Argus: Efficient Activity Detection System for Extended Video Analysis

Wenhe Liu†* Guoliang Kang†* Po-Yao Huang†* Xiaojun Chang‡
Lijun Yu† Yijun Qian† Junwei Liang† Liangke Gui† Jing Wen† Peng Chen
Alexander G. Hauptmann
Carnegie Mellon University†, Monash University‡
{wenhel,poyaoh,gkang,lijunyu,yijunqia,jingwen2,junweil,liangkeg}@andrew.cmu.edu,
{cxj273,1998cpcp}@gmail.com,Alex@cs.cmu.edu

Abstract

We propose an Efficient Activity Detection System, Argus, for Extended Video Analysis in the surveillance scenario. For the spatial-temporal event detection in the surveillance video, we first generate video proposals by applying object detection and tracking algorithm which shared the detection features. After that, we extract several different features and apply sequential activity classification with them. Finally, we eliminate inaccurate events and fuse all the predictions from different features. The proposed system wins Trecvid Activities in Extended Video (ActEV[1]) challenge 2019. It achieves the first place with 60.5 mean weighted P_{miss}, outperforming the second place system by 14.5 and the baseline R-C3D by 29.0. In TRECVID 2019 Challenge[2], the proposed system wins the first place with pAUDC@0.2tfa 0.48407.

1. Introduction

In recent years, the volume of video data from widely-deployed surveillance cameras has grown dramatically. However, camera network operators are overwhelmed with the data to be monitored, and usually cannot afford to view or analyze even a small fraction of their collections. For enabling timely response for critical surveillance events,such as traffic events [32, 31], 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. With the availability of large-scale video surveillance dataset such as VIRAT [24], the Activities in Extended Videos Prize Challenge (ActEV) seeks to encourage the development of real-time robust automatic activity detection algorithms in surveillance scenarios. Specifically, an activity is defined to be "one or more people (or vehicle) performing a specified movement or interacting with an object or group of objects". Figure 1 illustrates three "talking phone" and "vehicle turning" activities. For spatial object detection, as the common practice since Faster R-CNN [26], region-based object detectors employ proposal generation and classification networks. A few recent work applied this two-stage architecture for temporal action localization [8, 30, 19, 4], and demonstrated competitive performance. In particular, R-C3D network [30] closely follows the original Faster R-CNN but in the temporal domain. There is VideoCapsuleNet [9] which use an end-to-end segmentation network for action detection. For efficiency, there are also several previous works [25, 11, 17, 10, 15] focusing on online action detection or fine-grained action detection untrimmed videos. However,these methods may not generalize to a more challenging spatial-temporal activity detection problem, which is the central scenario for surveillance video analysis.

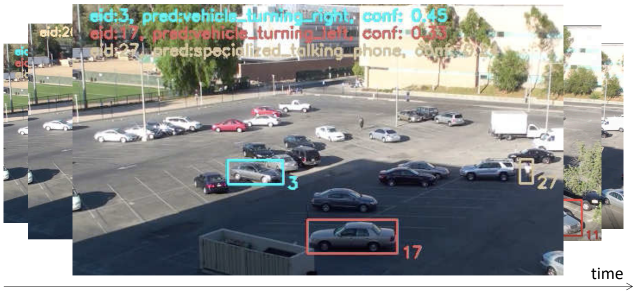
eid:3, pred:vehicle_turning_right, conf: 0.45
eid:17, pred:vehicle_turning_left, conf: 0.33
eid:27, pred:specialized_talking_phone,
eid:2
3
17
27
11
time

Figure 1. Activity detection in video surveillance scenarios.

To tackle the challenging spatial-temporal activity detection problem, we apply a divide-and-conquer strategy built on [5, 6]. We first generating a sparse set of class agnostic

[*] equal contribution
[1] https://actev.nist.gov/prizechallenge
[2] https://actev.nist.gov/trecvid19

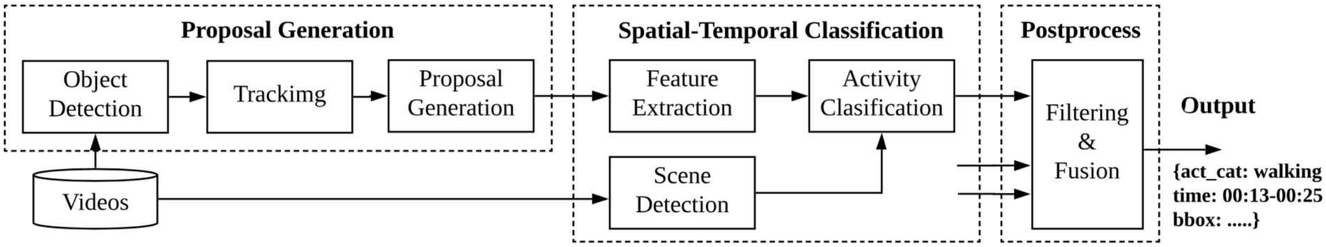

Figure 2. System architecture of Argus.

spatial-temporal proposals from the input video, followed by classifying and temporal localizing the action categories for each proposal. The proposal generation includes object detection, tracking to generated spatial-temporal tubes covering most activity priors for classification. Unlike prior spatial detection [26] or temporal localization work [8, 30, 19, 3], we incorporate domain knowledge to explicitly model human-object interaction in both spatial and temporal domains. We then employ sequential classifiers to temporally localize activities in the proposals. Our system employs and improves multiple recent methods in the sub-modules and achieves the state-of-the-art results for activity detection in video surveillance scenarios. We design a parallel framework to maximize the computation efficiency for large-scale surveillance video analysis. We term our spatial-temporal activity detection system **Argus**. We have dockerized Argus to enable SOTA surveillance video analysis with one script. In a nutshell, our contribution is twofold:

1. We propose Argus, an effective and efficient system for activity detection in extended Video analysis.
2. Argus yields SOTA results for spatial-temporal activity detection in video surveillance scenarios. It is easy to be applied to real-world large-scale surveillance event detection applications and be upgraded with future detection models.

2. The System

2.1. System Architecture

The overall system architecture is depicted in Fig. 2. We employ a two-stage system for activity detection. In the first stage we pre-process videos to generate event proposals to spatially and temporally localize candidates of activities. In the second stage, we extract features and perform temporal classification and postprocess to generate the activity detection outputs. The system is designed to achieve high recall in the first stage by increasing the proposal coverage whereas in the second stage the classification model aims to improve the precision. Argus is composed of three parts: (i) Activity proposal generation (ii) Classification (iii) Postprocess.

For Activity proposal generation, object detection model is first applied to detect person and vehicle objects. We then create tracklets and generate spatial-temporal activity proposals. To classify the activities in the proposals, we extract features and perform temporal classification to temporally localize activities. Additionally, a scene detection model is applied to provide scene information as the side-information for model switch. Lastly, results from multiple activity classifiers are filtered then being combined to generate the final outputs. In the following section we first introduce our pipeline implementation and elaborate individual module design.

2.2. Parallel Video Analysis Framework

The dataset is processed as chunks of videos. For each chunk, Argus operates parallel video analysis in the chunk.

2.2.1 Module Parallelization

Different modules require different amounts of CPU and GPU resources. For example, the proposal generation module (**P**) in Fig. 2 relies on the CPU resource, and the subsequent feature extraction module (**F**) mainly depends on the GPU resource. Based on the above reason, we can parallelize the **P** module and the **F** module. As shown in Fig. 3, we can largely reduce the additional time cost C_P brought by module **P**. Note that the length of time for extracting features by **F** is much longer than that for generating proposals of a video by **P**. Thus, the C_P approximately equals to the cost of processing only one video by **P**, which means that the **F** module doesn't need to wait for generating proposals except for those of the first video. Notably, C_P will not increase as the number of videos increases.

2.2.2 Pipeline Parallelization

Our system is a GPU-wise parallel computation system. In the experiments, we find that it is hard to predict and allocate the resource before we analysis the videos. For example, a short but dense video (i.e., a video with many proposals of events in a short time) may cost more than a long but sparse video. Therefore, we develop a GPU management subsystem to dynamically allocate GPU for pipelines. In this system, the GPU management system will monitor the GPU usage and dynamically create a new pipeline when an old one is finished. Please refer to Table 1 for details of implementations of our system.

Name	Model	Framework	GPU
Object Detection	CNN	TensorFlow	Yes
Tracking	D-SORT	TensorFlow	Yes
Proposal Generation	original	Python	No
Feature Extraction	CNN	Pytorch	Yes
Activity Classification	RNN	TensorFlow	Yes
Filtering	original	Python	No
Fusion	original	Python	No

Table 1. Implementation Detail. The model marked with 'original' is original implemented in this system.

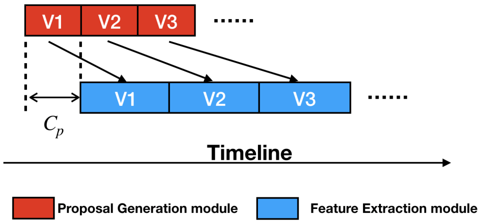

Figure 3. Parallelization between the Proposal Generation (**P**) module and the Feature Extraction (**P**) module. The v1, v2, and v3 denote different videos. For each video, the **F** module needs the proposals generated by the **P** module.

3. The Modules

3.1. Event Proposal Generation

The events of concern in ActEV are summarized in Table 3. These events involve either person or vehicle object, we use this prior knowledge to build the event proposal module starting from the object detection step. The output of this step is person and vehicle bounding box for each frame. The immediate natural next step is to associate detected object across frames, which is tracking. The output of this step is person tracklet and vehicle tracklet. Finally, we derive event proposal by designing heuristics on the tracklets. The output of this step is event proposal.

3.1.1 Object Detection

We utilize faster RCNN [26] with feature pyramid network [20] on ResNet-101 [12] as the backbone for object detection, in which RoIAlign is used to extract features for Region-of-Interest. We apply object detection on every k frame from the videos. Full resolution images are input to the model and we fine-tune our model using the full 15 object class annotation in the the VIRAT dataset.

3.1.2 Tracking

We compare the performance of deep SORT [29] and kernelized correlation filter (KCF) [13]. As shown in Table 2, deep SORT outperforms KCF for all the metrics except precision. As the tracking module is used to generate tracklets which are proposal candidates, we expect a high recall and low ID switches with a comparable precision. The results are reported in Table 2. In experiments, we also tried to using vehicle/person re-identification methods [23, 22, 21, 18] to future improve the quality of the tracklets. Re-id methods are applied to detect and merge the tracklets or the same person/vehicle. In the final system, we utilize deep SORT [29] to generate tracklets by associating detected objects across frames. We follow a similar track handling and Kalman filtering framework [29]. We use bounding box center position (u, v), aspect ratio γ, height h and their respective velocities in image coordinates as Kalman states. We compute the Mahalanobis distance between predicted Kalman states and newly arrived measurement to incorporate motion information. For each bounding box detection, we use the feature obtained from object detection module as a appearance descriptor. We compute the cosine distance between tracks and detections in appearance space. To build the association problem, we combine both metrics using a weighted sum. An association is defined admissible if it is within the gating region of both metrics.

3.1.3 Spatial-Temporal Proposal Generation

After obtaining the single object trajectories for person and vehicle respectively in videos, we generate event proposal. The event proposal can be treated as a sequence of bounding boxes corpped from each frame. We divide the events into three categories, namely: *person only proposal*, *vehicle only proposal* and *person-vehicle interaction proposal*. The categorization for the events on the VIRAT dataset is illustrated in Table 3. 1) The person and vehicle only proposals contains only events happened on a single object (i.e., either a person or a vehicle). 2) To generate proposals of person-vehicle interaction, we associate individual person and vehicle to model their interactions. We use a spatial-temporal regularization schema to obtain the interaction proposals. An intuitive illustration is shown in Figure 4 for event "person entering vehicle". Let the blue curve be the person trajectory and the red curve be the vehicle trajectory. The x-axis is the time dimension and the y-axis is the spatial dimension. In the black dashed line region, the spatial distance between person and vehicle trajectories are consistently close enough in space within the temporal window $[x1, x2]$. Finally, we use this regularization to generate event proposals from two object trajectories.

3.2. Spatial-Temporal Classification

3.2.1 Feature Extraction

We learn proposal-augmented I3D-Flow and I3D-RGB features by fine-tuning I3D [1] models for activity recognition

Models	Recall (%)	Precision (%)	ID switches	MOTA (%)	MOTAL (%)
KCF	93.5	**97.1**	2519	91.3	90.5
deep SORT	**95.2**	96.5	**909**	**91.7**	**91.8**

Table 2. Results of two multi-object tracking algorithms in the validation set of VIRAT

Type	**Events/Activities**
Person only	Transport_HeavyCarry, Riding, Talking, Activity_carrying, Specialized_talking_phone, Specialized_texting_phone, Entering, Exiting, Closing, Opening
Vehicle only	Vehicle_turning_left, Vehicle_turning_right, Vehicle_u_turn
Interaction	Open_Trunk, Loading, Closing_trunk, Unloading

Table 3. The events categorization according to proposal types on the VIRAT dataset

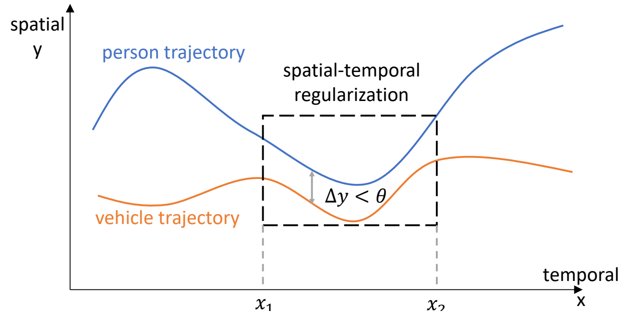

Figure 4. Illustration of the spatial-temporal regularization to obtain interaction proposals based on person and vehicle trajectories.

on VIRAT. The base models are pre-trained on ImageNet, Kinetics-600 [16], and Charades [27]. Recently, there are also latest works aim to learn the stream of action in order to replace the optical flow model [28, 7] but we use I3D-FLOW and I3D-RGB to learn the representative features. We fine-tune on the VIRAT dataset with the annotated positive event proposals and 5-times non-trivial background proposal as the negatives. We extract raw RGB and two types of raw optical flow frames (TVL1 and Farneback) from the spatial-temporal proposals for fine-tuning. The proposals are augmented by randomly scaling proposal in the temporal and spatial domain. After fine-tuning, we use the last convolutional layer as the feature for classification.

3.2.2 Activity Classification

We utilize a bi-directional LSTM [14] to perform temporal(sequential) classification to localize activities within spatial-temporal proposals. The spatial-temporal proposal generation in Sec 3.1.3 aims to cover most of the possible proposals (high recall) while the bi-LSTM classifier aims to achieve high precision. For training we temporally extend the proposals of positive events to supervise the classification model to capture the activity boundaries. Different from BSN [19], our model predict activities and locate activities boundaries simultaneously.

3.2.3 Scene Detection

To determine the scene (parking area, crossroads, etc) of a video, we apply a ResNet-101 [12] for classification. The frames of the first 20 seconds are extracted, predicted, and then averaged to determine the scene for classifier selection. To achieve the best performance on both accuracy and speed, the scene detection model use a isolate network and it did not share the feature with the action detection model.

3.3. Postprocess

3.3.1 Proposal Filtering

After classification and localization, the candidate proposals may have large spatial and temporal overlap. Thus we adopt spatial-temporal non-maximum suppression (NMS) to avoid redundant candidates. Empirically we find that the optimal IoU threshold set for suppression in NMS is high, which implies that our framework can generate less redundant proposals.

3.3.2 Fusion

To obtain the best performance, we apply late fusion in the postprocess stage. We take the prediction scores from individual proposals and heuristic average them if there intersection-over-union (IoU) is greater than a threshold. We repeat this process iteratively until the predictions converge. We fuse the models with a I3D-RGB model, and two types of I3D-Flow models.

4. Experiments

4.1. Experimental Setup

We conduct experiments on a subset of the widely used VIRAT [24] dataset which is of concern in the ActEV challenge. This subset consists of 18 event types distributed throughout 29 hours of videos. The videos are recorded using multiple models of HD video cameras at 1080p or 720p and the frame rates range between 25 and 30 Hz. The stationing cameras are mostly at the top of buildings and the view angles of cameras towards dominate ground planes range between 20 and 50 degree. The detailed events of concern can be found in Table 3. For our expeiments, all the models in the system are trained with official annotation provided for VIRAT dataset unless some of them are claimed to be fine-tuned on VIRAT with public pre-trained models.

Model	Closing	Closing Trunk	Entering	Exiting	Loading	Open Trunk	Opening	Transport Heavy Carry	Unloading	Vehicle turning left	Vehicle turning right	Vehicle u-turn	Pull	Riding	Talking	Activity carrying	Talking phone	Texting phone	mAP
I3D-RGB	66.06	35.26	17.26	23.14	12.54	16.28	40.48	28.95	15.11	48.29	60.99	33.46	55.47	48.33	52.14	23.35	1.29	0.28	32.15
I3D-Flow$_{\text{FB}}$	63.64	38.33	38.57	48.03	22.40	51.66	40.99	14.98	15.11	57.73	68.44	35.49	64.55	65.05	41.26	19.25	1.33	0.18	38.16
I3D-Flow$_{\text{TVL1}}$	58.38	45.18	46.50	57.91	21.01	51.75	47.02	21.37	27.45	55.99	70.65	29.40	58.41	79.94	45.63	23.68	2.44	0.36	41.28
Fusion	82.24	69.97	51.82	69.24	35.58	64.10	66.51	25.26	43.99	66.74	78.47	37.36	74.18	80.76	63.73	27.20	1.60	0.37	52.17

Table 4. Activity recognition results on the VIRAT testing set. (Higher is better)

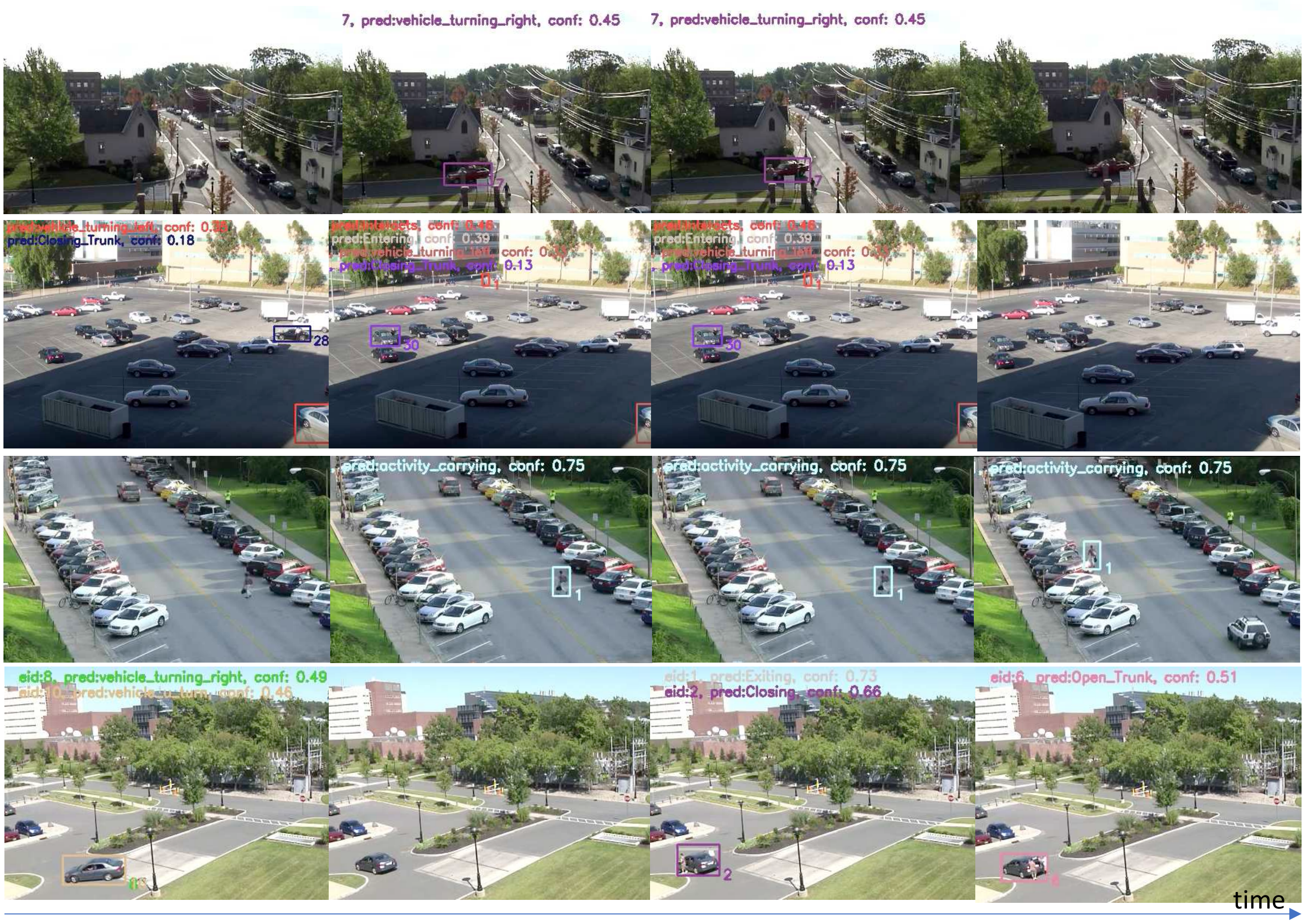
7, pred:vehicle_turning_right, conf: 0.45
7, pred:vehicle_turning_right, conf: 0.45
7
7
pred:vehicle_turning_left, conf: 0.30
pred:Closing_Trunk, conf: 0.18
28
pred:interacts, conf: 0.46
pred:Entering, conf: 0.39
pred:vehicle_turning_left, conf: 0.31
pred:Closing_Trunk, conf: 0.13
30
pred:interacts, conf: 0.46
pred:Entering, conf: 0.39
pred:vehicle_turning_left, conf: 0.31
pred:Closing_Trunk, conf: 0.13
30
pred:activity_carrying, conf: 0.75
pred:activity_carrying, conf: 0.75
pred:activity_carrying, conf: 0.75
1
1
1
eid:8, pred:vehicle_turning_right, conf: 0.49
eid:10, pred:vehicle_u_turn, conf: 0.46
eid:1, pred:Exiting, conf: 0.73
eid:2, pred:Closing, conf: 0.66
2
eid:6, pred:Open_Trunk, conf: 0.51
time

Figure 5. System output visualization.

For activity recognition, we use mean average precision (mAP) as the metric (higher is better). We use the spatial-temporal proposals defined in the VIRAT for evaluation. For activity detection, we use the P_{miss}@ metric (lower is better) defined in the ActEV challenge [3]. The system performance is evaluated using $P_{miss}(\tau)$ and $Rate_{FA}(\tau)$ which are defined as

$$P_{miss}(\tau) = \frac{8 + N_{MD}(\tau)}{10 + N_{\text{True\_Instance}}}, \tag{1}$$

and

$$Rate_{FA}(\tau) = \frac{N_{FA}(\tau)}{\text{Video\_Duration\_In\_Minutes}}. \tag{2}$$

Here, τ is the activity presence confidence score threshold, $P_{miss}(\tau)$ is the probability of missed detections at τ and $Rate_{FA}(\tau)$ is the rate of false alarms at τ. $N_{MD}(\tau)$ is the number of missed detections at τ, $N_{FA}(\tau)$ is the number of false alarms at τ, and $N_{TrueInstance}$ is the number of the true instances in the sequence. For ActEV-PC evaluations, the system performance will be evaluated using P_{miss} at $Rate_{FA} = 0.15$ for activities.

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

The implementation details is listed in Table 1. We use the data and the annotations defined in the standard training split in VIRAT to train or fine-tuning individual modules. The best model in the validation split is used for model selection. We report the activity recognition (spatial-temporal proposals are given) and activity detection (spatial-temporal proposals are generated by Argus) on the testing split.

Experiments	mean-P_{miss}@0.15rfa (%)
RC3D [30]	91.30
Team SRI	80.46
Team IBM and MIT	75.65
Team UMD	75.03
Team UCF	75.00
Argus (RGB)	79.25
Argus (I3D-Flow$_{\text{TVL1}}$)	71.52
Argus (Fusion)	**60.47**

Table 5. Activity detection results in Trecvid Activities in Extended Video (ActEV) Challenge 2019 (Lower is better). The best result is marked in bold.

Experiments	Partial AUDC@0.2fta
NIST-TEST	0.85649
NTT-CQUPT	0.60058
Hitachi	0.59889
BUPT-MCPRL	0.52408
team-arnet	0.49099
Argus (RGB)	0.49595
Argus (I3D-Flow$_{\text{TVL1}}$)	0.48615
Argus (Fusion)	**0.48407**

Table 6. Activity detection results on the VIRAT testing set on TRECVID 2019 (Lower is better). The best result is marked in bold.

4.2. Activity Classification Results

Table 4 summarizes the result of activity recognition on VIRAT. As can be seen, the augmented optical flow I3D models greatly outperform the RGB model in 13/18 events for TVL1 and Farneback. Events with smaller spatial proposals such as activities involving cell phones are harder to be recognized. Complex activities, which includes reasoning over multiple objects (i.e., “loading”, “transport heavy carry”) and longer temporal (i.e., “Vehicle u-trun”) are also challenging. With a cost of roughly 13x computation time, optical flow with TVL1 algorithm yields better performance over Farneback. The reason behind is that the I3D model weights are pre-trained TVL1 flow on Kinetics and Charades. The late fusion of the three models delivers the best activity recognition performance.

4.3. Activity Detection Results

For the challenge, we prepare our system on a four GPU (NVIDIA 1080Ti) cards machine with 128G memory and one 32-core CPU. The running time is 39,688 seconds on 246 test videos with the total duration around 6,731 seconds.

In Table 5 we show the results on mean-P_{miss}@0.15rfa in Trecvid Activities in Extended Video (ActEV) challenge 2019 [4]). In the challenge, the proposed system achieves the first place with 60.5 mean weighted P_{miss}, outperforming the second place system by 14.5 and the baseline R-C3D by 29.0.

Table 6 presents the comparisons Partial AUDC@0.2tfa [5] results of activity detection on VIRAT test dataset (reported on the official leaderboard on TRECVID 2019 Challenge [6]). As can been seen, Argus outperforms other teams by a large margin. We observed that the fusion of RGB and Flow_TVL1 feature reports the best result of 0.48407 in Partial AUDC@0.2tfa .

5. Conclusion

We propose an Efficient Activities Detection System for Extended Video Analysis in surveillance event detection. On the system level, the proposed system utilize both modular and pipeline level parallel computing strategies to optimize usage of GPU and video processing in order to perform efficient inference procedure. On the algorithm level, the proposed system is easy to implemented with state-of-the-art models and algorithm. We conducted thorough experiments on the VIRAT dataset and wins Trecvid Activities inExtended Video (ActEV) challenge 2019 and TRECVID 2019 Challenge. For future work, the proposed system can be easily adapted to other real-world applications in video analysis area.

6. Acknowledgment

This work was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00340. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation/herein.Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements,either expressed or implied, of IARPA, DOI/IBC, or the U.S.Government.

[4] https://actev.nist.gov/prizechallenge. Our Team is named as MUDSML, snapshot taken on March, 2019

[5] Partial AUDC is the area under the DET curve between a Time-based False Alarm rate of 0 and 0.2. Value of a perfect system is 0.

[6] https://actev.nist.gov/trecvid19, Our Team [2] is named as MUDSML, snapshot taken on Oct 1st, 2019

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[2] 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.

[3] X. Chang, Z. Ma, M. Lin, Y. Yang, and A. G. Hauptmann. Feature interaction augmented sparse learning for fast kinect motion detection. *IEEE Trans. Image Processing*, 26(8):3911–3920, 2017.

[4] X. Chang, Y. Yu, Y. Yang, and E. P. Xing. Semantic pooling for complex event analysis in untrimmed videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(8):1617–1632, 2017.

[5] J. Chen, P.-Y. Huang, J. Liu, J. Liang, T.-Y. Hu, W. Ke, W. Barrios, Vaibhav, X. Chang, H. Dong, A. Hauptmann, S. Chen, and Q. Jin. Informedia @ trecvid 2018: Ad-hoc video search, video to text description, activities in extended video. In *Proceedings of TRECVID 2018*. NIST, USA, 2018.

[6] J. Chen, J. Liu, J. Liang, T. Hu, W. Ke, W. Barrios, D. Huang, and A. G. Hauptmann. Minding the gaps in a video action analysis pipeline. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 41–46, Jan 2019.

[7] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid. Mars: Motion-augmented rgb stream for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7882–7891, 2019.

[8] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017.

[9] K. Duarte, Y. Rawat, and M. Shah. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems*, pages 7610–7619, 2018.

[10] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768, 2015.

[11] J. Gleason, R. Ranjan, S. Schwarcz, C. Castillo, J.-C. Chen, and R. 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.

[12] 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*, pages 770–778, 2016.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2014.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] R. Hou, C. Chen, and M. 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.

[16] 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.

[17] T. S. Kim, Y. Zhang, Z. Xiao, M. Peven, W. Qiu, J. Bai, A. Yuille, and G. D. Hager. Safer: Fine-grained activity detection by compositional hypothesis testing.

[18] R. Kuma, E. Weill, F. Aghdasi, and P. Sriram. Vehicle re-identification: an efficient baseline using triplet embedding. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2019.

[19] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[21] W. Liu, X. Chang, L. Chen, D. Phung, X. Zhang, Y. Yang, and A. G. Hauptmann. Pair-based uncertainty and diversity promoting early active learning for person re-identification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(2):1–15, 2020.

[22] W. Liu, X. Chang, L. Chen, and Y. Yang. Early active learning with pairwise constraint for person re-identification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 103–118. Springer, 2017.

[23] W. Liu, X. Chang, L. Chen, and Y. Yang. Semi-supervised bayesian attribute learning for person re-identification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[24] 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*, pages 3153–3160. IEEE, 2011.

[25] A. J. Rana, P. Tirupattur, M. N. Rizve, K. Duarte, U. Demir, Y. Rawat, and M. Shah. An online system for real-time activity detection in untrimmed surveillance videos.

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[27] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016.

[28] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.

[29] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.

[30] H. Xu, A. Das, and K. 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.

[31] 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)*. IEEE, 2019.

[32] 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)*, pages 1–6. IEEE, 2018.