UA-DETRAC 2017: Report of AVSS2017 & IWT4S Challenge on Advanced Traffic Monitoring

Siwei Lyu[1,17], Ming-Ching Chang[1], Dawei Du[2], Longyin Wen[3], Honggang Qi[2], Yuezun Li[1], Yi Wei[1], Lipeng Ke[2], Tao Hu[2], Marco Del Coco[4], Pierluigi Carcagnì[4], Dmitriy Anisimov[5], Erik Bochinski[6], Fabio Galasso[7], Filiz Bunyak[8], Guang Han[9], Hao Ye[10], Hong Wang[10], Kannappan Palaniappan[8], Koray Ozcan[11], Li Wang[12], Liang Wang[13], Martin Lauer[14], Nattachai Watcharapinchai[15], Nenghui Song[1], Noor M. Al-Shakarji[8], Shuo Wang[11], Sikandar Amin[7], Sitapa Rujikietgumjorn[15], Tatiana Khanova[5], Thomas Sikora[6], Tino Kutschbach[6], Volker Eiselein[6], Wei Tian[14], Xiangyang Xue[12], Xiaoyi Yu[9], Yao Lu[16], Yingbin Zheng[10], Yongzhen Huang[13], Yuqi Zhang[13]

[1]University at Albany, State University of New York, USA, [2]University of Chinese Academy of Sciences, China, [3]GE Global Research, USA, [4]National Research Council, Italy, [5]Intel, Nizhny Novgorod, Russia, [6]Technische Universität Berlin, Germany, [7]OSRAM GmbH, Germany, [8]University of Missouri Columbia, USA, [9]Nanjing University of Posts and Telecommunications, China, [10]Shanghai Advanced Research Institute, Chinese Academy of Sciences, China, [11]Iowa State University, USA, [12]Fudan University, China, [13]Institute of Automation, Chinese Academy of Sciences, China, [14]Karlsruhe Institute of Technology, Germany, [15]National Electronics and Computer Technology Center, Thailand, [16]University of Washington, USA, [17]Tianjin Normal University, China

Abstract

The rapid advances of transportation infrastructure have led to a dramatic increase in the demand for smart systems capable of monitoring traffic and street safety. Fundamental to these applications are a community-based evaluation platform and benchmark for object detection and multi-object tracking. To this end, we organize the AVSS2017 Challenge on Advanced Traffic Monitoring*, in conjunction with the* International Workshop on Traffic and Street Surveillance for Safety and Security *(IWT4S), to evaluate the state-of-the-art object detection and multi-object tracking algorithms in the relevance of traffic surveillance. Submitted algorithms are evaluated using the large-scale UA-DETRAC benchmark and evaluation protocol. The benchmark, the evaluation toolkit and the algorithm performance are publicly available from the website* http://detrac-db.rit.albany.edu.

1. Introduction

One problem plaguing most modern urban areas is the inefficient and ineffective transportation systems. With the advent of ubiquitous, inexpensive and smart camera systems, there is an urgent demand for efficient and effective video analysis technologies for the purpose of safety and security in traffic and street surveillance. Fundamental to these tasks are reliable object detection and multi-object tracking algorithms, and large-scale traffic surveillance benchmarks and evaluation protocols.

Over the past two decades, a large number of detection and tracking methods have been proposed [17, 12, 20, 5, 30, 1, 2, 11, 34, 3]. However, many challenging factors still affect the performance including scale changes, occlusions, and background clutters. It is crucial to evaluate the performance of the state-of-the-art algorithms to understand their strength/weakness and facilitate the development of improved algorithms.

Several benchmarks have been constructed to provide a unified platform to evaluate and compare various algorithms, which can be organized into two main categories: (1) detection benchmarks (*e.g.*, INRIA [9], ETH [15], Caltech [13], KAIST [24]) and (2) tracking benchmarks (*e.g.*, PETS2009 [18], MOT15 [25], and MOT16 [28]). Most existing detection benchmarks focus on person detection (*i.e.*, pedestrians). However, vehicles are rather central to smart transportation, traffic control and surveillance. The KITTI benchmark suite [19] provides evaluation on vehicle detection and tracking, however the views are collected from the viewpoint of (first person) moving cars rather than typical surveillance views. Finally, we note that high-quality groundtruth annotation may be labor extensive but is crucial for evaluations.

To this end, in conjunction with the *International Workshop on Traffic and Street Surveillance for Safety and Security* (IWT4S), we organize the *AVSS2017 Challenge on Advanced Traffic Monitoring* to evaluate the state-of-the-art object detection and multi-object tracking algorithms in the relevance of traffic surveillance. The challenge is based on the **U***niversity at* **A***lbany* **DE***tection and* **TRAC***king* (UA-DETRAC), which is a large scale traffic surveillance benchmark and dataset and the DETRAC evaluation protocol [33]. A dedicated website (http://detrac-db.rit.albany.edu) for the challenge was built and maintained by the *Computer Vision and Machine Learning Lab*[1] at University at Albany, State University of New York, USA, where the benchmark and evaluation toolkits are available to the public. This challenge will help advancing the state-of-the-art of detection and tracking algorithms, and motivate new applications of traffic surveillance video analysis.

2. The Challenge

The goal of the *AVSS2017 Challenge on Advanced Traffic Monitoring* is to provide a comprehensive performance evaluation to the state-of-the-art detection and multi-object tracking algorithms for traffic and street surveillance applications. Participants of the challenge are required to integrate their algorithms into the DETRAC toolkit and perform the corresponding evaluations. Once the required data are received on the DETRAC submission website (http://detrac-db.rit.albany.edu), the DETRAC evaluation protocol will then be automatically executed, and results will be sent to the participant team. Details on the dataset and evaluation protocol are available in [33], and challenge results are available at the challenge website (https://iwt4s.wordpress.com/challenge/).

To ensure fair evaluations, we request that parameter tuning can only be done on the training data of the UA-DETRAC benchmark. Only the best performing setting of a participant method can be submitted for testing evaluation on the UA-DETRAC server. Exploiting the evaluation server for parameter tuning on the test set is strictly forbidden by enforcing that (1) annotations of the testing data are not made available and (2) all submitted algorithms are required to run with fixed parameters on all experiments. Specifically, each individual algorithm is allowed to change its parameters internally. Yet the parameters have to be set automatically, *e.g.*, the aspect ratio of objects and the length of trajectories. Notably, parameters are not to be hand-tuned for a specific sequence.

[1] http://www.cs.albany.edu/cvml.

2.1. UA-DETRAC Benchmark

The UA-DETRAC benchmark [33] consists of 100 video sequences, which are selected from over 10 hours of videos taken with a Canon EOS 550D camera at 24 different locations in China, representing various common traffic types and conditions including urban highway, traffic crossings and junctions. The videos are recorded at 25 frames per seconds (fps), with a spatial resolution of 960×540 pixels. In total, there are more than $140,000$ frames in the UA-DETRAC benchmark that are manually annotated corresponding to $8,250$ different vehicles. Similar to the PASCAL VOC [16], we delineate *do-not-care* regions corresponding to far field of the videos where vehicles are small and with low spatial resolutions. The UA-DETRAC benchmark is divided into a training set (60 sequences) and a testing set (40 sequences). Training videos are taken at different locations from the testing videos, but with similar traffic conditions and attributes.

2.2. DETRAC Evaluation Protocol

The average precision (AP) score of the precision-recall (PR) curve is a standard metric used in the literature to indicate the performance of detection algorithms. Larger AP score indicates better detection performance. The *multi-object tracking accuracy* (MOTA) and *multi-object tracking precision* (MOTP) are standard tracking metrics. We use the DETRAC metrics [33], which are the extension and combination of standard metrics *i.e.*, PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores, to evaluate tracking performance. Note that the PR-MOTA curve [33] is a three dimension curve characterizing the relation between detection performance (precision and recall) and object tracking performance (MOTA) from the CLEAR MOT metrics [4]. From the PR-MOTA curve, we calculate the integral score to rank multi-object tracking methods. Specifically, we accumulate the MOTA score in different normalized detection score threshold[2]. In this way, we can compare different multi-object tracking algorithms by integrating the effect of detections. The scores of other seven metrics, *e.g.*, PR-MOTP and PR-IDS, are calculated similarly. Details are described in [33].

2.3. Challenge Rules

In the *AVSS2017 Challenge on Advanced Traffic Monitoring*, the evaluation set is divided into two difficulty levels (*beginner* and *experienced*):

- **Beginner**: participants of the 'beginner' difficulty should submit only the (detection or tracking) results from the evaluation set that are marked as 'easy' in the

[2] We vary the threshold in the range of $0.0 : 0.1 : 1.0$. Thus different detection inputs can be gradually generated corresponding to different values of precision and recall.

Table 1. **Detection performance:** *AP scores of submitted vehicle detection algorithms on the UA-DETRAC-test set in the beginner (easy) / experienced (medium and hard) levels in various environmental conditions. "−" indicates the data is not available. Bold faces are the top performer and underlines corresponds to second runner-up.*

Methods	Overall	Easy	Medium	Hard	Cloudy	Night	Rainy	Sunny
GP-FRCNN (D.1)	91.90/**76.57**	91.90/**91.79**	−/**80.85**	−/**66.05**	92.77/**81.23**	**92.91**/**77.20**	82.77/**68.59**	93.96/**85.16**
EB (D.2)	89.57/67.99	89.57/87.77	−/73.03	−/54.74	94.68/75.13	90.26/71.80	71.34/52.99	90.57/82.04
SSDR (D.3)	79.47/59.07	79.47/77.84	−/64.41	−/45.98	88.86/62.79	75.27/60.88	70.26/48.55	86.36/74.32
RCNN-SC (D.4)	**93.43**/−	**93.43**/−	−/−	−/−	**96.69**/−	92.54/−	**87.30**/−	**94.47**/−
FRCNN-Res (D.5)	82.90/61.65	82.90/82.90	−/66.89	−/48.14	82.93/61.97	82.49/65.88	86.44/59.13	83.14/59.17
DFCN (D.6)	86.86/65.82	86.86/86.83	−/72.96	−/50.47	98.05/69.90	83.96/69.41	70.71/54.11	88.45/80.79
DPM	34.63/25.70	34.63/34.42	−/30.29	−/17.62	32.54/24.78	36.71/30.91	40.26/25.55	50.53/31.77
ACF	54.80/46.35	54.80/54.27	−/51.52	−/38.07	71.95/58.30	45.20/35.29	43.76/37.09	73.07/66.58
R-CNN	59.71/48.95	59.71/59.31	−/54.06	−/39.47	74.14/59.73	94.46/39.32	90.81/39.06	76.64/67.52
CompACT	65.50/53.23	65.50/64.84	−/58.70	−/43.16	77.27/63.23	61.98/46.37	57.68/44.21	77.35/71.16

Table 2. **Detection speed:** *Average running speed of the detection algorithms (in FPS) when running on the UA-DETRAC-test set. "×" indicates the GPU is not used.*

Detectors	CPU	Frequency	GPU	RAM	Codes	Speed
GP-FRCNN (D.1)	12×Intel Xeon E5-2690v3	2.60GHz	Tesla K40	256GB	Python, C++	4.00
EB (D.2)	4×Intel Core i7-4770	3.40GHz	TitanX	16GB	C++	11.00
SSDR (D.3)	8×Intel Core i7-6700K	4.00GHz	GTX1080	32GB	C++	**34.00**
RCNN-SC (D.4)	2×Intel Xeon E5-2630v3	4.00GHz	2×Tesla K80	384GB	Python, Tensorflow	2.20
FRCNN-Res (D.5)	16×Intel Xeon X5570	2.93GHz	2×Titan X	96GB	Python, Tensorflow	1.00
DFCN (D.6)	11× Intel Xeon E5-1650v3	3.50GHz	Titan X	64GB	Python	11.00
DPM	4×Intel Core i7-6600U	2.60GHz	×	8GB	Matlab,C++	0.17
ACF	2×Intel Xeon E5-2470v2	2.40GHz	×	64GB	Matlab	0.67
R-CNN	2×Intel Xeon E5-2470v2	2.40GHz	Tesla K40	64GB	Matlab,C++	0.10
CompACT	2×Intel Xeon E5-2470v2	2.40GHz	Tesla K40	64GB	Matlab,C++	0.22

UA-DETRAC dataset. The threshold of metric baselines for detection and tracking are 65% AP score and 14% PR-MOTA, respectively.

- **Experienced**: participants of the 'experienced' difficulty should submit all detection results or tracking results from the evaluation set that are marked as 'medium' and 'hard' in the UA-DETRAC dataset. The threshold of metric baselines for detection and tracking are 55% AP score and 7% PR-MOTA, respectively.

We further analyze the performance of submitted algorithms under the 4 weather conditions provided in the UA-DETRAC dataset, *i.e.*, *cloudy*, *night*, *sunny*, and *rainy*.

3. Results and Analyses

As discussed in Section 2.3, in this section, we analyze all results included the baseline and submitted participant methods that scored more than the threshold in the challenge. We begin with a short overview of the algorithms considered in the challenge, present results, and discuss overall findings. Detailed descriptions of evaluated algorithms are Section 4.

3.1. Vehicle Detection

We have received 7 entries of detectors from various authors in the IWT4S 2017 challenge. One participant performed the beginner experiment, and the rest performed both the beginner and experienced experiments. The UA-DETRAC committee additionally performed both experiments with 4 baseline detection methods. Thus a total of 11 algorithms were included in the detection task of IWTS4 2017 challenge.

As shown in Table 1, significant progress has been made in vehicle detection by the aid of various kinds of state-of-the-art convolutional networks, such as R-CNN [31], SSD [27] and ResNet [21]. For beginner level, the best RCNN-SC (D.4) achieves 93.43% AP in easy sequences, which benefits from the prior knowledge of vehicle types (*i.e.*, car, bus, van and others). For experienced level, GP-FRCNN (D.1) is ranked number one in accuracy, which proposes an automatic estimation of the approximate scene geometry by ranking proposals to yield less false positives in background cluttered areas, and less false negatives for small vehicles. Besides, it is worth pointing out SSDR (D.3) uses lightweight feature extraction network to preserve reasonable number of computations for realtime application (*i.e.*, 34.00 FPS).

However, the results of the submitted detectors on the UA-DETRAC benchmark show that there is still room for improvement for vehicle detectors. Weather conditions, such as rainy and night, significantly affect the performance of detectors. Existing vehicle detectors do not perform well when the appearance changes caused by pool lighting conditions are significant at rainy days. Similar trends are found in cluttered scenes of hard sequences. Besides, different detection algorithms requires different platform for testing. As such, it is hard to compare the tracking speed of the evaluated detection algorithms directly, which are listed in Table 2 for reference.

Table 3. **Tracking performance:** *DETRAC metrics of submitted object tracking algorithms on the UA-DETRAC-test set in the beginner (easy)/ experienced (medium and hard) levels in various environmental conditions. "−" indicates the data is not available. "∗" indicates the tracking method uses the private detections for association. Bold faces are the top performer and underlines corresponds to second runner-up.*

Methods	PR-MOTA↑	PR-MOTP↑	PR-MT↑	PR-ML↓	PR-IDS↓	PR-FM↓	PR-FP↓	PR-FN↓
GOG+CompACT	23.9/11.7	47.4/34.4	20.5/10.8	21.0/21.1	829.9/2571.2	776.2/2463.8	6276.5/25352.8	36738.3/145257.5
CEM+CompACT	8.1/4.5	44.2/33.2	3.8/2.6	40.9/34.5	73.7/**198.1**	**88.3**/**267.5**	3236.0/**9047.6**	60393.3/200703.1
DCT+R-CNN	23.2/8.5	45.8/36.5	18.5/6.5	18.1/27.1	183.3/541.8	176.3/532.4	8976.3/24204.6	36484.9/180873.7
IHTLS+CompACT	20.8/8.7	46.5/34.2	20.2/10.7	21.6/21.1	178.0/774.0	735.8/2835.9	10484.0/42814.2	37172.1/145188.5
H^2T+CompACT	21.8/10.1	44.0/33.6	21.7/11.5	21.7/20.3	162.9/687.8	191.7/922.2	10278.4/41193.8	36115.2/139703.2
CMOT+CompACT	22.5/10.3	45.9/33.4	23.3/12.6	20.0/19.7	**40.7**/243.2	254.1/1255.9	11424.4/45619.6	34134.9/134568.6
HGFT(T.1)+CompACT	−/12.1	−/33.5	−/10.4	−/21.5	−/1927.5	−/2141.0	−/24160.0	−/145262.2
GM-PHD(T.2)+EB(D.2)*	−/14.4	−/26.5	−/12.3%	−/18.8%	−/994.3	−/1660.4	−/19627.3	−/139807.3
GM-PHD(T.2)+CompACT	21.8/10.9	**47.6**/35.0	16.2%/**15.1**%	20.4%/21.6%	641.8/556.4	2038.5/1674.9	37963.0/29687.1	186043.9/147257.0
CCM(T.3)+R-CNN/CompACT	25.2/10.7	45.8/33.8	23.8%/11.9%	**15.8**%/20.0%	179.9/514.7	590.6/1705.5	10155.4/35624.7	**32742.9**/142110.0
IOUT(T.4)+EB(D.2)*	**34.0**/**16.4**	37.8/26.7	**27.9**%/14.8%	20.4%/**18.2**%	573.6/1743.2	603.7/1846.3	**1617.0**/12627.0	33760.8/136077.8
IOUT(T.4)+R-CNN	29.3/11.8	47.2/**36.5**	25.0%/8.9%	17.3%/25.0%	1112.5/3693.1	1261.0/4228.3	3457.6/16634.7	33394.1/168527.2
JTEGCTD(T.5)+CompACT	28.4/14.2	47.1/34.4	23.1%/13.5%	18.3%/18.7%	69.4/415.3	260.6/1345.7	5034.0/26221.8	33093.8/**133867.4**
MTT(T.6)+CompACT	−/12.0	−/35.7	−/7.7%	−/23.2%	−/814.7	−/3158.9	−/14016.8	−/156997.0
GMPHD-KCF(T.7)+EB(D.2)*	−/14.1	−/25.9	−/12.5%	−/18.5%	−/909.9	−/1437.2	−/21863.7	−/139245.4
GMPHD-KCF(T.7)+CompACT	−/12.0	−/33.8	−/10.8%	−/19.5%	−/648.8	−/1300.2	−/30518.1	−/140669.4

3.2. Multi-Vehicle Tracking

On the other hand, we have received 8 entries of trackers from various authors in the IWT4S 2017 challenge. 5 participants performed both the beginner and experienced experiments, and the rest only performed the experienced experiment. The UA-DETRAC committee additionally performed both experiments with 6 baseline trackers with the best detection input. Thus a total of 14 trackers were presented in the challenge. All submitted algorithms use the provided R-CNN and CompACT or EB (D.2) detection as the input for their trackers. They focus on modeling appearance similarity (*e.g.*, MTT (T.6) and GMPHD-KCF (T.7)) and motion relation (*e.g.*, HGFT (T.1), GM-PHD (T.2), C-CM (T.3), IOUT (T.4) and JTEGCTD (T.5)) among detections in consecutive frames.

As shown in Table 3, all multi-object tracking systems perform disappointedly. Given the provided detections, the PR-MOTA scores of the best JTEGCTD (T.5)+CompACT combination are 28.4% and 14.2% (Perfect = 100%) in beginner and experienced level respectively. Although given the more accurate EB (D.2) detection input, the PR-MOTA score of the best IOUT (T.4)+EB (D.2) combination is only 16.4% in experienced level. Compared to other more sophisticated methods, IOUT only calculates the overlap intersection-over-union between detections without using image information, resulting in very low overall complexity. However, this method is not universally valid but depends on more accurate detection input (*e.g.*, EB (D.2)). Then, different from pedestrian datasets, the fixed-camera based vehicle dataset makes the size and aspect ratios of objects undergo not great changes in a few consecutive frames.

It can easily be seen that more accurate detection input leads to more accurate tracking performance (*e.g.*, IOUT (T.4) and GMPHD-KCF (T.7)). Besides, the correlation filters based visual tracker can further facilitate the tracking accuracy (see GM-PHD (T.2) and GMPHD-KCF (T.7) with CompACT). In Table 4, we report the average tracking speed and the corresponding platform configuration of all the evaluated object tracking algorithms.

4. Discussions

This paper concludes with summarized results and findings. All evaluations are conducted on the large scale UA-DETRAC benchmark, which is annotated per-frame with different attributes in complex traffic and street surveillance scenarios. Besides, the DETRAC-MOT toolkit is developed to evaluate either detection or tracking methods conveniently. In this challenge, a total of 7 new detectors and 8 new trackers are submitted from participants. For the beginner level, the best submitted detector and tracker of the IWT4S 2017 challenge are RCNN-SC (D.4) and IOUT (T.4) respectively. For the experienced level, the best detector and tracker are GP-FRCNN (D.1) and JTEGCTD (T.5) respectively. Other important cues such vehicles types and correlation filters can increase the performance.

The main goal of the IWT4S challenge is to establish a community-based platform for discussion of both detection and tracking performance evaluation for traffic and street surveillance. We hope it will contribute to the community with verified annotated benchmarks, evaluation protocols and evaluation toolkits. There are several important future works we would like to further investigate. Firstly, we will improve the evaluation toolkit for cross-platform integration, making it easier for different users. Secondly, we plan to enrich the current UA-DETRAC benchmark to include more sequences in more scenes and richer annotations of attributes. Thirdly, we would like to launch other challenges focused to narrow application domains, such as object recognition and behavior analysis.

Acknowledgments. This work is partly supported by the National Science Foundation under Grant No. CCF-1319800, and the Nvidia Corporation.

Table 4. **Tracking speed:** *Average running speed (in FPS) of the object tracking algorithms on the UA-DETRAC-test benchmark. "–" indicates the data is not avaliable, and "×" indicates the GPU is not used.*

Trackers	Codes	CPU	RAM	Frequency	GPU	Speed
HGFT	Matlab	-	32GB	-	×	3.97
GM-PHD	C++	Intel i7-4770	16GB	3.50GHz	×	947.24
CCM	Matlab	Intel i7-4720	16GB	2.60GHz	×	471.28
IOUT	Python	Intel i7-6700	32GB	3.40GHz	×	**6902.07**
JTEGCTD	Matlab	Intel i7-3720QM	8GB	2.70GHz	×	60.38
MTT	Python, C++	Intel E5-2650	128GB	2.00GHz	Titan X	24.30
GMPHD-KCF	C++	Intel i7-4770	32GB	3.50GHz	×	24.60
CEM	Matlab	Intel i7-3520M	16GB	2.90GHz	×	4.62
GOG	Matlab	Intel i7-3520M	16GB	2.90GHz	×	389.51
DCT	Matlab,C++	Intel i7-3520M	16GB	2.90GHz	×	0.71
IHTLS	Matlab	Intel i7-3520M	16GB	2.90GHz	×	19.79
H^2T	C++	Intel i7-3520M	16GB	2.90GHz	×	3.02
CMOT	Matlab	Intel i7-3520M	16GB	2.90GHz	×	3.79

Appendix: List of Challenge Submissions

All participant submission methods for detection and tracking are listed within chronological order in which they are received, and prefixed by **D** and **T**, respectively.

- **D.1. Geometric proposals for Faster R-CNN (GP-FRCNN).** *Sikandar Amin, Fabio Galasso.* {*s.amin, f.galasso*}*@osram.com*

  GP-FRCNN is an extension of the Faster R-CNN detector [31]. Faster R-CNN based methods are known to be limited in two aspects: (1) it is sensitive to background clutter; (2) performance decreases as the problem scales up. To address these issues, the proposed GP-FRCNN method re-ranks the generic object proposals with an approximate geometric estimate of the scene. However, this simple extension requires scale adjustments (*e.g.*, anchors, layer resolution) involved in the implementation details. GP-FRCNN performs equally well on smaller and larger objects, and does not require an explicit geometric formulation.

- **D.2. Evolving boxes for fast vehicle detection (EB).** *Li Wang, Yao Lu, Hong Wang, Yingbin Zheng, Hao Ye, Xiangyang Xue. wangli16@fudan.edu.cn, luyao@cs.washington.edu,* {*wang_hong, zhengyb, yeh*}*@sari.ac.cn, xyxue@fudan.edu.cn*

  EB is a deep learning framework that proposes and refines the object boxes under different feature representations. It is embedded with a light-weight proposal network to generate initial anchor boxes as well as to early discard unlikely regions. A fine-turning network produces detailed features for the candidate boxes. By applying different feature fusion techniques, the initial boxes can be refined for both localization and recognition. See [32] for more details.

- **D.3. Lightweight SSD based on ResNet10 with dilations (SSDR).** *Dmitriy Anisimov, Tatiana Khanova.* {*dmitry.anisimov, tatiana.khanova*}*@intel.com*

  SDDR focuses on design of fast object detection model based on ResNet10 [21], which still retains high quality. Besides, the dilation module [6] is extended to subsampling all layers in feature extraction network. After removing pooling operations, channels in convolution layers are sampled additionally to preserve small number of computations. To deal with high computation demand of neural networks, pruning can be done on per-channel basis by eliminating less important channels in convolution filters [26].

- **D.4. R-CNN with Sub-Classes (RCNN-SC).** *Nattachai Watcharapinchai, Sitapa Rujikietgumjorn.* {*sitapa.rujikietgumjorn, nattachai.watcharapinchai*}*@nectec.or.th*

  RCNN-SC is based on the Faster R-CNN architecture [31] with residual network. Instead of training from scratch, the COCO pre-trained model parameters [23] are used for the initialization of the models. The residual network is then fine-tuned on the UA-DETRAC dataset. Instead of using a single object class (*i.e.*, the vehicle) to train on the R-CNN, multiple sub-classes of vehicles (*i.e.*, car, van, bus, and others) are used, such that the R-CNN can better learn respective features of each vehicle class.

- **D.5. Faster R-CNN with ResNet101 (FRCNN-Res).** *Nenghui Song, Yi Wei, Ming-Ching Chang.* {*njsong, ywei2, mchang2*}*@albany.edu*

  FRCNN-Res employs the Faster R-CNN architecture in [31]. Specifically, the ResNet-101 model is used to fine-tune on the UA-DETRAC train set. More details can be found in [23].

- **D.6. Region-based Deformable Fully Convolutional Network (DFCN).** *Shuo Wang, Koray Ozcan.* {*shuowang, koray6*}*@iastate.edu*

  DFCN is the combination of Region-based Fully Convolutional Networks (R-FCN) [7] and Deformable Convolutional Networks (Deformable ConvNets) [8] using resnet-101 as the backbone. The model is fine-tuned based on the contents of UA-DETRAC dataset. The source code[3] of Microsoft Research Asia (MSRA) is modified and used to train a model suitable for UA-DETRAC dataset.

- **D.7. CERTH Single Shot multibox Detector (SSD) for vehicle detection (CERTH-SSD).** *Konstantinos Avgerinakis, Panagiotis Giannakeris, Alexia Briassouli, Ioannis Kompatsiaris.* {*koafgeri, giannakeris, abria, ikom*}*@iti.gr*

  CERTH-SSD uses a modification of SSD [27]. Specifically, the SSD Inception V2 architecture that is trained

[3] https://github.com/msracver/Deformable-ConvNets

on the COCO dataset is adopted. It uses a single feed-forward convolutional network to directly predict classes and bounding boxes of objects without requiring a second stage per-proposal classification operation. *Argmax* is used for classification purposes and *SmoothL1* to compute location loss function.

- **T.1. Higher-order Graph and Flow network based Tracker (HGFT).** *Xiaoyi Yu, Guang Han.* {*1215012310, hanguang8848*}*@njupt.edu.cn*

  HGFT is a variant of the min-cost flow network based tracker [30]. Different from the original algorithm, this method considers high-order temporal relations among detections in the confidence calculation, which results in better tracking performance. Specifically, the relations are calculated based on the overlap and height ratio among detections in several consecutive frames.

- **T.2. Gaussian Mixture Probability Hypothesis Density Filter (GM-PHD).** *Tino Kutschbach, Volker Eiselein, Thomas Sikora.* {*kutschbach, eiselein*}*@nue.tu-berlin.de*

  GM-PHD [14] is a new tree-based path extraction algorithm for a Gaussian Mixture Probability Hypothesis Density (PHD) filter in multi-object tracking. Specifically, the PHD filter is a multi-object Bayes filter with the advantage of linear complexity and ability to deal with high clutter especially in radar/sonar scenarios. Specifically, an eight-dimensional state-space is used, where a state is encoded by the position of the upper-left corner and the height/width of its bounding box in the image, plus their temporal derivatives. See [14] for further details.

- **T.3. Online distance based and offline appearance based tracker with correlated color dissimilarity matrix (CCM).** *Noor M. Al-Shakarji, Filiz Bunyak, Kannappan Palaniappan.* {*nmahyd, punyak, palaniappank*}*@mail.missouri.edu*

  CCM employs a process in managing the birth, death and temporary lose of object during visual tracking. Kalman filter is used to predict the location of targets on following frames. For local assignment, spatial distance is used to assign object using Hungarian algorithm [29]. Refinement process based on spatial distance and reliable appearance model is used for solving the global assignment. This method can filter out noisy detections from objects that are reliably detected.

- **T.4. Intersection-over-union tracker (IOUT).** *Erik Bochinski, Volker Eiselein, Thomas Sikora.* {*bochinski, eiselein, sikora*}*@nue.tu-berlin.de*

  IOUT is a simple tracker which can essentially maintain and expand a track by selecting the detection with the highest intersection-over-union to the last detection in the existing frame (if a certain threshold is met). The remaining detections (not assigned to an existing track) will start a new track. All tracks with no detection updates will end eventually. The performance can be further improved by filtering out all tracks with small length and/or no low-confidence detections.

- **T.5. Joint tracking with event grouping and constraints in time domain (JTEGCTD).** *Wei Tian, Martin Lauer.* {*fwei.tian, martin.lauerg*}*@kit.edu*

  JTEGCTD consists of two major components: the grouping approach for prediction events and track stitching by constraints in time domain. The first one is employed to reduce the drift of targets which is caused by mismatched objects in crowded scenes. The key idea is based on the motion relationship within multiple objects. The second step attempts to rediscover the target after long disappearance by solving subgraph models.

- **T.6. Multi-task Deep Learning for Fast Online Multiple Object Tracking (MTT)** *Yuqi Zhang, Yongzhen Huang, Liang Wang.* {*yuqi.zhang,yzhuang,wangliang*}*@nlpr.ia.ac.cn*

  MTT improves the original method in [35] by using the appearance feature extractor trained by triplet loss instead of cross entropy loss. In addition, the detection quality is judged first before performing tracking. The detection quality is simply defined as the intersection-over-union between the ground truth. If the intersection-over-union is low, it is labeled as bad quality. A two-class classification network is trained. The data association part only deals with detections with good quality. The two parts formulate a multi-task network and the computation of the convolutional layers is shared in the two tasks.

- **T.7. Gaussian Mixture Probability Hypothesis Density Filter extended by Kernelized Correlation Filters (GMPHD-KCF).** *Tino Kutschbach, Volker Eiselein, Thomas Sikora.* {*kutschbach, eiselein*}*@nue.tu-berlin.de*

  GMPHD-KCF is based on the work published in [14]. Furthermore, it has been extended by visual correlation filters (namely Kernelized Correlation Filters) presented in [22]. Specifically, the visual tracking scheme is similar to [10] and uses two separate models for estimating target translation and scale. This is possible because in most visual tracking applications, the change in target position between two frames is much larger than the change in scale. Therefore, translation estimation and scale estimation can be done subsequently. In this work, FHOG-features [17] from the computer

vision toolbox[4] are used. For translation estimation, a KCF with Gaussian kernel, and for scale estimation a filter with linear kernel is used.

- **T.8. CERTH Kernelized Correlation Filters (KCF) tracking algorithm for vehicle tracking (CERTH-KCF).** *Konstantinos Avgerinakis, Panagiotis Giannakeris, Alexia Briassouli, Ioannis Kompatsiaris.* {*koafgeri, giannakeris, abria, ikom*}*@iti.gr*

  CERTH-KCF is based on the KCF tracking algorithm [22] in order to localize the bounding boxes throughout sequential frames. The given detections are compensated by allowing creation of a new trajectory only after an intersection-over-union score check is performed with other already existing localized boxes in the frame and falls below a certain value. When a detection is missed, only the coordinates are updated to relocalize the bounding box, while when the algorithm could not localize the initial given vehicle id for 3 sequential video frames the vehicle was presumed to have traveled off the frame and it was ultimately deleted from the database. To handle with the case that a correctly tracked vehicle passes in front of another confusing the tracker, the trajectories are merged at the current frame and assign the oldest vehicle id to the resulting trajectory.

References

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR*, pages 1265–1272, 2011.

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, pages 1926–1933, 2012.

[3] S. H. Bae and K. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR*, pages 1218–1225, 2014.

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image and Video Processing*, 2008, 2008.

[5] Z. Cai, M. J. Saberian, and N. Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *ICCV*, pages 3361–3369, 2015.

[6] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[7] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016.

[8] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[11] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *ICCV*, pages 2304–2311, 2013.

[4] http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html

[12] P. Dollár, R. Appel, S. J. Belongie, and P. Perona. Fast feature pyramids for object detection. *TPAMI*, 36(8):1532–1545, 2014.

[13] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *TPAMI*, 34(4):743–761, 2012.

[14] V. Eiselein, D. Arp, M. Pätzold, and T. Sikora. Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors. In *AVSS*, pages 325–330, 2012.

[15] A. Ess, B. Leibe, and L. J. V. Gool. Depth and appearance for mobile scene analysis. In *ICCV*, pages 1–8, 2007.

[16] M. Everingham, S. M. A. Eslami, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[17] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[18] J. M. Ferryman and A. Shahrokni. PETS2009: dataset and challenge. In *AVSS*, pages 1–6, 2009.

[19] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[20] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[22] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015.

[23] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[24] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *CVPR*, pages 1037–1045, 2015.

[25] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *coRR*, abs/1504.01942, 2015.

[26] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *CoRR*, abs/1608.08710, 2016.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[28] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *coRR*, abs/1603.00831, 2016.

[29] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[30] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208, 2011.

[31] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[32] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue. Evolving boxes for fast vehicle detection. In *ICME*, 2017.

[33] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. UA-DETRAC: A new benchmark and protocol for multi-object tracking. *CoRR*, abs/1511.04136, 2015.

[34] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *CVPR*, pages 1282–1289, 2014.

[35] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. *CoRR*, abs/1703.07402, 2017.