Jianhuang Lai[1,2,3]

[1] School of Data and Computer Science, Sun Yat-sen University,
Guangzhou, China
xiexiaoh6@mail.sysu.edu.cn

[2] Guangdong Key Laboratory of Information Security Technology, Guangzhou,
China

[3] Key Laboratory of Machine Intelligence and Advanced Computing
of the Ministry of Education, Guangzhou, China

Abstract. The state-of-the-art performance for object detection has been significantly improved over the past two years. Despite the effectiveness on still images, something stands in the way of transferring the powerful detection networks to videos object detection. In this work, we present a fast and accurate framework for video object detection that incorporates temporal and contextual information using convolutional LSTM [27]. Moreover, an Encoder-Decoder module is made up based on the convolutional LSTM to predict the feature map. It is an end-to-end learning framework and is general and flexible when combining with still-image detection networks. It achieves significant improvement on both speed and accuracy. Our method significantly improves upon strong single-frame baselines in ImageNet VID [21], especially for more challenging moving objects at high speed.

Keywords: Video object detection · Convolutional LSTM
Encoder-Decoder module

1 Introduction

Deep learning has achieved significant success and been widely applied to various computer vision tasks such as image classification [7,25], object detection [1,3,4,17], semantic segmentation [6,13], video representation [14], dense captioning [8], etc. In the case of object detection, the performance has made a huge leap forward with the success of deep Convolutional Neural Networks (CNN). To make the object detection more challenging, ImageNet introduced a new task for object detection from videos (VID), which brings object detection from still image into the video domain. In this task, the object detection system is required to give the position and the class of the objects in each frame. VID play an

X. Xie—This project is supported by the Natural Science Foundation of China (61573387, 61672544), Guangzhou Project (201807010070).

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11257, pp. 99–109, 2018.
https://doi.org/10.1007/978-3-030-03335-4_9

important helping role in a number of applications on video analysis such as video representation, video caption and object tracking.

However, existing methods focus on detecting objects in still images, and directly applying them to solve the video object detection is clumsy. Different from the ImageNet object detection (DET) challenge in still image, VID shows objects in image sequences and comes with additional challenges such as motion blur due to rapid camera or object motion, illumination variation due to scene changing or different camera angles, partial occlusion or unconventional object-to-camera poses, etc (See some examples in Fig. 1). The broad range of appearances varying in video make recognizing the class of object more difficult. Besides, video is a kind of data with high density, which raises a higher demand to object detector's speed and accuracy.

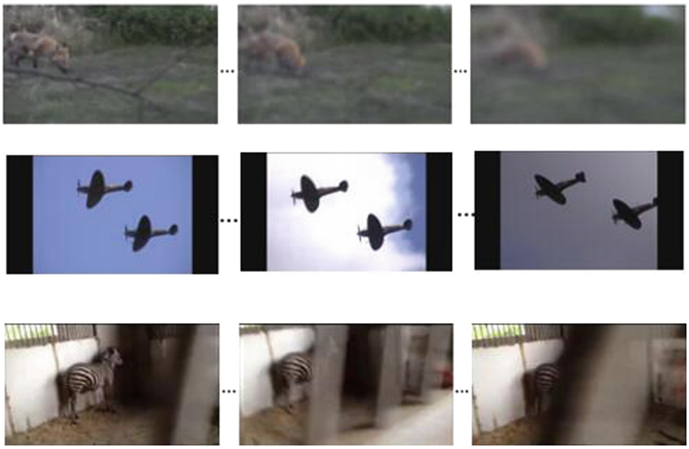

Fig. 1. Example special video images with motion blur, illumination variation and occlusion, respectively.

Although difficulties arise, videos have more rich temporal information than still image. How to exploit the relation among the image sequences become the crux of the video object detection methods. We seek to improve the video object detection quality by exploiting temporal information, in a principled way. As motivated by the success in precipitation nowcasting [27], using convolutional LSTM (ConvLSTM), we propose to improve the detection by spatiotemporal aggregation. Note that the ConvLSTM has convolutional structures in both the input-to-state and state-to-state transitions, so it can work with 2D spatial feature maps and solve the spatiotemporal sequence forecasting problem. This suggests that it may fit the video image sequences.

In this work, we propose a unified framework based on the ConvLSTM to tackle the problem of object detection in realistic video. The framework consists of three main modules: (1) firstly, a fully convolutional network, which can be some general pre-trained ImageNet models such as googlenet [25], resnet [7], to generate the feature map; (2) then a Encoder-Decoder module composed by two ConvLSTMs, one for the input feature maps of adjacent frames and another for the output feature map; (3) task module including RPN [3], the final classification subnetwork and regression subnetwork, just like the other two-stage detector. Finally, the entire architecture can be trained end-to-end.

2 Related Work

Object Detection from Still Image. State-of-the-art methods for general object detection [1,3,6,17,19,20] are mainly based on deep CNNs. In general, the detection networks are divided into two kinds according to whether the region proposals are needed. First, one-stage network that directly predict boxes for an image in one step such as YOLO [19], SSD [12] and second, two-stage network with Region Proposal Network such as Fast R-CNN [3], Faster R-CNN [20], R-FCN [1].

Our approach builds on R-FCN [1] which is a simple and efficient framework for object detection on region proposals with a fully convolutional nature. Unlike the Faster R-CNN [20], R-FCN reduces the cost for region classification by pushing the region-wise operations to the end of the network with the introduction of a position-sensitive RoI pooling layer which works on convolutional features that encode the spatially subsampled class scores of input RoIs.

Object Detection in Video. Since the object detection from video task has been introduced at the ImageNet challenge in 2015, it has drawn significant attention. Kang et al. [9,10] combined the still-image object detection with general object tracking method and proposed a tubelet proposal network to propagates predicted bounding boxes to neighboring frames and then generates tubelets by applying tracking algorithms from high-confidence bounding boxes. Seq-NMS [5] constructs sequences along nearby high-confidence bounding boxes from consecutive frames. Differing from these box-level post-processing methods, Zhu et al. [29,30] utilized a optical flow ConvNet for propagating the deep feature maps via a flow field instead of the bounding box.

Sequence Modeling. Recurrent neural networks, especially Long Short-Term Memory (LSTM), have been adopted to address many video processing tasks such as action recognition [16], video summarization [28],video representations [23] and object tracking [15]. However, limited by the fixed propagation route of existing LSTM structures where the input, cell output and states are all 1D vectors, most of these previous works can only learn some holistic information, which is impractical for image data.

Some recent approaches develop more complicated recurrent network structures. For instance, to apply the LSTM to image sequence, the ConvLSTM [27] was proposed for video prediction. In our method, we exploits spatiotemporal information by using ConvLSTM. Besides, the entire system is end-to-end trained for the task of video object detection.

3 Method

In this section, we first give an overview of object detection from video (Sect. 3.1) including the task setting and some base elements in the task. Then we give a detailed description of our framework design (Sect. 3.2). Section 3.3 describes the major component Encoder-Decoder module and introduces how to exploit the spatiotemporal information using ConvLSTM.

3.1 Overview

The ImageNet object detection from video (VID) task is similar to image object detection task (DET) in still images. There are 30 classes, which is a subset of 200 classes of the DET task. Given the input video images I_t where t is the time, the algorithms need to produce a set of annotations (r_t), which include class labels, confidence scores and bounding boxes. Therefore, a baseline approach is to apply an off-the-shelf object detector to each frame individually.

Most of the two-stage detection network include two major components: (1) a feature extraction subnetwork N_{feat} composed by a common set of convolutional layers which can generate the feature map $f_t = N_{feat}(I_t)$ on the input image; (2) a task-specific subnetwork N_{task} which executes the specific task such as classification, regression to output the result $r_t = N_{task}(f_t)$. Consecutive video frames are highly similar, likewise, their feature maps have a strong correlation. How to use the correlation information is what we present in the following sections.

3.2 Model Design

The proposed architecture takes every other frame $I_t \in R^{H_i \times W_i \times 3}$ at time t, and pushes them through a backbone N_{feat} (i.e. ResNet-101 [7]) to obtain feature maps $f_t \in R^{H_f \times W_f \times C_f}$ where H_f, W_f and C_f are the width, height and number of channels of the feature map, and then output the result r_t though the N_{task}. Our overall system builds on the R-FCN [1] object detector, specifically, the ResNet-101 models pre-trained for ImageNet classification as default. It works in two stages: first extracts candidate regions of interest (RoI) using a Region Proposal Network (RPN) [20]; and, second, performs region classification into different object categories and background by using a position-sensitive RoI pooling layer [1]. That is to say that every other frame needs to go through the whole R-FCN and get the result.

Let us now consider the other frames, which are not processed by the whole R-FCN. We extend this architecture by introducing a module named Encoder-Decoder to propagate the feature maps. It figures out how to properly fuse the features from multiple frames to get the feature map of current frame. Besides, we can control how many frames we want to fuse by defining the parameter T. Besides, to make the prediction more robust, a convolution layer follows behind the decoding ConvLSTM. Obviously, the module is much faster than the feature network. They are elaborated below (Fig. 2).

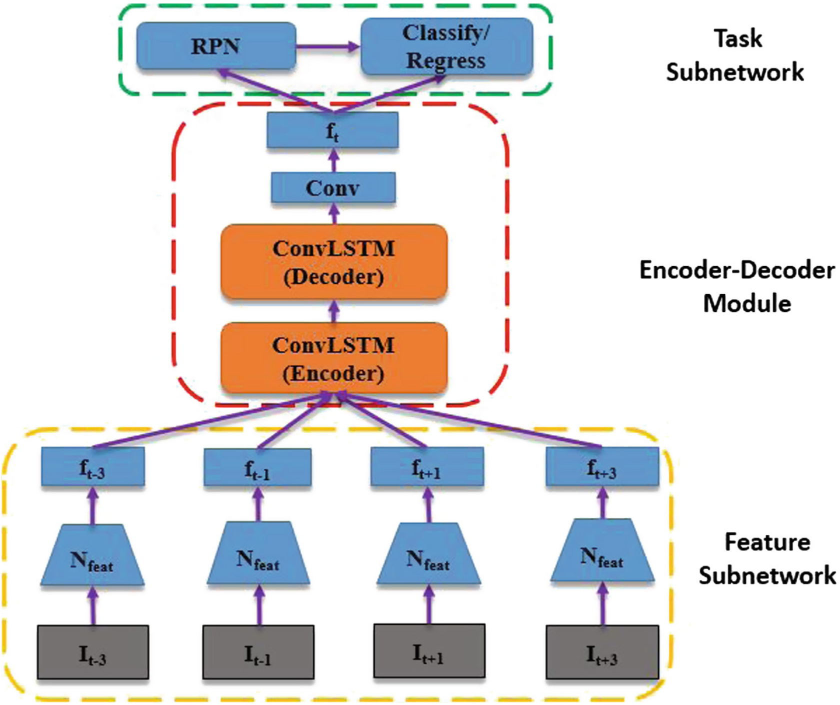

Fig. 2. Our proposed architecture based on ConvLSTM Encoder-Decoder module (see Sect. 3 for details).

3.3 Encoder-Decoder Module

The framework [24] provides a general framework for sequence-to-sequence learning problems, which include two stage: one to read the input sequence and the other to extract the output sequence, and its ability to capture long-term temporal dependencies makes it a natural choice for this application. Our spatiotemporal sequence, we use the Encoder-Decoder structure like in [24]. During the encoding step, use one ConvLSTM to read the input sequence feature maps, one timestep at a time, to compresses the whole input sequence into a hidden state tensor, and then to use another ConvLSTM to conduct the hidden state to give the prediction.

The equation of ConvLSTM are shown in Eqs. (1, 2) below, where '$*$' denotes the convolution operator. All the input-to-state kernels w_h and state-to-state kernels w_x are of size $3 \times 3 \times 512$ with the 1×1 padding. They are all randomly initialized. As we can see, the module is characterized by fewer parameters than the convolution feature network and the flow method [2,29]. Moreover, it is convenient to change the dependency scope, just adjust the parameter T.

For the start states, before the first input, we initialize the c_0 and h_0 of the encoding ConvLSTM to zero which means "no history", and the input x_t at each timestep are corresponding feature map. As well the initial state c_0 and cell output h_0 of the decoding ConvLSTM are copied from the last state of the encoding network, but its input are zeros.

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} \begin{pmatrix} w_h \\ w_x \end{pmatrix} * \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b \tag{1}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t, \;\; h_t = o_t \cdot tanh(c_t) \tag{2}$$

4 Experiments

4.1 Setup

ImageNet VID Dataset [21]**.** It is a prevalent large-scale benchmark for video object detection. Following the protocols in [10,30], model training and evaluation are performed on the 3,862 video snippets from the training set and the 555 snippets from the validation set, respectively. The snippets are fully annotated, and are at frame rates of 25 or 30 fps in general. There are 30 object categories. They are a subset of the categories in the ImageNet DET dataset. During training, besides the ImageNet VID train set, we also used a subset of the ImageNet DET train set which include the 30 categories.

Implementation Details. We use the stride-reduced ResNet-101 with dilated convolution in conv5 to reduce the effective stride and also increase its receptive field. The RPN is trained at 15 anchors corresponding to 5 scales and 3 aspect ratios, and apply non-maximum suppression (NMS) with an IoU threshold of 0.7 to select the top 300 proposals in each frame for training/testing our R-FCN detector. Then, like the Focal Loss [11] and online hard example mining method [22], we also select a certain number of hard region (with high loss) from the proposals produced by the RPN to make training more effective and efficient. By setting different weights for hard and non-hard proposals, the training can puts more focus on hard proposals. Note that, in this strategy, data forward and gradient backforward propagate through the same network.

In both training and testing, we use single scale images with shorter dimension of 400 pixels. In SGD training, 4 epoches (400K iterations) are performed on 2 GPUs, where the learning rates are 10^{-4} and 10^{-5} for the first 3 epoches and the last 1 epoch iterations, respectively.

For testing we apply NMS with IoU threshold of 0.3. For better analysis, the ground truth objects in validation set are divided into three types: slow, medium, fast according to their motion speed, just like [29] and we also report their mAP scores respectively, so we can do a more detailed analysis and in-depth understanding.

4.2 Results

Overall Results. Method R-FCN is the still-image method baseline which is trained on single-frame using ResNet-101. Note that we train the network on only two GPUs and do not add bells and whistles like multi-scale training/testing in order to facilitate comparison and draw clear conclusions. We investigate the effect of T, however, limited by the memory, we only test $T = 1, 2$ for encoding ConvLSTM. From the Table 1, the performance for single-frame testing is 73.19% mAP, but rises to 74.5% with our ConvLSTM based method. This 1.3% gain in accuracy shows that the ConvLSTM can effectively promotes the information from nearby frames in feature map lavel. Besides, when T increases (from 1 to 2), the performance also has an obvious growth (from 73.65% to 74.5%). As to runtime, the proposed ConvLSTM based method has about twice as fast, which is in accord with theory. Some example results are shown in Fig. 3.

Table 1. Performance comparison on the ImageNet VID validation set. The average precision (in %) for each class and the mean average precision over all classes is shown.

Method	airplane	antelope	bear	bicycle	bird	bus	car	cattle	dog	cat	elephant	fox
still(R-FCN)	88.11	83.26	83.33	63.55	70.29	74.40	56.81	69.64	74.15	78.98	77.06	89.64
ConvLSTM-based(T=1)	88.70	82.35	83.67	63.66	70.81	75.27	57.44	68.89	72.73	77.93	77.09	89.96
ConvLSTM-based(T=2)	89.30	83.43	84.21	64.75	71.61	76.54	58.29	69.95	73.56	78.86	77.67	90.55

Method	giant-panda	hamster	horse	lion	lizard	monkey	motor-cycle	rabbit	red-panda
still(R-FCN)	80.51	85.56	69.57	47.22	76.64	49.09	81.75	60.89	83
ConvLSTM-based(T=1)	81.2	87.0	69.36	54.64	76.94	47.99	81.72	62.78	82.72
ConvLSTM-based(T=2)	81.88	87.98	70.33	53.34	77.37	49.33	82.48	63.77	83.29

Method	sheep	snake	squirrel	tiger	train	turtle	water-craft	whale	zebra	**mAP(%)**	**speed(fps)**
still(R-FCN)	54.49	71.37	48.77	91.06	77.43	77.86	66.67	74.14	90.41	73.19	4.08
ConvLSTM-based(T=1)	56.55	71.9	48.2	91.27	78.5	78.4	67.03	74.46	90.36	73.65	7.9
ConvLSTM-based(T=2)	57.37	72.63	49.09	91.83	79.3	79.17	67.93	75.06	91.11	74.5	7.8

Table 2. Comparison of various approaches.

Method	mAP (%)
R-FCN	73.19
R-FCN + conv	73.25
ConvLSTM (T = 2)	74.5

Table 3. Detection accuracy of different motion speeds.

Method	mAP (%) slow	mAP (%) medium	mAP (%) fast
R-FCN	82.5	71.8	51.2
ConvLSTM-based (T = 1)	82.6	72.4	51.7
ConvLSTM-based (T = 1)	82.6	74.1	52.8

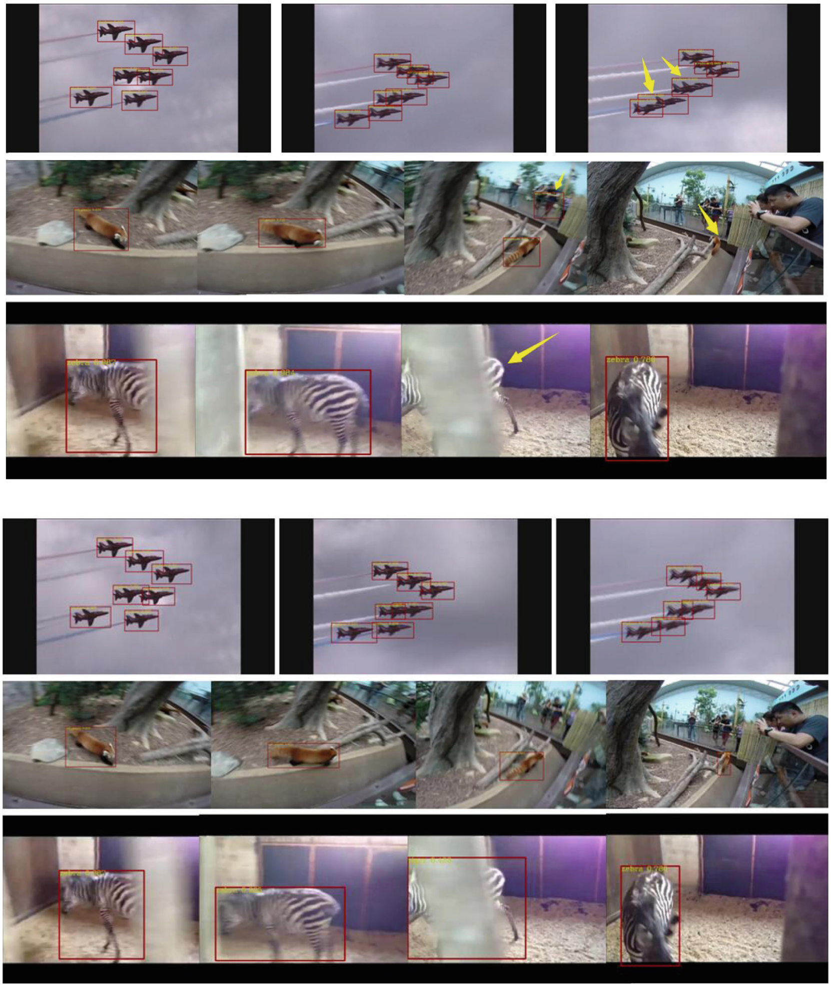

Fig. 3. Example video clips where the proposed ConvLSTM based method improves over the single-frame baseline (using ResNet-101). The first three lines are results by single-frame baseline and the last three lines are results by the proposed method.

When comparing our 74.5% mAP against the other methods, we make the following observations. The ILSVRC 2015 winner [9] combines two Faster R-CNN detectors, multi-scale training/testing, context suppression, high confidence tracking [26] and optical-flowguided propagation to achieve 73.8%. The deep feature flow [30], a recognition ConvNet (ResNet) is applied to key frames only and an optical flow FlowNet [2] is used for propagating the deep feature maps via a flow field to the rest of the frames, achieve 73.1% mAP at a higher detection speed.

Ablation Study. To take out the effect of the increased parameter size, we replace the ConvLSTM with two convolution layers, the Table 2, shows it only has a small increase in mAP. The fact is enough to prove that is ConvLSTM with gates control that aggregate the information in the image sequence.

Motion Speed. Evaluation on motion groups (Table 3) shows that detecting fast moving objects is very challenging: mAP is 82.5% for slow motion, and it drops to 51.2% for fast motion. It shows that "fast motion" is an intrinsic challenge and it is critical to consider motion in video object detection. When T changes, the medium speed objects improve the most increased by 2.3% (from 71.8% to 74.1%), while the fast have a little increment and the slow almost unchanged, that is to say T has a different influence on different speed. It is reasonable that T control the range of the dependence, when T increase, more motion information are catched.

5 Conclusion and Future Work

This work presents an accurate, end-to-end and principled learning framework for video object detection using ConvLSTM, and its main goal is to reach the accuracy-speedup tradeoff. Moreover, it would be complementary to existing box-level framework for better accuracy in video frames. More annotation data (e.g., YouTube-BoundingBoxes [18]) may be benefit to improvements. And there is still large room to be improved in fast object motion. We believe these open questions will inspire more future work.

References

1. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. CoRR, abs/1605.06409 (2016)
2. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2758–2766 (2015)
3. Girshick, R.: Fast R-CNN. arXiv preprint arXiv:1504.08083 (2015)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)

5. Han, W., et al.: Seq-NMS for video object detection. arXiv preprint arXiv:1602.08465 (2016)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision, ICCV, pp. 2980–2988. IEEE (2017)
7. 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)
8. Johnson, J., Karpathy, A., Fei-Fei, L.: DenseCap: fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4565–4574 (2016)
9. Kang, K., et al.: T-CNN: tubelets with convolutional neural networks for object detection from videos. IEEE Trans. Circ. Syst. Video Technol. (2017)
10. Kang, K., Ouyang, W., Li, H., Wang, X.: Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 817–825 (2016)
11. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. arXiv preprint arXiv:1708.02002 (2017)
12. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
13. 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)
14. Luo, Z., Peng, B., Huang, D.-A., Alahi, A., Fei-Fei, L.: Unsupervised learning of long-term motion dynamics for videos. arXiv preprint arXiv:1701.01821, 2 (2017)
15. Milan, A., Rezatofighi, S.H., Dick, A.R., Reid, I.D., Schindler, K.: Online multi-target tracking using recurrent neural networks. In: AAAI, pp. 4225–4232 (2017)
16. Ng, J.Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 4694–4702. IEEE (2015)
17. Ouyang, W., et al.: DeepID-Net: deformable deep convolutional neural networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2403–2412 (2015)
18. Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V.: YouTube-BoundingBoxes: a large high-precision human-annotated data set for object detection in video. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 7464–7473. IEEE (2017)
19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
20. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497 (2015)
21. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
22. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)
23. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using LSTMs. In: International Conference on Machine Learning, pp. 843–852 (2015)

24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
25. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)
26. 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)
27. Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., Woo, W.-C.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
28. Zhang, K., Chao, W.-L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 766–782. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_47
29. Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. arXiv preprint arXiv:1703.10025 (2017)
30. Zhu, X., Xiong, Y., Dai, J., Yuan, L., Wei, Y.: Deep feature flow for video recognition. In: Proceedings of CVPR, vol. 2, p. 7 (2017)