An Effective Detection Framework for Activities in Surveillance Videos

Ya Li[1], Shaoqiang Xu[1], Xiangqian Cheng[1], Liang Zhou[1], Yanyun Zhao[1,2],
Zhicheng Zhao[1,2], Fei Su[1,2], Bojin Zhuang[3]
[1]Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, Beijing University of Posts and Telecommunications, Beijing, China
[3]Ping An Technology Co.,Ltd
{liya, shao_qiang_xu, chengxiangqian, 2013210906, zyy, zhaozc, sufei}@bupt.edu.cn
zhuangbojin232@pingan.com.cn

Abstract

Recently some works are proposed to detect atomic actions of a single actor in unrealistic videos, such as movies. But they cannot be effectively applied to activity detection in surveillance videos. In this paper, we introduce an effective three-stage framework for spatiotemporal activity localization in surveillance videos. We devide the spatiotemporal activity localization task into three subtasks: spatial activity localization, activity tube tracking, and temporal activity detection. For spatial activity localization, we generalize Faster-RCNN to 3D form to localize activity regions in a short video clip. To the best of our knowledge, we are the first to extend object detector to detect activity regions in videos. For activity tube linking, we extend the object tracking algorithm to track activity tubes. For temporal activity localization, we build efficient temporal localization systems to detect activity instances. Experiments on VIRAT dataset demonstrate the effectiveness of our model. Besides, our framework is efficient as we don't need to process every frame of videos. With this effective and efficient framework, we got w_pmiss@0.15rfa 0.693 on TRECVID-ActEV test set in ActEV-Prize challenge.

1. Introduction

With the great power of CNN, we have witnessed advances in image classification, object detection and action recognition in recent years. But the problem of identifying and localizing activities in untrimmed videos is much more challenging, especially in surveillance videos. We still have not seen effective approaches to detection activities in continuous videos.

The VIRAT [21] dataset was introduced to advance activity detection in continuous videos, it was collected from realistic surveillance scenes. Several challenges make it more difficult than other spatiotemporal action localization dataset, eg., AVA [13]. The first challenge is that VIRAT dataset focuses on detecting activities rather than actions of a single person. Since many people and other objects may get involved in one activity, it's necessary to understand the interaction among actors and the interaction among actor and objects. Secondly, the spatial range and temporal length of activities vary drastically compared with actions in AVA dataset, which makes methods [8, 10, 35] designed for other datasets unsuitable for VIRAT. Thirdly, the sparsity and dense overlapping of activities both exist in VIRAT dataset. On one hand, surveillance videos contain significant spans without any activities. On the other hand, activities such as `opening`, `entering`, `closing` may occur sequentially and overlap with each other. Moreover, the small size of actors makes VIRAT more challenging, which explains why works [12, 37, 38] rely on frame-wise person detection can't perform well on VIRAT dataset.

In this work, we propose an effective three-stage framework to handle this challenging task. We devide this challenging task into three sequential subtasks: spatial activity localization, activity tube tracking, and temporal activity detection, each subtask is handled by a specific module independently. For spatial activity localization, we generalize Faster-RCNN to 3D form to localize activity regions in a short video clip, since 3D convolution can capture both appearance feature and motion feature and shows promising results on video recognition. What's important, we propose to detect activity regions rather than actors, which greatly simplify spatial activity localization problem as we don't need to generate activity regions based on detected actors. Given activity tubes, we extend the object tracking algorithm to track activity tubes. The Hungarian Algorithm [16] is used to link tubes into activity tracks for later temporal

detection. Finally, two effective temporal localization systems are built to detect activity instances on activity tracks.

To summarizes, our contributions are as followings: (1) We propose an effective and efficient framework for spatiotemporal activity localization in surveillance videos, experiments on VIRAT dataset demonstrate the effectiveness of our model. (2) We adapt Faster-RCNN into 3D form to detect activity regions in a short video clip. To the best of our knowledge, we are the first to extend object detector to detect activity regions in videos. (3) We extend object tracking algorithm to track activity tubes.

2. Related work

Action Recognition. Action recognition aims to classify a trimmed video clip into actions of interest, which is fundamental to video understanding. Previous works [6, 30] take raw images to get appearance information and optical flow to get the motion feature and achieve fruitful results on UCF101. Tran et al. [5] introduce 3D architecture to learn spatiotemporal features by 3D ConvNets. Carreira et at. [2] inflate 2D kernel of bn-inception to 3D kernel to model spatiotemporal feature. Xie et at. [32] propose that 3D kernel can be separated into 2D convolution on space and followed with 1D convolution along the temporal axis without losing accuracy and make network slighter and faster. Zolfaghari et at. [39] advise ECO, which applies a 3D convolution head on feature maps extracted by 2D convolution to trade off expressiveness and computation costs. We use 3D convolution to extract spatiotemporal features for activity detection.

Temporal Action Detection. The goal of temporal action detection is to identify the start and end times as well as the action label for each action instance in long, untrimmed videos. There are two main different lines to address this problem. One line of works predict action label at frame level or segment level, then use these labels to find temporal boundaries of actions [4, 17, 22, 34]. The other line works generate proposals by densely distributed anchors at first, then classify proposals into actions and refine proposals [3, 7, 26, 33, 36], which is inspired by recent region based object detector. We follow the second line to build our temporal action detection system.

Spatiotemporal Activity localization. A few deep learning based algorithms have been proposed for activity detection recently. [12, 27, 37, 38] detect objects at frame level at first, then generate action proposals by data association or clustering, after that temporal action detection is applied on action proposals. These approaches are limited by frame-wise object detector. Besides, they are inefficient because object detection needs to be performed frame by frame. Recently some works try to utilize spatiotemporal feature by 3D convolution [8, 9, 14, 15, 35], however, most of them only focus on atomic action detection in actor-centered video, eg., AVA dataset [13], thus can't be applied to detection activity in surveillance video. T-CNN [14] is the closest to ours, they detect class-agnostic action tubes by Tube Proposal Network(RPN), and classify each associated action track into action instance. However, they only detect class-agnostic action tubes, which makes the linking procedure more difficult. Besides, they just classify action tracks into actions without temporal localization, which is critical for activity detection.

Object Detection. Object detection is one of the key components in computer vision. Modern deep-learning based detector can be divided into two streams, one is two-stage detectors including Faster-RCNN [24] and many follow-up improvements. The two-stage detector use Region Proposal Network(RPN)[24] to generate proposals, then classify proposals into objects of interest and perform box refinement. The one-stage detector [19, 23] directly predict boxes classes at every position of the feature map. In this work, we generalize Faster-RCNN into 3D form to utilize the spatiotemporal feature to detect activity regions in video clips.

Data association. Data association is an important procedure of the multi-object tracking problem, which can be formulated as a bipartite graph matching problem. Most online processing methods use Hungarian Algorithm[16] or minimum-cost-network-flow to solve this problem. In our approach, we extend the task of object tracking to activity tube tracking.

3. Approach and Models

Our three-stage framework is designed to detect activities in surveillance videos. In section3.1, we firstly give a brief analysis of activities in VIRAT dataset, then present our overall framework and explain our design decisions. Then we describe modules of each stage in detail in the following sections.

3.1. Spatiotemporal activity localization framework

Activity analysis. According to the attributes of different activities, we divide 18 activities into 3 groups, as illustrated in Table1, ie., `vehicle-person`, `turning`, `person-centered`. Activities of person-centered group often last a long time. The appearance and motion pattern of an activity are pretty similar at every segment, so that we can distinguish these activities in a short glimpse. For activities of turning and vehicle-person groups, their temporal lengths vary widely, ranging from a fraction of a second to dozens of seconds. Besides, they may overlap with each other in space and time, thence it's infeasible to distinguish an activity of these two groups except we have seen the whole span of the activity.

To detect activities that vary widely in space and time in surveillance videos, we design a three-stage detection

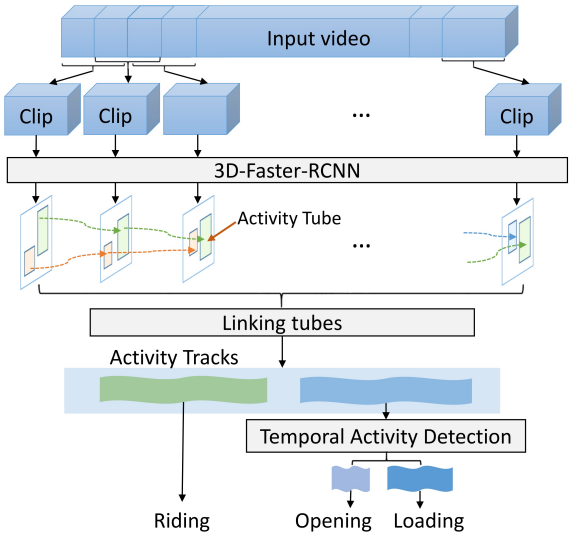

Figure 1. The overall structure of our framework. We slice video clips along the temporal axis and detect activity tubes of each clip in space by 3D-Faster-RCNN with rough classification. Then, activity tubes are linked into activity tracks by the data association module. Tracks of different groups will be treat differently. Vehicle-related tracks will be sent to the temporal activity detection module to detect fine-grained activity instances. Tracks of the person-centered group will be submitted as final output.

Activities	super-category
Closing	vehicle-person
Closing_trunk	vehicle-person
Entering	vehicle-person
Exiting	vehicle-person
Loading	vehicle-person
Open_Trunk	vehicle-person
Opening	vehicle-person
Transport_HeavyCarry	person-centered
Unloading	vehicle-person
Vehicle_turning_left	turning
Vehicle_turning_right	turning
Vehicle_u_turn	turning
Pull	person-centered
Riding	person-centered
Talking	person-centered
activity_carrying	person-centered
specialized_talking_phone	person-centered
specialized_texting_phone	person-centered

Table 1. According to the attributes of different activities, 18 activities are divided into 3 groups.

framework, its overall structure is illustrated in Figures 1. First of all, we localize possible activity regions of each clip. A 32-frame sliding window is used to slice clips from continuous videos with stride 16. We sample 8 frames from each clip and put them into 3D-Faster-RCNN to detect activity tubes with rough classification. The 3D-Faster-RCNN is described in section 3.2. Notably, we treat activity categories of vehicle-person group as one class, activity categories of turning group as another class when detecting activity regions. In other words, there are nine categories of detection targets, ie., `vehicle-person`, `turning`, `Pull`, `Riding`, `Talking`, `Transport HeavyCarry`, `activity carrying`, `specialized talking phone`, `specialized texting phone`. Next, given activity tubes of these nine categories, we use the Hungarian Algorithm to associate tubes into activity tracks, which is described in section 3.3. Finally, we employ different temporal detection approach for tracks of each activity group. For tracks that belong to the person-centered group, we directly output a track as an activity instance. For tracks of the vehicle-person group, a two-stage temporal activity localization system is built to detect fine-grained activity instance on tracks. Moreover, a one-stage temporal acitivity detection model is employed to detect activity instances of turning group. The temporal detection part is described in section 3.4.

3.2. 3D-Faster-RCNN

To detect activities in untrimmed videos, it's imperative to localize activity in space at first. To this end, we extend the image object detector to localize activity regions in a short video clip. Inspired by the success in object detection [24] and action recognition [2], we adapt Faster-RCNN [24] into 3D form to utilize spatiotemporal feature, which can significantly improve activity spatial localization performance compared with single-frame detector.

For activity detection, appearance information and motion information are equally important. To better capture the spatiotemporal information in video, we replace the 2D convolution backbone of Faster-RCNN with I3D. The I3D is implemented by [31], who inflate 2D convolution of ResNet50 into 3D convolution. The spatial resolution of each Conv stage output feature map keep consistent with ResNet, the temporal dimension is only downsampled once at $pool2$ that following Conv2, we refer the reader to [31] for more details.

FPN is adopted to fuse low-level appearance features with high-level semantic features to facilitate small activity region detection, as most activities only occupy a small part of a frame. To accumulate long-range information, we squeeze the 4D feature map $(channel \times time \times height \times width)$ to 3D $(channel \times height \times width)$ at FPN lateral connection by convolution and squeeze operation. Therefore, the RPN and R-CNN head remain the same as [18]. Specifically, we inflate the lateral connection module from 2D convolution to 3D convolution with kernel $(4, 1, 1)$ and no padding, thus the time dimension of lateral connection

output feature map would be 1, then squeeze operation is applied to remove time dimension. Same as Faster-RCNN, we use the cross-entropy loss for classification and smooth L1 loss for region regression.

Discussion. We detect activity regions rather than actors, activity region is defined as a tight box that covers every object that involved in specific activity over the clip time span. We believe this will bring several benefits. The first benefit is that the activity region is bigger than a person area, thus more appearance features and motion features are accumulated in the activity region, therefore it's easier to detect an activity region than an actor. The second one is that we can model interactions among actors, which can help detect activities that involved multi-actors. Models of [8, 9, 14] are similar to our 3D-Faster-RCNN but different in several ways: 1) FPN [18] is adopted in our network to facilitate small targets detection. We argue that FPN is important as most activities only occupy a small part of a frame, whereas FPN is absent in theirs; 2) We perform detection on the fused feature map of multiple frames rather than sliced feature map [8, 9] to utilize long-range information, which makes our network more effective.

3.3. Data association

Given activity tubes of nine categories, we then associate them to activity tracks for spatiotemporal localization of the entire video. We extend the task of object box tracking to activity tube tracking, which can be formulated as a data association problem. Inspired by [11], the Hungarian Algorithm [16] is adopted to link tubes of adjacent clips. Activity tubes can be represented as nodes of a bipartite graph. We use Intersection over Union (IoU) between the two boxes of adjacent clips as cost of edge.

Tubes of each category are processed separately. Given activity tubes, we initialize tracks on the first clip, then update tracks by tubes of the next clip that matched. Any tubes that not matched to an existing activity track will instantiate a new track.

This data association method is very light, and the main computational overhead comes from calculating IoU. Compared with other stages, the time cost can be ignored. Besides, we find it is effective to link activity tubes.

3.4. Temporal activity localization

Given activity tracks produced by the data association module, our next step is to perform temporal detection based on tracks. We employ a different strategy for each activity group. For tracks that belong to the person-centered group, we have already known the specific category at the spatial activity localization stage, thus no further analysis required, we just recognize a track as an activity. For activity tracks of vehicle-person and turning groups, multiple activities of different subclasses may occur in one track,

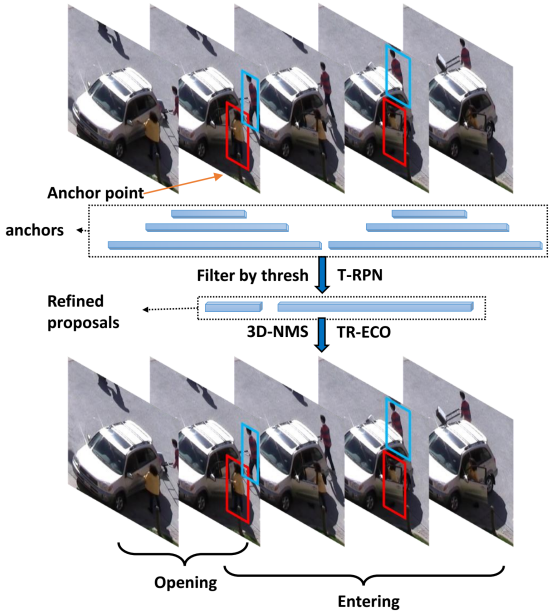

Figure 2. Temporal activity detection pipeline for vehicle-person activities. Multi-anchors are generated of every person at each anchor point.

therefore further temporal detection and fine-grained classification are required. Our approaches for vehicle-person and turning are similar but not identical, we will present each approach separately in the following sections.

3.4.1 Vehicle-person activity detection

Given activity tracks of the vehicle-person group, we detect fine-grained activity instance on these tracks. One choice is just classify activity tracks into fine-grained classes as [12, 14], but we find it's sub-optimal for two reasons: 1) Multiple activities may contain in one track, thus it's impossible to achieve high recall with few tracks. 2) Too much background noise would be included if we just put an entire track into the classifier. In our approach, we formulate this as a temporal activity detection task. A two-stage temporal activity detection system is developed to detect activity along the temporal axis on activity tracks.

Our vehicle-person activity detection system consists of two stages, models of two stage are trained independently. The first stage is an RPN-like module to propose possible 3D-proposals, named Time Region Proposal Network(T-RPN). A 3D-anchor is sent through T-RPN to predict probability that the content of the anchor corresponds to a valid activity and regress its time boundaries. T-RPN return sparse 3D-proposals as we filter out most 3D-anchors that "activityness" lower than a specific threshold. We set different thresholds during training and testing. The second stage module classifies 3D-proposals into activities of inter-

est or background and perform further time refinement, we call it Time Regression ECO [40] (TR-ECO). Both stages use ECO as the backbone, 16 frames evenly sampled from an anchor or proposal are sent to the backbone network. An extra branch is added to the final layer of ECO to predict temporal offsets relative to predefined anchors. Notably, our temporal regression of TR-ECO is independent of category. We didn't find better performance with class-dependent temporal regression.

The proposal generation stage and activity classification stage share a same form of multi-task loss. Same as TAL-Net[3], we use cross-entropy loss of classification and smooth L1 loss for temporal regression:

$$\mathcal{L} = \sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \lambda \sum_i [p_i^* \geq 1]\mathcal{L}_{reg}(t_i, t_i^*) \quad (1)$$

where i is the index of an anchor or proposal in a mini-batch. p is predicted probability of the proposal or activity, p* is ground-truth label, $\mathcal{L}_{cls}$ is the cross-entropy loss. For regression, t is the predicted offset relative to anchor or proposal, $t*$ is ground-truth offset, and $\mathcal{L}_{reg}$ is the smooth L1 loss. Offsets t is defined same as TAL-Net[3], we refer reader to [3] for more details. We set $\lambda = 0.2$ for proposal generation, $\lambda = 1$ for activity classification.

Anchor generation. As we can see from Figure2, multiple persons may interact with the same vehicle and involved in different activities in one track, therefore, it's necessary to process each person separately. Given activity tracks, we define anchor points that evenly distributed on tracks along the temporal axis with interval as 16 frames. At each anchor point, 3D-anchors with 15 different time scales are generated for each person in the track, ie., {32, 40, 50, 64, 80, 101, 128, 161, 203, 256, 322, 406, 512, 645, 812} frames. 3D-anchors of the same person enjoy together the same spatial location, which is defined as an enlarged square region that centered at the person's box. The size of the square region is the short side of the track spatial region. As our 3D-Faster-RCN only detects union region of each activity, a separately trained Faster-RCNN is used to detect persons. If no person is detected at the anchor point, we just enlarge the track region into square as the spatial location of anchors.

Chao et al. [3] have pointed that it's sub-optimal to use same features to classify multiple anchors, as the receptive field is fixed but the temporal length of activities may vary drastically. To avoid this problem, we evenly sample 16 frames from each anchor no matter how long the anchor is and put them through T-RPN to predict "activityness" score and temporal offsets. Compared with TAL-Net, our network has fewer parameters as we don't need the multi-tower network to deal with different anchor scales, thus overfitting is alleviated.

3.4.2 Turning acitivity detection

Vehicle turning group consists of left turn, right turn, and U-turn activities. We build a simple one-stage temporal activity detection system to detect turning activity instances. ECO is used to classify anchors into activity of interest or background. We evenly sample 16 frames for each anchor.

Anchor generation. Given an activity track of turning group, anchor points are evenly distributed on the track along the temporal axis with interval as 48 frames. We generate 3 3D-anchors with time scales {64, 128, 256} at each anchor point. The spatial location of each 3D-anchor is defined as a tight box that covers the vehicle active area in the anchor time range.

4. Experiments

In this section, we firstly introduce the VIRAT dataset and the evaluation metric, then we present experiments of each stage in detail in the following subsections.

4.1. VIRAT dataset

The VIRAT[21] dataset was introduced to assess the performance of activity detection algorithms in realistic scenes. The main scenes of the dataset are parking lots, streets and other outdoor spaces. All 18 activities are listed in the first column of Table 1. Boxes of persons and vehicles that involved in each activity are annotated exhaustedly during activity span, other objects, eg., bike, prop are partially annotated. The train set contains 64 videos. The validate set has 54 videos.

4.2. ActEV metric

The performance is evaluated by the probability of missed detection P_{miss} at a fixed rate of false alarm per minute $Rate_{FA}$. Detected activities are mapped to ground truth by Hungarian Algorithm. A detected activity that doesn't correspond to any ground-truth activity is a false alarm, a ground-truth activity that isn't paired to any detection is a miss. For details of the evaluation metric, we refer the reader to [1].

4.3. 3D-Faster-RCNN

A 32-frame sliding window with stride 16 is used to slice clips during training and testing. To save GPN memory, we sample 8 frames from a 32-frame clip at a regular interval and put them through 3D-Faster-RCNN to detect activity tubes in the clip. The target of 3D-Faster-RCNN is a tight box that covers all objects that get involved in a specific activity over the clip span. A box is kept only if the corresponding activity span in this clip longer than 16 frames or longer than half of the activity's length.

Training. We initialize the RPN and Fast-RCNN heads by Faster-RCNN that pretrained on COCO dataset [20].

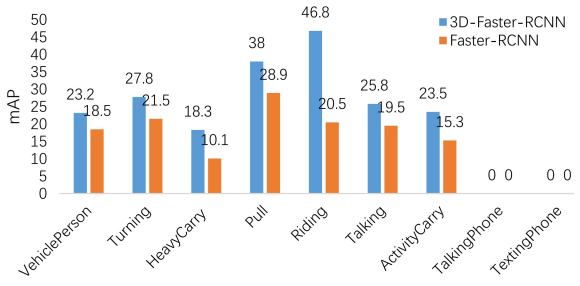

Figure 3. Spatial activity localization results. We can see there is a significant improvement in each class over regular Faster-RCNN that perform detection on a single frame.

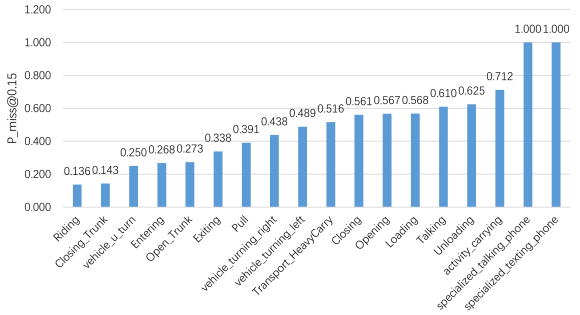

Figure 4. P_miss@0.15 for each activity category of validate set.

The I3D backbone is pretrained on Kinetics dataset [2]. Multi-scale training and horizontal flipping are used as data augmentation. The short side of a clip is randomly scaled to {832, 864, 896, 928, 960, 992, 1024, 1056}. We train on a 2-GPU machine where each GPU has 1 video clip as mini-batch. The total batch size is 2. The model is trained on the train set for 50000 steps with learning rate 0.0025 at first. We finetune it on trainval union set for another 50000 steps with learning rate 0.0025 for our final submitted results.

Inference. We scale the frame short side to 1056 at the inference stage. A tube with a score lower than 0.05 are filtered out. NMS is applied with threshold 0.1.

We evaluate the performance of 3D-Faster-RCNN by mAP[20] on validate set, and get 22.6%. The AP of each category on validate set is shown in Figure 3. We compare our model with Faster-RCNN that perform detection on a single frame to verify the effectiveness of our 3D-Faster-RCNN. As we can see from Figure 3, there is a significant improvement in each category.

4.4. Temporal activity localization

We present experiment details of vehicle-person and turning activity detection separately.

4.4.1 Vehicle-person activity detection

Label assignment. For proposal generation, an anchor is assigned a positive label if it overlaps with a ground-truth activity cuboid with temporal Intersection-over-Union(tIoU) and spatial Interaction-over-Union(sIoU) both higher than 0.7. An anchor is recognized as negative if the tIoU is lower than 0.3 or the sIoU is lower than 0.3 with all ground-truth activities. Anchors that neither are positive nor negative are filtered out when training. For activity classification, a proposal is assigned the activity label of its most overlapped ground-truth activity cuboid, if the tIoU and the sIoU both higher than 0.5. Otherwise, a background label is assigned.

We try diverse data augmentation methods to avoid overfitting. Center-corner cropping and scale jittering with horizontal flipping are employed on scaled images. Images are resized to 224×224 before sending to the network.

Several design decisions are examined to make our model perform better. We double the time span of all category ground-truth activities except trunk-related activities to include more temporal context. Besides, `Open trunk`, `Close trunk` are divided into four categories, ie., `Open trunk car`, `Close trunk car`, `Open trunk pika`, `Close trunk pika`, as we find trunk of car and trunk of pika have opposite move direction.

Training. We train T-RPN on the train set for 10 epochs with learning rate 0.001. We only use proposals with a score higher than 0.15 as input to TR-ECO. For activity classification, we train TR-ECO on the train set for 50 epochs with a learning rate 0.001. The batch size of both stages are same at 96. In order to mitigate the serious foreground and background sample imbalance problem, we set the foreground and background sample ratio to 3:1 at both training stages.

Inference. Proposals with a score lower than 0.3 are filtered out before sending them to TR-ECO when testing. We ensemble activity classification results of epoch 20, 30, 40, 50. Spatiotemporal 3D-NMS is applied to ensembled results with spatial threshold 0.3 and temporal threshold 0.5.

4.4.2 Turning activity detection

Label assignment. An anchor is assigned the activity label of its most overlapped groud-truth activity cuboid, if both the tIoU and the sIoU are higher than 0.5. An anchor is assigned a background label if the sIoU is lower than 0.3 or the tIoU is lower than 0.2 with all ground-truth activities. Anchors that neither are positive nor negative are filtered out during training.

The data augmentation of turning activity detection is the same as vehicle-person activity detection except that there is no horizon flipping. We train the network on the train set for 40 epochs with a learning rate of 0.001. The 3D-NMS is applied with spatial threshold 0.1 and temporal threshold 0.1 at the inference stage.

4.5. Results

As we can see from Figure4 , our framework achieve good results in most activity categories on the validation set. However, the performance on `specialized talking phone` and `specialized texting phone` is poor, we think the reason is that actors are too small to distinguish phone-related activities, and the lacking of training samples. We get w_pmiss@0.15rfa 0.693 on TRECVID-ActEV test set in ActEV Prize Challenge.

5. Conclusion

In this paper, we introduce an effective and efficient framework to detect activities in surveillance videos. Our framework consists of three parts. We generalize Faster-RCNN into 3D form to detect activity regions in a short video clip, which shows promising results on the spatial activity localization task. We extend object boxes tracking algorithm to track activity tubes, and it saves the cost of pedestrian detection and tracking. Different approach is employed for each activity group during the temporal activity detection stage.We believe that our approach has a strong generalization ability that can be applied to many scenarios.

ACKNOWLEDGEMENTS This work is supported by Chinese National Natural Science Foundation (61532018).

References

[1] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, David Joy, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Joao Magalhaes, David Semedo, and Saverio Blasi. Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *Proceedings of TRECVID 2018*. NIST, USA, 2018.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. 2017.

[3] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.

[4] Achal Dave, Olga Russakovsky, and Deva Ramanan. Predictive-corrective networks for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 981–990, 2017.

[5] Tran Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision*, 2015.

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[7] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*, 2017.

[8] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for ava. *arXiv preprint arXiv:1807.10066*, 2018.

[9] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *CoRR*, abs/1812.02707, 2018.

[10] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *arXiv preprint arXiv:1812.02707*, 2018.

[11] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 350–359, 2018.

[12] Joshua Gleason, Rajeev Ranjan, Steven Schwarcz, Carlos Castillo, Jun-Cheng Chen, and Rama Chellappa. A proposal-based solution to spatio-temporal action detection in untrimmed videos. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 141–150. IEEE, 2019.

[13] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[14] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5822–5831, 2017.

[15] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017.

[16] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[17] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[19] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160, June 2011.

[22] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018.

[23] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[25] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017.

[26] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017.

[27] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017.

[28] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5620–5629, 2017.

[29] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[30] Limin Wang, Yuanjun Xiong, Wang Zhe, Qiao Yu, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, 2016.

[31] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[32] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[33] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[34] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2017.

[35] Yubo Zhang, Pavel Tokmakov, Cordelia Schmid, and Martial Hebert. A structured model for action detection. *CoRR*, abs/1812.03544, 2018.

[36] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.

[37] Kaihui Zhou, Yandong Zhu, and Yanyun Zhao. A spatio-temporal deep architecture for surveillance event detection based on convlstm. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[38] Yandong Zhu, Kaihui Zhou, Menglai Wang, Yanyun Zhao, and Zhicheng Zhao. A comprehensive solution for detecting events in complex surveillance videos. *Multimedia Tools and Applications*, 78(1):817–838, 2019.

[39] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, 2018.

[40] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 695–712, 2018.