CMU Informedia at TRECVID 2020: Activity Detection with Dense Spatio-temporal Proposals

Lijun Yu,* Yijun Qian,* Wenhe Liu, and Alexander G. Hauptmann

Carnegie Mellon University

lijun@cmu.edu, yijunqia@andrew.cmu.edu, {wenhel,alex}@cs.cmu.edu

Abstract

We propose an action recognition system for surveillance scenarios, which wins TRECVID 2020 [1] Activities in Extended Video (ActEV[2]) Challenge with a large advantage of 23.8% ahead the runner up system. Our system develops a dense spatial-temporal proposal generation model which collaborates with the state-of-the-art action classifiers. The proposed system utilizes multiple state-of-the-art modules and is trained on VIRAT Dataset with only released annotations. In this paper, we demonstrate the architecture and algorithms with technique details of our winner system.

1 Introduction

In the past decade, widely-deployed surveillance cameras have grown gradually. As a result, the volume of streaming surveillance videos becomes overwhelmingly large dramatically, which makes it difficult to process and analyze by human being. Meanwhile, it is a request of public safety for critical surveillance events detection in real-time. There is thus strong incentive to develop fully-automated methods to identify and localize activities in extended video collections and provide the capability to alert and triage emergent videos. These methods will alleviate the current manual process of monitoring by human operators and scale up with the growth of sensor proliferation in the near future. An efficient and effective functionality to spatially and temporally detect or localize human activities is central in surveillance video analysis. The Activities in Extended Videos Prize Challenge (ActEV) seeks to encourage the development of real-time robust automatic activity detection algorithms in surveillance scenarios. With the availability of large-scale video surveillance datasets such as VIRAT [2] it aims to test and evaluate the surveillance activity detection systems on both detection performance and processing speed.

To tackle the challenge, we propose a system which is able to generate dense spatio-temporal proposals followed by varies of activity classification models. Moreover, we adopt a asynchronous parallel design to further optimize the processing speed of the system. For proposal generation, we develop an algorithm which contains spatial object detection and object tracking models to crop object (person/vehicle) centred proposal cubes from the input videos. For activity classification, we implements and optimize several spatial-aware activity classification algorithms and apply a fusion and filtering method in the post-processing stage.

To foster further advantage of the field, we summarize our contribution as twofold:

1. We propose a dense spatio-temporal cube proposal paradigm to precisely localize activities in surveillance videos and reduce false alarms.

*Contributed equally

[2] https://actev.nist.gov

2. Our system has achieved the first place in the NIST TRECVID ActEV benchmark with $nAUDC@0.2T_{fa} = 0.42$, which reports 23.8% ahead of the runner up system.

2 System

2.1 Problem Statement

In the series of NIST Activities in Extended Video (ActEV) evaluations, the task is activity detection in videos with extended metadata. Given a set of untrimmed videos $\mathcal{V} = \{V_i\}$, the system should identify a set of activity instances $\mathcal{A} = \{A_i\}$ Each activity instance is defined by a three-tuple $A_i = (V_i, L_i, C_i)$, meaning an activity of type C_i occurs at a spatio-temporal area L_i in video V_i.

In the current dataset, VIRAT [2], $\mathcal{V}$ is limited within single-view videos from a surveillance camera with a fixed point of view. In the current evaluation plan [3], the spatial localization precision is not measured. The idea was that, after processed by the system, we still have human reviewers to inspect the activity instances with the highest confidence scores for further usages. The performance is thus measured by the recall of activity instances within a time limit of all positive frames plus T_{fa} of negative frames, where T_{fa} is referred to as time-based false alarm rate.

2.2 Architecture

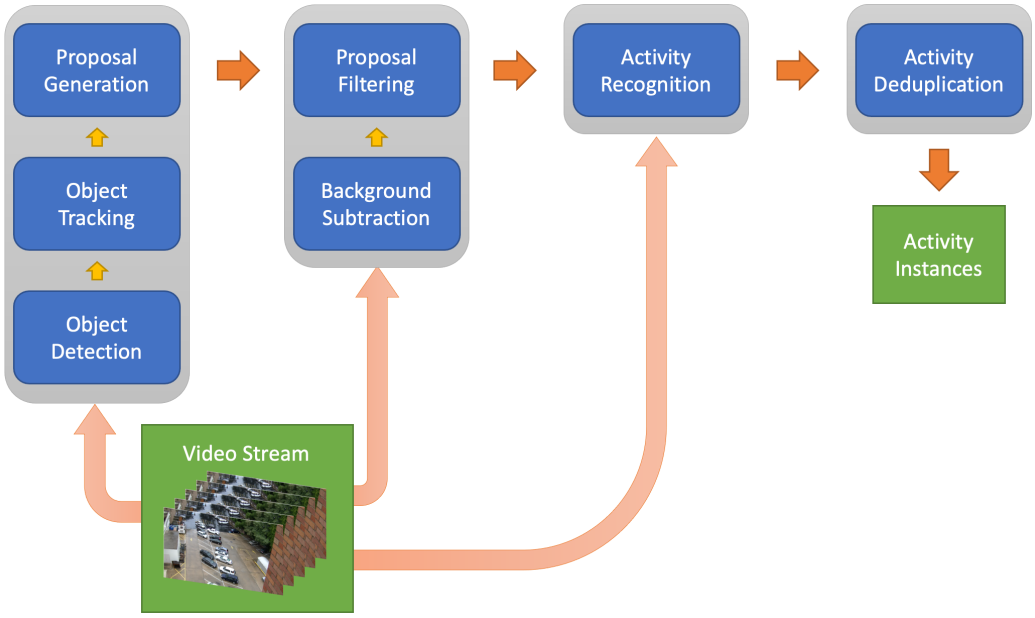

Figure 1: System Architecture

The architecture of the proposed system. To tackle the task of activity recognition, we adopt an intermediate concept of *spatio-temporal cube proposal*. The system first generates candidate proposals with frame-wise information such as objects. These proposals are filtered with a background subtraction model. Then, action recognition models are applied on the proposals to predict per-class confidence scores. Finally, a post-processed method is applied to merge and filter the scores and output final activity instances.

2.3 Proposal Generation

In this section, we introduce the proposal generation stage of our system. First, we apply detection and tracking methods to identify the candidate objects in the video; Second, we generate the spatio-temporal cube proposals for activity classification.

Detection and Tracking To conduct activity recognition, we first identify the candidate objects (person, vehicle) in the video. For each frame i, we apply an object detection model to get objects $O_i = \{o_{i,j} \mid j = 1, \cdots, n_i\}$ with object types $c_{i,j}$ and bounding boxes $(x_0, x_1, y_0, y_1)_{i,j}$.Objects are detected in a stride of every S_{det} frames. Object detection methods [4, 5] optimized for video scenario can also be used. A multiple object tracking algorithm is applied on the detected objects to assign track ids to each of them as $tr_{i,j}$.

Cube Proposal Previous works such as [6, 7, 8, 9] utilize the whole trajectory of each tracked object as the proposal, which are *spatio-temporal tube proposals*. It generates proposals on resized bounding boxes across frames. Tube proposal has several drawbacks . First, in action recognition task, such tube proposals still require temporal activity localization in the later stage to determine the existence of the activities on video clips, which would be more difficult than classification on fix length video clips. Second, the objects in the tube proposal will suffer from the shape change when being resized frame by frame. Third, the bounding boxes shift across frames also could harm visual features extracting. All these problems could result in a high false alarm rate on action recognition.

In this work, we adopt a different form of proposals, which we define as *spatio-temporal cube proposals*. A spatio-temporal cube proposal is defined by a six-tuple

$$p_i = (x_0, x_1, y_0, y_1, t_0, t_1)_i \tag{1}$$

corresponding to boundaries in three dimensions.

Proposal Sampling For input videos with variable length, one straightforward approach is to cut them into non-overlapping proposals with D_{prop} frames and process each one sequentially. Such intuitive approach, such as in [10], would result in significant performance drop at the boundary between clips, because it might break the completeness of activities when cutting proposals. To handle the problem, we propose a dense overlapping proposals sampling algorithm. As illustrated in Figure 2, a dense overlapping proposal system is defined by two parameters, i.e. duration D_{prop} and stride S_{prop}. Proposals are generated within temporal windows of D_{prop} frames. For every temporal window in $S_{prop} \leq D_{prop}$ frames, video clips are sampled densely with overlapping. Generally, non-overlapping proposal system can be treated as a downgraded case when $S_{prop} = D_{prop}$.

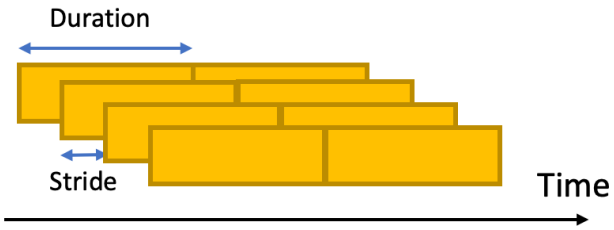

Figure 2: Dense Overlapping Proposals

Proposal Refinement To generate proposals in a temporal window from t_0 to $t_1 = t_0 + D_{prop}$, we select seed track ids Tr_{t_0} from the central frame $t_c = \frac{t_0+t_1}{2}$. Their bounding boxes are enlarged as the union across the temporal window

$$\begin{aligned}(x_0, x_1, y_0, y_1) = union(\{(x_0, x_1, y_0, y_1)_{i,j} \mid t_0 \leq i \leq t_1, tr_{i,j} = tr_{t_c,k}\}) \\ k = 1, \cdots, n_{t_c}\end{aligned} \tag{2}$$

This algorithm is robust through identity switch in the tracking algorithm as it uses the stable seeds from the central frame. It also ensures the coverage of moving objects by enlarging the bounding box when it's successfully tracked.

For now, the proposal generation system applies a frame-wise object detection with slight aid of tracking in each of short video clips. The motion information is not yet explored. To product high quality proposals, we apply a proposal filtering algorithm to eliminate the proposals that are unlikely to contain activities.

For each proposal, a foreground segmentation algorithm is implemented to generate a binary mask for every S_{bg} frames for each video clip. We average the value of pixel masks in its cube to get its foreground score f_i. For proposals generated by object type c, those proposals with $f_i \leq F_c$ will be filtered out, where F_c is a threshold. In experiment, F_c can be tuned on the training set.

2.4 Activity Recognition

In this section, we will introduce details about the implementation of activity recognition modules. Given an input sequence of proposals, We followed the sparse-sampling strategy mentioned in [11] to sample N frames for training and inference. To be specific, the video is evenly separated into N segments. From each segment, we randomly selected 1 frame to generate the sampled clip $\mathbb{C}_i$ for training.

The sampled clip is fed to an action recognition module $\mathbb{V}$ to get classification scores X_i.

$$X_i \quad = \quad Sigmoid(\mathbb{V}(T_i)) \tag{3}$$

Where $\mathrm{X}_i = \{x_i^1, x_i^2, ..., x_i^C\}$ and $X_i \in \mathbb{R}^C$. To transform the action recognition modules from previous multi-class task to the realm of multi-label recognition, we modified the loss function for optimization. Instead of traditional cross entropy loss (XE), we implemented a weighted binary cross entropy loss (wBCE). In which, two weight parameters are adopted, the activity-wise weight $\mathrm{W}_a = \{w_a^1, w_a^2, ..., w_a^C\}$ and the positive-negative weight $\mathrm{W}_p = \{w_p^1, w_p^2, ..., w_p^C\}$. W_a balances the training samples of different activities and W_p balances the positive and negative samples of a specific activity. With the aligned label sequence of i^{th} batch represented as $Y_i = \{y_i^1, y_i^2, ..., y_i^C\} \in \mathbb{R}^C$. The calculation of w_a^c is derived as:

$$\hat{w}_a^c = \frac{1}{\sum_{i \in [B]} y_i^c} \tag{4}$$

$$w_a^c = C \times \frac{\hat{w}_a^c}{\sum_{c \in [C]} \hat{w}_a^c} \tag{5}$$

And the derivation of w_p^c is:

$$w_p^c = \frac{\sum_{i \in [B]} \mathbf{1}_{y_i^c = 0}}{\sum_{i \in [B]} y_i^c} \tag{6}$$

With the definition of these two weights, the derivation of loss_{wBCE} is:

$$loss_{wBCE} \quad = \quad \frac{1}{C} \sum_{c \in [C]} loss_{i,c} \tag{7}$$

$$loss_{i,c} \quad = \quad -w_p^c [w_a^c sgn(y_i^c) \cdot \log(x_i^c + (1 - sgn(y_i^c)) \log(1 - x_i^c))] \tag{8}$$

In which, sgn represents the signal function. Compared with vanilla BCE loss, we found wBCE loss can significantly improve the fincal performance on internal validation set. The detailed results are provided in Section 4.5.

Furthermore, we tried multiple action recognition modules and made late fusion action-wisely according to the results on the validation set. We found each classifier does show superiority on certain actions. Through the feedback from the online leaderboard, such fusion strategy can improve the final performance with noticeable margins.

2.5 Activity Deduplication

As the system generates overlapping proposals, it could have duplicate predictions for some of the proposals. This would result in a large amount of false alarms unless we deduplicate them.

For each proposal, there are C scores corresponding to each activity. We duplicate it into C proposals, each with one activity score and perform deduplication in each type. Figure 3 is a diagram for our deduplication algorithm.

1. Split the overlapping cubes of duration D_{prop} and stride S_{prop} into non-overlapping cubes of duration S_{prop}. An output cube relies on all original cubes in the temporal window, with an averaged score and an intersected bounding box.

2. Merge the non-overlapping cubes of duration S_{prop} back into $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$ groups of non-overlapping cubes of duration D_{prop}. An output cube is merged from $\lfloor \frac{D_{prop}}{S_{prop}} \rfloor$ cubes with an averaged score and the union of bounding boxes.
3. Select the group where the maximum score resides.

The deduplication algorithm performs an interpolation upon the overlapping cubes. Each group in step 3 contains information from every classification results, maximizing the information utilization.

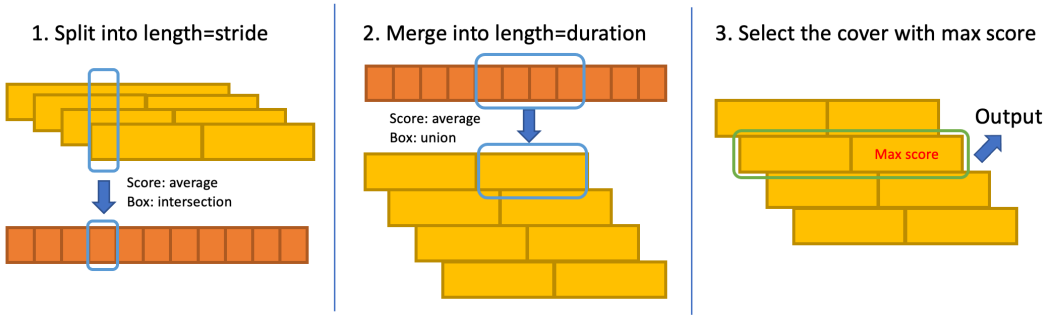

Figure 3: Activity Deduplication Algorithm

3 Experiments

3.1 Dataset and Metrics

In TRECVID 2020, a new partition of the VIRAT [2] dataset is introduced with augmented annotation of 35 activities. It contains 64 videos for training, 54 videos for validation, and 246 videos for testing. The main metrics are $nAUDC@0.2T_{fa}$ and $P_{miss}@0.15T_{fa}$ according to the evaluation plan[3]. $P_{miss}@0.15T_{fa}$ measures the recall of activity instances within a time limit of all positive frames plus 15% of negative frames. $nAUDC@0.2T_{fa}$ is the integration of P_{miss} on $T_{fa} \in [0, 0.2]$. The challenge leaderboard scores the submitted predictions on the test set. In the next section, we report the experimental results on both test set (leaderboard result) and on validation set.

3.2 Implementation Details

In the winner system, we apply Mask R-CNN [12] with a ResNet-101 [13] backbone from Detectron2 [14] pre-trained on the Microsoft COCO dataset [15] as the object detector, with $S_{det} = 8$. Only person and vehicle classes are conducted. For tracking algorithm, we apply the work in [16] and reuse the region-of-interest from the ResNet backbone as in [17, 18]. The proposals are generated with $D_{prop} = 64$ and $S_{prop} = 16$.

The original annotation in VIRAT is object bounding-boxes in each activity, which cannot be directly used to train activity classifiers. To prepare the training set for the activity classifiers, we develop a label assignment algorithm. To match the proposals with the original annotation, we convert the annotation of object bounding boxes into the cube proposals with duration D_{prop} and stride S_{prop} with a matching algorithm. The proposal sampling algorithm and the proposal generation algorithm are the same as described in former sections. For each proposal, we estimate the spatial intersection-over-union (IoU) between it and ground truth annotations in the same temporal window. Then, the label for this proposal is assigned as the each activity class with the maximum IoU score. Generally, a proposal with at least one positive score is considered a positive proposal. For negative proposals, we adopted the background subtraction, with a Gaussian mixture model from [19], algorithm to select them. The proposal filter is set with a tolerance of $P_{pos} = 0.05$.

For activity classifiers, we re-implement multiple state-of-the-art models including R(2+1)D [20], X3D [21], and Temporal Relocation Module (TRM). During training procedure, frames are cropped with jittering [11]. For X3D and TRM, we trained modules with weights pre-trained on Kinetics [22]. For R(2+1)D modules, we trained modules with weighst pre-trained on IG65M [23].

3.3 Quality analyze of Dense Spatio-temporal Proposals

First, we analyze the quality of our dense spatio-temporal proposals. We analyze the upper bound performance of both overlapping and non-overlapping proposals on VIRAT validation set. The experiments are performed by directly converting the ground truth annotations into the proposal format with activity labels, and being scored by official metrics. The results are shown in Table 1 and Table 2.

Table 1: Upper Bounds of Non-overlapping Proposals on VIRAT Validation Set

Duration (# frame)	$nAUDC@0.2T_{fa}$
32	0.1208
64	**0.0673**
96	0.0688
128	0.0788

Table 2: Upper Bounds of Overlapping Proposals on VIRAT Validation Set

Duration / Stride (# frame)	16	32
32	0.0705	-
64	**0.0127**	0.0621
96	0.0275	0.0504

It is shown in the experiments that the non-overlapping proposal format reports 6.7%. The overlapping proposals with duration 64 and stride 16 only have 1.3% systematic errors.

3.4 Performance of Proposal Filtering

We examine the quality of the proposals with and without the filter, as shown in Table 3 and 4. The metrics are calculated by assigning ground truth labels to the proposals, simulating a perfect classifier. The proposals are further filtered by IoU and reference coverage levels from 0, 0.1, to 0.9 to calculate a partial result.

Table 3: Statistics of Proposals on VIRAT Validation Set

Name	Unfiltered Proposals	Filtered Proposals
Number of Proposals	211271	62831
Positive rate	0.1704	**0.5204**
Rate of unique label	0.4558	0.4415
Rate of two labels	0.4127	0.4252
Rate of three labels	0.1017	0.1060

Table 4: Proposal Quality Metrics on VIRAT Validation Set

$nAUDC@0.2T_{fa}$	IoU			Reference Coverage		
Threshold	Average	≥ 0	≥ 0.5	Average	≥ 0.5	≥ 0.9
Unfiltered Proposals	0.2358	0.0772	0.1518	0.1562	0.1125	0.4211
Filtered Proposals	0.2352	0.0772	0.1469	0.1563	0.1099	0.4280

With the dense cube proposals, the best $nAUDC@0.2T_{fa}$ we can achieve with a ideal classifier is 0.08, as indicated in the IoU ≥ 0 column. The IoU and reference coverage bounded scores are used to measure the spatial matching quality of proposals, as the $nAUDC@0.2T_{fa}$ does not consider spatial dimensions. We can see that even with a condition of IoU ≥ 0.5, our proposal can achieve up to 0.15, which indicates the spatial preciseness. The proposal filtering is also proved effective, which removed 70% of original proposals without dropping the recall level.

3.5 Performance of Classification and Fusion

In this section, we would introduce the results of R(2+1)D [20], X3D [21], and TRM on validation set and reported the step-by-step fusion results on the official leader board.

According to the results in Table 5, we found R(2+1)D performed best on the validation set. However, as shown in the figure below, every model shows better performance on certain activities than the others. We merged these models' outputs activity-wisely for the final submission according to the experimental results on the validation set. We listed the history of milestone submissions on the

Table 5: Results of Activity Recognition Models on VIRAT Validation Set

Model	Pretraining	Input	$nAUDC@0.2T_{fa}$	Mean $P_{miss}@0.15T_{fa}$
R(2+1)D	IG65M	32×112×112	**0.356**	**0.256**
X3D	Kinetics	16×312×312	0.383	0.284
TRM	Kinetics	8×224×224	0.394	0.303

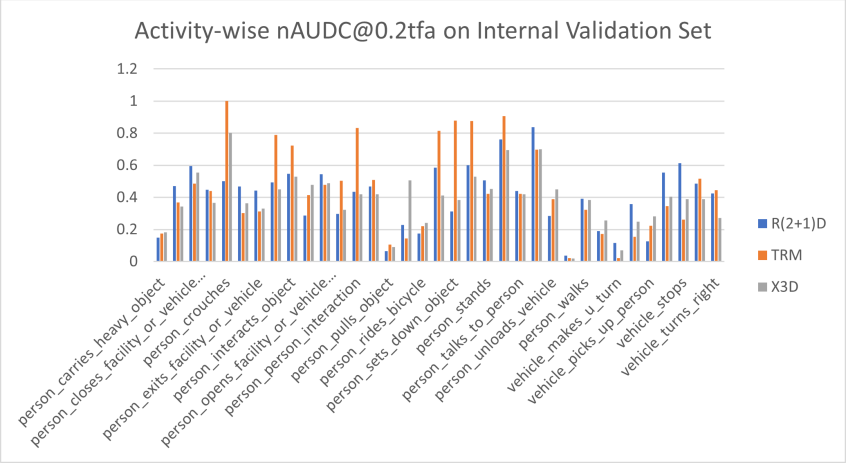

Figure 4: Activity-wise $nAUDC@0.2T_{fa}$ results of three models on the validation set

leaderboard. The online evaluation system won't give out results if the submission does not surpass the previous system. So we can not give out the performance of every single model on the official leaderboard. However, since each milestone submission surpassed the previous one with one extra model added, we think it demonstrates that each model contributes to the final performance and surpasses the others in certain activities.

Table 6: Results of Fusion Models on the Leaderboard

Model	Training Data	$nAUDC@0.2T_{fa}$
R(2+1)D	Training set	0.438
R(2+1)D	Training+validation sets	0.436
R(2+1)D+TRM	Training set	0.431
R(2+1)D+TRM	Training+validation sets	0.429
R(2+1)D+TRM+X3D	Training set	0.424
R(2+1)D+TRM+X3D	Training+validation sets	**0.423**

3.6 Leaderboard Results

The TRECVID 2020 ActEV Leaderboard result [3] is shown in Table 7. Our system (INF) achieved an $nAUDC@0.2T_{fa}$ of 0.42307, which is 23.8% better than the runner up. We also achieved a mean $P_{miss}@0.15T_{fa}$ of 0.33241, which is 31.9% better than the runner up. These extraordinary results fully verify the effectiveness of our system with dense spatio-temporal cube proposals.

Table 7: TRECVID 2020 ActEV Leaderboard

Rank	Team	Best System	$nAUDC@0.2T_{fa}$	Mean $P_{miss}@0.15T_{fa}$
1	INF	INF (Ours)	**0.42307**	**0.33241**
2	BUPT-MCPRL	MCPRL_S1	0.55515	0.48779
3	UCF	UCF-P	0.58485	0.54730
4	TokyoTech_AIST	TTA-SF2	0.79753	0.75502
5	CERTH-ITI	P	0.86576	0.84454
6	Team UEC	UEC	0.95168	0.95329
7	kindai_kobe	kind_ogu_baseline	0.96820	0.96443

4 Conclusion

In this paper, we present an action recognition system for surveillance scenarios, which wins TRECVID ActEV Challenge 2020 with a large advantage of 23.8% over the runner up. In our system, we propose a dense spatio-temporal proposal paradigm to precisely localize activities in surveillance videos and reduce false alarms. We proved the effectiveness and advantage of our novel proposal format. With state-of-the-art activity recognition models, it achieved the new state-of-the-art performance in TRECVID ActEV with $nAUDC@0.2T_{fa} = 0.42$.

Acknowledgments and Disclosure of Funding

This research is supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340. This research is supported in part through the financial assistance award 60NANB17D156 from U.S. Department of Commerce, National Institute of Standards and Technology. This project is funded in part by Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the US Department of Transportation.

References

[1] G. Awad, A. A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, J. Zhang, E. Godard, L. Diduch, J. Liu, A. F. Smeaton, Y. Graham, G. J. F. Jones, W. Kraaij, and G. Quénot, "Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains," in *Proceedings of TRECVID 2020*, NIST, USA, 2020.

[2] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, pp. 3153–3160, IEEE, 2011.

[3] N. A. team, "Trecvid actev 2020 evaluation plan." https://actev.nist.gov/pub/TRECVID_ActEV_2020_EvaluationPlan.pdf, 2020.

[4] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 408–417, 2017.

[5] Y. Qian, L. Yu, W. Liu, G. Kang, and A. G. Hauptmann, "Adaptive feature aggregation for video object detection," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pp. 143–147, 2020.

[3] https://actev.nist.gov/trecvid20#tab_leaderboard

[6] X. Chang, W. Liu, P.-Y. Huang, C. Li, F. Zhu, M. Han, M. Li, M. Ma, S. Hu, G. Kang, *et al.*, "Mmvg-inf-etrol@ trecvid 2019: Activities in extended video.," in *TRECVID*, 2019.

[7] W. Liu, G. Kang, P.-Y. Huang, X. Chang, Y. Qian, J. Liang, L. Gui, J. Wen, and P. Chen, "Argus: Efficient activity detection system for extended video analysis," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pp. 126–133, 2020.

[8] L. Yu, P. Chen, W. Liu, G. Kang, and A. G. Hauptmann, "Training-free monocular 3d event detection system for traffic surveillance," in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 3838–3843, IEEE, 2019.

[9] L. Yu, D. Zhang, X. Chen, and A. Hauptmann, "Traffic danger recognition with surveillance cameras without training data," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2018.

[10] M. N. Rizve, U. Demir, P. Tirupattur, A. J. Rana, K. Duarte, I. Dave, Y. S. Rawat, and M. Shah, "Gabriella: An online system for real-time activity detection in untrimmed security videos," *arXiv preprint arXiv:2004.11475*, 2020.

[11] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, pp. 20–36, Springer, 2016.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[14] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[16] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.

[17] L. Yu, Q. Feng, Y. Qian, W. Liu, and A. G. Hauptmann, "Zero-virus: Zero-shot vehicle route understanding system for intelligent transportation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 594–595, 2020.

[18] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann, "Electricity: An efficient multi-camera vehicle tracking system for intelligent city," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 588–589, 2020.

[19] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.

[20] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

[21] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 203–213, 2020.

[22] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[23] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12046–12055, 2019.