Title
UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking

Authors
Wen, Longyin
Du, Dawei
Cai, Zhaowei
et al.

Publication Date
2020-04-01

DOI
10.1016/j.cviu.2020.102907

Peer reviewed

Computer Vision and Image Understanding
journal homepage: www.elsevier.com

UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking

Longyin Wen[a], Dawei Du[b], Zhaowei Cai[c], Zhen Lei[d], Ming-Ching Chang[b], Honggang Qi[e], Jongwoo Lim[f], Ming-Hsuan Yang[g], Siwei Lyu[b,**]

[a] *JD Finance America Corporation, Mountain View, CA, USA*
[b] *Computer Science Department, University at Albany, State University of New York, Albany, NY, USA*
[c] *Department of Electrical and Computer Engineering, University of California, San Diego, CA, USA*
[d] *National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[e] *School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China*
[f] *Division of Computer Science and Engineering, Hanyang University, Seoul, Korea*
[g] *School of Engineering, University of California at Merced, CA, USA*

ABSTRACT

Effective multi-object tracking (MOT) methods have been developed in recent years for a wide range of applications including visual surveillance and behavior understanding. Existing performance evaluations of MOT methods usually separate the tracking step from the detection step by using one single predefined setting of object detection for comparisons. In this work, we propose a new University at Albany DEtection and TRACking (UA-DETRAC) dataset for comprehensive performance evaluation of MOT systems especially on detectors. The UA-DETRAC benchmark dataset consists of 100 challenging videos captured from real-world traffic scenes (over 140,000 frames with rich annotations, including illumination, vehicle type, occlusion, truncation ratio, and vehicle bounding boxes) for multi-object detection and tracking. We evaluate complete MOT systems constructed from combinations of state-of-the-art object detection and tracking methods. Our analysis shows the complex effects of detection accuracy on MOT system performance. Based on these observations, we propose effective and informative evaluation metrics for MOT systems that consider the effect of object detection for comprehensive performance analysis.

1. Introduction

Multiple object tracking (MOT), which aims to extract trajectories of numerous moving objects in an image sequence, is a crucial task in video understanding. A robust and reliable MOT system is the basis for a wide range of applications including video surveillance, autonomous driving, and sports video analysis. To construct an automatic tracking system, most effective MOT approaches, *e.g.*, (Khan et al., 2005; Zhang et al., 2008; Benfold and Reid, 2011; Breitenstein et al., 2011; Izadinia et al., 2012; Yang and Nevatia, 2012; Huang et al., 2013; Yang et al., 2014; Wen et al., 2014; Dehghan et al., 2015), require a pre-trained detector, *e.g.*, (Felzenszwalb et al., 2010; Dollár et al., 2014; Girshick et al., 2014; Yan et al., 2014; Cai et al., 2015; Redmon et al., 2016) to discover the target objects in the video frames (usually with bounding boxes). As such, a general MOT system entails an object detection step to find target locations in each video frame, and an object tracking step that generates target trajectories across video frames[1].

**Corresponding author: Tel.: +1-518-437-4938.
e-mail: slyu@albany.edu (Siwei Lyu)

[1] In this work, we use the explicit definition of MOT system, *i.e.*, *MOT system = detection + tracking*.

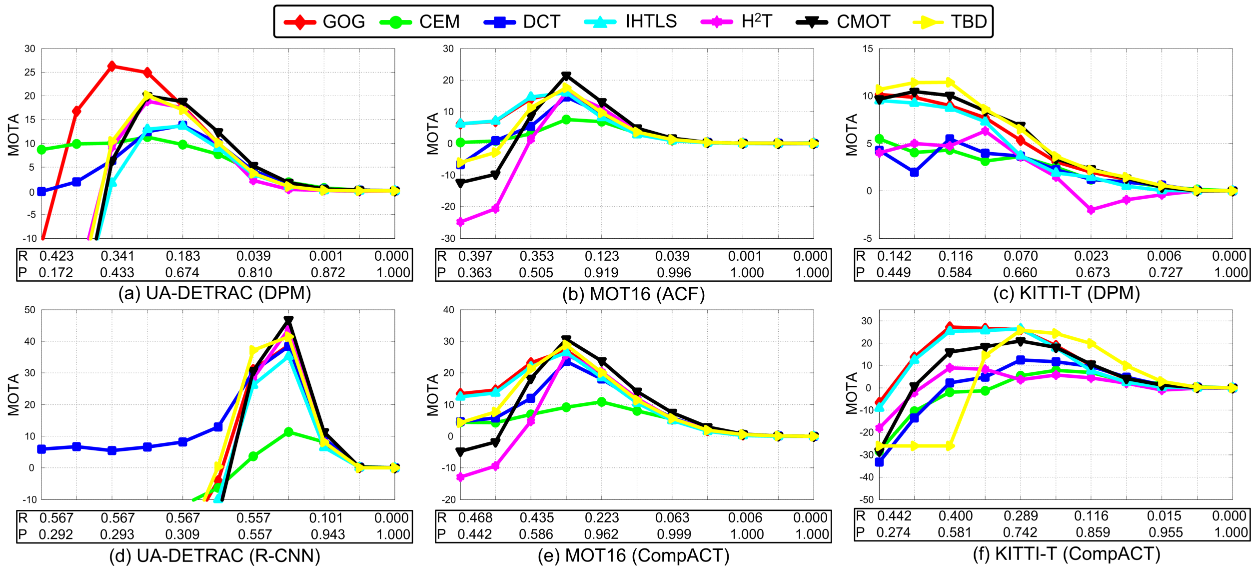

Fig. 1: Precision-recall curves corresponding to the MOT systems on the UA-DETRAC, MOT16 (Milan et al., 2016), and KITTI-T (Geiger et al., 2012) datasets, constructed by four object detection algorithms: DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), and CompACT (Cai et al., 2015) with seven object tracking algorithms: GOG (Pirsiavash et al., 2011), CEM (Andriyenko and Schindler, 2011), DCT (Andriyenko et al., 2012), IHTLS (Dicle et al., 2013), H^{2}T (Wen et al., 2014), CMOT (Bae and Yoon, 2014), and TBD (Geiger et al., 2014). The x-axis corresponds to different precision/recall scores of detectors obtained by varying the detection score threshold. The y-axis is the MOTA score in traditional CLEAR MOT metrics (Stiefelhagen et al., 2006) of the MOT system constructed by detection and tracking methods. Note that with different detection score thresholds, the performance differences of different MOT systems vary significantly according to the MOTA scores.

Despite significant advances in recent years, relatively less effort has been made to large scale and comprehensive evaluations of MOT methods, especially for the effect of object detection to MOT performance. Existing MOT evaluation methods usually separate the object detection (*e.g.*, (Everingham et al., 2015; Dollár et al., 2012; Geiger et al., 2012; Russakovsky et al., 2015)) and object tracking steps (*e.g.*, (Ferryman and Shahrokni, 2009; Bashir and Porikli, 2006; Geiger et al., 2012; Milan et al., 2013; Leal-Taixé et al., 2015)) in comparisons. While this evaluation strategy is widely adopted in the literature, it is insufficient for analyzing complete MOT systems (see Figure 1). In particular, it is important to understand the effect of detection accuracy on the complete MOT system performance, which can only be revealed in a comprehensive quantitative study on object detection and tracking steps *jointly*.

In this work, we propose a new large-scale University at Albany DEtection and TRACking (UA-DETRAC) dataset. The UA-DETRAC dataset includes 100 challenging videos with more than $140,000$ frames of real-world traffic scenes. These videos are manually annotated with a total of 1.21 million labeled bounding boxes of vehicles and useful attributes, *e.g.*, illumination of scenes, vehicle type, and occlusion. Different from other self-driving car datasets (*e.g.*, KITTI (Geiger et al., 2012), Berkeley DeepDrive BDD100k (Yu et al., 2018), Baidu Apolloscapes (Huang et al., 2018) and Oxford Robotic Car (Maddern et al., 2017) datasets), the proposed dataset focuses on detecting and tracking vehicles, which is a thoroughly annotated MOT evaluation dataset containing traffic scenes. Moreover, it poses new challenges for object detection and tracking algorithms. Please see Table 1 for a detailed comparison to other benchmark datasets.

We evaluate the complete MOT systems constructed from combinations of ten object tracking schemes ((Andriyenko and Schindler, 2011; Pirsiavash et al., 2011; Andriyenko et al., 2012; Dicle et al., 2013; Wen et al., 2014; Bae and Yoon, 2014; Geiger et al., 2014; Kim et al., 2015; Bochinski et al., 2017; Lyu et al., 2018)) and six object detection methods ((Felzenszwalb et al., 2010; Dollár et al., 2014; Girshick et al., 2014;

Table 1: Summary of existing object detection or tracking datasets. First six columns: the number of training/testing data ($1k = 10^3$) indicating the number of images containing at least one object, the number of object tracks, and the number of unique object bounding boxes. Remaining columns: additional properties of each dataset, *i.e.*, "D": detection task, "T": tracking task, "P": target object is pedestrian, and "C": target object is vehicle.

Dataset	Training Set			Testing Set			Properties						
	Frame	Tracks	Boxes	Frame	Tracks	Boxes	Color	Video	Task	Object	Illumination	Occlusion	Year
INRIA (Dalal and Triggs, 2005)	1.2*k*	-	1.2*k*	741	-	566	✓		D	P			2005
ETH (Ess et al., 2007)	490	-	1.6*k*	1.8*k*	-	9.4*k*	✓	✓	D	P			2007
NICTA (Overett et al., 2008)	-	-	18.7*k*	-	-	6.9k	✓		D	P			2008
TUD-B (Wojek et al., 2009)	1.09*k*	-	1.8*k*	508	-	1.5*k*	✓		D	P			2009
Caltech (Dollár et al., 2012)	67*k*	-	192*k*	65*k*	-	155*k*	✓	✓	D	P		✓	2012
CUHK (Ouyang and Wang, 2012)	-	-	-	1.06*k*	-	-	✓		D	P			2012
KITTI-D (Geiger et al., 2012)	7.48*k*	-	40.6*k*	7.52*k*	-	39.7*k*	✓	✓	D	P,C		✓	2014
KAIST (Hwang et al., 2015)	50.2*k*	-	41.5*k*	45.1*k*	-	44.7*k*	✓	✓	D	P	✓	✓	2015
BU-TIV (Wu et al., 2014)	-	-	-	6556	-	-	✓	✓	T	P,C			2014
MOT15 (Leal-Taixé et al., 2015)	5.5*k*	500	39.9*k*	5.8*k*	721	61*k*	✓	✓	T	P	✓		2015
MOT16 (Milan et al., 2016)	5.3*k*	467	110*k*	5.9*k*	742	182*k*	✓	✓	T	P,C	✓	✓	2016
TUD (Andriluka et al., 2008)	610	-	610	451	31	2.6*k*	✓	✓	D,T	P			2008
PETS2009 (Ferryman and Shahrokni, 2009)	-	-	-	1.5*k*	106	18.5*k*	✓	✓	D,T	P	✓		2009
KITTI-T (Geiger et al., 2012)	8*k*	-	-	11*k*	-	-	✓	✓	T	C		✓	2014
UA-DETRAC	84*k*	5.9*k*	578*k*	56*k*	2.3*k*	632*k*	✓	✓	D,T	C	✓	✓	2015

Cai et al., 2015; Ren et al., 2017; Wang et al., 2017)), on the UA-DETRAC, MOT16 (Milan et al., 2016), and KITTI-T (Geiger et al., 2012) datasets[2]. While existing performance evaluation protocols use a single predefined setting of object detection to compare different object tracking methods, our experimental results (see Figure 1) show that the performance (*e.g.*, relative rankings of different methods) of MOT systems vary significantly using different settings for object detection. For example, as shown in Figure 1(a), the CEM tracker obtains higher MOTA score than the DCT tracker at the precision-recall values $(0.433, 0.341)$, but lower MOTA score at the precision-recall values $(0.674, 0.183)$. Similar results are observed for other trackers in the MOT16 (Milan et al., 2016) and KITTI-T (Geiger et al., 2012) datasets. As such, using a single predefined setting of object detection is not sufficient to reveal the full behavior of the whole MOT systems and can lead to uninformative evaluations and conclusions.

Based on these observations, we propose a new evaluation protocol and metrics for MOT. The proposed UA-DETRAC protocol considers the effect of object detection from the perspective of system evaluation. One recent work (Solera et al., 2015) also addresses the issue of MOT performance evaluation with a single predefined setting of detection results and suggests to use multiple perturbed ground truth annotations as detection inputs for analysis. However, evaluation with perturbed ground truth annotations does not reflect the performance of an object detector in practice. In contrast, our analysis is based on the actual outputs of the state-of-the-art object detectors with full range of precision-recall rates. From this perspective, our analysis and evaluation protocol reflect how a complete MOT system performs in practice. The main contributions of this work are summarized as follows. 1) We present a large scale UA-

[2]Since the testing sets from the MOT16 (Milan et al., 2016) and KITTI-T (Geiger et al., 2012) datasets are not publicly available, the experiments are carried out on the training sets. For the MOT16 (Milan et al., 2016) dataset, we train the pedestrian detectors on the INRIA dataset (Dalal and Triggs, 2005) and the first 4 sequences of the training set of MOT16 (Milan et al., 2016), and train the tracking models on the first 4 sequences of the training set in MOT16. The remaining 3 sequences are used for evaluation. For the KITTI-T (Geiger et al., 2012) dataset, the first 13 sequences are used to train the vehicle detection and tracking models, and the remaining 8 sequences are used for evaluation.

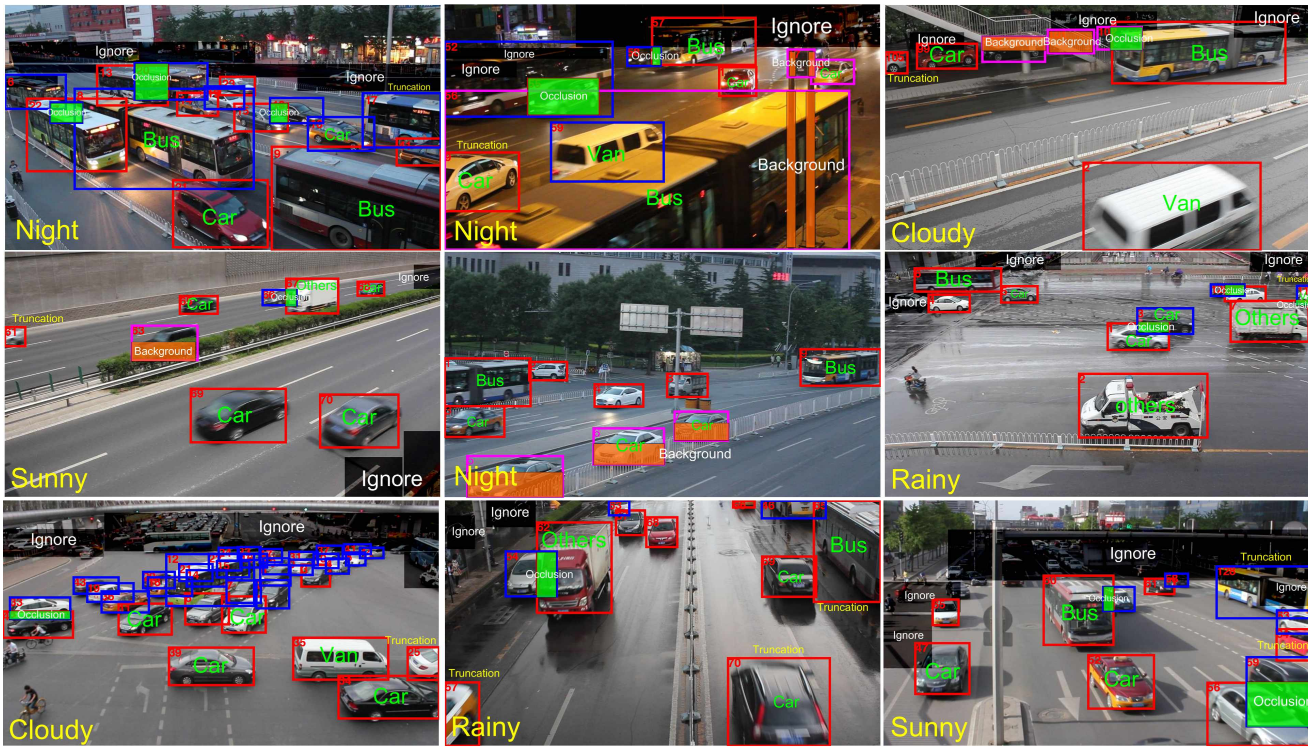

Fig. 2: Sample annotated frames in the UA-DETRAC dataset. Bounding box colors indicate the occlusion level, as fully visible (red), partially occluded by other vehicles (blue), or partially occluded by background (pink). Black opaque areas are background regions that are not used in the benchmark dataset; green opaque areas are regions occluded by other vehicles; and orange opaque regions are areas occluded by background clutters. The illumination conditions are indicated by the texts in the bottom left corner of each frame.

DETRAC dataset for both vehicle detection and MOT evaluation, which differs from existing databases significantly in terms of data volume, annotation quality, and difficulty (see Table 1). 2) We propose a new protocol and evaluation metrics for MOT by taking the effect of object detection module into account. 3) Based on the UA-DETRAC dataset and evaluation protocol, we thoroughly evaluate complete MOT systems by combining the state-of-the-art detection and tracking algorithms, and analyze the conditions under which the existing object detection and tracking methods may fail.

2. UA-DETRAC Benchmark Dataset

The UA-DETRAC dataset consists of 100 videos, selected from over 10 hours of image sequences acquired by a Canon EOS 550D camera at 24 different locations, which represent various traffic patterns and conditions including urban highway, traffic crossings and T-junctions. Notably, to ensure the diversity, we capture the data at different locations with various illumination conditions and shooting angles. The videos are recorded at 25 frames per seconds (fps) with the JPEG image resolution of 960×540 pixels. A website[3] is constructed for performance evaluation of both detection and tracking methods on the UA-DETRAC dataset using a submission protocol similar to that of the MOT15 benchmark dataset (Leal-Taixé et al., 2015).

2.1. *Data Collection and Annotation*

Video Annotation. More than $140,000$ frames in the UA-DETRAC dataset are annotated with $8,250$ vehicles, and a total of 1.21 million bounding boxes of vehicles are labeled. We ask over 10 domain experts to annotate the collected data for more than two months. We carry out several rounds of cross-check to ensure high quality annotations. Similar to PASCAL VOC (Everingham et al., 2015), there are some regions discarded in each frame, which cover vehicles that cannot be annotated due

[3] `http://detrac-db.rit.albany.edu.`

to low resolution. Figure 2 shows sample frames with annotated attributes in the UA-DETRAC dataset.

The UA-DETRAC dataset is divided into training (UA-DETRAC-train) and testing (UA-DETRAC-test) sets, with 60 and 40 sequences, respectively. We select training videos that are taken at different locations from the testing videos, but ensure the training and testing videos have similar traffic conditions and attributes. This setting reduces the chances of detection or tracking methods to overfit to particular scenarios. All the evaluated detection and tracking algorithms are trained on the UA-DETRAC-train set and evaluated on the UA-DETRAC-test set.

The UA-DETRAC dataset contains videos with large variations in scale, pose, illumination, occlusion and background clutters. For evaluation on object detection, similar to the KITTI detection (Geiger et al., 2012) and WIDER FACE (Yang et al., 2016) datasets, we define three levels of difficulties in the UA-DETRAC-test set, *i.e.*, *easy* (10 sequences), *medium* (20 sequences), and *hard* (10 sequences) based on the recall rate of the EdgeBox method (Zitnick and Dollár, 2014). Figure 4 shows the distribution of the UA-DETRAC-test set in terms of detection difficulty. The average recall rates of these three levels are 97.0%, 85.0%, and 64.0%, respectively, with $5,000$ proposals per frame.

For MOT evaluation, we also define three levels of difficulties among the 40 testing sequences, *i.e.*, *easy* (10 sequences), *medium* (20 sequences), and *hard* (10 sequences) based on the average PR-MOTA score (defined in Section 3.2) of the MOT systems constructed from combinations of six representative object tracking methods (*i.e.*, GOG (Pirsiavash et al., 2011), CEM (Andriyenko and Schindler, 2011), DCT (Andriyenko et al., 2012), IHTLS (Dicle et al., 2013), H^2T (Wen et al., 2014), and CMOT (Bae and Yoon, 2014)) and four representative object detection methods (*i.e.*, DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), and CompACT (Cai et al., 2015)). Figure 5 shows the distribution of the UA-DETRAC-test set in terms of tracking difficulty.

To analyze the performance of object detection and tracking algorithms thoroughly, we annotate sequences with several attributes:

- **Vehicle type.** We annotate four types of vehicles as *car*, *bus*, *van*, and *others* (including other vehicle types such as trucks and tankers)[4]. The distribution of vehicle type is shown in Figure 3(a).
- **Illumination.** We consider four categories of illumination conditions, *i.e.*, *cloudy*, *night*, *sunny*, and *rainy*. The distribution of video sequences based on illumination attribute is presented in Figure 3(b).
- **Scale.** We define the scale of the annotated vehicle bounding boxes as the square root of the area in pixels. The distribution of vehicle scale in the dataset is presented in Figure 3(c). We label vehicles with three scales: *small scale* (0-50 pixels), *medium scale* (50-150 pixels), and *large scale* (more than 150 pixels).
- **Occlusion ratio.** We use the fraction of vehicle bounding box being occluded to define the occlusion ratio. We annotate the occlusion relations between vehicle bounding boxes and compute the occlusion ratio. Vehicles are annotated with three categories: *no occlusion*, *partial occlusion*, and *heavy occlusion*. Specifically, a vehicle is considered partially occluded if the occlusion ratio of a vehicle is in the range of 1% to 50%, and heavily occluded if the occlusion ratio is larger than 50%. The distribution of occluded vehicles in videos is shown in Figure 3(d).
- **Truncation ratio.** The truncation ratio indicates the degree of vehicle parts appears outside a frame[5]. If a vehi-

[4]Some vehicles in our dataset are rarely occurring special vehicles, and the number of them are limited. To facilitate model training, we combine them together to form the “other” category.

[5]Note that it is difficult to annotate the outside region of objects accurately. We thus approximately estimate the outside regions of objects by referring the last complete bounding boxes of objects before exiting the scenes. Meanwhile, the truncation ratio is only used to determine the validation of the annotated objects. That is, for both detection and tracking, we only consider the objects with the ratio smaller than 50% in training. Thus, the annotation errors in truncation ratio have little impact on the benchmark.

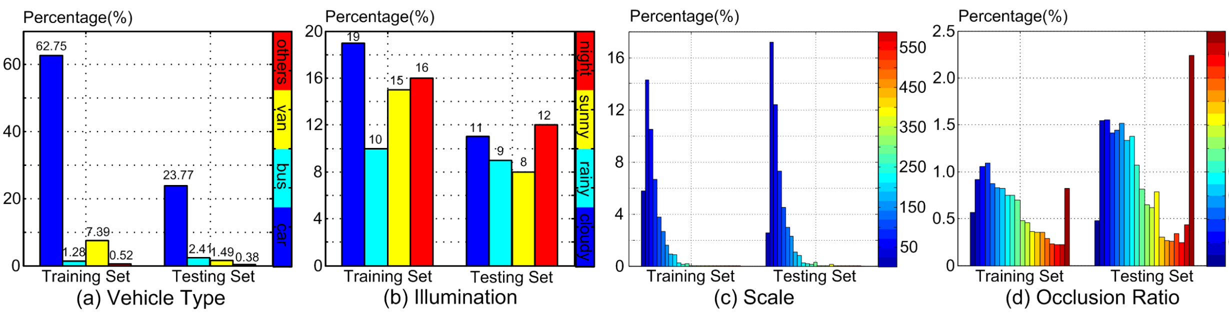

Fig. 3: Attribute statistics of the UA-DETRAC benchmark dataset.

cle is not fully captured within a frame, we annotate the bounding box across the frame boundary and compute the truncation ratio based on the region outside the image. The truncation ratio is used in the training process of the evaluated detection and tracking algorithms where we discard the annotated bounding box when its ratio is larger than 50%.

2.2. *Relevance to Existing Benchmark Datasets*

2.2.1. *Object Detection Datasets*

Numerous benchmark datasets, *e.g.*, PASCAL VOC (Everingham et al., 2015), ImageNet (Russakovsky et al., 2015), Caltech (Dollár et al., 2012), KITTI-D (Geiger et al., 2012), and KAIST (Hwang et al., 2015), have been developed for object detection. These datasets are mainly developed for object detection in single images that can be used to train detectors for MOT systems. Recently, to facilitate the research in autonomous driving field, the Berkeley DeepDrive BDD100k (Yu et al., 2018) contains over $100,000$ videos with annotations of image level tagging, object bounding boxes, drivable areas, lane markings, and full-frame instance segmentation. The Baidu Apolloscapes (Huang et al., 2018) provides high density 3D point cloud map, per-pixel, per-frame semantic image label, lane mark label, and semantic instance segmentation annotations. The Oxford Robotic Car (Maddern et al., 2017) is a autonomous driving dataset with approximate 20 million images with LIDAR, GPS and INS ground-truth in all weather conditions.

2.2.2. *Object Tracking Datasets*

Several multi-object tracking benchmarks have also been collected for evaluating object tracking methods. Some of the most widely used multi-object tracking evaluation datasets include the PETS09 (Ferryman and Shahrokni, 2009), PETS16 (Patino), KITTI (Geiger et al., 2012), MOT15 (Leal-Taixé et al., 2015) and MOT16 (Milan et al., 2016), and UAVDT (Du et al., 2018) datasets. The PETS09 and PETS16 datasets focus on multi-pedestrian detection, tracking as well as counting. The KITTI dataset is designed for object tracking and detection, which are acquired from a moving vehicle with viewpoint of the driver. The MOT15 dataset aims to provide a unified platform and evaluation protocol for object tracking. It includes a dataset of 22 videos mostly from surveillance cameras where the targets of interest are pedestrians. In addition, it also provides a platform where new datasets and multi-object tracking methods can be incorporated in a plug-and-play manner. The MOT16 benchmark dataset is an extension of MOT15 with more challenging sequences and thorough annotations. Recently, the UAVDT dataset is proposed to advance object tracking algorithms applied in drone based scenes.

Compared to existing multi-object tracking datasets, the UA-DETRAC benchmark dataset is designed for vehicle surveillance scenarios with significantly more video frames, annotated bounding boxes and attributes. The vehicles in the videos are acquired at different view angles and frequently occluded. Meanwhile, the UA-DETRAC benchmark is designed for performance evaluation of both object detection and multi-object tracking. Table 1 summarizes the differences between existing and proposed UA-DETRAC benchmarks in various aspects.

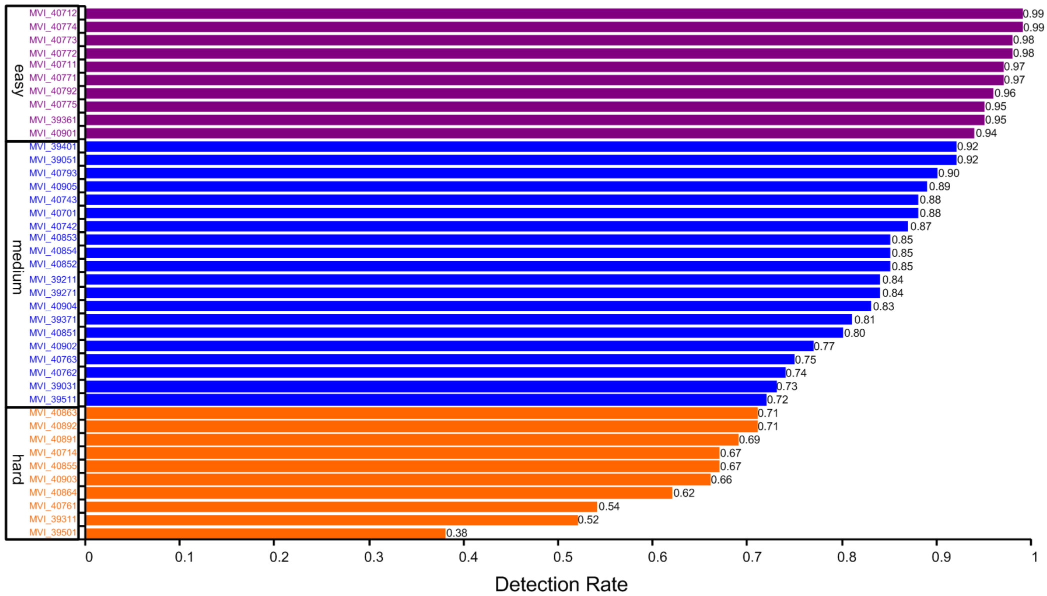

Fig. 4: Recall rates for different sequences in the UA-DETRAC-test set. Each sequence is ranked in the descending order based on the recall rate of the EdgeBox method (Zitnick and Dollár, 2014) with the number of proposal fixed at 5,000. The sequences of three levels of difficulties, *i.e.*, ***easy***, ***medium***, and ***hard*** are denoted in purple, blue, and orange, respectively.

2.3. *Object Detection Algorithms*

We review the state-of-the-art object detection methods in the context of MOT (*e.g.*, humans, faces and vehicles), and describe 12 evaluated algorithms in the UA-DETRAC dataset.

2.3.1. *Review of Object Detection Methods*

Viola and Jones (2004) develop an adaptive boosting algorithm based on a cascade of classifiers and Haar-like feature to detect faces effectively and efficiently. To achieve robust performance, Zhang et al. (2007) propose to use the discriminative multi-block local binary pattern (MB-LBP) features, which capture more image structural information than Haar-like features, to represent face images. Gradient features are important cues for detection. Dalal and Triggs (2005) use the histogram of oriented gradients (HOG) to describe local dominant edge cues for pedestrian detection in single images. In addition, both optical flows and HOG features have been used to detect pedestrians in videos (Dalal et al., 2006). In (Dollár et al., 2014), it has been demonstrated that significant performance gain for pedestrian can be achieved when features from multiple channels are used.

In contrast to hand-crafted features, *e.g.*, Haar (Viola and Jones, 2004), HOG (Dalal and Triggs, 2005), and MB-LBP (Zhang et al., 2007), data-driven hierarchical features (*e.g.*, CNN features (Krizhevsky et al., 2012)) have recently been shown to be effective in numerous vision tasks. The region-based CNN (R-CNN) method (Girshick et al., 2014) combines region proposals with convolutional neural networks (CNNs), and achieves better performance in the PASCAL VOC challenge (Everingham et al., 2015) than systems based on hand-crafted features. Recently, R-CNN are extended (Girshick, 2015; He et al., 2015) to attend to RoIs on feature maps using the RoIPool operation, achieving fast speed and better accuracy. Faster R-CNN (Ren et al., 2017) attempt to learn the attention mechanism with a region proposal network (RPN) to further improve the methods (Girshick, 2015; He et al., 2015). Cai et al. (2015) develop a boosting approach to solve the cascade learning problem of combining features of different complexities, which exploits features of high complexity in the later stages, where only a few difficult candidate patches remain to be classified. As such, the method performs well in terms of accuracy and speed. Redmon et al. (2016) formulate object detection as a regression problem to efficiently predict bounding boxes and associate class probabilities from an image by a single neural

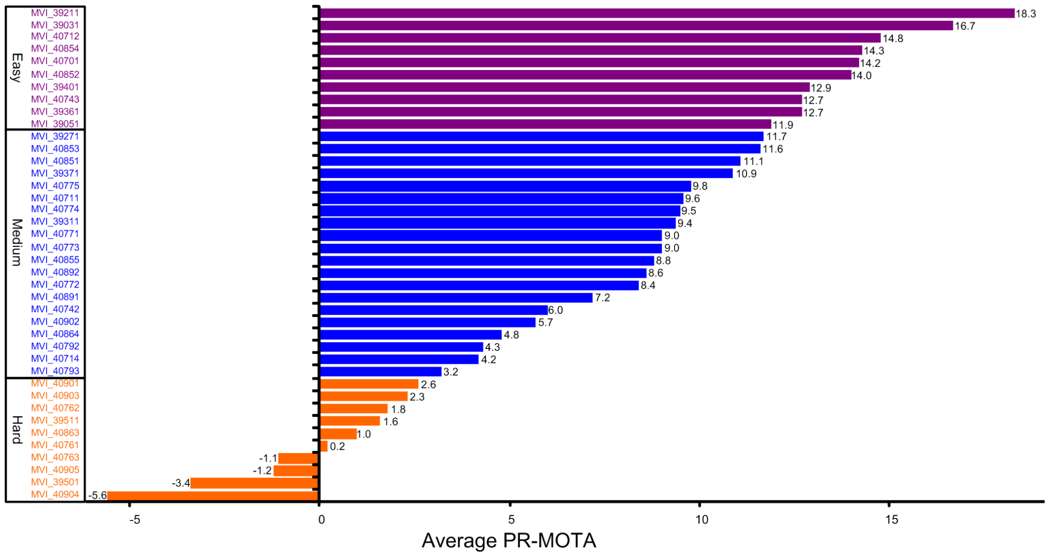

Fig. 5: Average PR-MOTA scores for different sequences in the UA-DETRAC-test set. Each sequence is ranked in descending order based on average PR-MOTA score of the MOT systems constructed from combinations of six representative object tracking methods, *i.e.*, GOG (Pirsiavash et al., 2011), CEM (Andriyenko and Schindler, 2011), DCT (Andriyenko et al., 2012), IHTLS (Dicle et al., 2013), H^2T (Wen et al., 2014), and CMOT (Bae and Yoon, 2014), and four representative object detection methods, *i.e.*, DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), and CompACT (Cai et al., 2015). The sequences of three levels of difficulties, *i.e.*, ***easy***, ***medium***, and ***hard*** are denoted in purple, blue, and orange, respectively.

network in one evaluation. To account for the scale issue in object detection, Cai et al. (2016) present a unified multi-scale CNN algorithm, where object detection is performed at multiple output layers with each focusing on objects within certain scale ranges.

Parts-based representations have been widely used in object detection. Felzenszwalb et al. (2010) present the deformable part-based model (DPM), which describes the part positions as latent variables in the structural SVM framework, for object detection. They extend this strategy and develop a multi-resolution method operating as a deformable parts model and a rigid template to handle large and small objects, respectively. However, the DPM-based methods are computationally expensive for practical applications. To improve the efficiency, Yan et al. (2014) propose an algorithm by constraining the root filter to be low rank, designing a neighborhood-aware cascade to capture the dependence among regions for aggressive pruning, and constructing look-up tables to replace HOG feature extraction with simpler operations.

Considerable efforts have also been made to improve the quality of object proposals for object detection. Instead of scanning through regions sequentially for object detection, Lampert et al. (2008) propose a branch-and-bound scheme to efficiently maximize the classifier function over all possible sliding windows based on bag-of-words image representation (Sivic and Zisserman, 2003). van de Sande et al. (2011) develop a selective search approach using over-segmented regions to generate limited number of possible object locations. Zitnick and Dollár (2014) exploit the number of contours in a bounding box to facilitate generating object proposals. Ren et al. (2017) introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, which enables region proposals efficiently.

2.3.2. Evaluated Object Detectors

We evaluate 12 state-of-the-art object detection algorithms in this work using the UA-DETRAC dataset[6] including the DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), CompACT (Cai et al., 2015), Faster R-CNN (Ren et al., 2017), EB (Wang et al., 2017), YOLOv3 (Redmon and Farhadi, 2018), GP-FRCNN (Amin and Galasso,

[6]We use the original codes for evaluation of object detection.

2017), CSP (Liu et al., 2019), HAT (Wu et al., 2019), FG-BR_Net (Fu et al., 2019) and RD2 (Zhang et al., 2018) methods. Specifically, GP-FRCNN (Amin and Galasso, 2017) and RD2 (Zhang et al., 2018) are the winners in the UA-DETRAC Challenges in 2017 (Lyu et al., 2017) and 2018 (Lyu et al., 2018). We retrain these methods on the UA-DETRAC-train set and evaluate the performance on the UA-DETRAC-test set.

The DPM method is trained using a mixture of 3 star-structured models where each one has 2 latent orientations. The ACF cascade uses $2,048$ decision trees of depth 4. For the CompACT scheme (Cai et al., 2015), we train a cascade of $2,048$ decision trees of depth 4. For the ACF and CompACT methods, the template size is set to 64×64 pixels. To detect vehicles with different aspect ratios, the original images are resized to six different aspect ratios before being scanned by the detectors such that only a single model is needed. A bounding box regression model based on the ACF features is trained for the ACF and CompACT detectors for better performance. For the R-CNN algorithm, we fine-tune AlexNet (Krizhevsky et al., 2012) on the UA-DETRAC-train set. Instead of using the selective search method (Uijlings et al., 2013) to generate proposals[7], the output bounding boxes of the ACF method are warped to 227×227 pixels and then fed into the R-CNN model for classification. For the Faster R-CNN algorithm, we fine-tune the VGG-16 backbone (Simonyan and Zisserman, 2015) on the UA-DETRAC-train set. We use the default 3 scales and 3 aspect ratios to set the anchors, and top-2000 ranked proposals generated by the region proposal network (RPN) are used to train the second stage Fast R-CNN. The positive samples are all annotated vehicles from the UA-DETRAC-train set with less than 50% occlusion and truncation ratios, and the KITTI-D dataset (Geiger et al., 2012) is used for mining hard negatives. The minimum size of the detected object is set to 25×25 pixels for all detectors.

The EB method (Wang et al., 2017) is constructed by the pre-trained VGG-16 network (Simonyan and Zisserman, 2015) on the ImageNet classification dataset. The YOLOv3 (Redmon and Farhadi, 2018) scheme uses the Darknet-53 backbone to perform feature extraction. The GP-FRCNN (Amin and Galasso, 2017) model is an extension of the Faster R-CNN detector (Ren et al., 2017) by re-ranking the generic object proposals with an approximate geometric estimation of the scenes. The CSP (Liu et al., 2019) approach uses the ResNet-50 network as the backbone, which is also pre-trained on the ImageNet dataset (Deng et al., 2009). The HAT (Wu et al., 2019) algorithm is constructed based on the VGG-16 model, and uses the LSTM method for category-specific attention with 128 hidden cells. The FG-BR_Net (Fu et al., 2019) method uses the OMoGMF (Yong et al., 2018) model as the base block of the proposed background subtraction-recurrent neural network. The RD2 (Zhang et al., 2018) scheme is a variant of the RefineDet (Zhang et al., 2018) method with the Squeeze-and-Excitation Network (SENet) (Hu et al., 2018). It averages the results of two detectors with different backbones, *i.e.*, SEResNeXt-50 and ResNet-50.

2.4. *Object Tracking Algorithms*

We briefly review the multi-object tracking algorithms, and then describe ten state-of-the-art object tracking approaches evaluated in this work.

2.4.1. *Review of MOT Methods*

Early multi-object tracking methods formulate the task as the state estimation problem using Kalman (Leven and Lanterman, 2009; Pellegrini et al., 2009) and particle filters (Isard and Blake, 1998; Khan et al., 2005; Mikami et al., 2009; Yang et al., 2014). These methods typically predict the target states in short duration effectively but do not perform well in complex scenarios.

Recently, more effective multi-target tracking algorithms are developed based on the tracking-by-detection framework. Typically, detection results from consecutive frames are linked

[7] Since the selective search method is less effective in generating accurate region proposals of vehicles, we use the outputs of ACF as proposals (Hosang et al., 2015) for the R-CNN method. We need to ensure the proposals generated by ACF with high recall. If a proposal is not generated in the vicinity of an object, it will not be detected by R-CNN.

based on similarities in appearance and motion to form long tracks *e.g.*, joint probabilistic data association filter (JPDAF) (Fortmann et al., 1983) and multiple hypotheses tracking (MHT) (Reid, 1979) methods. The JPDA method (Fortmann et al., 1983) considers all possible matches between the tracked targets and detections in each time frame to compute the joint probabilistic score to complete the tracking task. However, with an increasing number of targets in the sequence, the computational complexity of the method becomes intractable. Rezatofighi et al. (2015) present a computationally tractable approximation to the original JPDA algorithm based on a recent method to find the m-best solutions of an integer linear program. The MHT method (Reid, 1979) builds a tree of potential track hypotheses for each candidate target, and evaluates the likelihoods of the hypothesized matches over several time steps. To further improve MHT in exploiting higher-order information, Kim et al. (2015) train an online appearance model for each track hypothesis. By design, the MHT method is more effective than the JPDAF scheme for long-term association problem at the expense of computational loads.

Several algorithms consider associations of detection/tracklet pairs as an optimization task based on the K-shortest path (Berclaz et al., 2011), maximum weight independent sets (Brendel et al., 2011), maximum multi-clique optimization (Dehghan et al., 2015), tensor power iterations (Shi et al., 2014), network flows (Zhang et al., 2008; Pirsiavash et al., 2011; Leal-Taixé et al., 2014), linear programs (Jiang et al., 2007), Hungarian algorithm (Bae and Yoon, 2014), generalized linear assignment optimization (Dicle et al., 2013), and subgraph decomposition (Tang et al., 2015). To exploit motion information of targets, Wen et al. (2014) formulate the multi-object tracking task as exploring dense structures on a hypergraph, whose nodes are detections and hyper-edges describe the corresponding high-order relations. The run-time bottleneck of (Wen et al., 2014) is addressed in (Wen et al., 2016a) using a RANSAC approach to extract the dense structures on hypergraph efficiently. Andriyenko and Schindler (2011) formulate multi-object tracking as an energy minimization problem by using physical constraints such as target dynamics, mutual exclusion, and track persistence. Yamaguchi et al. (2011) develop an agent-based behavioral model of pedestrians to improve tracking performance, which predicts human behavior based on an energy minimization problem. Andriyenko et al. (2012) tackle multi-object tracking as a discrete-continuous optimization problem that integrates data association and trajectory estimation in an energy function in a way similar to (Delong et al., 2012).

2.4.2. Evaluated Object Trackers

Using the UA-DETRAC dataset, we evaluate performance of MOT systems constructed by different combinations of 6 state-of-the-art object detection algorithms including DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), CompACT (Cai et al., 2015), Faster R-CNN (Ren et al., 2017), and EB (Wang et al., 2017), and 10 object tracking approaches including GOG (Pirsiavash et al., 2011), CEM (Andriyenko and Schindler, 2011), DCT (Andriyenko et al., 2012), IHTLS (Dicle et al., 2013), H^{2}T (Wen et al., 2014), CMOT (Bae and Yoon, 2014), TBD (Geiger et al., 2014), MHT (Kim et al., 2015), IOU (Bochinski et al., 2017) and KIOU (Lyu et al., 2018). All codes of the object detection and tracking algorithms are publicly available or provided by the authors of the corresponding publications. All these methods take object detection results in each frame as inputs and generate target trajectories to complete tracking task. We use the UA-DETRAC-train set to determine the parameters of these methods empirically[8], and the UA-DETRAC-test set for performance evaluation.

3. UA-DETRAC Evaluation Protocol

As discussed in Section 1, existing multi-object tracking evaluation protocols that use a single predefined object detec-

[8]We use the grid search of one parameter over a range of values while keep other parameters fixed. For each setting of parameters of the tracker, we generate a PR-MOTA curve and compute the corresponding PR-MOTA score. Then, we determine the parameters of the tracker corresponding to the maximum PR-MOTA score.

tion setting as input may not reflect the complete MOT performance well. In this section, we introduce the evaluation protocol for object detection and MOT that better reveal complete performance.

3.1. Evaluation Protocol for Object Detection

Evaluation metric. We generate the full *precision vs. recall* (PR) curve for each object detection algorithm. The PR curve is generated by varying the score threshold of an object detector to generate different precision and recall values. Per-frame detector evaluation is performed as in the KITTI-D benchmark (Geiger et al., 2012), where the hit/miss threshold of the overlap between a pair of detected and ground truth bounding boxes is set to 0.7.

Ranking detection methods. The *average precision* (AP) score of the PR curve is used to rank the performance of a detector. We follow the strategy in the PASCAL VOC challenge (Everingham et al., 2015) to compute the AP score, *i.e.*, calculate the average precisions at the fixed 11 recall values from 0 to 1: $\{0, 0.1, 0.2, \cdots, 0.9, 1.0\}$.

3.2. Evaluation Protocol for Object Tracking

Existing evaluation metric. We first introduce a set of performance evaluation metrics for object tracking widely used in the literature including *mostly tracked* (MT), *mostly lost* (ML), *identity switches* (IDS), *fragmentations of target trajectories* (FM), *false positives* (FP), *false negatives* (FN), and two CLEAR MOT metrics (Stiefelhagen et al., 2006), *multi-object tracking accuracy* (MOTA) as well as *multi-object tracking precision* (MOTP). The FP metric describes the number of false alarms by a tracker, and FN is the number of targets missed by any tracked trajectories in each frame. The IDS metric describes the number of times that the matched identity of a tracked trajectory changes, while FM is the number of times that trajectories are disconnected. Both IDS and FM metrics reflect the accuracy of tracked trajectories. The ML and MT metrics measure the percentage of tracked trajectories less than 20% and more than 80% of the time span based on the ground truth respectively. The MOTA metric for all sequences in the benchmark is defined by (Solera et al., 2015),

$$\mathrm{MOTA} = 100 \cdot (1 - \frac{\sum_v \sum_t (\mathrm{FN}_{v,t} + \mathrm{FP}_{v,t} + \mathrm{IDS}_{v,t})}{\sum_v \sum_t \mathrm{GT}_{v,t}})[\%], \quad (1)$$

where $\mathrm{FN}_{v,t}$ is the number of false negatives, and $\mathrm{FP}_{v,t}$ is the number of false positives at time index t of sequence v, with the hit/miss threshold of the bounding box overlap between an output trajectory and the ground truth set to be 0.7. In addition, $\mathrm{IDS}_{v,t}$ is the number identity switches of a trajectory, and $\mathrm{GT}_{v,t}$ is the number of ground truth objects. The MOTP metric is the average dissimilarity between all true positives and the corresponding ground truth targets, as the average overlap between all correctly matched hypotheses and respective objects. We note that the MOTA score is computed by the FN, FP and IDS scores of the tracking results.

Proposed evaluation metric. In this work, we show it is necessary to consider the effect of detection performance on MOT evaluation and introduce the UA-DETRAC metrics[9], *i.e.*, the PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores, to take the effect of object detection into account, based on the basic evaluation metrics. First, we take the basic evaluation metric MOTA as an example to describe the PR-MOTA score. The PR-MOTA curve (see Figure 6) is a three-dimensional curve characterizing the relationship between object detection (precision and recall) and tracking (MOTA). In the following, we describe the steps to generate a PR-MOTA curve and calculate the score:

1. We first vary the detection threshold[10] gradually to generate different object detection results (bounding boxes) cor-

[9] Notably, it is not sufficient to compare two MOT systems based on MOTA scores that generated by two settings of input detection with different FN and FP values. Similar to the case in object detection, it is not meaningful to compare the performance of two detectors based on the different points on the PR curves. Thus, the maximum value on the PR-MOTA curve is not a good choice to compare the performance of the trackers. We expect a tracker achieving better performance as it can generate better performance with any different settings of detections. In this work, we use the average MOTA score over the PR curve of a detector, *i.e.*, PR-MOTA, for comparison.

[10] Specifically, we vary the threshold 10 times with equal interval from the minimal to the maximum detection scores to generate the PR-MOTA curve.

responding to different values of precision p and recall r. The two-dimensional curve corresponding to (p, r) is the *precision-recall* (PR) curve C that delineates the region of possible PR values of a detector.

2. For a set of detection results determined by (p, r), we apply an object tracking algorithm and compute the resulting MOTA score $\Psi(p, r)$. The MOTA scores for (p, r) values on the PR curve form a three-dimensional curve, *i.e.*, the PR-MOTA curve, as shown in Figure 6.

3. From the PR-MOTA curve, we compute the integral score Ω^* to measure MOT performance, *i.e.*, the PR-MOTA score $\Omega^* = \frac{1}{2}\int_C \Psi(p, r)\mathrm{d}s$[11] ($\Omega^*$ is the line integral along the PR curve C). In other words, the PR-MOTA score Ω^* corresponds to the (signed) area of the curved surface formed by the PR-MOTA curve along the PR curve, as shown by the shaded area in Figure 6.

Using the PR-MOTA score, we can compare different multi-object tracking algorithms by considering the effect of detection modules. The scores of other seven metrics, *e.g.*, PR-MOTP and PR-IDS, are similarly computed.

Ranking MOT methods. We rank the performance of MOT methods based on the PR-MOTA scores (larger PR-MOTA score indicates higher rank). If the PR-MOTA scores of two MOT methods are the same, we rank them based on the PR-MOTP scores (larger PR-MOTP score indicates higher rank).

3.3. Comparisons with Existing Evaluation Protocols

It has been shown recently (Milan et al., 2013) that the widely used MOT evaluation metrics including the MOTA, MOTP or IDS scores (Stiefelhagen et al., 2006; Li et al., 2009), do not fully reveal how a MOT system performs. Furthermore, there

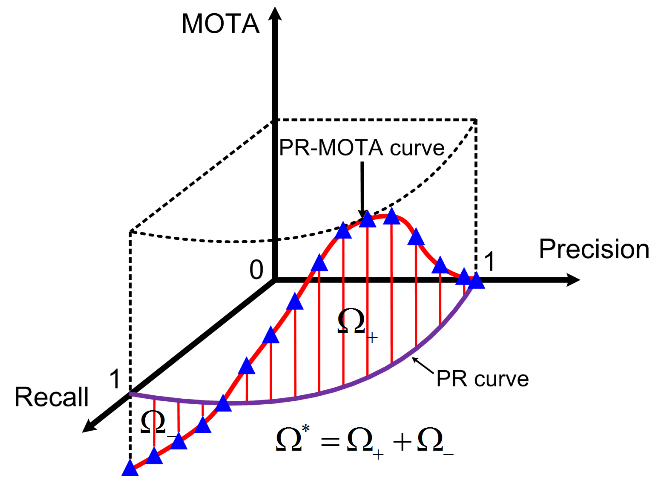

Fig. 6: Proposed UA-DETRAC metric Ω^* of the PR-MOTA curve: the purple curve is the precision-recall curve describing the performance of object detection and the red one is the PR-MOTA curve. The blue triangles represent the sampling points used to generate the PR-MOTA curve.

are several issues associated with the MOT evaluation protocols. Early studies (Zamir et al., 2012; Andriyenko et al., 2012) use different object detection methods for evaluating MOT methods. It is well known that detection results affect the performance of MOT methods significantly. Most recent evaluation methods (*e.g.*, (Huang et al., 2013; Wen et al., 2014; Leal-Taixé et al., 2015; Milan et al., 2016)) adopt a protocol that uses the same predefined setting of detection results to evaluate MOT methods, in order to make the evaluation independent of variations of detection results. It has been shown in (Solera et al., 2015) that the performance of MOT systems cannot be clearly reflected with a predefined setting of detection inputs, and multiple synthetic detections generated by controlled noise are used for comparisons. However, these synthetically generated detection results do not fully correspond to how detectors perform in real world scenarios. In addition, in (Solera et al., 2015), the detections are randomly perturbed independently for each frame, which is different from how real detectors operate. In contrast, the UA-DETRAC protocol considers the complete performance of a detector for MOT evaluation. Using the three-dimensional curve of detection (PR) and tracking scores (*e.g.*, MOTA and MOTP), the UA-DETRAC protocol can better reflect the behavior of the whole MOT systems.

[11] Note that we have $\int_C \Psi(p, r)\mathrm{d}s \in (-\infty, 200\%]$. The proof can be found in the appendix. To convert it to percentage, we multiply it by $\frac{1}{2}$ to ensure the PR-MOTA score is within the range of $(-\infty, 100\%]$. As it is difficult to compute the integration directly, we approximate it with additions over sampled locations over the PR curve C.

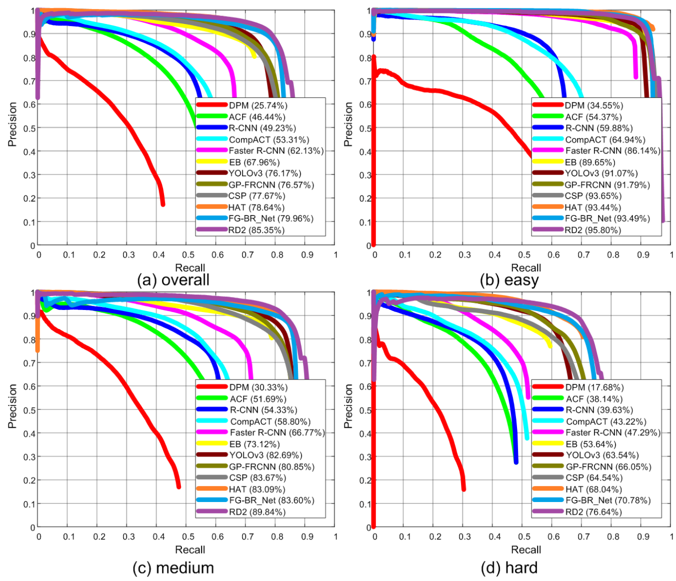

Fig. 7: Precision *vs.* recall curves of the detection algorithms in ***overall/easy/medium/hard* subsets** of UA-DETRAC benchmark dataset. The scores in the legend are the AP scores for evaluating the performance of object detection algorithms.

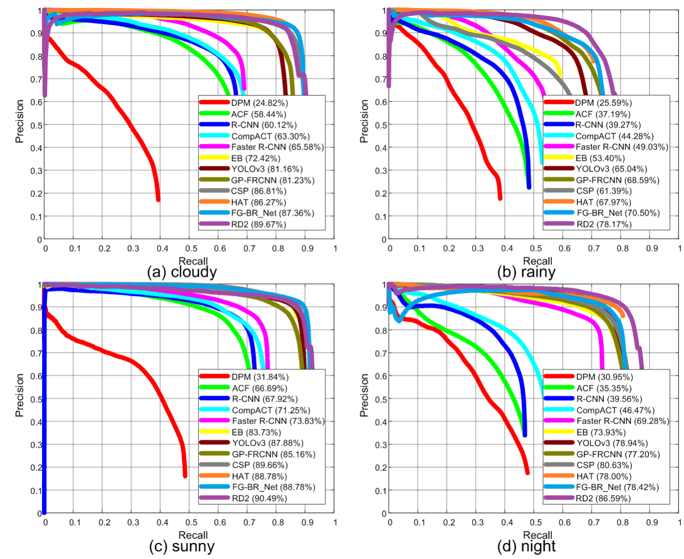

Fig. 8: Precision *vs.* recall curves of the detection algorithms in ***cloudy/rainy/sunny/night* subsets** of UA-DETRAC benchmark dataset. The scores in the legend are the AP scores for evaluating the performance of object detection algorithms.

4. Analysis and Discussion

4.1. Object Detection

Overall performance. The results of five state-of-the-art object detectors on the UA-DETRAC dataset, shown in Figure 7(a) with the PR curves, indicate that there remains much room for improvement for object detection algorithms. Specifically, the DPM and ACF methods do not perform well on vehicle detection with only 25.74% and 46.44% AP scores respectively. The R-CNN algorithm performs slightly better than the ACF method with AP score of 49.23%. The CompACT algorithm achieves more accurate results with 53.31% AP score than the aforementioned methods by learning complexity-aware cascades. The recent proposed detectors achieve more than 60% AP scores, *i.e.*, Faster R-CNN with 62.13% AP score, YOLOv3 with 76.17% AP score, and RD2 with 85.35% AP score. As shown in Figure 7(b)-(d), from the *easy* to *hard* subsets, the AP scores of detectors drop 15 to 20%. For example, the best detector RD2 only achieves 76.64% AP score on the *hard* subset, which demonstrates that more effective detectors are needed for the challenging scenarios in the UA-DETRAC dataset.

Illumination. Most detectors are developed based on the assumption that objects can be spotted in the scenes with poor lighting conditions. Figure 8 shows that all methods achieve AP scores below 80% in the *rainy* and *night* scenes, except for the RD2 algorithm, which achieves 86.59% AP in the *night* scene. In contrast, object detectors perform relatively well in other scenes with better lighting conditions. For example, the RD2 detector achieves 89.67% and 90.49% AP scores in the *cloudy* and *sunny* days, respectively.

Vehicle type. As shown in Figure 9(a)-(d), the detectors perform relatively well only on cars among all kinds of vehicles, *e.g.*, the RD2 method achieves 84.49% AP score. It is worth mentioning that the AP score of the Faster R-CNN method is only 17.07% in terms of the *other* category, better than the other detectors including the RD2 method. The reason can be that the region proposal network (RPN) is effective in reducing the number of candidate object locations and filtering out most background proposals to address the class imbalance issue. Moreover, the poor results can be attributed to two factors. First, it is difficult to handle large variations of scale and aspect ratio for different vehicles. Second, the limited amount of training samples affects the performance of object detectors,

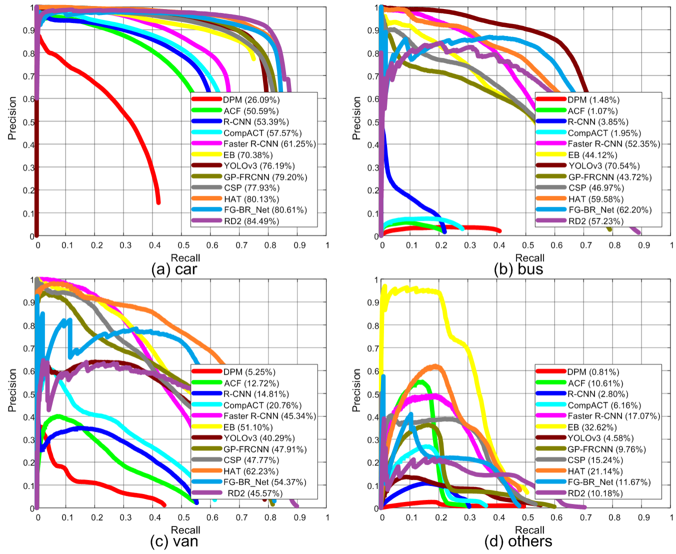

Fig. 9: Precision *vs.* recall curves of the detection algorithms in **car/bus/van/others subsets** of UA-DETRAC benchmark dataset. The scores in the legend are the AP scores for evaluating the performance of object detection algorithms.

i.e., only 0.52% vehicles are in the *others* category in the training set, see Figure 3(a).

Scale. Figure 10 shows the detection results for each scale of vehicles in the UA-DETRAC-test set. For *small* scale vehicles, most detectors achieve over AP scores of 30% except the DPM method, where the RD2 method obtains the best AP score of 66.72%. The CSP approach employs the anchor-free framework to represent objects by points, resulting in inferior AP score of 40.08%. For *medium* scale vehicles, the RD2 method obtains the best AP score of 86.63%, which benefits from the proposed anchor refinement module to filter out negative anchors and refine positive anchors to provide better initialization for the location regression. The YOLOv3 method achieves the best AP score of 76.53% for large scale vehicles. These results show that more effective detectors need to be developed to deal with small scale vehicles.

Occlusion ratio. Figure 11 shows the effect of occlusion on detection performance in three categories, *i.e.*, no occlusion, partial occlusion, and heavy occlusion, as described in Section 2.1. When partial occlusion occurs (occlusion ratio is between 1% − 50%), the performance of detectors drops significantly (more than 15% AP score). Furthermore, when heavy occlusion occurs (occlusion ratio is over 50%), the AP scores of all detectors are less than 20%. Significant efforts need to be made for vehicle detection under heavy occlusions.

Table 2: Comparison results based on the MOTA/MOTP and PR-MOTA/MOTP metrics.

Detection	Tracking	PR-MOTA	MOTA	PR-MOTP	MOTP
EB	KIOU	21.1	62.1	28.6	81.9
Faster R-CNN	MHT	14.5	58.2	32.5	78.4
CompACT	GOG	14.2	44.4	37.0	80.8
R-CNN	DCT	11.7	38.4	38.0	80.6
ACF	GOG	10.8	35.7	37.6	80.3
DPM	GOG	5.5	26.2	28.2	76.2

4.2. Multi-Object Tracking

The tracking results of the MOT systems constructed by six object detection and ten object tracking methods on different subsets of the UA-DETRAC benchmark are presented in the following tables: *overall* (Table 3); *easy* (Table 4), *medium* (Table 5), and *hard* (Table 6); *cloudy* (Table 7), *rainy* (Table 10), *sunny* (Table 8), and *night* (Table 9). In addition, we present the performance trends of the MOT systems constructed by the evaluated detection and tracking algorithms, on the *overall* dataset and three subsets (*i.e.*, *easy*, *medium*, and *hard*) in Figure 12. In Table 2, we compare the PR-MOTA score and the corresponding best MOTA score among all the detection thresholds for every detector. Note that the results of the best performing trackers are reported.

As shown in Table 3, existing MOT systems do not perform well, *e.g.*, the top PR-MOTA score (defined in Section 3.2) is only 21.1%, *i.e.*, the EB+KIOU method. These experimental results demonstrate that more efforts are needed to improve the current tracking methods to handle challenging scenarios in the UA-DETRAC dataset. Besides, the Faster R-CNN+CMOT (14.2% PR-MOTA score), Faster R-CNN+TBD (14.4% PR-MOTA score), Faster R-CNN+MHT (14.5% PR-MOTA score), and CompACT +GOG (14.2% PR-MOTA score) methods perform equally well while the DPM+CMOT scheme performs worst among all evaluated systems with the lowest PR-MOTA score of −3.4%.

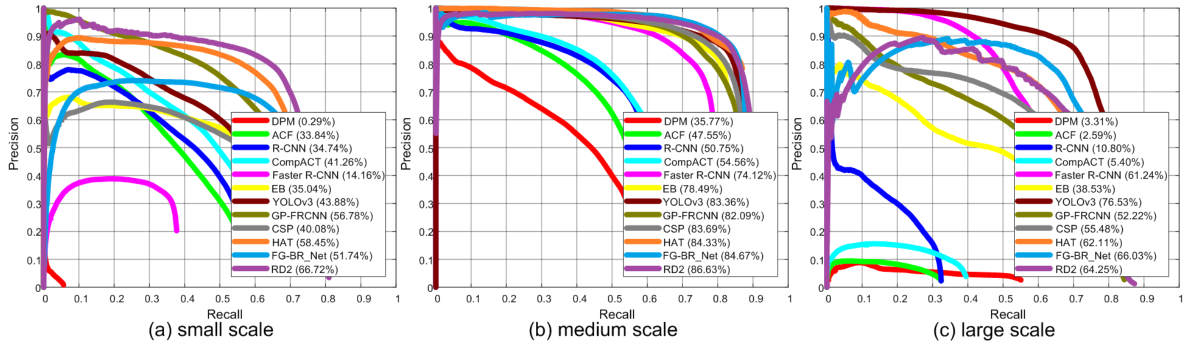

Fig. 10: Precision *vs.* recall curves of the detection algorithms in ***small/medium/large* scale subsets** of UA-DETRAC benchmark dataset. The scores in the legend are the AP scores for evaluating the performance of object detection algorithms.

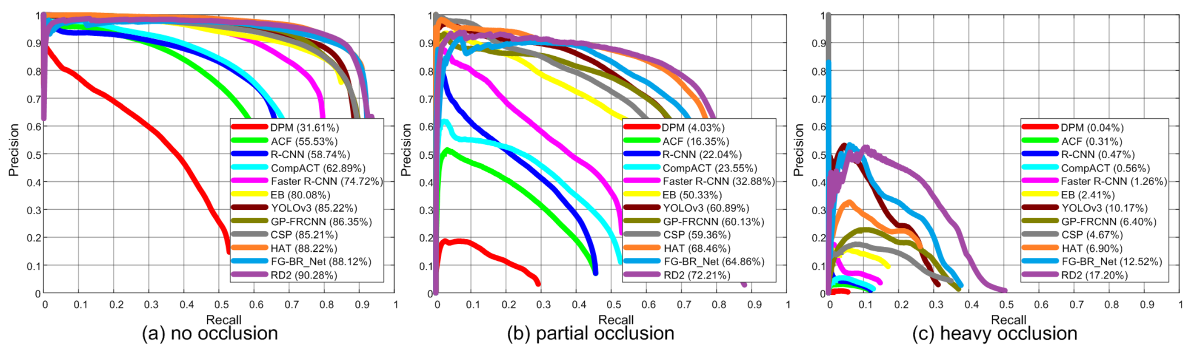

Fig. 11: Precision *vs.* recall curves of the detection algorithms in **no/partial/heavy occlusion subsets** of UA-DETRAC benchmark dataset. The scores in the legend are the AP scores for evaluating the performance of object detection algorithms.

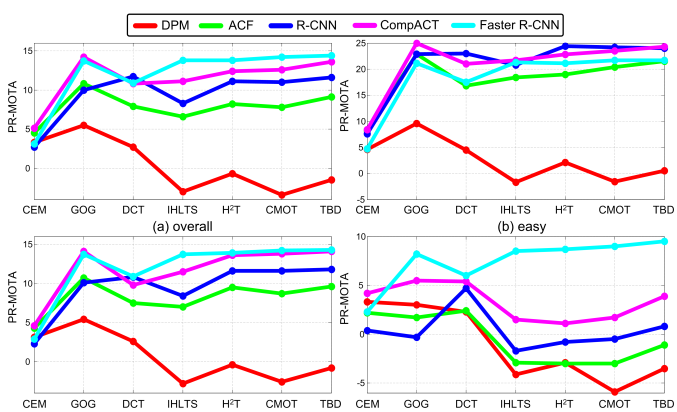

Fig. 12: Performance trends of the MOT systems constructed by different detection and tracking algorithms. The *x*-axis corresponds to different tracking algorithms, and the *y*-axis is the PR-MOTA scores of different MOT systems. Different colors of the curves indicate different object detection algorithms.

Figure 12 shows that a complete MOT system in general achieves better performance with better detections. The average PR-MOTA scores of all object tracking methods with the DPM, ACF, R-CNN, CompACT, and Faster R-CNN detectors are 0.41%, 7.84%, 9.49%, 11.40%, and 11.98% respectively. For example, the difference of PR-MOTA scores of the DPM+H^2T and Faster R-CNN+H^2T methods (−0.7% and 13.8%) is significant. On the other hand, the CEM method performs relatively stably with different detectors than other trackers, *e.g.*, the difference of the PR-MOTA scores of the ACF+CEM and CompACT+CEM (*i.e.*, 0.6%) methods is much smaller than the difference of the CompACT+CMOT and ACF+CMOT (*i.e.*, 4.8%) methods, and the difference of the CompACT+GOG and ACF+GOG (*i.e.*, 3.4%) methods. The H^2T, DCT and IHTLS tracking algorithms use local to global optimization strategies to associate input detections, and do not resolve false positives well. When effective detections are used, the effective appearance (*e.g.*, CMOT) or motion models (*e.g.*, H^2T and IHTLS), and trajectory refining mechanism (*e.g.*, DCT) adopted in these methods help tracking the objects accurately. The TBD algorithm uses a generative model to learn the high-level semantics in terms of traffic patterns. Thus, when combined with an effective detector, it can learn an accurate generative model to gen-

Table 3: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the **overall** UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	19.4	28.9	17.7	18.4	2311.3	2445.9	14796.5	171806.8
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	**21.1**	28.6	**21.9**	**17.6**	462.2	712.1	19046.9	159178.3
Faster R-CNN	GOG	13.7	33.7	12.2	20.2	2213.4	2466.5	11941.6	165757.8
	CEM	3.1	33.4	1.9	30.9	503.8	672.8	17152.9	228871.7
	DCT	10.9	18.9	11.9	19.9	630.3	530.8	32104.7	164767.6
	IHTLS	13.8	18.9	12.5	20.0	456.6	2011.3	13048.7	165695.8
	H^2T	13.8	18.9	11.2	20.5	686.6	841.3	9522.3	168479.6
	CMOT	14.2	18.9	14.1	20.0	157.3	647.5	15854.1	160390.4
	TBD	14.4	19.5	13.5	19.9	1332.8	1483.6	10613.2	163363.1
	MHT	14.5	32.5	15.9	19.1	492.3	576.7	18141.4	**156227.8**
CompACT	GOG	14.2	37.0	13.9	19.9	3334.6	3172.4	32092.9	180183.8
	CEM	5.1	35.2	3.0	35.3	267.9	352.3	12341.2	260390.4
	DCT	10.8	37.1	6.7	29.3	141.4	132.4	13226.1	223578.8
	IHTLS	11.1	36.8	13.8	19.9	953.6	3556.9	53922.3	180422.3
	H^2T	12.4	35.7	14.8	19.4	852.2	1117.2	51765.7	173899.8
	CMOT	12.6	36.1	16.1	18.6	285.3	1516.8	57885.9	167110.8
	TBD	13.6	37.3	15.3	19.3	2026.9	2467.3	43247.8	173837.3
R-CNN	GOG	10.0	38.3	13.5	20.1	7834.5	7401.0	58378.5	192302.7
	CEM	2.7	35.5	2.3	34.1	778.9	1080.4	34768.9	269043.8
	DCT	11.7	38.0	10.1	22.8	758.7	742.9	36561.2	210855.6
	IHTLS	8.3	38.3	12.0	21.4	1536.4	5954.9	68662.6	199268.8
	H^2T	11.1	37.3	14.6	19.8	1481.9	1820.8	66137.2	184358.2
	CMOT	11.0	37.0	15.7	19.0	506.2	2551.1	74253.6	177532.6
	TBD	11.6	**38.7**	14.6	20.3	4110.2	4427.7	56027.6	188676.9
ACF	GOG	10.8	37.6	12.2	22.3	3950.8	3987.3	45201.5	197094.2
	CEM	4.5	35.9	2.9	37.1	265.4	366.0	15180.3	270643.2
	DCT	7.9	37.9	4.8	34.4	108.1	101.4	13059.7	251166.4
	IHTLS	6.6	37.4	11.5	22.4	1243.1	4723.0	72757.5	198673.5
	H^2T	8.2	36.5	13.1	21.3	1122.8	1445.8	71567.4	189649.1
	CMOT	7.8	36.8	14.3	20.7	418.3	2161.7	81401.4	183400.2
	TBD	9.1	38.0	14.1	21.6	2689.0	3101.0	64555.7	189346.3
DPM	GOG	5.5	28.2	4.1	27.7	1873.9	1988.5	38957.6	230126.6
	CEM	3.3	27.9	1.3	37.8	265.0	317.1	13888.7	270718.5
	DCT	2.7	29.3	0.5	42.7	**72.2**	**68.8**	**7785.8**	280762.2
	IHTLS	-3.0	27.9	1.1	29.8	1583.6	4153.5	79197.5	244232.8
	H^2T	-0.7	28.8	2.1	28.4	1738.8	1525.6	71631.0	236520.9
	CMOT	-3.4	28.4	5.1	26.6	447.5	1040.5	104768.3	221991.7
	TBD	-1.5	30.3	5.5	27.0	1914.3	1707.2	88863.0	224179.7

erate accurate object trajectories for better MOT performance in traffic scenes and achieve the best results (*i.e.*, 14.4% PR-MOTA score with the Faster R-CNN scheme). Different from these methods, the CEM scheme uses a global energy minimization strategy to reduce false positives, which helps achieve better performance than other trackers with less accurate input detections (*e.g.*, DPM). Since the CEM scheme does not exploit target appearance information, the performance gain is not significant when more accurate detectors are used. These results show the importance of exploiting the strength of detectors and trackers in developing robust MOT systems.

However, there also exist some counter-examples. For example, the CompACT+GOG method performs better than the Faster R-CNN+GOG method with PR-MOTA of 14.2% and 13.7%, respectively (Faster R-CNN better than CompACT); the ACF+CEM method performs better than the R-CNN+CEM method with PR-MOTA of 4.5% and 2.7%, respectively (R-CNN performs better than ACF); and the R-CNN+DCT method performs better than the CompACT+DCT method, with PR-MOTA of 11.7% and 10.8%, respectively (CompACT performs better than R-CNN). As shown in Figure 12, the performance curves of different detectors over seven trackers on the overall dataset and three subsets (*i.e.*, *easy*, *medium*, and *hard*) are intertwined with each other. Specifically, we note that different trackers perform best with different detectors, *e.g.*, GOG achieves the best performance with CompACT on the overall

Table 4: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***easy* subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

easy subset									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	34.0	37.8	27.8	20.4	573.0	602.9	1579.6	33765.7
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	**36.4**	37.6	**33.7**	20.6	78.1	121.6	2103.8	**30722.4**
Faster R-CNN	GOG	21.1	41.1	16.4	21.9	582.2	609.3	1300.3	33751.3
	CEM	4.7	39.7	3.0	34.9	121.1	154.0	3818.8	52270.9
	DCT	17.5	22.8	15.4	21.8	131.2	105.1	5840.0	34153.8
	IHTLS	21.3	23.3	16.5	21.7	87.3	480.3	1418.2	33826.7
	H^2T	21.1	23.3	14.4	22.3	120.0	143.7	**786.1**	34666.3
	CMOT	21.7	23.6	18.5	21.8	25.1	112.3	2241.7	32648.9
	TBD	21.7	24.1	17.3	21.7	356.8	385.4	1128.3	33444.6
	MHT	23.0	24.1	21.0	21.5	38.8	48.8	1804.3	31442.1
CompACT	GOG	25.0	47.0	21.4	18.8	852.3	795.8	6493.7	34383.1
	CEM	8.4	43.7	4.0	39.6	75.9	91.2	3365.3	59106.9
	DCT	21.0	45.7	11.4	26.4	48.1	44.0	4187.6	42475.8
	IHTLS	21.7	46.6	21.0	19.4	184.0	761.0	10848.4	34806.9
	H^2T	22.8	44.1	22.6	19.5	168.7	198.6	10563.3	33690.7
	CMOT	23.5	45.9	24.3	17.8	42.3	263.4	11820.6	31687.2
	TBD	24.3	47.2	22.7	18.7	443.3	555.7	8615.8	33516.7
R-CNN	GOG	22.9	47.7	23.3	18.6	1603.0	1515.2	9530.0	35296.2
	CEM	7.6	43.6	4.0	35.6	171.8	223.2	6825.4	58504.9
	DCT	23.0	46.3	18.1	19.2	180.8	173.5	8319.1	37772.2
	IHTLS	20.8	47.7	20.5	19.8	297.8	1306.5	11487.9	37172.3
	H^2T	24.4	46.7	24.3	18.3	288.7	320.6	10318.7	33920.7
	CMOT	24.2	46.8	26.0	**17.7**	71.4	404.9	12250.1	32384.1
	TBD	24.0	48.2	23.8	19.1	943.1	950.1	8776.4	35263.3
ACF	GOG	22.9	**48.6**	20.3	20.9	1070.1	1039.9	9086.1	37899.7
	CEM	8.2	45.3	4.2	42.3	73.3	92.9	3501.1	62814.7
	DCT	16.8	47.4	8.2	34.4	43.0	39.0	4100.3	51456.3
	IHTLS	18.4	48.3	19.2	21.5	287.4	1157.4	14449.8	38849.3
	H^2T	19.0	45.7	21.4	22.2	234.1	282.0	14873.0	37750.6
	CMOT	20.4	47.6	23.5	19.7	66.4	390.0	16193.1	34825.4
	TBD	21.5	48.9	22.5	20.8	611.2	702.8	12499.0	36667.2
DPM	GOG	9.6	34.9	4.9	37.8	550.6	573.8	8584.1	60684.7
	CEM	4.6	35.5	1.3	51.6	64.6	71.0	3321.4	72758.6
	DCT	4.5	37.2	0.6	57.1	**22.0**	**21.0**	2411.5	73868.8
	IHTLS	-1.7	29.3	1.1	40.3	428.9	1184.1	18573.3	64988.9
	H^2T	2.1	35.1	2.7	38.8	503.7	420.8	15708.8	62983.8
	CMOT	-1.6	34.7	6.1	36.6	92.9	253.1	24864.9	58973.8
	TBD	0.5	38.6	6.3	60.7	474.4	443.5	21151.2	88645.3

set, while DCT achieves the best performance with R-CNN on the overall set. These results suggest that it is important to choose effective detector for each object tracking algorithm when constructing an MOT system. On the other hand, it is necessary to develop different types of detectors for evaluating object tracking methods comprehensively and fairly (rather than one specific detector).

On the UA-DETRAC-test set (see Table 3), the EB+KIOU method achieves the best result, *i.e.*, 21.1% PR-MOTA score. Followed by the EB+KIOU method, the EB+IOU and Faster R-CNN+MHT methods achieve higher PR-MOTA scores (*i.e.*, 19.4% and 14.5% respectively) than other combinations of MOT systems, while the CEM scheme performs relative worse with five different detection algorithms, *i.e.*, Faster R-CNN, CompACT, R-CNN, ACF, and DPM.

On the *easy* subset (see Table 4), both the IOU and KIOU methods perform well with approximate 35.0% PR-MOTA score using the EB object detector. The GOG algorithm combined with the CompACT scheme performs inferior with 25.0% PR-MOTA score, while the H^2T method achieves comparable PR-MOTA score of 24.4% with the R-CNN detection algorithm. It is worth mentioning that these approaches outperform the Faster R-CNN+MHT method with PR-MOTA of 23.0%, which also indicates the importance of selecting combinations

Table 5: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***medium*** **subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

medium **subset**									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	18.2	26.6	15.5	18.3	1201.0	1271.9	5779.6	94920.7
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	**19.9**	26.3	**19.6**	**17.3**	229.2	356.2	7628.6	**88101.0**
Faster R-CNN	GOG	13.7	17.6	12.2	19.9	1151.7	1287.6	4917.1	91880.0
	CEM	2.9	16.8	1.3	30.9	262.2	355.4	8866.1	127841.6
	DCT	10.9	18.1	10.4	19.8	355.1	307.3	16017.9	91936.0
	IHTLS	13.7	18.2	12.1	20.0	242.8	1060.0	5600.0	92061.5
	H^2T	13.9	17.9	10.8	20.5	364.3	467.7	**3940.1**	93228.5
	CMOT	14.2	18.1	13.5	19.9	86.8	359.8	7033.9	89273.1
	TBD	14.3	18.7	12.9	19.8	697.6	776.5	4542.5	90846.1
	MHT	14.1	31.8	15.0	19.1	291.3	342.9	8348.6	88117.0
CompACT	GOG	14.1	35.1	11.6	20.4	1879.4	1756.1	14534.6	98318.8
	CEM	4.6	33.8	2.4	34.6	137.2	182.9	6371.9	142184.8
	DCT	9.8	35.4	5.3	29.7	75.2	70.9	6781.1	123203.6
	IHTLS	11.5	34.9	11.7	20.5	527.4	1946.6	25120.6	98255.7
	H^2T	13.6	34.3	12.4	19.6	458.1	607.1	21843.7	94097.3
	CMOT	13.8	34.3	13.5	19.1	161.5	846.8	24875.4	90813.1
	TBD	14.1	34.9	11.7	20.3	1155.2	1407.2	17533.0	94782.5
R-CNN	GOG	10.1	35.8	10.7	19.8	4538.5	4234.0	27352.7	102789.4
	CEM	2.3	33.4	1.7	33.2	432.7	600.9	17734.4	144310.0
	DCT	10.8	35.7	7.4	22.8	409.8	401.1	16512.6	115166.2
	IHTLS	8.4	35.8	9.8	21.0	863.3	3290.8	33694.8	106288.7
	H^2T	11.6	32.6	12.1	19.5	800.1	986.9	30594.7	97768.2
	CMOT	11.6	34.5	13.0	18.6	302.8	1510.4	34643.8	94466.6
	TBD	11.8	**36.2**	11.1	21.3	2286.7	2529.7	25102.6	104902.7
ACF	GOG	10.7	35.1	10.0	22.6	2241.0	2244.9	21540.0	106318.5
	CEM	4.3	34.0	2.6	35.5	148.2	204.7	7907.1	144650.8
	DCT	7.5	35.7	4.1	33.4	58.8	55.3	6184.6	135241.2
	IHTLS	7.0	34.9	9.5	22.5	700.4	2634.0	35626.0	106777.5
	H^2T	9.5	34.3	10.9	21.0	618.2	801.8	32650.6	101111.4
	CMOT	8.7	34.3	11.8	20.8	255.3	1292.6	38341.8	98554.0
	TBD	9.6	35.0	10.5	22.2	1571.5	1825.9	29180.9	102010.9
DPM	GOG	5.4	29.4	3.5	28.7	1061.7	1104.7	21447.7	131043.5
	CEM	3.1	28.8	1.0	38.1	153.7	188.1	8581.5	152847.9
	DCT	2.6	30.1	0.5	42.7	**42.7**	**41.1**	4838.6	158656.3
	IHTLS	-2.8	27.9	1.1	30.8	842.1	2227.8	43458.0	138436.8
	H^2T	-0.4	29.1	1.8	29.5	946.9	841.9	38930.3	134364.9
	CMOT	-2.6	29.6	4.3	27.7	249.4	580.2	55505.1	126376.9
	TBD	-0.8	31.4	5.0	28.1	1085.9	972.0	48089.8	128647.7

of detectors and trackers. On the *hard* subset (see Table 6), none of the evaluated methods perform well, *e.g.*, with best PR-MOTA score of 13.1% (EB+KIOU). The PR-MOTA scores of twelve MOT systems, *e.g.*, R-CNN+GOG, ACF+CMOT, DPM+TBD, are all less than 0%, which demonstrates the difficulty of the proposed UA-DETRAC dataset and the badly need of developing more robust methods for real-world applications.

On the *cloudy* subset (see Table 7) and *rainy* subset (see Table 10), the compACT+GOG method achieves higher PR-MOTA score than the tracking methods with the input detections generated by the EB detector. It shows that the necessity to use different detectors on different scenes for better accuracy.

In terms of other metrics, the GOG method achieves good PR-MOTA scores combined with the CompACT scheme in the UA-DETRAC benchmark but with the higher PR-IDS and PR-FM scores. For a given detection method, the MOT systems using the GOG method achieve almost more than twice larger PR-IDS scores comparison to other methods, *e.g.*, Faster R-CNN+GOG (2213.4 PR-IDS) *vs.* Faster R-CNN+H^2T (686.6 PR-IDS), and CompACT+GOG (3334.6 PR-IDS) *vs.* CompACT+CMOT (285.3 PR-IDS). The highest PR-IDS (7834.5) and PR-FM (7401.0) scores are all generated by the R-CNN+GOG approach, which are also almost twice larger than other tracking methods. The GOG scheme uses a

Table 6: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***hard* subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

***hard* subset**									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	11.9	26.2	12.8	18.7	565.3	598.4	6704.9	42709.8
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	**13.1**	25.6	**16.2**	**17.4**	145.3	217.7	8443.2	39633.6
Faster R-CNN	GOG	8.2	30.1	8.1	21.1	507.1	586.1	5357.1	41423.9
	CEM	2.3	30.7	1.4	29.8	124.7	169.5	4508.6	51601.1
	DCT	6.0	17.2	11.1	20.2	147.9	124.7	10285.5	40274.1
	IHTLS	8.5	16.8	9.6	20.4	126.9	485.4	5664.8	41150.7
	H^2T	8.7	17.1	9.3	20.7	203.4	230.3	4536.6	41893.7
	CMOT	9.0	16.8	11.2	20.6	43.1	169.2	6287.1	39792.5
	TBD	9.5	17.3	11.1	20.2	299.7	340.7	4645.5	40434.6
	MHT	9.5	29.0	13.4	18.3	147.9	169.9	7466.0	**37809.8**
CompACT	GOG	5.5	32.8	7.6	24.8	651.7	669.5	10047.6	49483.2
	CEM	4.2	32.3	3.1	34.9	59.2	81.6	2522.9	59497.1
	DCT	5.4	34.0	3.6	35.4	21.9	20.3	1966.0	58331.5
	IHTLS	1.5	31.9	7.3	24.6	228.6	830.3	16364.7	49486.5
	H^2T	1.1	31.6	8.0	24.3	217.9	296.5	18241.6	48204.0
	CMOT	1.7	31.5	9.3	23.0	76.0	380.2	19313.0	46389.6
	TBD	3.9	33.1	8.9	23.7	457.5	569.3	14124.4	48020.5
R-CNN	GOG	-0.3	34.9	6.4	25.0	1583.4	1555.5	18856.2	52261.9
	CEM	0.4	33.0	1.9	34.7	159.4	234.9	9269.2	62186.8
	DCT	4.7	**35.7**	6.3	28.3	165.2	164.4	10498.5	54459.2
	IHTLS	-1.7	32.5	5.4	26.5	326.2	1219.4	20631.5	53732.0
	H^2T	-0.8	31.3	7.0	24.3	358.7	468.0	22443.2	50609.6
	CMOT	-0.5	31.3	8.1	23.4	112.3	550.7	24039.8	48712.5
	TBD	0.8	35.1	7.5	24.9	835.3	945.7	18547.4	51516.6
ACF	GOG	1.7	32.8	5.4	29.0	670.0	722.5	12661.8	54198.7
	CEM	2.2	31.7	2.0	38.5	46.5	69.5	3506.8	63274.2
	DCT	2.4	34.2	1.6	40.2	13.7	13.3	2606.0	63907.1
	IHTLS	-2.9	31.9	4.8	29.0	231.7	864.1	19763.8	54480.2
	H^2T	-3.0	32.0	6.1	27.8	243.8	326.3	21673.6	52671.8
	CMOT	-3.0	31.6	7.0	26.8	81.5	417.2	23304.4	51213.2
	TBD	-1.1	33.1	6.9	28.0	506.7	611.4	18449.9	52744.6
DPM	GOG	3.0	30.3	4.9	31.6	298.6	355.7	9361.4	55825.4
	CEM	3.3	30.2	2.3	38.5	51.9	64.7	2053.4	62950.7
	DCT	2.3	30.2	0.6	42.5	**10.4**	**9.9**	**679.0**	65790.5
	IHTLS	-4.1	30.2	1.0	33.3	317.1	784.4	17543.5	58228.9
	H^2T	-2.9	29.6	2.6	31.7	304.4	278.7	17837.4	56158.0
	CMOT	-5.9	29.5	5.8	29.8	113.9	230.4	24917.4	53682.9
	TBD	-3.5	32.3	5.6	30.4	451.4	402.7	20267.4	54478.8

greedy algorithm to solve the optimization problem on a flow network, which may generate more false trajectories of objects, indicated by higher PR-IDS and PR-FM scores than other trackers. Thus, it is less effective for surveillance scenarios when accuracy of trajectories (*i.e.*, lower PR-IDS and PR-FM scores) is of great importance.

For surveillance scenarios, our results suggest that the EB+KIOU and EB+KIOU methods are more effective than other alternatives with higher PR-MOTA scores and lower PR-IDS and PR-FM scores (see Table 3). While for the traffic safety monitoring scenarios, where false negatives (indicated by PR-FN) are more of the concern than identity switches and trajectory fragmentations, the Faster R-CNN+MHT or EB+KIOU tracking systems seem more suitable and lead to reliable performance.

5. Run-time Performance

We report the run-time of the evaluated object detection algorithms in Table 11. Since object detection algorithms are developed on various platforms (*e.g.*, the R-CNN (Girshick et al., 2014) and Faster R-CNN (Ren et al., 2017) methods requires a GPU for both training and testing), it is difficult to compare the running time efficiency fairly.

For the object tracking algorithms, given the input detection

Table 7: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***cloudy* subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

***cloudy* subset**									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	16.0	22.0	13.1	17.9	530.7	551.9	1637.4	47778.5
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	17.4	21.8	16.5	17.5	83.5	123.7	2248.3	**44824.6**
Faster R-CNN	GOG	14.7	18.8	11.1	19.9	672.5	753.6	1807.0	51264.1
	CEM	2.7	17.2	1.7	31.4	153.1	214.8	4603.2	72521.6
	DCT	12.1	19.1	10.5	20.0	212.1	174.0	7412.2	51210.0
	IHTLS	14.7	19.3	11.1	20.0	125.5	605.1	2101.2	51490.4
	H^2T	14.7	19.0	9.8	20.5	121.7	177.0	**1292.7**	52195.0
	CMOT	15.3	19.2	13.0	19.7	44.6	189.6	2938.5	49427.3
	TBD	15.0	19.8	11.7	19.9	423.7	469.0	1817.3	50939.6
	MHT	14.9	33.3	14.2	18.6	168.9	195.4	4365.1	48682.4
CompACT	GOG	**19.4**	41.6	20.3	16.5	1124.9	1074.2	11435.8	54070.5
	CEM	6.0	38.8	3.8	34.6	113.1	150.8	5544.4	87167.1
	DCT	16.6	41.3	11.9	21.9	73.2	70.5	5874.5	66122.3
	IHTLS	15.9	41.3	19.9	16.7	321.3	1064.0	18422.9	54696.4
	H^2T	17.8	39.6	21.4	16.3	214.3	322.6	16976.1	52638.8
	CMOT	17.8	40.6	22.9	15.4	90.6	445.5	19320.9	50260.6
	TBD	18.7	41.7	21.6	16.3	593.9	682.9	14864.3	52584.0
R-CNN	GOG	15.5	41.4	20.9	16.7	2301.5	2129.8	18389.2	55330.5
	CEM	1.9	37.3	2.5	35.7	282.9	400.8	12314.4	90119.0
	DCT	15.1	40.5	13.7	20.1	239.8	232.5	11866.7	64729.8
	IHTLS	13.3	38.5	19.1	17.7	553.7	1795.3	21932.2	57962.8
	H^2T	17.7	37.2	22.2	**15.7**	386.7	520.3	18618.0	52843.4
	CMOT	16.3	40.5	23.4	**15.7**	178.6	744.5	23241.4	51077.1
	TBD	17.0	**42.7**	**21.9**	20.4	1174.2	1177.9	17672.9	61218.5
ACF	GOG	16.6	42.4	19.2	17.8	1419.5	1427.4	15848.4	58228.8
	CEM	5.7	39.6	3.8	36.8	121.3	166.5	6274.1	90616.5
	DCT	13.8	42.3	8.6	27.2	65.0	60.2	5345.3	75672.1
	IHTLS	12.0	42.2	18.1	18.2	459.2	1526.3	24547.2	59523.1
	H^2T	14.4	40.5	20.1	17.7	297.7	429.7	22640.9	56851.3
	CMOT	13.9	41.4	21.6	16.5	144.1	643.7	26604.7	54021.6
	TBD	15.0	42.6	20.8	17.5	841.0	911.9	21684.6	56224.8
DPM	GOG	5.6	28.2	3.0	25.2	585.3	614.1	11827.8	70211.2
	CEM	2.6	27.2	1.0	35.8	103.1	124.3	5646.3	82784.7
	DCT	3.6	28.6	0.8	37.1	**35.3**	**35.1**	4236.5	82335.9
	IHTLS	-2.7	26.9	0.8	27.9	471.4	1237.6	23659.1	74763.0
	H^2T	1.4	28.6	1.7	25.6	465.1	428.0	18898.6	71484.7
	CMOT	-1.9	28.6	3.6	24.2	115.9	318.8	29176.7	68000.8
	TBD	-3.5	32.3	5.6	30.4	451.4	402.7	20267.4	54478.8

generated by different detection algorithms (*e.g.*, DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), CompACT (Cai et al., 2015), and Faster R-CNN (Ren et al., 2017)) with the largest F-score, the average run-time on 40 sequences in the UA-DETRAC-test set are presented in Table 12. The run-time is measured on a computer with a 2.9 GHz Intel i7 processor and 16 GB memory.

The detection approaches based on deep learning are more attractive than other methods (*e.g.*, DPM and ACF) in terms of accuracy when computing resources are not constrained. For the surveillance applications, taking both accuracy and speed into account (see Table 3, 11, and 12), the EB+IOU and EB+KIOU systems lead to the most accurate results with relative high efficiency based on our evaluation. However, for applications with constrained computing resources, the ACF+CMOT and ACF+H^2T methods achieve higher PR-MOTA score and lower PR-IDS and PR-FM scores with relative high efficiency.

6. Conclusions and Future Research Directions

In this work, we present a large scale multi-object tracking benchmark (UA-DETRAC) consisting of 100 videos with rich annotations. We carry out extensive experiments to evaluate the performance of twelve object detection and ten object track-

Table 8: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***sunny* subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

***sunny* subset**									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	35.6	42.8	32.7	19.3	401.0	426.9	2240.4	21065.2
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	**38.3**	42.5	**38.8**	19.4	59.0	97.4	2769.1	18899.4
Faster R-CNN	GOG	20.9	22.3	19.9	20.9	352.7	396.0	1496.0	20087.9
	CEM	5.8	21.7	3.8	33.0	90.4	107.2	2833.2	30197.5
	DCT	16.6	23.4	17.8	20.8	80.6	69.5	4464.9	20600.8
	IHTLS	20.6	24.0	20.1	20.9	74.1	324.7	1898.0	20162.8
	H^2T	21.4	24.1	17.5	20.6	119.4	145.2	992.8	20467.6
	CMOT	21.4	24.5	21.5	21.0	21.5	90.8	1934.8	19569.8
	TBD	22.0	24.9	21.1	20.8	208.0	232.2	1142.1	19763.0
	MHT	23.0	25.0	24.0	20.6	28.8	40.8	1706.2	**18650.1**
CompACT	GOG	22.5	44.6	21.5	16.3	386.1	361.6	3561.2	20674.2
	CEM	7.1	41.5	4.6	37.2	36.0	38.6	1725.5	34204.4
	DCT	16.3	43.0	10.2	26.1	23.1	21.2	1958.8	27218.9
	IHTLS	18.9	44.4	21.1	16.8	112.9	402.3	6181.7	20950.1
	H^2T	18.9	43.8	22.0	16.9	132.4	138.2	6800.5	20338.8
	CMOT	20.2	43.8	23.5	**15.4**	27.5	133.8	6877.3	19377.2
	TBD	21.5	45.5	22.8	15.9	252.3	321.7	4905.6	20150.4
R-CNN	GOG	20.1	47.6	23.1	19.6	940.3	898.2	6706.5	22922.1
	CEM	7.8	43.7	4.8	35.5	113.9	142.5	4226.1	35310.9
	DCT	21.1	45.9	19.1	20.3	111.1	113.4	5372.9	24327.4
	IHTLS	18.1	44.2	19.3	21.0	203.6	779.4	7675.2	24179.7
	H^2T	18.7	42.7	23.5	21.6	209.4	209.8	8622.5	22738.9
	CMOT	21.0	46.8	25.5	18.7	44.5	251.6	8495.2	21375.7
	TBD	21.3	**49.8**	22.9	27.5	612.1	633.8	5526.5	28281.8
ACF	GOG	20.5	45.8	20.4	17.3	526.9	513.6	5292.7	22036.3
	CEM	9.5	42.9	5.7	36.2	45.7	52.1	2195.5	33771.0
	DCT	14.6	44.1	8.7	31.1	20.8	20.5	2153.3	30084.2
	IHTLS	16.0	44.8	18.9	17.8	168.0	615.9	8579.1	22476.0
	H^2T	16.2	44.3	21.0	18.1	184.0	183.2	9383.8	21531.4
	CMOT	17.2	44.9	23.5	16.2	37.4	221.0	9834.3	20491.2
	TBD	19.1	46.8	23.0	16.7	354.9	409.7	7388.1	21178.9
DPM	GOG	7.2	29.4	6.3	16.8	316.2	334.0	6017.9	22758.3
	CEM	3.3	28.0	1.6	32.6	26.2	27.5	2060.1	29890.9
	DCT	2.2	31.9	0.3	40.2	**8.5**	**7.5**	**838.3**	31962.1
	IHTLS	-3.1	29.4	1.2	19.6	242.8	666.5	11080.1	25429.3
	H^2T	-1.6	29.4	3.2	19.0	364.3	301.3	10876.6	24378.6
	CMOT	-5.1	29.7	7.5	15.8	62.1	171.8	16312.8	21810.8
	TBD	-2.3	34.3	8.7	22.5	345.1	290.7	13655.1	26388.1

ing methods. We show it is necessary to understand the effect of detection accuracy on the complete MOT system performance. Using the proposed UA-DETRAC metrics, we analyze the quantitative results and conclude with a discussion of the state of the art in both object detection and tracking approaches. Based on the analysis and discussion, there are several research directions worth exploring:

Protocol. The proposed evaluation protocol can only be used to evaluate tracking methods taking detection boxes as inputs. For tracking methods that operate on likelihood maps, we can use thresholds to determine bounding boxes for each frame, such that the proposed evaluation protocol can also be used. In addition, to reduce the computational complexity in evaluation, it is necessary to design a new protocol without tuning the precision and recall rates of input detections for MOT.

Metrics. The current evaluation metrics are designed for general tracking applications. However, in surveillance applications, we need reliably trajectories of objects. Thus, more emphasis should be made on the identity switches in evaluation. While for autonomous driving, we should avoid false negatives at all cost, but can live with identity switches. Thus, it is interesting to design some comprehensive metrics to replace the current ones, *e.g.*, PR-MOTA and PR-MOTP, to adapt to the requirements of different application scenarios.

Table 9: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***night* subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

***night* subset**									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	26.8	35.5	23.3	18.9	772.4	829.5	4139.3	44702.2
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	**28.5**	35.0	**28.7**	**18.2**	181.7	276.8	5926.0	40838.6
Faster R-CNN	GOG	17.2	**36.5**	14.8	21.1	670.4	702.4	3028.5	40709.1
	CEM	5.3	**36.5**	2.1	33.0	155.7	206.9	4291.1	58646.4
	DCT	14.8	20.3	14.7	20.4	173.2	154.7	7978.2	39980.2
	IHTLS	17.6	20.2	15.9	20.5	151.5	583.5	3209.4	40374.0
	H^2T	17.7	20.1	14.3	21.1	290.0	319.2	2078.0	41237.6
	CMOT	18.3	20.2	17.4	20.8	45.3	191.7	3591.8	38945.3
	TBD	18.7	20.7	17.4	20.3	362.0	403.4	2187.2	39499.6
	MHT	19.5	20.7	19.9	20.2	90.3	119.9	3052.0	**37570.5**
CompACT	GOG	10.9	33.2	6.2	22.6	831.3	805.1	4692.6	44419.8
	CEM	5.5	32.6	1.6	33.9	38.0	47.9	**1206.9**	57282.7
	DCT	6.9	34.1	1.7	34.3	12.4	9.0	2298.5	53994.2
	IHTLS	8.6	32.8	6.6	22.2	190.9	1002.9	9683.2	43678.6
	H^2T	9.6	30.9	6.9	21.5	250.0	316.2	9995.4	41819.3
	CMOT	10.9	32.0	8.3	20.4	52.2	453.1	9888.6	40081.7
	TBD	10.9	33.1	7.6	21.4	573.2	774.5	7188.8	42231.5
R-CNN	GOG	7.2	33.0	5.2	22.1	1961.4	1874.6	8673.6	46152.3
	CEM	4.6	31.3	1.6	29.2	178.1	239.6	6785.0	53971.9
	DCT	9.3	33.3	4.4	23.1	187.3	182.7	6785.0	46554.3
	IHTLS	6.0	32.8	5.2	23.5	258.6	1391.7	11238.1	47199.5
	H^2T	7.7	31.6	6.1	21.9	371.6	469.0	11767.8	43851.8
	CMOT	9.5	31.3	6.9	20.9	73.2	644.3	11033.5	42052.3
	TBD	9.0	32.9	6.3	22.2	942.3	1154.0	8023.6	45063.3
ACF	GOG	6.8	32.1	3.8	25.5	889.1	885.8	5845.5	48213.3
	CEM	3.4	31.0	0.9	35.3	26.8	40.5	1949.4	58206.8
	DCT	3.4	32.5	1.0	36.4	**6.5**	**5.1**	2515.3	57697.3
	IHTLS	4.2	31.4	4.3	24.9	196.2	1193.3	11318.5	47441.2
	H^2T	4.8	30.3	4.9	23.6	272.5	368.3	12896.2	44873.1
	CMOT	6.0	31.0	5.3	23.0	65.6	608.2	12288.1	43784.3
	TBD	6.3	31.7	5.3	24.2	687.2	915.3	9189.5	45802.3
DPM	GOG	8.0	31.6	4.6	28.3	515.6	602.2	9933.9	54420.5
	CEM	5.2	31.8	1.3	38.0	60.6	73.4	3141.3	66044.1
	DCT	3.5	34.2	0.5	45.3	16.2	15.3	1945.7	69830.9
	IHTLS	-2.4	32.4	1.4	29.6	443.5	1303.5	22572.0	58075.6
	H^2T	-1.6	31.2	2.2	28.4	567.8	487.8	23406.8	55813.0
	CMOT	-2.9	31.3	5.8	26.8	136.6	321.5	30127.8	51544.9
	TBD	-0.2	34.2	6.5	27.3	613.9	549.4	24528.6	52590.9

Joint detection and tracking. Performance of object detection significantly affects the tracking performance, and the temporal coherency in tracking can help detection vice versa. It is of great interest to combine object detection and tracking in a unified framework to boost each other for better performance. We expect additional gains in performance of object detection and tracking from continued research on combining them.

Real-time issue. The deep learning approaches surpass other methods by a large margin in terms of performance, especially in object detection field. However, the requirements of computational resources are very harsh to some extent. Some recent approaches (Zhang et al., 2015; Rastegari et al., 2016; Wen et al., 2016b; Howard et al., 2017) focus on pruning, compressing, or low-bit representing a "basic" network to adapt to embedded platforms, which aim to achieve high efficiency while maintaining comparable accuracy. We expect a stream of works coming out, focusing more on real-time constraints for object detection and MOT.

Data. We expect to extend the current UA-DETRAC dataset to include more sequences and richer annotations for evaluation on pedestrian detection and tracking approaches. We will also conduct studies to examine the effect of quantity and type of training data versus performance for both object detection and tracking fields.

Table 10: PR-MOTA, PR-MOTP, PR-MT, PR-ML, PR-IDS, PR-FM, PR-FP, and PR-FN scores of the MOT systems constructed by five object detection algorithms and seven object tracking algorithms on the ***rainy* subset** of the UA-DETRAC benchmark dataset. The evaluation results of the winners in the UA-DETRAC Challenge 2017 and 2018 are also reported. Bold faces correspond to the best performance of the MOT systems on that metric. The pink, cyan, and gray rows denote the trackers ranked in the first, second, and third places based on the PR-MOTA score with the corresponding detector.

***rainy* subset**									
Detection	Tracking	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDS	PR-FM	PR-FP	PR-FN
UA-DETRAC 2017 Challenge Winner									
EB	IOU	8.8	18.3	7.2	17.6	512.5	533.0	4381.4	48462.9
UA-DETRAC 2018 Challenge Winner									
EB	KIOU	10.0	18.0	10.3	**16.4**	103.8	151.6	5438.7	**45341.3**
Faster R-CNN	GOG	8.4	29.3	7.5	20.5	550.0	639.8	5191.1	53530.4
	CEM	1.5	29.2	0.9	29.5	114.9	154.6	5333.6	68080.7
	DCT	5.8	16.1	8.3	20.3	165.7	136.2	11567.3	52910.9
	IHTLS	8.5	16.0	7.6	20.6	112.6	518.6	5477.6	53501.5
	H^2T	8.5	16.1	7.2	21.3	172.3	216.2	4676.7	54339.2
	CMOT	8.6	15.9	9.0	20.4	46.4	178.2	6743.1	52141.1
	TBD	8.9	16.4	8.3	20.3	351.8	391.7	4959.0	52948.0
	MHT	8.4	27.7	10.8	18.9	186.6	204.3	8223.9	50866.5
CompACT	GOG	**10.4**	34.1	10.9	20.2	1183.2	1112.5	11993.6	61194.0
	CEM	4.5	33.3	3.2	35.2	93.0	126.1	3929.3	82490.1
	DCT	7.7	35.2	4.6	32.3	43.9	41.1	3138.7	76723.7
	IHTLS	7.0	33.8	10.6	20.4	347.3	1232.1	19588.9	61250.8
	H^2T	8.9	33.5	12.0	19.2	297.5	379.7	18311.1	58742.4
	CMOT	8.5	33.1	**13.0**	18.8	118.7	530.9	21476.6	56650.2
	TBD	9.8	34.9	12.5	19.6	731.5	840.0	15887.3	58987.5
R-CNN	GOG	3.8	36.3	9.8	21.8	2529.9	2397.6	24592.1	66215.2
	CEM	0.3	33.9	1.4	37.7	194.9	283.1	10826.2	89416.9
	DCT	7.1	36.7	7.1	26.9	218.1	212.5	12174.4	74121.5
	IHTLS	2.3	34.7	8.0	23.2	531.1	1910.8	27517.5	68421.8
	H^2T	5.1	34.4	11.2	20.5	489.3	574.7	26781.2	63481.3
	CMOT	3.8	35.1	12.1	20.5	211.3	836.9	31568.7	61486.0
	TBD	5.4	36.5	10.8	21.8	1329.3	1386.8	22967.6	64593.4
ACF	GOG	6.4	35.4	8.6	23.9	1306.9	1341.9	17473.7	68161.9
	CEM	3.6	34.9	2.7	38.5	82.7	119.0	4900.1	87777.9
	DCT	4.7	**36.8**	2.4	39.9	28.2	26.5	3234.7	87241.7
	IHTLS	1.5	35.0	7.8	24.1	432.4	1515.6	27693.8	68772.8
	H^2T	3.5	34.6	9.9	22.0	403.0	498.0	26793.2	65664.9
	CMOT	2.1	34.4	10.9	21.8	169.5	722.9	31718.4	63816.6
	TBD	4.0	36.2	10.7	23.0	928.8	1013.1	25173.5	65689.2
DPM	GOG	5.1	30.5	5.1	27.5	596.5	596.9	11988	71805.7
	CEM	4.0	30.3	2.2	36.2	83.0	101.7	3201.4	83325.6
	DCT	2.6	30.2	0.6	41.8	**18.3**	**16.7**	**1007.0**	88439.9
	IHTLS	-2.2	30.5	1.5	29.5	471.6	1193.3	23492.5	75360.0
	H^2T	-0.3	30.9	2.8	27.7	471.3	409.8	21537.0	73429.6
	CMOT	-2.7	29.9	6.3	26.1	162.5	296.9	31485.3	68800.3
	TBD	-1.0	32.3	6.0	26.6	648.1	541.1	26449.2	69884.8

Acknowledgement

This work was supported in part by US Natural Science Foundation under Grant IIS1816227, and in part by National Nature Science Foundation of China under Grant 61472388 and Grant 61771341.

References

Amin, S., Galasso, F., 2017. Geometric proposals for faster R-CNN, in: AVSS, pp. 1–6.

Andriluka, M., Roth, S., Schiele, B., 2008. People-tracking-by-detection and people-detection-by-tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Andriyenko, A., Schindler, K., 2011. Multi-target tracking by continuous energy minimization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1265–1272.

Andriyenko, A., Schindler, K., Roth, S., 2012. Discrete-continuous optimization for multi-target tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1926–1933.

Bae, S.H., Yoon, K., 2014. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1218–1225.

Bashir, F., Porikli, F., 2006. Performance evaluation of object detection and tracking systems, in: PETS.

Benfold, B., Reid, I., 2011. Stable multi-target tracking in real-time surveillance video, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3457–3464.

Table 11: Average frame-per-second of the object detection algorithms on the UA-DETRAC-test set.

Detectors	DPM	ACF	R-CNN	CompACT	Faster R-CNN	GP-FRCNN	EB
Platform	CPU: 4×Intel Core i7-6600U (2.60GHz)	CPU: 2×Intel Xeon E5-2470v2 (2.40GHz)	CPU: 2×Intel Xeon E5-2470v2 (2.40GHz) GPU: Tesla K40	CPU: 2×Intel Xeon E5-2470v2 (2.40GHz) GPU: Tesla K40	CPU: E5-2695v3 (2.30GHz) GPU: TitanX	CPU: 12×Intel Xeon E5-2690v3 (2.60GHz) GPU: Tesla K40	CPU: 4×Intel Core i7-4770 (3.40GHz) GPU: TitanX
Codes	Matlab,C++	Matlab	Matlab,C++	Matlab,C++	C++	Python, C++	C++
Speed	0.17	0.67	0.10	0.22	11.11	4.0	11.00

Table 12: Average frame-per-second of the object tracking algorithms on the UA-DETRAC-test set with the largest F-score detection results generated by five different detection algorithms, *i.e.*, DPM (Felzenszwalb et al., 2010), ACF (Dollár et al., 2014), R-CNN (Girshick et al., 2014), CompACT (Cai et al., 2015), and Faster R-CNN (Ren et al., 2017).

Trackers	Codes	DPM	ACF	R-CNN	CompACT	Faster R-CNN	Average
CEM	Matlab	4.49	3.74	5.40	4.62	5.71	4.79
GOG	Matlab	476.52	319.29	352.80	389.51	484.95	404.61
DCT	Matlab,C++	2.85	1.29	0.71	2.19	1.32	1.67
IHTLS	Matlab	7.94	5.09	11.96	19.79	38.92	16.74
H^2T	C++	1.77	1.08	2.78	3.02	6.29	2.99
CMOT	Matlab	4.48	3.12	3.59	3.79	3.50	3.70
TBD	Matlab	0.60	3.17	3.17	4.88	8.99	4.16

Berclaz, J., Fleuret, F., Türetken, E., Fua, P., 2011. Multiple object tracking using k-shortest paths optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 1806–1819.

Bochinski, E., Eiselein, V., Sikora, T., 2017. High-speed tracking-by-detection without using image information, in: AVSS, pp. 1–6.

Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.J.V., 2011. Online multi-person tracking-by-detection from a single, uncalibrated camera. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 1820–1833.

Brendel, W., Amer, M.R., Todorovic, S., 2011. Multiobject tracking as maximum weight independent set, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1273–1280.

Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N., 2016. A unified multi-scale deep convolutional neural network for fast object detection, in: Proceedings of European Conference on Computer Vision, pp. 354–370.

Cai, Z., Saberian, M., Vasconcelos, N., 2015. Learning complexity-aware cascades for deep pedestrian detection, in: Proceedings of the IEEE International Conference on Computer Vision.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893.

Dalal, N., Triggs, B., Schmid, C., 2006. Human detection using oriented histograms of flow and appearance, in: Proceedings of European Conference on Computer Vision, pp. 428–441.

Dehghan, A., Assari, S.M., Shah, M., 2015. GMMCP-Tracker:globally optimal generalized maximum multi clique problem for multiple object tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Delong, A., Osokin, A., Isack, H.N., Boykov, Y., 2012. Fast approximate energy minimization with label costs. International Journal of Computer Vision 96, 1–27.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F., 2009. Imagenet: A large-scale hierarchical image database, in: CVPR, pp. 248–255.

Dicle, C., Camps, O.I., Sznaier, M., 2013. The way they move: Tracking multiple targets with similar appearance, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2304–2311.

Dollár, P., Appel, R., Belongie, S., Perona, P., 2014. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 1532–1545.

Dollár, P., Wojek, C., Schiele, B., Perona, P., 2012. Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 743–761.

Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., Tian, Q., 2018. The unmanned aerial vehicle benchmark: Object detection and tracking, in: ECCV, pp. 375–391.

Ess, A., Leibe, B., Gool, L.J.V., 2007. Depth and appearance for mobile scene analysis, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1–8.

Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision 111, 98–136.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1627–1645.

Ferryman, J.M., Shahrokni, A., 2009. PETS2009: dataset and challenge, in: Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance, pp. 1–6.

Fortmann, T., Shalom, Y.B., Scheffe, M., 1983. Sonar tracking of multiple targets using joint probabilistic data association. IEEE J. Oceanic Engineering 8, 173–184.

Fu, Z., Chen, Y., Yong, H., Jiang, R., Zhang, L., Hua, X., 2019. Foreground gating and background refining network for surveillance object detection. TIP 28, 6077–6090.

Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R., 2014. 3d traffic scene understanding from movable platforms. IEEE Transactions on Pattern Analysis and Machine Intelligence 36, 1012–1025.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the KITTI vision benchmark suite, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361.

Girshick, R.B., 2015. Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision.

Girshick, R.B., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 1904–1916.

Hosang, J.H., Omran, M., Benenson, R., Schiele, B., 2015. Taking a deeper look at pedestrians, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4073–4082.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: CVPR, pp. 7132–7141.

Huang, C., Li, Y., Nevatia, R., 2013. Multiple target tracking by learning-based hierarchical association of detection responses. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 898–910.

Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R., 2018. The apolloscape dataset for autonomous driving, in: CVPRWorkshops, pp. 954–960.

Hwang, S., Park, J., Kim, N., Choi, Y., Kweon, I.S., 2015. Multispectral pedestrian detection: Benchmark dataset and baseline, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Isard, M., Blake, A., 1998. Condensation - conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28.

Izadinia, H., Saleemi, I., Li, W., Shah, M., 2012. $(\text{MP})^2$T: Multiple people multiple parts tracker, in: Proceedings of European Conference on Computer Vision, pp. 100–114.

Jiang, H., Fels, S., Little, J.J., 2007. A linear programming approach for multiple object tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Khan, Z., Balch, T.R., Dellaert, F., 2005. MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1805–1918.

Kim, C., Li, F., Ciptadi, A., Rehg, J.M., 2015. Multiple hypothesis tracking revisited, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 4696–4704.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, pp. 1106–1114.

Lampert, C.H., Blaschko, M.B., Hofmann, T., 2008. Beyond sliding windows: Object localization by efficient subwindow search, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S., 2014. Learning an image-based motion context for multiple people tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3542–3549.

Leal-Taixé, L., Milan, A., Reid, I.D., Roth, S., Schindler, K., 2015. Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint abs/1504.01942.

Leven, W.F., Lanterman, A.D., 2009. Unscented kalman filters for multiple target tracking with symmetric measurement equations. IEEE Transactions on Automatic Control 54, 370–375.

Li, Y., Huang, C., Nevatia, R., 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2953–2960.

Liu, W., Liao, S., Ren, W., Hu, W., Yu, Y., 2019. High-level semantic feature detection: A new perspective for pedestrian detection, in: CVPR, pp. 5187–5196.

Lyu, S., Chang, M., Du, D., Li, W., Wei, Y., Coco, M.D., Carcagnì, P., *et al.*, 2018. UA-DETRAC 2018: Report of AVSS2018 & IWT4S challenge on advanced traffic monitoring, in: AVSS, pp. 1–6.

Lyu, S., Chang, M., Du, D., Wen, L., Qi, H., Li, Y., Wei, Y., Ke, L., Hu, T., Coco, M.D., Carcagnì, P., *et al.*, 2017. UA-DETRAC 2017: Report of AVSS2017 & IWT4S challenge on advanced traffic monitoring, in: AVSS, pp. 1–7.

Maddern, W., Pascoe, G., Linegar, C., Newman, P., 2017. 1 year, 1000 km: The oxford robotcar dataset. I. J. Robotics Res. 36, 3–15.

Mikami, D., Otsuka, K., Yamato, J., 2009. Memory-based particle filter for face pose tracking robust under complex dynamics, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 999–1006.

Milan, A., Leal-Taixé, L., Reid, I.D., Roth, S., Schindler, K., 2016. Mot16: A benchmark for multi-object tracking. arXiv preprint abs/1603.00831.

Milan, A., Schindler, K., Roth, S., 2013. Challenges of ground truth evaluation of multi-target tracking, in: Workshops in Conjunction with IEEE Conference on Computer Vision and Pattern Recognition, pp. 735–742.

Ouyang, W., Wang, X., 2012. A discriminative deep model for pedestrian detection with occlusion handling, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3258–3265.

Overett, G., Petersson, L., Brewer, N., Andersson, L., Pettersson, N., 2008. A new pedestrian dataset for supervised learning, in: Proceedings of IEEE Intelligent Vehicles Symposium, pp. 373–378.

Patino, L., . PETS2016 dataset website. http://www.cvg.reading.ac.uk/PETS2016/a.html. [Online; accessed 23-Februay-2017].

Pellegrini, S., Ess, A., Schindler, K., Gool, L.J.V., 2009. You'll never walk alone: Modeling social behavior for multi-target tracking, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 261–268.

Pirsiavash, H., Ramanan, D., Fowlkes, C.C., 2011. Globally-optimal greedy algorithms for tracking a variable number of objects, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1201–1208.

Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A., 2016. Xnor-net: Imagenet classification using binary convolutional neural networks, in: Proceedings of European Conference on Computer Vision, pp. 525–542.

Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. CoRR abs/1804.02767.

Reid, D.B., 1979. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 24, 843–854.

Ren, S., He, K., Girshick, R.B., Sun, J., 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1137–1149.

Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A.R., Reid, I.D., 2015. Joint probabilistic data association revisited, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 3047–3055.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision , 1–42.

van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M., 2011. Segmentation as selective search for object recognition, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1879–1886.

Shi, X., Ling, H., Hu, W., Yuan, C., Xing, J., 2014. Multi-target tracking with motion context in tensor power iteration, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3518–3525.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: ICLR.

Sivic, J., Zisserman, A., 2003. Video google: A text retrieval approach to object matching in videos, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1470–1477.

Solera, F., Calderara, S., Cucchiara, R., 2015. Towards the evaluation of reproducible robustness in tracking-by-detection, in: Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance.

Stiefelhagen, R., Bernardin, K., Bowers, R., Garofolo, J.S., Mostefa, D., Soundararajan, P., 2006. The CLEAR 2006 evaluation, in: International Evaluation Workshop on Classification of Events, Activities and Relationships, pp. 1–44.

Tang, S., Andres, B., Andriluka, M., Schiele, B., 2015. Subgraph decomposition for multi-target tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5033–5041.

Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M., 2013. Selective search for object recognition. International Journal of Computer Vision 104, 154–171.

Viola, P.A., Jones, M.J., 2004. Robust real-time face detection. International Journal of Computer Vision 57, 137–154.

Wang, L., Lu, Y., Wang, H., Zheng, Y., Ye, H., Xue, X., 2017. Evolving boxes for fast vehicle detection, in: ICME, pp. 1135–1140.

Wen, L., Lei, Z., Lyu, S., Li, S.Z., Yang, M.H., 2016a. Exploiting hierarchical dense structures on hypergraphs for multi-object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 1983–1996.

Wen, L., Li, W., Lei, Z., Yi, D., Li, S.Z., 2014. Multiple target tracking based on undirected hierarchical relation hypergraph, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3457–3464.

Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H., 2016b. Learning structured sparsity in deep neural networks, in: Advances in Neural Information Processing Systems, pp. 2074–2082.

Wojek, C., Walk, S., Schiele, B., 2009. Multi-cue onboard pedestrian detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 794–801.

Wu, S., Kan, M., Shan, S., Chen, X., 2019. Hierarchical attention for part-aware face detection. IJCV 127, 560–578.

Wu, Z., Fuller, N.W., Theriault, D.H., Betke, M., 2014. A thermal infrared video benchmark for visual analysis, in: Workshops in Conjunction with IEEE Conference on Computer Vision and Pattern Recognition, pp. 201–208.

Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L., 2011. Who are you with and where are you going?, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1345–1352.

Yan, J., Lei, Z., Wen, L., Li, S.Z., 2014. The fastest deformable part model for object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2497–2504.

Yang, B., Nevatia, R., 2012. Online learned discriminative part-based appearance models for multi-human tracking, in: Proceedings of European Conference on Computer Vision, pp. 484–498.

Yang, M., Liu, Y., Wen, L., You, Z., Li, S.Z., 2014. A probabilistic framework for multitarget tracking with mutual occlusions, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1298–1305.

Yang, S., Luo, P., Loy, C.C., Tang, X., 2016. WIDER FACE: A face detection benchmark, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5525–5533.

Yong, H., Meng, D., Zuo, W., Zhang, L., 2018. Robust online matrix factorization for dynamic background subtraction. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, 1726–1740.

Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T., 2018. BDD100K: A diverse driving video database with scalable annotation tooling. CoRR abs/1805.04687.

Zamir, A.R., Dehghan, A., Shah, M., 2012. GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs, in: Proceedings of European Conference on Computer Vision, pp. 343–356.

Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.Z., 2007. Face detection based on multi-block LBP representation, in: Proceedings of International Conference on Biometrics, pp. 11–18.

Zhang, L., Li, Y., Nevatia, R., 2008. Global data association for multi-object tracking using network flows, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z., 2018. Single-shot refinement neural network for object detection, in: CVPR, pp. 4203–4212.

Zhang, X., Zou, J., Ming, X., He, K., Sun, J., 2015. Efficient and accurate approximations of nonlinear convolutional networks, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1984–1992.

Zitnick, C.L., Dollár, P., 2014. Edge boxes: Locating object proposals from edges, in: Proceedings of European Conference on Computer Vision, pp. 391–405.

Appendix

We present the proof of the range of the PR-MOTA score Ω^*. The PR-MOTA score Ω^* is defined as the line integral along the PR curve, *i.e.*,

$$\Omega^* = \frac{1}{2}\int_C \Psi(p, r)\mathrm{d}\mathbf{s} \tag{2}$$

where C is the PR curve, and $\Psi(p, r)$ is the MOTA value corresponding to the precision p and recall r on the PR curve. Since any MOTA score $\Psi(p, r) \in (-\infty, 100]$, we have the lower bound of Ω^*: $-\infty$. We determine the upper bound of Ω^* as follows. Let $C_0, \cdots, C_n$ be the dividing points on the PR curve, where the i-th point is $C_i = (p_i, r_i)$ corresponding to the precision p_i and recall r_i on the PR curve, and C_0 and C_n are two end points on the PR curve. We denote the length of the i-th arc determined by $C_{i-1}C_i$ as Δs_i. Let $\lambda = \max_{1\leq i\leq n} \Delta s_i$. Thus, we have $\Delta s_i = \sqrt{\Delta p_i^{\,2} + \Delta r_i^{\,2}}$, if $\lambda \to 0$. Then, the PR-MOTA score Ω^* can be rewritten as

$$\Omega^* = \frac{1}{2}\int_C \Psi(p, r)\mathrm{d}\mathbf{s} = \frac{1}{2}\lim_{\lambda\to 0}\sum_{i=1}^{n} \Psi(p_i, r_i)\Delta s_i \tag{3}$$

$\forall i$, we have $\Psi(p_i, r_i) \leq 100$. Thus, we can get

$$\begin{aligned}
\Omega^* &\leq \frac{1}{2}\cdot 100 \cdot \lim_{\lambda\to 0}\sum_{i=1}^{n}\Delta s_i \\
&= 50 \cdot \lim_{\lambda\to 0}\sum_{i=1}^{n}\sqrt{\Delta p_i^{\,2} + \Delta r_i^{\,2}} \\
&\leq 50 \cdot \lim_{\lambda\to 0}\sum_{i=1}^{n}(|\Delta p_i| + |\Delta r_i|) \\
&= 50 \cdot \lim_{\lambda\to 0}(\sum_{i=1}^{n}|\Delta p_i| + \sum_{i=1}^{n}|\Delta r_i|)
\end{aligned}$$

Since the precision p and recall r on the PR curve are in the interval $[0, 1]$, *i.e.*, $p \in [0, 1]$ and $r \in [0, 1]$, we have $\sum_{i=1}^{n}|\Delta p_i| \leq 1$ and $\sum_{i=1}^{n}|\Delta r_i| \leq 1$. Thus, we obtain $\Omega^* \in (-\infty, 100]$. Note that the equal sign is achieved under two constraints: 1) when precision $p \neq 0$, recall $r = 0$, and recall $r \neq 0$, precision $p = 0$; and 2) for any precision p and recall r values of the input detection, the object tracking method can always achieve the MOTA score of 100. The two constraints are the idea cases that usually do not hold in real-world applications.