Learning for Graph Matching based Multi-object Tracking in Auto Driving

Yihao Yin\textsuperscript{1*}, Xin Feng\textsuperscript{1} and Haoming Wu\textsuperscript{1}

\textsuperscript{1}College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China
\textsuperscript{*}Corresponding author’s e-mail: bleach@2018.cqut.edu.cn

Abstract. Multi-object tracking in autonomous driving aims to represent trajectories of moving objects for planning system of the vehicle. In this paper, we propose a new tracking-by-detection scheme based on deep neural networks for multi-object detection and tracking in autonomous driving scene. We first introduce a light-weight neural network branch for fast object detection, and based on the detection results on each frame, we build two object graphs for consecutive frames separately, where the vertices of the graph represent the objects in the image, and the edges of the graph represent the spatial relations between the objects. We then formulate the multi-object tracking problem as the graph matching process by learning the relevance between objects from another object association network branch. Experiments results on the MOT multi-object tracking dataset show that the proposed object detection and tracking approach achieves comparable results with state-of-the-art deep learning based multi-object tracking methods, and outperforms them in tracking efficiency, which ensures real-time multi-object tracking for autonomous driving.

1. Introduction
Visual perception is the vision system of autonomous driving, where multi-object detection and tracking are the essential modules in perceiving the object instances, such as pedestrians and cars, and their moving trajectories in the driving scenario. Following the object detection results in current frame, the task of multi-object tracking (MOT) is to estimate the location of the object in the consecutive frames. The derived trajectories that contain the orientation and speed of the moving objects are the key inputs of collision distance forecast and planning system of the vehicle.

Due to the high detection accuracy of the current object detection, therefore, tracking-by-detection based multi-object tracking method has become a research hotspot in the field of multi-object tracking. In the “tracking-by-detection” scheme, object detection is the first process to generate object localization result for each frame, then to associate objects represented in bounding boxes across frames. DeepSort\textsuperscript{[1]} uses Faster R-CNN\textsuperscript{[2]} as the object detection algorithm and combined with Kalman filtering to predict the bounding box of object, uses an appearance characteristics extraction model to extract the apparent feature of object, through the appearance feature similarity between objects and cross union ratio of between two objects to achieve MOT. DAN\textsuperscript{[3]} is an end-to-end tracking network which uses VGG-16\textsuperscript{[4]} to extract spatial feature of the detected object. Without using Hungarian algorithm for data association, it proposes to learn the matching matrix between objects of consecutive frames by convolutional neural networks. JDE\textsuperscript{[5]} adds the feature extraction branch to the object detection network YOLOv3\textsuperscript{[6][7]} and feature maps are shared for the multi-tasks of object detection and object feature extraction.
In this paper, we first propose a light-weight deep CNN based object detection network for fast object localization in autonomous driving. Before object association, we build another simple object feature extraction network to extract object appearance features. Based on the detected object bounding boxes, we define an object graph-based frame descriptor, where the detected objects in each frame are regarded as nodes, and the connections between two objects in the same frame are taken as edges. Based on the object graph definition, we then have two object graphs for consecutive frames. Naturally, we can formulate the object associate problem to a graph matching. The scheme of multi-object tracking framework proposed in this paper is shown in Figure 1.

2. Method

2.1. Object Detection

Compared with general object detection, the scenario of autonomous driving is much more complex, and the objects such as pedestrians and cars, are always with diversity of posture and appearance features. Furthermore, small pedestrian objects and variance weather conditions brings additional difficulties to visual object detection and tracking task. Thereupon, general object detection algorithm is not good for detection, we then design a special lightweight one-stage pedestrian object detection model, which is shown in Figure 2, to detect object bounding boxes on multi-scale feature maps. In order to reduce information loss during down-sampling and enrich object feature information, two multi-scale down-sampling modules are designed when performing down-sampling operations, so that the features contain as much information as possible after down-sampling. After the last down-sampling, spatial pyramid pooling[8] is added to extract the feature information of different receptive fields for prediction. Compared with the detection algorithm used in most tracking papers, our detection model can achieve real-time detection speed and almost the same accuracy.
2.2. **Object Feature Extraction Network**

The feature map of detection task also can be used to achieve classification task due to the similarity of features. At this point, the backbone of our light weight detection network can be used as backbone of feature extraction network. Our object feature extraction network is shown in Figure 3, for the input of the feature extraction network is a whole image that contains multiple objects, the feature vector of where the object’s relative center point coordinates are located is regarded as the appearance feature of object. Finally, it of great importance to perform dimensionality reduction through the $1 \times 1$ convolution according to the importance of feature layers and to extract object features from different feature layers to get more robust appearance features.

![Figure 3. Feature extraction network.](image)

2.3. **Learning for Object Graph Matching**

The spatial information and appearance features between the objects are full used to construct the object graph, each node contains the object appearance feature, each edge between two frames contains the direction vector between two objects, each edge in the same frame contains the direction vector between two objects. In order to get the object matching matrix, the nodes and edges are matched in the object graph structure separately.

![Figure 4. The multi object matching process on hyper object graph.](image)

The matching matrix is obtained from object graph shown in Figure 4. The objects in two frames are combined in pairs according to the node and edge information to obtain the node feature similarity matrix and the edge feature similarity matrix. In Figure 4, $SF_{t1,t2}^N$ represents the pairwise combination of node features between two frames, $SF_{t1,t2}^E$ represents the pairwise combination of related edge features of node between two frames, $S_{t1,t2}$ represents the spatial similarity matrix by object displacement difference indicated by red edge. $S_{t1,t2}$ represents the object feature similarity matrix from the addition of node similarity matrix $S_{t1,t2}^N$ and edge similarity matrix $S_{t1,t2}^E$. $M_{t1,t2}$ represents the matching matrix. Node feature similarity matrix and edge feature similarity matrix can directly learn node and edge similarity through matching network, in which the 5 convolutions are used due to
the matching network only calculates the similarity between two objects. Through the first 4
convolutions, the input node similarity matrix and edge similarity matrix channel are halved each time
to make the feature better mapped. The last convolution is used to convert the similarity matrix into
the form of a matching matrix. The edge similarity matrix and the node similarity matrix are added to
represent the object feature similarity matrix.

The spatial similarity matrix between the objects is a constraint on the results obtained by the
object feature similarity matrix. The object similarity matrix is obtained by the Hadamard product of
the spatial similarity matrix and the object feature similarity matrix, in order to prevent objects
between two frames with similar appearance features but spatially far away in space being wrongly
associated. The calculation method of the spatial similarity matrix is as the equation (1).

$$S_{ij} = \frac{e^{-d_{ij}} - e^{-1}}{1 - e^{-1}}$$

where $d_{ij}$ is the normalized distance between object $i$ and object $j$, the value is the ratio of the
Euclidean distance between objects to the diagonal distance of the image.

3. Experiment
Without loss of generality, the performance of the proposed tracking algorithm is verified in MOT17
test set. The MOT17 test set contains seven video sequences and some video sequences obtained by
moving camera. For this is a test set without labels, we need to submit our tracking results to MOT
challenge official website to get the evaluation metrics result of our tracking results. The main
evaluation metrics of MOT are multiple objects tracking accuracy (MOTA) which is made up with
redundant objects (FP), missed objects (FN) and identity switches (ID SW), and the mismatch degree
of label and predicted bounding box (MOTP), in the automatic driving scenario, the inference speed
metric Hz is also very important.

The training set for the detection model is a collection of pictures selected from the BDD100K
training set at a 1:1 ratio of positive and negative samples and all City Person training set. Before
training, the K-Means clustering on all training set labels is used to obtain 9 anchors. These 9 anchors
are divided into 3 groups according to the size of the bounding box and used to represent the
distribution of objects in the data set, which are used for bounding box regression in different layer.
The total training iterations are 400K, the batch size is 64, and the learning rate begins with 0.0026
and is divided by 10 at the 150K, 180K and 300K iterations. Except the source detection model SPD,
we also designed an object detection model without multi-scale down-sampling module SPD-N and a
more light weight object detection model with separable convolution SPD-lite. TABLE 1 shows the
detection results of the proposed detection model and the famous object detection YOLOv3 on test set
of BDD100K and City Person.

| Model      | Size (MB) | Memory (MB) | FLOPS (BFLOPS) | mAP (%) | Time (ms) |
|------------|-----------|-------------|----------------|---------|-----------|
| YOLOv3     | 237       | 52.43       | 65.88          | 49      | ~13       |
| YOLO Tiny  | 34        | 52.43       | 5.57           | 22      | ~3        |
| SPD(ours)  | 26        | 7.37        | 13.06          | 68      | ~5        |
| SPD-N(ours)| 21        | 7.37        | 10.23          | 54      | ~4.5      |
| SPD-lite(ours) | 6 | 1.84    | 4.47           | 64      | ~4        |

The training set for the feature extraction and matching Network is MOT17. The total training
process iterations are 80,000 steps, the batch size is 8, the learning rate begins with 0.01 and is divided
by 10 at the 33200, 53120 and 66400 iterations. The comparison results with the advanced multi-
object tracking algorithm in the MOT17 test set are shown in Table 2. ↑ means that the higher the
index, the better. ↓ means that the lower the index, the better. In Table 2, it can be seen that our
algorithm is superior to other advanced algorithms in five indicators, and in summary, our algorithm is also the best.

Figure 5 shows a comparison of the tracking results before and after adding spatial information, the figure above shows tracking result without considering spatial information, the figure below shows tracking result with spatial information taken into account. In the figure above, the object in the red area has an abnormal trajectory due to wrong matched object in the distance. In the figure above, because of the spatial relationship between the objects, the wrong matched phenomenon disappeared.

| Tracker       | Type       | MOTA↑ | MOTP↑ | IDF1↑ | MT↑  | ML↓  | FP↓  | FN↓  | ID_SW↓ | Hz↑  |
|---------------|------------|-------|-------|-------|------|------|------|------|--------|------|
| LSST17        | offline    | 54.7  | 75.9  | 62.3  | 20.4 | 40.1 | 26091| 228434| 1243   | 1.5  |
| GMOT          | offline    | 55.4  | 76.7  | 57.9  | 22.7 | 34.7 | 20608| 229511| 1403   | 5.9  |
| MHT DAM       | offline    | 50.7  | 77.5  | 47.2  | 20.8 | 36.9 | 22875| 252889| 2314   | 0.9  |
| PHD_GSDL17    | online     | 48.0  | 77.2  | 49.6  | 17.1 | 35.6 | 23199| 265954| 3998   | 6.7  |
| DeepMOT       | online     | 48.1  | 76.5  | 43.0  | 17.6 | 38.6 | 26490| 262578| 3696   | -    |
| Tracktor17    | online     | 53.5  | 78.0  | 52.3  | 19.5 | 36.6 | 12201| 248047| 2072   | 1.5  |
| Tracktor++v2  | online     | 56.3  | 78.8  | 55.1  | 21.1 | 35.3 | 8866 | 235449| 1987   | 1.5  |
| DAN           | online     | 52.4  | 76.9  | 49.5  | 21.4 | 30.7 | 25423| 234592| 8431   | 6.3  |
| Ours          | online     | 56.6  | 76.9  | 50.4  | 43.6 | 30.7 | 22824| 213267| 8994   | 9.0  |

4. Conclusion
In this paper, a light-weight multi-object detection model is first proposed to get the location of objects in real time. Based on the object detection results, we formulate the multi-object tracking problem into a hyper graph matching process. The object graph is mainly constructed by the object appearance features as nodes and the relative relationship between objects within a frame and across frames, which are defined as edges properties. In addition, the spatial matching matrix and the object matching matrix obtained from object graph structure are integrated to effectively prevent two different objects are matched due to similar appearance.
Acknowledgments
This work is supported by the Humanities and Social Sciences of Ministry of Education Planning Fund (Grant No. 17YJCZH043) and the Graduate Innovation Fund of Chongqing University of Technology (Grant No. ycx20192061).

References
[1] Wojke N, Bewley A, Paulus D, “Simple online and realtime tracking with a deep association metric,” IEEE International Conference on Image Processing (ICIP). 2017.
[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” In NIPS, 2015.
[3] D. Mitzel, E. Horbert, A. Ess, and B. Leibe, “Multi-person tracking with sparse detection and continuous segmentation,” In Proc. Eur. Conf. Comput. Vis., 2010, pp. 397–410.
[4] Simonyan, K. and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” CoRR abs/1409.1556 (2015): n. pag.
[5] Wang Z., Zheng L., Liu Y., Li Y., Wang S. (2020) Towards Real-Time Multi-Object Tracking. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020.
[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
[7] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” In CVPR.
[8] He K, Zhang X, Ren S, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.” IEEE Transactions on Pattern Analysis and Machine Intelligence 37.9 (2015): 1904.