DT-3DResNet-LSTM: An Architecture for Temporal Activity Recognition in Videos

Li Yao[1,2] and Ying Qian[1]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, 211189, P.R. China
[2] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, 211189, P.R. China

Abstract. Due to participation in TRECVID ActEV[1] competition, we conducted research on temporal activity recognition. In this paper, we propose an architecture DT-3DResNet-LSTM to classify and temporally localize activities in videos. D represents that our system firstly detects objects in video frames. T represents that we use these detected results, as input to object tracking model, which can achieve data association information among adjacent frames of multiple objects. Lastly, we input clipped video frames of different objects into 3D Convolutional Neural Network to achieve features and then train a recurrent neural network that learns to classify video clips. What's more, we process the output of RNN model to get the final classification of input video and determine the temporal localization of input video.

Keywords: Activity recognition, 3D Convolutional Neural Network, LSTM.

1 Introduction

In recent years, activity recognition of videos has important applications in many scenarios, such as video surveillance, content-based video retrieval, and automotive autopilot technology.

Traditionally, video activity recognition [2,3,4,5] is completed by extracting the features from the video frames and building a mathematical model of the status corresponding to the features. Along with the rapid growth of deep learning, convolutional neural networks have been generally used in computer vision and activity recognition [6,7,8,9,10,11].

Significant progress has been made in research on video classification [11,12,13,14], which is supervised learning based on given labels. While this task has been very challenging, the current video datasets have been preprocessed to clear temporal information. However, a complete video recognition system needs to identify the activity in the unprocessed video and locate the start and end frames of the activity.

In this paper, we would like to solve the problem: given an unprocessed video, identify the activity and find the temporal localization of the activity. We proposed an

architecture to solve the problem in three processes, which is object detection, object tracking and activity classification. Using this and other related work as a baseline, we then make the following two contributions:

1.3DResNet LSTM network: we use 3D ResNet CNN [15] model pretrained in Kinetics [16] dataset to get the features of input video, and then we feed the feature into Long short-term memory (LSTM) [7] network to find the actual temporal localization. We show that combine CNN and RNN will get more accurate result of activity classification and temporal localization.

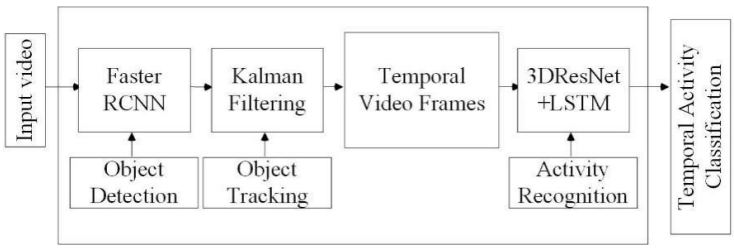
2.Object tracking in video: our proposed tracking model ignore this objects that are predicted to be the same object but has a far distance between frames. Also, we compute the Intersection over Union (IoU) between predicted bounding box and detected bounding box to get object type of tracking objects.

As far as we know, this work is the first time to combine deeper 3D CNNs with RNN for activity classification task. Previous research shows deeper 2D CNNs has a good performance on the ImageNet dataset [17]. However, it is not taken for granted that deeper 3D CNN will also perform well in video activity recognition because the number of video datasets is less than the number of image datasets. The results of this study, which indicate deeper 3D CNNs are effective on activity classification, can be expected to promote the development of video recognition. And combined with LSTM, we can more accurately find the temporal localization of activities.

2 Proposed Architecture

As Fig. 1. shows, we promoted an activity recognition architecture that contains three different sections. First, we use Faster-RCNN [22] to detect the target object in video frames. Second, we use Kalman filter [39] to track detected different objects and then generate numerous continuously clipped frames which contains tracked objects. Third, we use 3DResNet [15] and LSTM [7] to classify activities and accurately temporally localize activities.

In addition, in the Kalman filter target tracking process, we compute the IoU (Intersection over Union) between Kalman filter prediction frame and actual frame. The



predicted bounding box whose IoU value is greater than a certain threshold will be output, and the target class corresponding to the frame detected by the video frame is actually given, thereby improving the overall recognition accuracy.

Fig. 1. The proposed activity recognition architecture which contains three sub-processes. Object detection model is applied to generate the correct bounding box of objects. Object tracking model uses the results of object detection model and then track different objects among frames. Activity recognition model works as the final process of activity recognition architecture to output the final prediction of temporal localization and classification of input videos.

2.1 Faster RCNN Object Detection

Since the main targets of the ActEV surveillance video task are pedestrians and vehicles, target detection is an important basis for subsequent activity identification. We use the Faster RCNN [22] with VGG16 [40] as the bottom feature of the video frame for object detection. The Faster RCNN network framework is shown in Fig. 2. For an arbitrary input image, the VGG16 model is used to obtain image features. The last layer of feature map is conv5-3. The RPN (Region Proposal Network) network performs a 3×3 convolution on the conv5-3 layer, followed by a 512-dimensional full-connection layer, and the full-connection layer is followed by two sub-connection layers, which are used for the classification and regression of anchors, and then get the proposals through calculation screening. The anchors are a set of fixed-size reference windows, there are 3 scales and 3 aspect ratios. The ROI Pooling layer uses the generated proposal to extract the feature from the feature maps for pooling, and then enters the Fast RCNN network for classification and regression. Fast RCNN [21] identifies and classifies the proposals extracted from the RPN network, and then adjusts the regression parameters to obtain the precise location of the target.

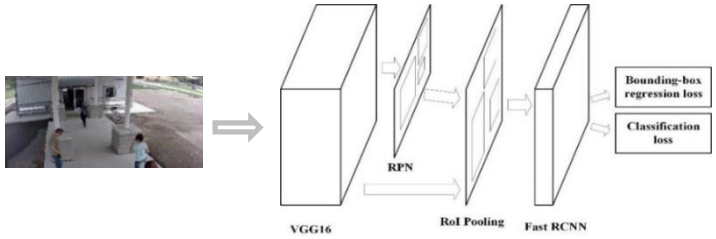


Fig. 2. The architecture of Faster RCNN. The features extracted by VGG16 were fed into RPN to generate proposals and also used with the generated proposals to ROI pooling. ROI pooling layer make the input proposals resize to the same output size and then feed into FAST RCNN to achieve classification and bounding box.

2.2 Kalman Filter Object Tracking

As Algorithm 1 shows, we use the results of Faster RCNN as the input of object tracking model. The system output is tracking positions and types of detected objects. The system records the previous processed frame number and to prevent the frame number of two adjacent frames is greater than a threshold value which can reducing the possibility of false tracking. What's more, when the frame number difference between two adjacent frames is smaller than a certain threshold, we compute the IoU (Equation 1) value between the predicted results and actual bounding box positions. If the IoU is bigger than a certain value, the tracking result will be added to final results.

The final results include valid frame number, object type and object position of detected objects in each video frame.

$$\mathrm{IoU} = \frac{\mathrm{DetectionResult} \bigcap \mathrm{GroundTruth}}{\mathrm{DetectionResult} \bigcup \mathrm{GroundTruth}} \tag{1}$$

We describe the object model, i.e. the representation and the motion model used to propagate a target's identity into the next frame. We estimate the inter-frame movements of each object via a linear constant velocity model which is irrelevant of other objects and camera motion. The status of each object is modeled as:

$$x = [u, v, s, r, u^{'}, v^{'}, s^{'}]^{T} \tag{2}$$

where u and v indicate the horizontal and vertical pixel location of the center of the target object, while the scale s and r indicate the scale (area) and the aspect ratio of the target object's bounding box respectively. When a detection is related to a target, the detected bounding box is used to regenerate the target status where the velocity elements are computed optimally through a Kalman filter [38]. If no detection is related to the target, its status is briefly forecasted without rectification using liner velocity model.

Algorithm 1: Detected objects tracking

```
Input: N video frames and information about detected objects which
       containing positions and object types.
Output: Tracking positions and types of detected objects.
Initialize previous frame number = 1;
for n ← 1 to N do
    if current frame number - previous frame number > 5 then
        Clean kalman filter tracking records;
    else
        Update predicted trackers;
        for t ← trackers do
            for d ← original positions do
                Compute IoU of t and d;
                if IoU ≥ 0.3 then
                    Add tracking results to final results.;
Return final tracking results.
```

2.3 3DResNet+LSTM Activity Classification

Previous research [16,41] shows that 3D CNNs does not perform well on UCF-101, HMDB-51, and ActivityNet datasets whereas 3D CNNs trained on Kinetics performs well. Deeper 3D CNNs may have good performance compared to shallow 3D CNNs. However, deep 3D CNNs have more parameters needed to learn through training, as a result, huge datasets are required to prevent overfitting when training this deep CNNs. Kinetics is a big enough dataset so we use this dataset to pre-train our ResNet model.

A basic ResNets block includes two convolutional layers, which is followed by batch normalization and a ReLU. A shortcut connection is between the input of the block and the layer before last ReLU model. To prevent many parameters needed to learning of superficial networks, we apply identity connections and zero padding for the shortcuts in basic blocks.

ResNeXt add a different component in terms of depth and breadth, which is called cardinality. Different from the original bottleneck block, the ResNeXt block partitions feature maps into small groups, which is called group convolutions. Cardinality represents the number of middle convolutional layer groups in the bottleneck block. In their study, Xie et al. showed that using more cardinality in 2D architectures can achieve more effectively compared with using wider or deeper ones [42]. In this study, we using the cardinality of 32 to assess the result of ResNeXt-101 on activity recognition, as shown in Fig. 3.

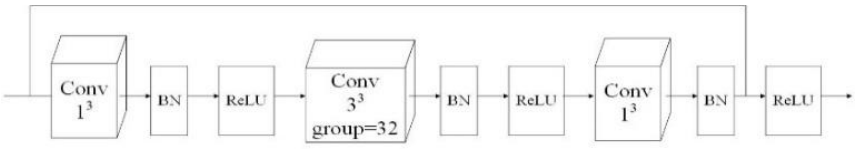


Fig. 3. Block of ResNeXt architecture. We represent conv, x^3, F as the kernel size,and the number of feature maps of the convolutional filter are $x \times x \times x$ and F, respectively, and group as the number of groups of group convolutions, which partition the feature maps into small groups. BN represents batch normalization.

To improve the accuracy of temporal activity detection, as Fig. 4. depicted, we add LSTM (Long Short-Term Memory units) behind the 3DResNet output layers to classify a sequence of video frames. LSTMs are a type of RNNs that are able to exploit long and short temporal correlations in sequences, which makes them suitable for video applications. LSTMs have been used alongside CNNs for video classification [14] and activity localization in videos [43].

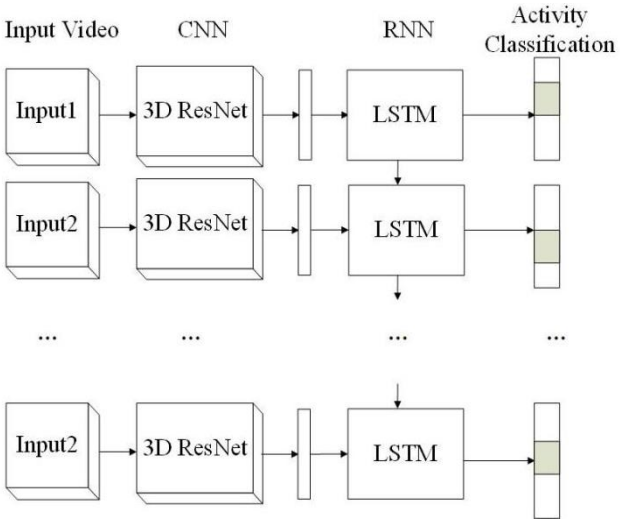


Fig. 4. Overview of proposed activity classification framework. Temporal features of inputted clipped video frames were extracted from a temporal CNN using 3D ResNet(middle-left) pre-trained on Kinetics. The features were then fed into a stack of recurrent sequence models (LSTMs, middle-right), which finally produce a prediction (right).

We design a network that extracts a sequence of C3D-f6 features of input video, and outputs a sequence of class probabilities for each 16-frames clip. We use LSTM layers, trained with dropout with probability $p = 0.5$ and a fully connected layer with a softmax as activation function. Fig. 4. shows our proposed architecture.

Given a processed clipped video, the prediction of our model is a series of class probabilities for each 16-frame video clip. We processed the output to predict the activity class and get temporal localization. First, to obtain the activity prediction of the whole video, we average the class probabilities over all video clips generated by object tracking model. Second, we choose the class which has the maximum probability among all candidate classes.

In order to achieve the temporal localization of predicted activity clipped video, we first apply a mean filter of k samples to the predicted series to make the values become smoothly through time (see Equation 3). Then, for each 16-frames clip we predict the probability of activity and no activity, and the activity probability is the summation of all probabilities of activity classes, and the no activity probability is the probability that this video clip belongs to background class. Finally, only clips with a probability value bigger than a certain threshold γ can be saved and marked as previously predicted class. Notice that, for each video clip, all predicted temporal results are activity class type.

$$\tilde{p}_i(x) = \frac{1}{2k} \sum_{j=i-k}^{i+k} p_i(x) \tag{3}$$

3 Experiments

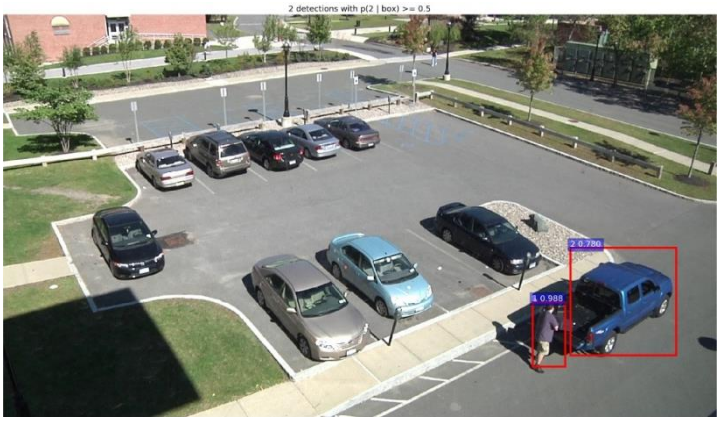


Fig. 5. Result of object detection model. We use the given bounding box information of person and vehicle to train a Faster RCNN model and use this model to detect objects in video frames. Number one in image represents person and number two represents vehicle.

We train a Faster RCNN model to detect specified objects such as person and vehicle in VIRAT Video Dataset and detected result is shown in Fig. 5. We use detection bounding box results which has confidence more than 0.5.

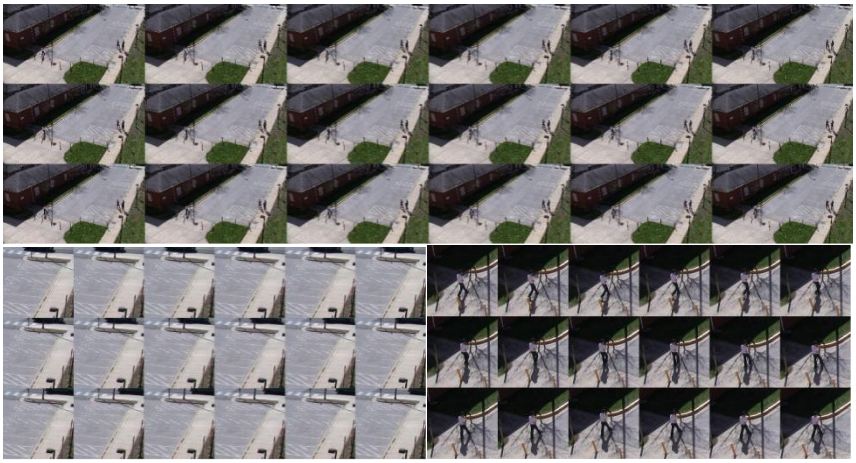


Fig. 6. Results of object tracking model in VIRAT video dataset. The top graph is original video frames. And the left graph is the tracking result of a moving vehicle on the top of the image. And the right graph is the tracking result of a moving pedestrian.

We feed the results of object detection model into object tracking model, and track the motion of different objects among adjacent frames to achieve multiple se-

quences of video frames which clipped according to detected bounding box information as shown in Fig. 6.

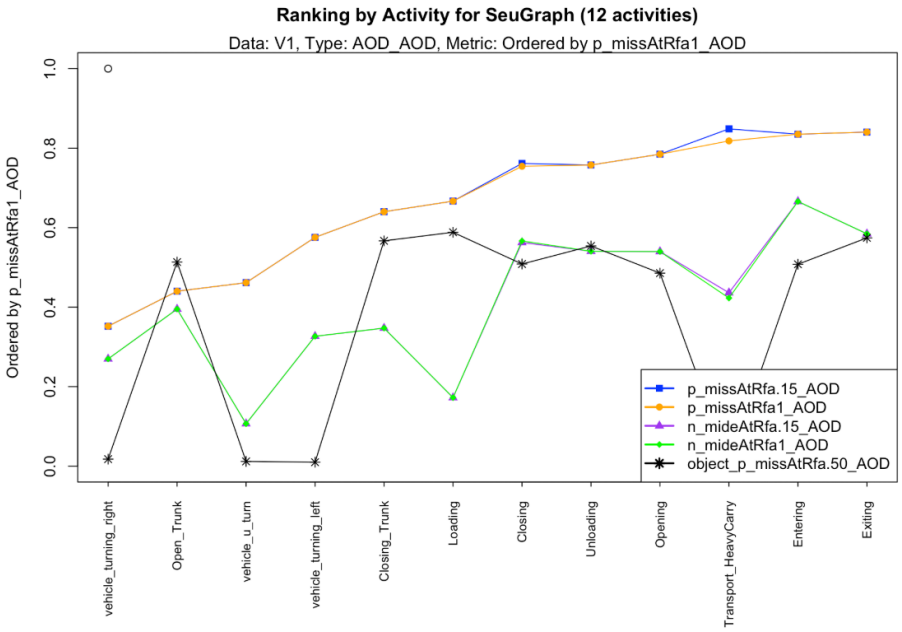


Fig. 7. Results of p_missAtRfa_AOD in 12 activities.

4 Conclusion

In this study, we propose an architecture for classification and temporal localization of activities in videos. Our architecture contains three inter-related processes. First, we use Faster RCNN as object detection model to detect objects which probably become a participant of a specific activity. Second, we feed the results of object detection into object tracking model to generate several sequences of video frames about detected objects. Third, we use 3DResNet to get features of temporal video frames and LSTM to locate the temporal position of activities more accurately.

References

1. George Awad, Asad Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Joao Magalhaes, David Semedo, Saverio Blasi. TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search. 2018.
2. Wang H, Kläser A, Schmid C, et al.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition, CVPR, 2011 IEEE Conference on, pp.3169-3176. IEEE(2011).

3. Scovanner P, Ali S, Shah M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM international conference on Multimedia, pp.357-360. ACM (2007).
4. Wang H, Schmid C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp. 3551-3558. IEEE (2013).
5. Li W, Zhang Z, Liu Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops, CVPRW, 2010 IEEE Computer Society Conference on, pp. 9-14. IEEE (2010).
6. Simonyan K, Zisserman A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp. 568-576. NIPS (2014).
7. Ng J Y H, Hausknecht M, Vijayanarasimhan S, et al.: Beyond short snippets: Deep networks for video classification. In: Computer Vision and Pattern Recognition, CVPR, 2015 IEEE Conference on, pp. 4694-4702. IEEE (2015).
8. Feichtenhofer C, Pinz A, Zisserman A P.: Convolutional two-stream network fusion for video action recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition 2016, CVPR, pp. 1933-1941. IEEE (2016).
9. Zha S, Luisier F, Andrews W, et al.: Exploiting image-trained CNN architectures for unconstrained video classification. arXiv preprint arXiv, 1503.04144 (2015).
10. Sun L, Jia K, Yeung D Y, et al.: Human action recognition using factorized spatio-temporal convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4597-4605. IEEE(2015).
11. Tran D, Bourdev L, Fergus R, et al.: Deep end2end voxel2voxel prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 17-24. IEEE (2016).
12. Tran D, Bourdev L, Fergus R, et al.: Deep end2end voxel2voxel prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp.17-24. IEEE(2016).
13. Wang L, Xiong Y, Wang Z, et al.: Towards good practices for very deep two-stream convnets. arXiv preprint arXiv, 1507.02159 (2015).
14. Yao L, Torabi A, Cho K, et al.: Describing videos by exploiting temporal structure. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4507-4515. IEEE(2015).
15. Hara K, Kataoka H, Satoh Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet , In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018 , pp. 18-22. IEEE, Salt Lake City (2018).
16. Kay W, Carreira J, Simonyan K, et al.: The kinetics human action video dataset. arXiv preprint arXiv, 1705.06950, (2017).
17. He K, Zhang X, Ren S, et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 770-778. IEEE(2016).
18. Benenson R, Omran M, Hosang J, et al.: Ten years of pedestrian detection, what have we learned?. In: European Conference on Computer Vision. pp. 613-627. Springer, Cham (2014).
19. Liu W, Anguelov D, Erhan D, et al.: Ssd: Single shot multibox detector. In: European conference on computer vision, pp. 21-37. Springer, Cham (2016).
20. Dai J, Li Y, He K, et al.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp: 379-387 (2016).
21. Girshick R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision 2015, pp. 1440-1448 (2015).
22. Ren S, He K, Girshick R, et al.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91-99 (2015).

23. Uijlings J R R, Van De Sande K E A, Gevers T, et al.: Selective search for object recognition. International journal of computer vision 104(2), 154-171 (2013).
24. Zhang L, Li Y, Nevatia R.: Global data association for multi-object tracking using network flows. In: Computer Vision and Pattern Recognition 2008, IEEE Conference on, pp. 1-8. IEEE(2008).
25. Li Y, Huang C, Nevatia R.: Learning to associate: Hybridboosted multi-target tracker for crowded scene. In: Computer Vision and Pattern Recognition 2009, IEEE Conference on, pp. 2953-2960. IEEE(2009).
26. Niebles J C, Han B, Fei-Fei L.: Efficient extraction of human motion volumes by tracking. In: Computer Vision and Pattern Recognition, CVPR, 2010 IEEE Conference on, pp.655-662. IEEE(2010).
27. Okuma K, Taleghani A, De Freitas N, et al.: A boosted particle filter: Multitarget detection and tracking. In: European conference on computer vision, pp.28-39. IEEE(2004).
28. Khan Z, Balch T, Dellaert F.: MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE transactions on pattern analysis and machine intelligence 27(11), 1805-1819 (2005).
29. Oh S, Russell S, Sastry S.: Markov chain Monte Carlo data association for multi-target tracking. IEEE Transactions on Automatic Control 54(3), 481-491 (2009).
30. Munkres J.: Algorithms for the assignment and transportation problems. Journal of the society for industrial and applied mathematics 5(1), 32-38 (1957).
31. Kim S, Kwak S, Feyereisl J, et al.: Online multi-target tracking by large margin structured learning. In: Asian Conference on Computer Vision. pp. 98-111. Springer. Berlin, Heidelberg (2012).
32. Breitenstein M D, Reichlin F, Leibe B, et al.: Online multiperson tracking-by-detection from a single, uncalibrated camera. IEEE transactions on pattern analysis and machine intelligence 33(9), 1820-1833 (2011).
33. Feichtenhofer C, Pinz A, Wildes R.: Spatiotemporal residual networks for video action recognition. In: Advances in neural information processing systems, pp. 3468-3476 (2016).
34. Feichtenhofer C, Pinz A, Wildes R P.: Spatiotemporal multiplier networks for video action recognition. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition 2017, CVPR,IEEE, pp. 7445-7454, IEEE (2017).
35. Wang L, Xiong Y, Wang Z, et al.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision, pp. 20-36. Springer, Cham (2016).
36. Karpathy A, Toderici G, Shetty S, et al.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1725-1732. IEEE(2014).
37. Varol G, Laptev I, Schmid C.: Long-term temporal convolutions for action recognition. IEEE transactions on pattern analysis and machine intelligence 40(6), 1510-1517 (2017).
38. Hochreiter S, Schmidhuber J.: Long short-term memory. Neural computation 9(8), 1735-1780 (1997).
39. Yoon J H, Yang M H, Lim J, et al.: Bayesian multi-object tracking using motion context from multiple objects. In : Applications of Computer Vision, WACV, 2015 IEEE Winter Conference on, pp. 33-40. IEEE (2015).
40. Simonyan K, Zisserman A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv,1409.1556 (2014).
41. Hara K, Kataoka H, Satoh Y.: Learning spatio-temporal features with 3D residual networks for action recognition. In: Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition. vol. 2, No.3, pp. 4. (2017).

42. Xie S, Girshick R, Dollár P, et al.: Aggregated residual transformations for deep neural networks. In: Computer Vision and Pattern Recognition, CVPR, IEEE Conference on. pp. 5987-5995. IEEE (2017).
43. Yeung S, Russakovsky O, Jin N, et al.: Every moment counts: Dense detailed labeling of actions in complex videos. International Journal of Computer Vision 126(2-4), 375-389 (2018).
44. Caba Heilbron, F., Escorcia, V., Ghanem, B., & Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, pp. 961-970. IEEE (2015).
45. Oh S, Hoogs A, Perera A, et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: Computer vision and pattern recognition, CVPR, 2011 IEEE conference on, pp. 3153-3160. IEEE (2011).