MMVG-INF-Etrol@TRECVID 2019: Activities in Extended Video

Xiaojun Chang† **Wenhe Liu** **Po-Yao Huang** **Changlin Li†** **Fengda Zhu†**

Mingfei Han† **Mingjie Li†** **Mengyuan Ma†** **Siyi Hu†** **Guoliang Kang**

Junwei Liang **Liangke Gui** **Lijun Yu** **Yijun Qian** **Jing Wen**

Alexander Hauptmann
Monash University† and Carnegie Mellon University

Abstract

We propose a video analysis system detecting activities in surveillance scenarios which wins Trecvid Activities in Extended Video (ActEV[1]) challenge 2019. For detecting and localizing surveillance events in videos, Argus employs a spatial-temporal activity proposal generation module facilitating object detection and tracking, followed by a sequential classification module to spatially and temporally localize persons and objects involved in the activity. We detail the design challenges and provide our insights and solutions in developing the state-of-the-art surveillance video analysis system.

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, 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 Oh et al. (2011), the Activities in Extended Videos Prize Challenge (ActEV-PC) 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 Ren et al. (2015), region-based object detectors employ proposal generation and classification networks. A few recent work applied this two-stage architecture for temporal action localization Dai et al. (2017); Xu et al. (2017); Lin et al. (2018); Chang et al. (2017b), and demonstrated competitive performance. In particular, R-C3D network Xu et al. (2017) closely follows the original Faster R-CNN but in the temporal

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

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

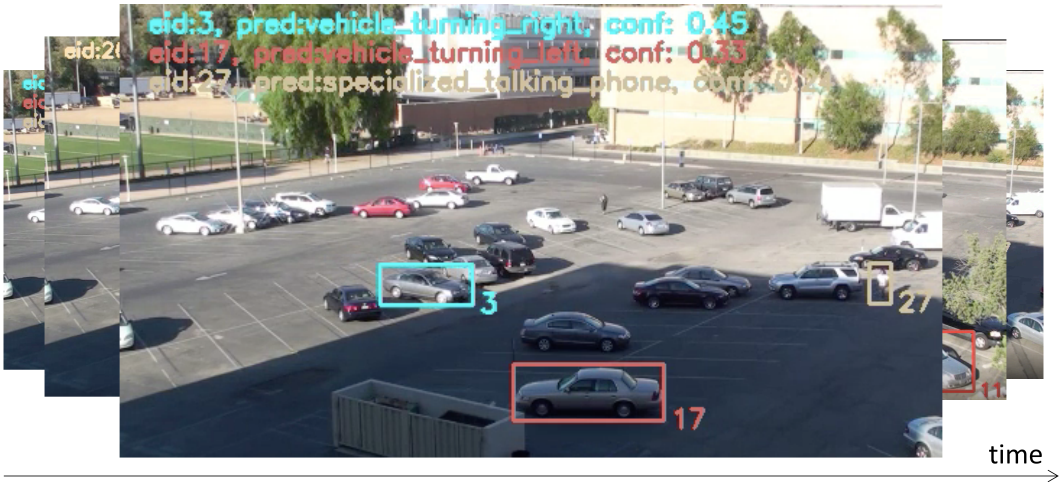

Figure 1: Activity detection in video surveillance scenarios.

domain. However, these methods do not generalize to a more challenging spatial-temporal activity detection problem, which is the central scenario for surveillance video analysis.

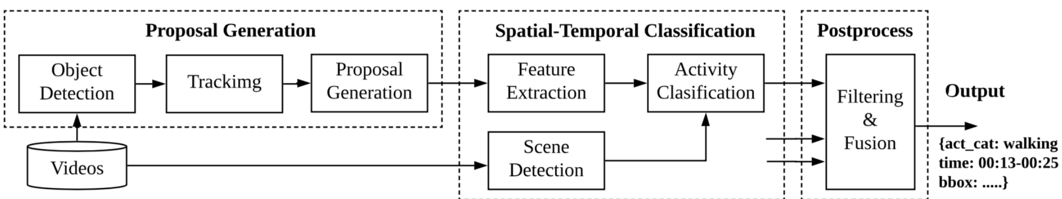

Figure 2: System architecture of Argus.

To tackle the challenging spatial-temporal activity detection problem, we apply a divide-and-conquer strategy built on Chen et al. (2018); Chen et al. (2019). We first generating a sparse set of class agnostic 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 Ren et al. (2015) or temporal localization work Dai et al. (2017); Xu et al. (2017); Lin et al. (2018); Chang et al. (2017a), 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. Argus yields SOTA results for spatial-temporal activity detection in video surveillance scenarios.
2. We open-source Argus to reproduce our results and accelerate research for surveillance video analysis.

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. Additionaly, 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 ensembled 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. Then 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 the documentation in our open-source repository for more details.

3 The Modules

3.1 Event Proposal Generation

The events of concern in ActEV are summarized in Table 1. 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 Ren et al. (2015) with feature pyramid network Lin et al. (2017) on ResNet-101 He et al. (2016) 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 utilize deep SORT Wojke et al. (2017) to generate tracklets by associating detected objects across frames. We follow a similar track handling and Kalman filtering framework Wojke et al. (2017). 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.

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 1: The events categorization according to proposal types on the VIRAT dataset.

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 1. 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 3 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.

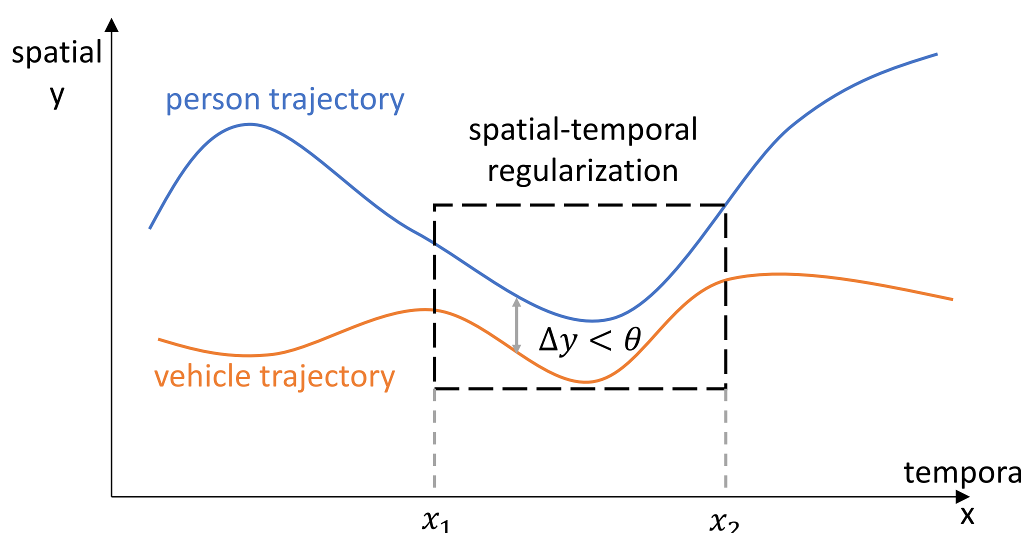

Figure 3: Illustration of the spatial-temporal regularization to obtain interaction proposals based on person and vehicle trajectories.

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$_{FB}$	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$_{TVL1}$	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 2: Activity recognition results on the VIRAT testing set. (Higher is better)

3.2 Spatial-Temporal Classification

3.2.1 Feature Extraction

We learn proposal-augmented I3D-Flow and I3D-RGB features by fine-tuning I3D Carreira and Zisserman (2017) models for activity recognition on VIRAT. The base models are pre-trained on ImageNet, Kinetics-600 Kay et al. (2017), and Charades Sigurdsson et al. (2016). 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 Spatial-Temporal Classification

We utilize a bi-directional LSTM Hochreiter and Schmidhuber (1997) 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 Lin et al. (2018), 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 He et al. (2016) for classification. The frames of the first 20 seconds are extracted, predicted, and then averaged to determine the scene for classifier selection. The detailed scene-classifier mapping could be found in our open-source repository.

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.

Experiments	mean-P_{miss}@0.15rfa
RC3D Xu et al. (2017)	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 3: Activity detection results on the VIRAT testing set. (Lower is better). The best result is marked in bold.

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 4: Implementation Detail. The model marked with 'original' is original implemented in this system.

4 Experiments

4.1 Experimental Setup

We conduct experiments on a subset of the widely used VIRAT Oh et al. (2011) 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 1.

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 [2]. 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.

The implementation details is listed in Table 4. 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.

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

Figure 4: System output visualization.

4.2 Activity Classification Results

Table 2 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.

Table 3 presents the comparisons of mean-P_{miss}@0.15rfa results of activity detection on VIRAT test dataset (reported on the official leaderboard [3]). 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 60.47 in mean-P_{miss}@0.15rfa.

5 Conclusion

We presented Argus, the state-of-the-art system for spatial-temporal activity detection in video surveillance scenarios. We conducted thorough experiments on the VIRAT dataset. The system presented in this paper can be easily adapted to other real-world applications. We hope that open-sourced Argus would accelerate research in the field of activity detection in surveillance videos. Please refer to *https://github.com/wenhel/argus* for the detailed documentation, scripts, and the dockerized video analysis tools.

[3]https://actev.nist.gov/prizechallenge#tab_leaderboard. Our Team is named as MUDSML

References

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

Xiaojun Chang, Zhigang Ma, Ming Lin, Yi Yang, and Alexander G. Hauptmann. 2017a. Feature Interaction Augmented Sparse Learning for Fast Kinect Motion Detection. *IEEE Trans. Image Processing* 26, 8 (2017), 3911–3920.

Xiaojun Chang, Yaoliang Yu, Yi Yang, and Eric P. Xing. 2017b. Semantic Pooling for Complex Event Analysis in Untrimmed Videos. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 8 (2017), 1617–1632.

Jia Chen, Po-Yao Huang, Jiang Liu, Junwei Liang, Ting-Yao Hu, Wei Ke, Wayner Barrios, Vaibhav, Xiaojun Chang, Huang Dong, Alexander Hauptmann, Shizhe Chen, and Qin Jin. 2018. Informedia @ TRECVID 2018: Ad-hoc Video Search, Video to Text Description, Activities in Extended video. In *Proceedings of TRECVID 2018*. NIST, USA.

J. Chen, J. Liu, J. Liang, T. Hu, W. Ke, W. Barrios, D. Huang, and A. G. Hauptmann. 2019. Minding the Gaps in a Video Action Analysis Pipeline. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*. 41–46. https://doi.org/10.1109/WACVW.2019.00015

Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. 2017. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*. 5793–5802.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).

Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 3–19.

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

Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. 2011. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*. IEEE, 3153–3160.

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

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*. Springer, 510–526.

Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP*.

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

Inf@TRECVID 2019: Instance Search Task

En Yu[1,2]**, Wenhe Liu**[2]**, Guoliang Kang**[2]**, Xiaojun Chang**[3]**,**
Jiande Sun[1] **and Alexander Hauptmann**[2]

[1]Shandong Normal University, [2]Carnegie Mellon University, [3]Monash University

Abstract

We participated in one of the two types of Instance Search task in TRECVID 2019: Fully Automatic Search, without any human intervention. Firstly, the specific person and action are searched separately, and then we re-rank the two sorts of search results by ranking the one type scores according to the other type, as well as the score fusion. And thus, three kinds of final instance search results are submitted. Specifically, for the person search, our baseline consists of face detection, alignment and face feature selection. And for the action search, we integrate person detection, person tracking and feature selection into a framework to get the final 3D features for all tracklets in video shots. The official evaluations showed that our best search result gets the 4th place in the Automatic search.

1 Task Description

In TRECVID 2019 [1], a query type in Instance Search task is proposed to retrieve specific person doing specific actions [2]. And it also derives two submission types, i.e., Fully Automatic (F) runs and Interactive (I) runs, depending on human intervention is involved or not. In detail, 30 queries are released, and each of them includes 4 example images for person containing the person of interest and example videos for the corresponding action. Besides, a mass of video shots segmented from BBC Eastenders test videos are given as the retrieved samples, while the type of training data can be chosen according to official requirements by ourselves. And all teams should demonstrate the types of training data by the notations of 'A' and 'E', in which "A" means video examples are not used while "E" is the opposite.

We only focus on the Fully Automatic search and the video examples are also used in our method. And thus, Table 1 shows all of our submissions and the evaluation results (MAP) according to different ranking strategies. Also, the comparisons with other teams are illustrated in Table 2 under the same setting, and we can find our team gets the 4th place in Fully Automatic task.

2 Our Method

In this time, we focus on the Fully Automatic runs and design two baselines for person search and action search respectively, and also three re-ranking strategies are used to get the final submissions [3].

2.1 Person Search

For person search, as shown in Figure 1, we firstly utilize the MTCNN model [4] to detect and crop the faces from frames, and then the cropped faces are fed into the face recognizer VGG-Face2 [5] for feature selection. Finally, we use the Cosine Distance to measure the similarities between queries and retrieved samples.

Table 1: Results of our submissions

Task Type	Submission ID	MAP	Ranking Type
F	Inf_run1_E	0.017	Person-Based
	Inf_run2_E	0.013	Action-Based
	Inf_run3_E	0.001	Fusion-Based

Table 2: Comparison with other teams at the same setting

Task Type	Team+Submission (best)	MAP
F	PKU_ICST + run_F_E	0.239
	BUPT_MCPRL + run_F_E	0.119
	NII_Hitachi_UIT + run_F_E	0.024
	Inf + run_F_E (ours)	0.017
	WHU_NERCMS + run_F_E	0.017
	HSMW_TUC +run_F_E	0.009

2.2 Action Search

For action search, from Figure 2, we can find that the Faster-RCNN [6] model pre-trained on MSCOCO dataset [7] is firstly used for person detect. In order to include the actions and objects completely, the proposals generated by person detection are expanded by 15% to the periphery. And then the tracklets for each person are generated via DeepSort [8] Tracking algorithm. After that, we fine-tune the RGB benchmark of I3D [9] model on the combination Charades dataset [10] and the offered video shots to extract the features of tracklets. Similarly, the Cosine similarity is used for action ranking.

2.3 Re-Ranking

Since we have got the person ranking and action respectively, three re-ranking methods are proposed to obtain the final results. Here we regard the query, ‘Ian + Holding_Glass’, as an example to describe these strategies. (1) Person-Search-Based: this strategy aims at using the person ranking to re-rank the action search rank list. We first select the shotID list about ‘Ian’ from person search list and the ShotID list about ‘Holding_Glass’ from action search list, respectively. And then the intersection of these two shotID lists. Finally, the final rank list is based on the intersection from the selected person search rank list. (2) Action-Search-Based: the first step is same with Person-Search-Based strategy, while the third step is based on the intersection from the selected action search rank list. (3) Fusion-Based: it is re-ranked by the average similarities between person search and action search.

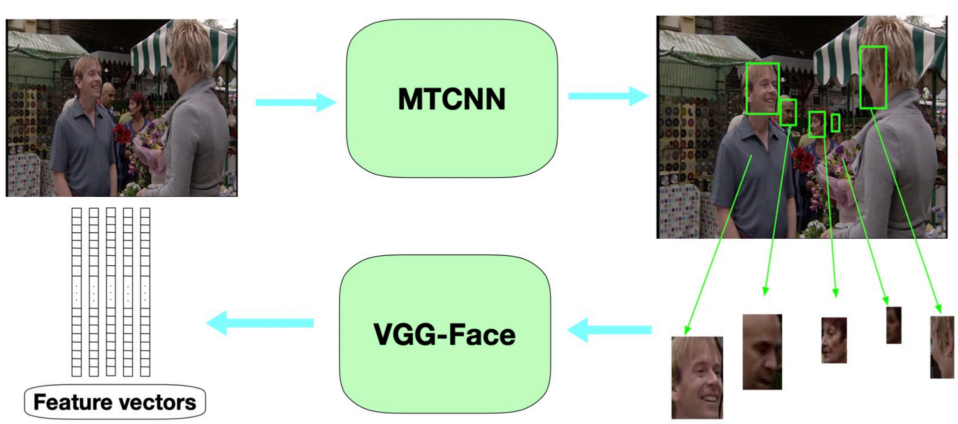

Figure 1: Framework of the face feature selection

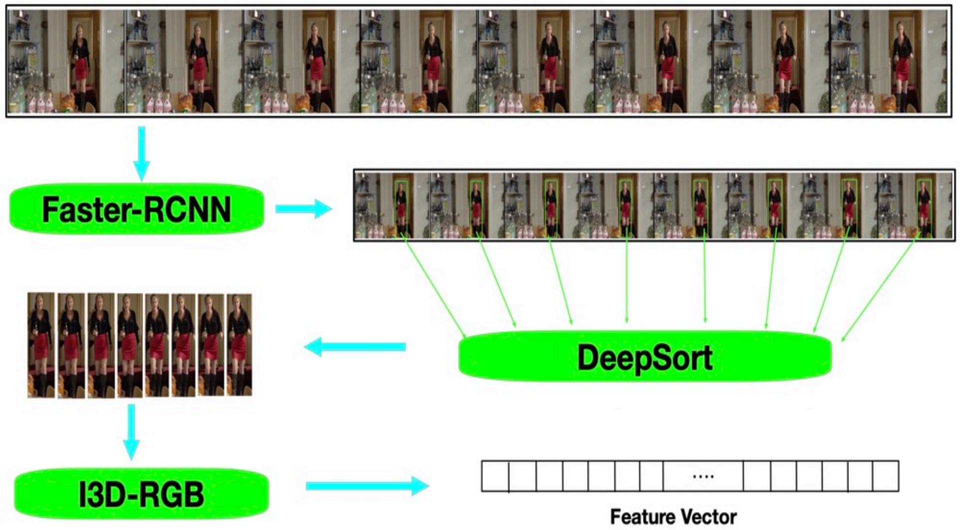

Figure 2: Framework of the action feature selection

3 Conclusion

We designed the person search and action search baseline for feature selection, and three re-ranking strategies were used for final submission. By doing the INS task in TRECVID 2019, we find that many target people only have side faces or it is very blurred in the testing dataset. So in future work, we will add super-resolution processing and person ReID technology to improve the accuracy of person search.

From the evaluation results, we can find the Person-Search-Based strategy got a better performance compared with the other two results. It demonstrated that our method can get accurate result for person search, while action search baseline can not reach the accurate representations for actions. Thus, we will also improve the action search baseline in future works.

Acknowledgments

This work was supported by Natural Science Foundation for Distinguished Young Scholars of Shandong Province (JQ201718), Natural Science Foundation of China (U1736122), Shandong Provincial Key Research and Development Plan (2017CXGC1504), Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00340.

References

[1] Awad G, Butt A, Curtis K, et al. TRECVID 2019: An evaluation campaign to benchmark Video Activity Detection, Video Captioning and Matching, and Video Search & retrieval. *Proceedings of TRECVID 2019*, 2019.

[2] Awad G, Kraaij W, Over P, et al. Instance search retrospective with focus on TRECVID. *International journal of multimedia information retrieval*, 2017, 6(1): 1-29.

[3] Yu E, Sun J, Li J, et al. Adaptive semi-supervised feature selection for cross-modal retrieval. *IEEE Transactions on Multimedia*, 2018, 21(5): 1276-1288.

[4] Zhang K, Zhang Z, Li Z, et al. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016, 23(10): 1499-1503.

[5] Cao Q, Shen L, Xie W, et al. Vggface2: A dataset for recognising faces across pose and age. *IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018: 67-74.

[6] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015: 91-99.

[7] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context. *European conference on computer vision*, 2014: 740-755.

[8] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. *IEEE International Conference on Image Processing (ICIP)*, 2017: 3645-3649.

[9] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 6299-6308.

[10] Sigurdsson G A, Varol G, Wang X, et al. Hollywood in homes: Crowdsourcing data collection for activity understanding. *European Conference on Computer Vision*, 2016: 510-526.