Vehicle Tracking Using Deep SORT with Low Confidence Track Filtering

Xinyu Hou, Yi Wang, and Lap-Pui Chau
School of Electrical and Electronic Engineering, Nanyang Technological University
50 Nanyang Ave, 639798, Singapore
{houx0008, wang1241}@e.ntu.edu.sg, elpchau@ntu.edu.sg

Abstract

Multi-object tracking (MOT) becomes an attractive topic due to its wide range of usability in video surveillance and traffic monitoring. Recent improvements on MOT has focused on tracking-by-detection manner. However, as a relatively complicated and integrated computer vision mission, state-of-the-art tracking-by-detection techniques are still suffering from issues such as a large number of false-positive tracks. To reduce the effect of unreliable detections on vehicle tracking, in this paper, we propose to incorporate a low confidence track filtering into the Simple Online and Realtime Tracking with a Deep association metric (Deep SORT) algorithm. We present a self-generated UA-DETRAC vehicle re-identification dataset which can be used to train the convolutional neural network of Deep SORT for data association. We evaluate our proposed tracker on UA-DETRAC test dataset. Experimental results show that the proposed method can improve the original Deep SORT algorithm with a significant margin. Our tracker outperforms the state-of-the-art online trackers and is comparable with batch-mode trackers.

1. Introduction

The advances of machine learning bring new possibilities to traditional multi-object tracking problem and introduces novel tracking-by-detection approaches. Specifically, tracking-by-detection algorithms consist of two parts: i.e., an object detection algorithm which gives detection results, usually in the form of bounding box coordinates, in every frame; and, a data association algorithm to determine whether the newly detected object can be associated with the estimated position of existing tracks. The association can be done by spatial information [5, 4] or by both spatial and appearance information [6, 20, 14, 1].

Tracking with only spatial data association is considered as the baseline of tracking-by-detection approach, where the input of tracker is the output of detector. In IOU tracker proposed by E. Bochinski *et al*. in [5], detection results from consequent frames whose intersection-over-union (IOU) is larger than a threshold are associated as a track using greedy algorithm. In [4] by A. Bewley *et al*., Kalman Filter is used to estimate the location of the tracked object from last frame. The Kalman Filter is an algorithm that is able to use measurements from detections and previous states of tracks that contain uncertainty to estimate the current states. The new detection results are assigned to the estimated tracks using Hungarian algorithm [13]. This tracker can achieve real time speed and is called the Simple Online Realtime Tracker (SORT).

Adding appearance information in data association can significantly increase the robustness of the tracker. E. Bochinski *et al*. [6] extends the IOU tracker to V-IOU tracker by adding visual single-object tracking, which helps IOU tracker to deal with missing detections and reduces the number of ID switches and fragmentations. The visual tracker is initialized when there is no detections to associate with current track and stopped when a new detection satisfies the IOU threshold within *ttl* frames, where *ttl* is a predefined threshold. Simple Online and Realtime Tracking with a Deep association metric (Deep SORT) from [20] by N. Wojke *et al*. is an extension of SORT. The appearance of new detections is compared with that of previous tracked objects in each tracks to help the data association problem. The appearance information is extracted by a convolutional neural network (CNN) trained on re-identification dataset and compared by cosine similarity metric [19]. We will discuss this method in details in Section 2 as it is the baseline of our proposed method. C. Long *et al*. [14] propose a hierarchical data association, which includes spatial information and appearance representation learned from re-identification dataset. The hierarchical data association is applied to detections and tracks that successfully passed a candidates selection. In [1], a multi-steps data association is proposed, which contains a short-term local association using spatial distance, a global data association with appearance model and a occlusion handling in tracklet level.

978-1-5386-9294-3/18/$31.00 ©2019 IEEE

We may notice that in all the above methods, a detection confidence threshold is used to filter out all detections with low confidence. This is based on the assumption that the detections with confidence lower than the threshold are very likely to be false positives and those with confidence higher than threshold should be true positives. However, the state-of-the-art detection methods have not yet perfectly follow this assumption. So, in this paper, we propose a low confidence track filtering extension in Deep SORT, which checks the average detection confidence of each track within the first several frames after their initialization. Tracks with low average confidence are filtered out to reduce false positive tracks. Average confidence can prevent missing out true positive tracks with few low-confidence detections caused by occlusion or noisy environment. Meanwhile, false positive tracks with relatively high-confidence wrong detections are more likely to be dropped. We evaluate this proposed method on vehicle tracking mission using UA-DETRAC [17] dataset. Our main distributions in this project are:

- Generate a vehicle re-identification dataset from UA-DETRAC training dataset, containing 122234 images and 1809 identities and train the cosine metric learning model [19] on it
- Incorporate a low confidence track filtering extension to Deep SORT tracking algorithm and evaluate it on UA-DETRAC testing dataset

This paper is organized as follow: in Section 2, the detailed methodology of our proposed tracking method is introduced, including the baseline Deep SORT algorithm and our low confidence track filtering extension. The proposed re-identification dataset are described in Section 3. In Section 4, the experiments and results are shown. Lastly, we conclude this paper in Section 5.

2. Proposed Method

Our proposed Deep SORT with Low Confidence track Filtering (DSLCF) tracking method is an extension of Deep SORT tracking method in [20]. The motivation of this extension is to allow the tracking algorithm to better deal with unreliable detection results such as low confidence true positives and high confidence false positives. These unreliable detections are very likely to occur in real-world object detection due to complex environment. With this extension, the number of false positive tracks will be greatly reduced.

2.1. Baseline Deep SORT Algorithm

Track Estimation The Deep SORT algorithm uses the Kalman filter to estimate existing tracks in current frame. The state used in the Kalman filter is represented as $(u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$, where (u, v, γ, h) is the bounding box position and $(\dot{x}, \dot{y}, \dot{\gamma}, \dot{h})$ is the velocity of each coordinates. The Deep SORT employs a simple and standard version of Kalman filter, which uses constant velocity and linear observation. In this way, the position of every existing track is estimated based on its previous locations when every new frame comes. The track estimation uses spatial information only.

Appearance Descriptor To obtain the appearance information of detections and tracks, an appearance descriptor is used to extract features from detection images and track images from previous frames. The appearance descriptor is a CNN trained on large-scale re-identification dataset. It is able to extract features in a way that features from the same identity are close together and features from different identities are far away from each other in the feature space.

Data Association With the estimated position of the existing tracks and the appearance descriptor, we can now associate new detection results to the existing tracks in each coming frame. A detection confidence threshold t_d is used to filter out all the detections with confidence lower than the threshold. New detections have to pass this threshold to be candidates of data association. The Deep SORT algorithm uses a cost matrix to represent the spatial and appearance similarities between each new detections and existing tracks It is integrated by two distance values. The first distance is shown in Eq. (1) representing the spatial information:

$$d^{(1)}(i,j) = (\mathbf{d}_j - \mathbf{y}_i)^T \mathbf{S}_i^{-1} (\mathbf{d}_j - \mathbf{y}_i) \tag{1}$$

where $(\mathbf{y}_i, \mathbf{S}_i)$ are the projection of i-th track in measurement space and $\mathbf{d}_j$ is the j-th new detection. This is the Mahalanobis distance between j-th new detection and estimated position of i-th track. The second distance is shown in Eq. (2) representing the appearance information:

$$d^{(2)}(i,j) = \min\left(1 - \mathbf{r}_j^T \mathbf{r}_k^{(i)} | \mathbf{r}_k^{(i)} \in R_i\right) \tag{2}$$

where $\mathbf{r}$ is the appearance descriptor and R_i is the appearances of the last 100 object associated with the i-th track. Each distance is accompanied with a gate function $b_{i,j}^{(1)}$ and $b_{i,j}^{(2)}$ which are equal to 1 if the distance is smaller than predefined threshold and 0 otherwise. The integrated cost matrix is shown in Eq. (3):

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{3}$$

With a gate matrix $b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)}$ which equals to 1 only when both spatial and appearance gate function are equal to 1 and otherwise 0, indicating whether (i, j) is a valid match for both spatial and appearance. In each new frame, the new detections are associated with existing tracks using this cost matrix and gate matrix.

Track Handling Every time a new detection is successfully associated with an existing track, the detection is

added to the track and the unassociated age of the track is zero. When new detections fail to associate with existing tracks in frame f, the new detections are initialized as *Tentative* tracks. The original Deep SORT algorithm checks that the *Tentative* tracks are associated with new detections in each of the $(f+1)$, $(f+2)$, ... $(f+t_{tentative})$ frames. If successfully associated, the track is updated as *Confirmed* track. Otherwise, the *Tentative* track is deleted immediately. As for the existing tracks that fail to associate with new detections in each frame, their unassociated ages will increase by one. If the unassociated age exceeds the max age threshold, the track will also be deleted.

2.2. Low Confidence Track Filtering

False positive tracks caused by unreliable detection results are badly affecting the performance of trackers. In order to better address this issue, we propose a low confidence track filtering extension to baseline Deep SORT tracker. In our extension, apart from the detection confidence threshold t_d, we calculate the average detection confidence of the new detections associated to the *Tentative* track in $(f+1)$, $(f+2)$, ... $(f+t_{tentative})$ frames. Only if the average detection confidence is larger than our predefined average detection confidence threshold t_{ave_d}, the *Tentative* track can be updated to *Confirmed* track. Otherwise, the *Tentative* track is deleted. Detailed algorithm of our extended low confidence track filtering for *Tentative* tracks is shown in Algorithm.1. In this way, the detections result are not simply filtered by only a t_d threshold but by two stages of filtering with both t_d and t_{ave_d}. Thus, t_d can be set to lower value to avoid missing detections and t_{ave_d} can help to suppress the false positive tracks generated by low t_d.

Algorithm 1 Low Confidence Track Filtering

Input: Tentative tracks T_t; Tentative threshold $t_{tentative}$; Average detection confidence threshold t_{ave_d}; Associated detection confidence p_t.
Output: Confirmed tracks T_c; Deleted tracks T_d

for sequential frames **do**
  for $t \in T_t$ **do**
    if t is new in T_t **then**
      $hits = 0$
      $total\_prob = 0$
    $hits = hits + 1$
    $total\_prob = total\_prob + p_t$
    if $hits \geq t_{tentative}$ **then**
      if $\frac{total\_prob}{hits} < t_{ave_d}$ **then**
        $T_d = T_d \cup t$ and $T_t = T_t \setminus t$
      else
        $T_c = T_c \cup t$ and $T_t = T_t \setminus t$

3. Proposed Re-Identification Dataset

The original Deep SORT is used for person tracking mission. To apply it on vehicle tracking problem, we first need a large scale vehicle re-identification dataset to train our appearance descriptor. As there are few vehicle re-identification datasets available, we use the identity information provided in UA-DETRAC annotations to generate a new vehicle re-identification dataset. We discard those boxes with either truncation ratio over 0.5 or occlusion ratio over 0.5 as they may contain noise. In addition, we discard those identities with total number of occurrence

Figure 1: Images from the self-generated UA-DETRAC re-identification dataset. Images in the same row is from the same identity.

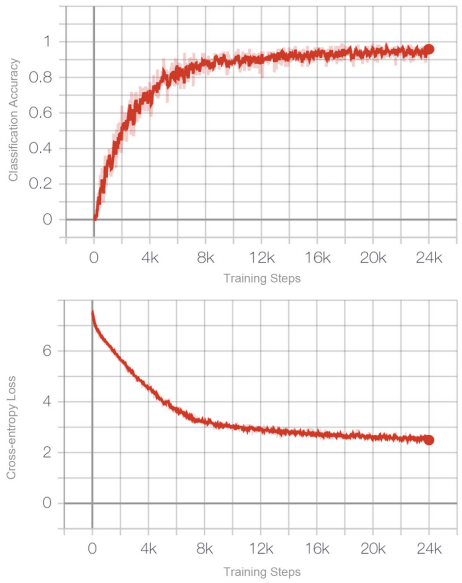

Figure 2: Curve of classification accuracy and cross entropy loss during training. Top: classification accuracy; Bottom: cross entropy loss.

less than 100 as we assume their positions do not vary enough in less than 100 frames, which makes them unhelpful for appearance descriptor training. For those identities with occurrence more than 100, we select roughly 100 frames with same interval in between for each identity. Images of vehicles are cropped from the surveillance images using the bounding box information and resized to the same size $(100, 100)$. As a result, there are in total 1809 identities and 122234 images in this vehicle re-identification dataset. The images in this dataset are named by Market1501 person re-identification dataset [21] format. Use `00892_c40752s1_01530_01.jpg` as an example. This is an image of identity 892 from the 1530-th frame of UA-DETRAC's `MVI_40752` training sequence. The `s1` and `01` are not applicable thus unchanged. Examples of images in the re-identification dataset are illustrated in Fig.1

4. Experiments

4.1. Vehicle Appearance Descriptor

Having obtained the vehicle re-identification dataset, we can train the CNN on it to obtain a vehicle appearance descriptor. The CNN structure we use is the same as the one in [19], which contains two convolutional layers followed by a max pooling layer and six residual blocks. l_2 normalization is used. We train the CNN on Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz for 24k steps. The classification accuracy is 0.9584 and the cross entropy loss is 2.492 upon the end of training. The curve of classification accuracy and cross entropy loss is shown in Fig.2.

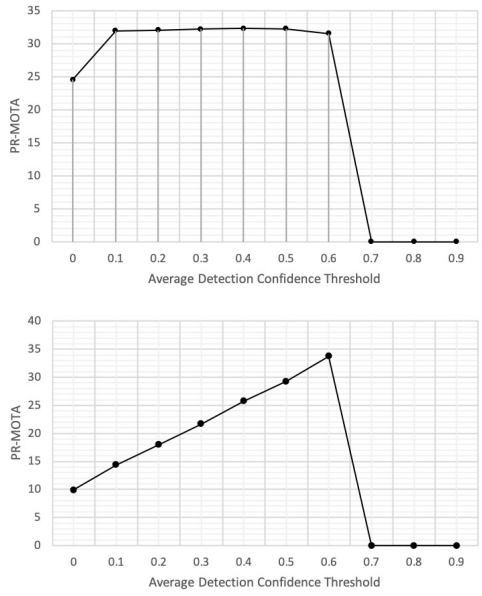

Figure 3: Comparison of MOTA value of 6 training sequences tracking results under different average detection confidence threshold t_{ave_d}. Top: EB [16] detection; Bottom: Mask R-CNN [12] detection.

4.2. Average Detection Confidence Threshold Selection

The average detection confidence threshold t_{ave_d} is selected by experiments. We select 6 sequences (`MVI_39851`, `MVI_39861`, `MVI_40752`, `MVI_40871`, `MVI_41063`, `MVI_63544`) from the UA-DETRAC training dataset which are filmed in relatively more complex environment such as raining, night and heavy traffic. We test $t_{ave_d} = 0.0 \sim 0.9$ on these 6 sequences. Both EB [16] and Mask R-CNN [12] detection methods, which are two state-of-the-art object detection methods we used as the detection input for our tracker, are tested separately. The final tracking results are evaluated using DETRAC MOT evaluation method. The comparison of the results is shown in Fig.3.

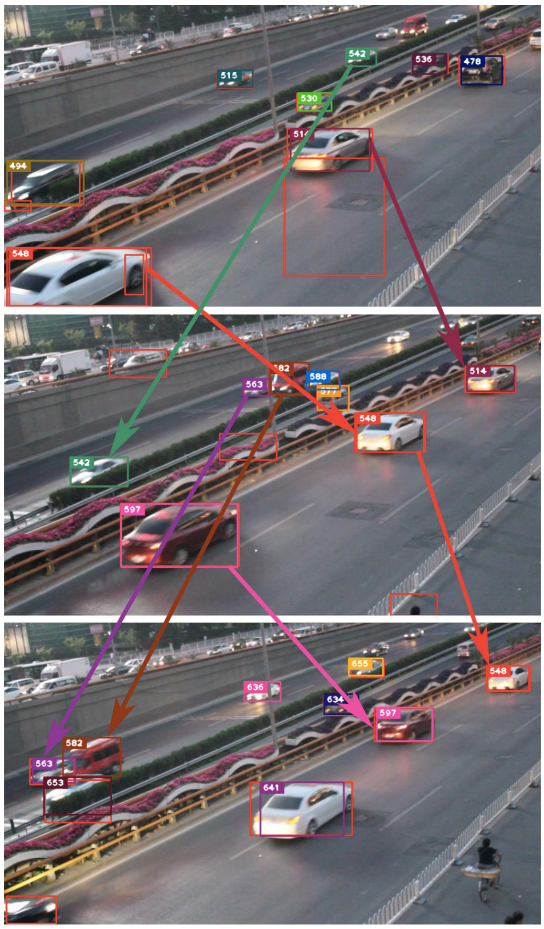

Figure 4: The tracking results on UA-DETRAC training sequence "MVI_40752" using EB detections. $t_d = 0.0$ and $t_{ave_d} = 0.4$.

Tracker	Detector	Method	PR-MOTA	PR-MOTP	PR-MT	PR-ML	PR-IDs	PR-FM	PR-FP	PR-FN
IHTLS[9]	CompACT [7]	Batch	11.1%	36.8%	13.8%	19.9%	953.6	3556.9	53922.3	180422.3
DCT[2]	R-CNN [11]	Batch	11.7%	38.0%	10.1%	22.8%	758.7	742.9	336561.2	210855.6
H^2T[18]	CompACT [7]	Batch	12.4%	35.7%	14.8%	19.4%	852.2	1117.2	51765.7	173899.8
GOG[15]	CompACT [7]	Batch	14.2%	37.0%	13.9%	19.9%	3334.6	3172.4	32092.9	180183.8
IOU[5]	R-CNN [11]	Batch	16.0%	**38.3%**	13.8%	20.7%	5029.4	5795.7	22535.1	193041.9
IOU[5]	CompACT [7]	Batch	16.1%	37.0%	14.8 %	19.7%	2308.1	3250.4	24349.4	176752.8
V-IOU[6]	CompACT [7]	Batch	17.7%	36.4%	17.4%	18.8%	363.8	1123.5	26413.3	166571.7
IOU[5]	EB[16]	Batch	19.4%	28.9%	17.7%	18.4%	2311.3	2445.9	**14796.5**	171806.8
K-IOU[8]	EB[16]	Batch	21.1%	28.6%	21.9%	**17.6%**	462.2	721.1	19046.8	**159178.3**
IOU[5]	Mask R-CNN [12]	Batch	**30.7%**	37.0%	30.3%	21.5%	668.0	733.6	17370.3	179505.9
V-IOU[6]	Mask R-CNN [12]	Batch	**30.7%**	37.0%	**32.0%**	22.6%	**162.6**	**286.2**	18046.2	179191.2
CMOT[3]	CompACT [7]	Online	12.6%	36.1%	16.1%	18.6%	**285.3**	1516.8	57885.9	167110.8
Deep SORT	EB	Online	16.8%	**41.4%**	18.0%	**18.2%**	548.7	2036.4	35059.5	169994.3
DSLCF (*ours*)	EB	Online	19.1%	**41.4%**	17.9%	18.3%	523.7	2004.4	**20049.0**	170374.1
Deep SORT	Mask R-CNN	Online	25.4%	36.1%	**32.4%**	19.9%	783.8	1508.4	63584.2	**166384.9**
DSLCF (*ours*)	Mask R-CNN	Online	**30.3%**	36.3%	30.2%	21.0%	388.8	**1260.4**	20263.4	179317.5

Table 1: Comparison of our experiment results with other method results on UA-DETRAC overall testing dataset.

As we can see in Fig.3, the MOTA when t_{ave_d} exists (not equal to 0) is better than when there is no t_{ave_d} (equals to 0). The tracking performances on EB are similar when t_{ave_d} is set to $0.1 \sim 0.6$. Thus either one of them can be selected as our final t_{ave_d}. We choose to use $t_{ave_d} = 0.3$ for experiments of our tracker with EB detection method. As for Mask R-CNN method, the tracking accuracy keeps increasing until $t_{ave_d} = 0.6$. Thus we choose $t_{ave_d} = 0.6$ for experiments of our tracker on Mask R-CNN detection results. Note that for both detection methods, when t_{ave_d} is larger than 0.6, the tracker does not work because all tracks are filtered out.

4.3. Comparison with State-of-the-arts

We test our proposed method on the overall testing dataset of UA-DETRAC, which contains 40 sequences. As mentioned in previous section, we use EB [16] and Mask R-CNN [12] detection results as our detection input. We select these two detectors because they are two baseline detection methods used by other trackers and they give relatively good performance. The performance of our tracker and the comparison with other state-of-the-art trackers are shown in Table.1. We classified all the compared trackers to Batch or Online classes. Batch tracker means the tracker uses both previous and future information to generate tracks in current frame and online tracker means the tracker only uses previous information to generate tracks. The meaning of standard multi-object tracking evaluation metrics mentioned in Table.1 are: multi-object tracking accuracy (PR-MOTA↑), multi-object tracking precision (PR-MOTP↑), mostly tracked (PR-MT↑), mostly lost (PR-ML↓), identity switches (PR-IDs↓), fragmentation (PR-FM↓),false positive (PR-FP↓) and false negative (PR-FN↓) [17]. The upper arrow ↑ means that larger value of this metric shows better performance. The down arrow ↓ means that smaller value of this metric shows better performance.

As we can see in Table.1, our proposed DSLCF tracker achieves PR-MOTA of 30.3% on Mask R-CNN detection, which is the highest among all online trackers and comparable with highest PR-MOTA of batched V-IOU tracker. It may be noted that the original Deep SORT algorithm with appearance descriptor trained on our re-identification dataset can already achieve good PR-MOTA on Mask R-CNN detection, which is 25.4%. With low confidence track filtering, our tracker can still improve the PR-MOTA of the origianl tracker by around 6%. Our proposed DSLCF method also achieves highest 41.4% PR-MOTP on EB detection. In addition, the PR-FP can be significantly reduced using our extension as we can observe 15010.5 PR-FP drop on EB and 43320.8 drop on Mask R-CNN between Deep SORT and DSLCF. Meanwhile, the IDs of DSLCF on Mask R-CNN is reduced to nearly half of the Deep SORT IDs.

An illustration of the tracking results generated by our proposed tracker in UA-DETRAC training dataset is shown in Fig.4. We can see from the images that there are a lot of red boxes with no identity labels. These red boxes are the false positive detections that are filtered out by our low confidence track filtering algorithm. As $t_d = 0.0$, there are a lot of red boxes appearing but the final tracking result will not be affected.

5. Conclusion

In this paper, we propose a low confidence track filtering extension on Deep SORT tracking algorithm, which can significantly reduce false positive tracks generated by Deep SORT. Tracks with low average detection confidence in their initial several frames will be deleted. In this way, the detection confidence can be set to lower value and even

zero to avoid missing detections. We also generate a vehicle re-identification dataset from UA-DETRAC dataset to train the Deep SORT for vehicle data association. Experiments on UA-DETRAC test dataset shows that our proposed extension can achieve promising results by notable margins against state-of-the-art trackers.

Acknowledgement

We wish to acknowledge the funding for this project from Nanyang Technological University under the Undergraduate Research Experience on CAmpus (URECA) programme.

References

[1] N. M. Al-Shakarji, F. Bunyak, G. Seetharaman, and K. Palaniappan. Multi-object tracking cascade with multi-step data association and occlusion handling. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018. 1

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 5

[3] S. Bae and K. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1218–1225, June 2014. 5

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 1

[5] E. Bochinski, V. Eiselein, and T. Sikora. High-speed tracking-by-detection without using image information. In *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*, Lecce, Italy, Aug. 2017. 1, 5

[6] E. Bochinski, T. Senst, and T. Sikora. Extending iou based multi-object tracking by visual information. In *IEEE International Conference on Advanced Video and Signals-based Surveillance*, pages 441–446, Auckland, New Zealand, Nov. 2018. 1, 5

[7] Z. Cai, M. J. Saberian, and N. Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. *CoRR*, abs/1507.05348, 2015. 5

[8] S. Chen and C. Shao. Python implementation of the kalman-iou tracker. https://github.com/siyuanc2/kiout. Accessed August 6, 2019. 5

[9] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *2013 IEEE International Conference on Computer Vision*, pages 2304–2311, Dec 2013. 5

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sep. 2010.

[11] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. 5

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 4, 5

[13] H. W. Kuhn. The hungarian method for the assignment problem. In *50 Years of Integer Programming*, 2010. 1

[14] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, 2018. 1

[15] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208, June 2011. 5

[16] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue. Evolving boxes for fast vehicle detection. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1135–1140, 2017. 4, 5

[17] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136, 2015. 2, 5

[18] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1282–1289, June 2014. 5

[19] N. Wojke and A. Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018. 1, 2, 4

[20] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. 1, 2

[21] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, Dec 2015. 4