Real-time Multiple Object Tracking with Deep SORT and CenterNet

Zhengke Wu, Jiabin Fang, Tianxiao Shen

***Abstract*—This is the report of the course project of EI339 Artificial Intelligence, concerning the problem of multiple object tracking. Simple Online and Real-time Tracking (SORT) is a pragmatic approach to multiple object tracking with a focus on simple, effective algorithms. Deep SORT integrates appearance information to improve the performance of SORT. As Detection-Based Tracking methods, they depend on good object detectors. CenterNet is a fast and powerful recently proposed approach, reaching state-of-the-art in real-time objection detection. In this paper, we studied and re-implemented the Deep SORT method, with the integration of CenterNet, and proposed some improvements to it. Also, our code has been made open source on Github.** [1]

I. Introduction

This is the report of the course project of EI339 Artificial Intelligence. As for the three evaluation levels, we think our work can be classified into **Excellent**. The reasons are:

- We have carefully studied and analyzed SORT [1] and deep SORT [2], reading both the papers and code.
- We have done a comprehensive investigation into currently popular object detection approaches including Faster R-CNN [3], SSD [4], YOLOv3 [5], CornerNet [6] and CenterNet [7].
- We have integrated deep SORT and CenterNet, which is currently the state-of-the-art approach in **real-time** object detection, achieving some engineering success with proper and good use of existing open source code.
- We have obtained some notable real-time performance improvements.

Multiple Object Tracking (MOT) plays an important role in solving many fundamental problems in video analysis and computer vision. Its main task is to find and identify moving objects in a sequence of images. Depending on how objects are initialized, the strategies of MOT can be classified into two sets: Detection-Based Tracking (DBT) and Detection-Free Tracking (DFT), of which the first one is the leading paradigm in this field of research. The DBT methods employ two steps: Objection Detection and Data Association, i.e., detecting objects of interest in each frame of a video first and then obtain the tracks of the detected objects across frames according to their correspondence. In this way, the MOT problem can be viewed as a data association problem.

MOT can also be categorized into online tracking and offline tracking. The difference is whether or not observations from future frames are utilized when handling the current frame. Online algorithms generally performs worse than offline algorithms, which is intuitive and natural. However, it is essential in many real-time scenarios. Therefore, in this project, our focus lies in online tracking.

Deep SORT [1] is an extension to Simple Real time Tracker (SORT) [8], and is one of the most popular and the most widely used object tracking frameworks. It achieved competitive performance with simple and elegant ideas, and can act as a baseline in the field of MOT. In this project, we study and re-implement the classical Deep SORT algorithm and achieve some improvements based on the Deep SORT framework.

[1] https://github.com/keithnull/ShallowSORT

II. Background

A. Deep SORT

Simple online and real-time tracking (SORT) is a very simple framework that performs Kalman filtering in image space and frame-by-frame data association using the Hungarian method with an association metric that measures bounding box overlap. This simple approach achieved favorable performance at high frame rates. It used state of the art convolutional neural network (CNN) based object detector Faster R-CNN [3].

To be more specific, Hungarian algorithm [9] is a very classic algorithm to find a maximum cardinality matching of minimum cost in the bipartite graph matching problem. And Kalman filter [10] is an algorithm that uses a series of measurements observed over time to predict the motion of an object. SORT approximates the inter-frame displacements of each object with a linear constant velocity model which is independent of other objects and camera motion. The state of each target is modelled as:

$$\boldsymbol{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T,$$

where u and v represent the horizontal and vertical pixel location of the centre of the target, while the scale s and r represent the scale (area) and the aspect ratio of the target's bounding box respectively.

Now the idea behind SORT seems rather simple, but it did make some contributions to the development of MOT at that time. Its code was also made open source, and provided a new baseline for the field of MOT.

Deep SORT took another step forward. The main contribution of Deep SORT beyond the original framework of SORT is a deep association metric learned on a large scale person re-identification data-set. Besides, it integrates appearance information to recover identities after long-term occlusions, when motion is less discriminate. It extensions reduce the

TABLE I
TOP RANKED REAL-TIME OBJECT DETECTOR ON COCO 2017

Rank	Method	FPS	MAP	Paper	Year
1	CenterNet ResNet-18	142	28.1	Objects as Points	2019
2	TTFNet	54.4	35.1	Training-Time-Friendly Network for Real-Time Object Detection	2019
3	CenterNet DLA-34	52	37.4	Objects as Points	2019
4	CenterMaskLite-MobileNetV2	50	28.8	CenterMask : Real-Time Anchor-Free Instance Segmentation	2019
5	YOLOv3-320	45	28.2	YOLOv3: An Incremental Improvement	2018
6	SSD512-HarDNet85	39	35.1	HarDNet: A Low Memory Traffic Network	2019
7	YOLOv3-418	34	31.0	YOLOv3: An Incremental Improvement	2018
8	CornerNet-Squeeze	33	34.4	CornerNet-Lite: Efficient Keypoint Based Object Detection	2019
9	RefineDet320 + VoVNet-57	21.2	33.9	An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection	2019
10	YOLOv3-608	20	33	YOLOv3: An Incremental Improvement	2018

number of identity switches by 45%, which improves the defects of SORT, achieving overall competitive performance at high frame rates.

We have to stress that Detection-Based Tracking algorithms like SORT and Deep SORT rely highly on the results of object detection, and appealing to more powerful detector or one trained in a more targeted way can easily provide better tracking results, as the authors of SORT admitted.

B. CenterNet

As have been stated at the end of previous section, object detection plays a vital role in deep SORT. On one hand, the quality and accuracy of object detection affects the final quality of object tracking; on the other hand, the real-time performance of object tracking largely relies on the speed of object detection. Therefore, to achieve real-time improvements with deep SORT, we focus on the real-time performance of object detection.

After some investigation, we compare the performance of a few currently popular real-time object detectors. According to PapersWithCode's data [2], on COCO 2017 data-set, top ranked real-time object detectors are listed in Table I. Eventually, we decide to adopt CenterNet [7] in our implementation.

Within the scope of our knowledge, CenterNet is currently the state-of-the-art real-time object detector. It treats each object as a single point and then uses keypoint estimation to get all its other properties like bounding box. In such way, it achieves a good trade-off between accuracy and speed. For example, on COCO 2017 data-set, its maximal speed is 142 FPS with 29.1% AP, or 52 FPS with 37.4% AP.

III. EXPERIMENTS

We evaluate the performance of our tracking implementation on a diverse set of testing sequences as set by the MOT16 data-set [11] which contains both moving and static camera sequences. To show the advantage of CenterNet, we have also tried YOLOv3 detector to provide a comparison.

[2]https://paperswithcode.com/sota/real-time-object-detection-on-coco

A. Metrics

Since it is difficult to use one single score to evaluate multi-target tracking performance, we utilise the standard MOT metrics [12].

B. Performance Evaluation

Tracking performance is evaluated using the MOT16 benchmark [11] where the ground truth for 7 training sequences is withheld. We just follow the evaluation method described in py-motmetrics. Table II is the results we get by using different detectors with different backbones. From the table we have found there is a certain gap between the benchmark and our project with the main indices of accuracy, MOTA and MOTP. As for the YOLOv3, since the model we get has been trained with the original MOT data, the accuracy is a little better than our model, which has not been trained specifically for this data-set. For the two different backbones, the accuracy of DLA-34 is better than the accuracy of Resnet-18, which is expected. From this we can find that choosing a better detector is a way to improve tracking results. However, since the training and the parameter adjustments cost much time, we didn't dig more in this aspect.

C. Runtime

For a 1920×1080 video, the program runs in 3.6 FPS with CenterNet DLA-34, 9.1 FPS with CenterNet Resnet-18 and 3.0 FPS with YOLOv3, on average. The reason why the FPS is not so high, we think, is that our equipment is not so good, with only 4GB graphics memory. But by the comparison with YOLOv3 detector, we have found that the CenterNet does

TABLE II
EVALUATION RESULTS OF DIFFERENT DETECTORS WITH DIFFERENT BACKBONES

	MOTA	MOTP	IDF1	IDP	IDR	Rcll	Prcn	MT	PT	ML	FP	FN	IDs	FM
Benchmark	51.60%	0.189	60.20%	64.40%	56.60%	70.10%	79.90%	214	235	68	19490	32985	935	1439
YOLOv3	33.10%	0.235	41.80%	65.90%	30.60%	40.00%	86.00%	92	208	217	7212	66270	427	1269
CenterNet Resnet-18	23.90%	0.247	34.50%	49.90%	26.30%	38.60%	73.10%	89	223	205	15663	67817	584	1707
CenterNet DLA-34	29.50%	0.217	44.30%	56.20%	36.50%	47.60%	73.20%	105	251	161	19180	57905	753	1676

perform better in speed, which can possibly reach the goal of real-time with a more powerful machine.

IV. Conclusion

In this project, we dove into the field of Multiple Object Tracking (MOT). It is both interesting and challenging. Our work is based on the credible Deep SORT method and currently state-of-the-art objection detection approach CenterNet. We made a step towards real-time multiple object tracking from an engineering prospective.

References

[1] 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)*. IEEE, 2017, pp. 3645–3649.

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

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

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[6] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[7] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," in *arXiv preprint arXiv:1904.07850*, 2019.

[8] 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)*, 2016, pp. 3464–3468.

[9] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[11] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[12] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.