Evaluation of 3D Object Detection Methods in Autonomous Driving using the KITTI Dataset

Anthony Shen

Received January 05, 2024
Accepted February 28, 2024
Electronic access March 15, 2024

A number of research papers have been published on 3D Object Detection methods of autonomous driving. However, these papers often describe approaches to specific problems or scenarios without providing comparison and evaluation of the existing methods and their limitations. The aim of the study is to compare and evaluate three methods for the 3D object detection benchmark in autonomous driving: Deep Learning and Geometry, Triangulation Learning Network and Monocular 3D Object Detection. Each of the three methods are evaluated from testing on the KITTI dataset and from examination of the techniques and algorithms used. The results demonstrate the reliability and accuracy of the Triangulation Learning Network, and it outperforms the other two methods. The study also provides a discussion of the benefits and drawbacks of the three novel approaches for 3D object detection.

Keywords: Evaluation, Autonomous Driving, Deep Learning, 3D Object Detection Methods

Backround & Objectives

3D object detection plays a crucial role for improving the performance and safety of autonomous driving systems. Compared to other benchmarks, such as 2D object detection or semantic segmentation, 3D object detection provides more accurate and robust information for high-level decision making, such as path planning, motion prediction, and collision avoidance. For example, 3D object detection can estimate the distance, size, orientation and shape of other vehicles, pedestrians, cyclists and obstacles, which is essential for collision avoidance and safe navigation. Without 3D object detection autonomous driving will not be sufficient and accidents will occur.

According to a study by Waymo, a leading company in self-driving technology, 3D object detection reduced the false positive rate of pedestrian detection by 75% compared to 2D object detection[1]. This is also shown in a report by KITTI, which shows that 3D object detection helps achieve an average precision of 96.9% for cars, 89.5% for pedestrians, and 88.4% for cyclists, which are much higher than the previous benchmarks[2]. This demonstrates its crucial role in autonomous driving.

The aim of the study is to compare and evaluate approaches for the 3D Object Detection benchwork in autonomous driving: Deep Learning and Geometry, Triangulation Learning Network and Monocular 3D Object Detection, and find the one that achieves the best precision and accuracy. It will evaluate the methods based on the techniques used in each program and the results from testing on the KITTI dataset.

Methods

Three Approaches

To test and evaluate the approaches to 3D object detection, there must be a general understanding for each method.

First, Deep Learning and Geometry presents an approach for 3D object detection and pose estimation from a single image. It obtains relatively accurate 3D object properties using a deep neural network and then combines the approximations with constraints provided by a 2D bounding box to form a 3D bounding box[3]. The first network output approximates the 3D object orientation, and the second network obtains the 3D object dimensions. These estimates, integrated with the constraints provided by the 2D bounding box, can recover a solid and accurate 3D pose[3]. This method is simpler compared to the other two methods because it does not require pre-processing stages or 3D object models.

The next method, Triangulation Learning Network, effectively utilizes stereo information resulting in lower costs for hardware and can adapt to different scales of objects[4]. It employs 3D anchors to establish correspondences between the regions of interest in stereo images, from which the neural network learns to detect and triangulate the targeted object in 3D space[4]. Additionally, it has a cost-effective channel reweighting strategy that biases the network towards key parts of the object and benefits triangulation learning.

Finally, the monocular approach performs 3D object detection from a single monocular image. The method seeks to generate a set of candidate class-specific object proposals, which run through a convolutional neural network to obtain high-quality

detections[5]. In particular, it places object candidates in 3D, and then scores each candidate box displayed to the image plane via several intuitive potentials[5]. They are then further processed by a convolutional neural network resulting in a fast 3D object detection.

Overall, these methods each provide unique approaches to object detection and are beneficial in their own way. Next, are the steps for preparing the data, obtaining and tweaking the autonomous driving programs, and running the code on Google Colab.

Dataset and Data Processing

The KITTI dataset was used to test the three methods. The KITTI dataset is a widely used benchmark for autonomous driving tasks, such as stereo vision, optical flow, scene flow, visual odometry, and 3D object detection. The dataset consists of high-resolution images and videos captured by a calibrated camera mounted on a car. It covers diverse urban scenarios and driving conditions, such as highways, residential areas, city centres, and country roads. The KITTI does not cover all environments such as rural and aerial views, however it overall provides a realistic and challenging testbed for evaluating and comparing different methods for autonomous driving applications.

The 3D Object Detection 2017 data including the left color images, right color images, camera calibration matrices, and training labels from the KITTI dataset were utilized.

The three methods generated results by processing the left and right color images and camera calibration matrices. The training labels containing the correct object name, exact location, dimensions, and orientation of the 3D bounding box, were compared with the results of each method. From the 7400 training images, camera calibration files, and training label files, 1000 of each were used to test and compare the results.

AI Model

Open-source programs for the three autonomous driving models were obtained from GitHub and imported into Google Colab. Changes in the programs were made to match the features and environment of Google Colab and to provide the key data and results.

First, efforts were made to optimize the hyperparameters, however there was little to no change in performance for each of the three methods indicating that optimization was reached. The three programs already had the most optimal hyperparameter settings targeting the accuracy, speed, and memory usage of each method. The hyperparameters involved in these 3D object detection methods included:

1. The learning rate, batch size, weight decay, and optimizer for training the model.
2. The type and parameters of the region proposal network, such as anchor shapes, scales, and ratios.

These hyperparameters as described were already tuned carefully in the program code to achieve optimal results.

Next, in each method, there was a main program that utilized modules including torch_lib.Dataset, library.Math, library.Plotting, and torch_lib.ClassAverages. The modules were added as program files below each of the main programs to make them executable in Google Colab. Next, data files were imported to each of the three Google Colabs including a camera calibration matrix file (with 1,000 data information of traffic) for testing, pretrained weight files for reliable functioning and results, and a label dataset file to evaluate the results. Then, each of the programs were truncated, and unnecessary variables and code segments were deleted. Functions were tweaked to provide key data about the 3D bounding box including: the object detected (car, vehicle, pedestrian, etc.), the azimuth value, the coordinates of the bounding box, and its dimensions. In a separate program, the calculated results from the methods were stored and compared with the labels. Finally, methods were scored based on the KITTI metric which is discussed next in Evaluation Metric.

Evaluation Metric

The official KITTI metric was used for calculating the following for each method: Average Orientation Estimation (AOS), Average Precision (AP), and Orientation Score (OS). The Average Orientation Estimation is a value between 0 and 1, where 1 represents a perfect prediction, and is calculated as the average of the cosine similarity between the estimated orientation and the ground truth orientation of the matched detections. A high AOS score means that the detected objects have similar orientations as the ground truth objects. The Average Precision is a value between 0 and 1 that evaluates the localization algorithm and the performance of the object detection, and is calculated under the area of the precision recall curve. A high AP score means that the detected objects have high IoU (Intersection over Union) with the ground truth objects. The Orientation Score is the ratio of AOS over AP and represents the error averaged across all test images. A high OS score means that the detected objects have both a high overlap and a high orientation similarity with the ground truth objects, which is the ultimate goal of object detection.

Results

Evaluation Results

All the bar graphs start at 0.9, because the methods all performed at high standards with scores above 0.9. A portion of the bars

Method	Avg AOS	Avg AP	Avg OS
DLG	0.9134	0.9245	0.9879
TLNet	0.9789	0.9801	0.9987
Mono3D	0.9423	0.9530	0.9887

Table 1 Provides the average AOS, AP, and OS values for the three methods: Deep Learning and Geometry (DLG), Triangulation Learning Network (TLNet), and Moncular 3D Object Detection (Mono3D).

Method	Alpha	Standard Deviation	Data Size	Confidence
AOS (DLG)	0.05	0.03782	1000	0.00234
AOS (TLNet)	0.05	0.00734	1000	0.000455
AOS (Mono3D)	0.05	0.01686	1000	0.00104

Table 2 Shows the statistics of the AOS for the three methods: the alpha (representing the certainty of the confidence interval), the standard deviation of, the data size, and the confidence (calculated based on the alpha, standard deviation, and data size).

Method	Alpha	Standard Deviation	Data Size	Confidence
AP (DLG)	0.05	0.08546	1000	0.00530
AP (TLNet)	0.05	0.00956	1000	0.000593
AP (Mono3D)	0.05	0.04749	1000	0.002970

Table 3 Like Table 2, provides the statistics for the AP of the three methods with the alpha, standard deviation, data size, and the confidence.

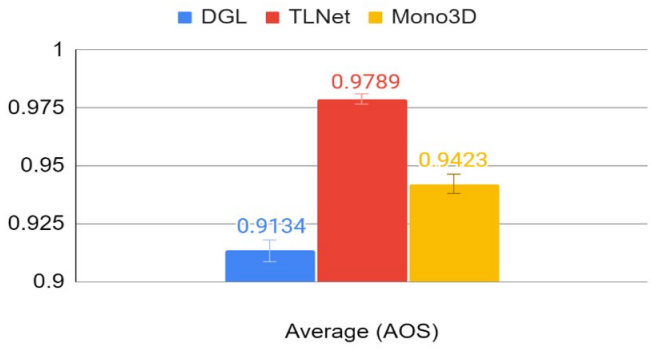

Fig. 1 Shows the average AOS and confidence interval for each method. The blue bar is Deep Learning and Geometry, the red bar is Triangulation Learning, and the yellow Monocular 3d Object Detection.

below 0.9 are removed in order to present clear differences between the methods and to show the confidence intervals. The nonoverlap between the error bars in the graphs shows the difference between the methods and demonstrate the credibility and significance of the results.

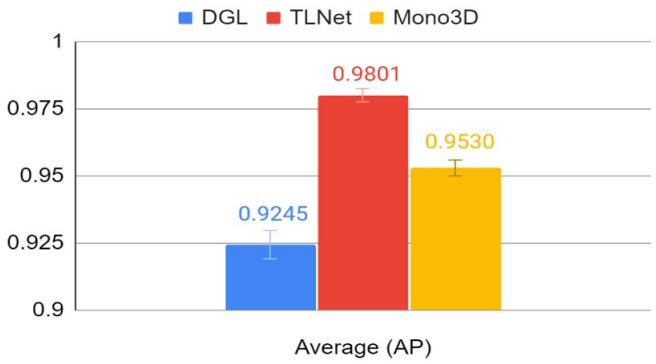

Fig. 2 Provides a visual of the average AP and confidence interval for the three methods.

The Deep Learning and Geometry method had the least AOS, AP, and Orientation Score with values of 0.9134, 0.9245, and 0.9879 respectively. The Monocular 3D Object Detection performed decently with values of 0.9423, 0.9530, and 0.9887 for each of the three metrics. Finally, the Triangulation Learning Network had the highest percentages for AOS and AP, 0.9789 and 0.9801, and the highest Orientation Score, 0.9987.

The Deep Learning and Geometry method detected less objects and had less precision and accuracy compared to the other methods. The Monocular 3D Object Detection performed better, with around a 3% increase in both AOS and AP. The Triangulation Learning Network ultimately performed the best with the highest scores in every metric. The TLNet's high AOS meant it could estimate the pose of the object the most accurately, and its high AP meant that it could accurately locate and segment the objects. Finally, its high OS meant that the detected objects had a high intersection and orientation similarity with the ground truth objects, which makes it a very precise and accurate detection algorithm. In addition, the Triangulation Learning Network had the smallest standard deviation and the narrowest confidence interval in the AOS and AP metrics. It therefore was more stable and produced more consistent values compared to the other two methods. The DLG and Mono3D methods had higher standard deviations and wider confidence intervals demonstrating less stability and uniformity in its results.

Discussion

Overall, the three methods all performed well with decent AOS, AP, and AOS. According to the results, they were all relatively stable and consistent and did not produce any major outliers, anomalies, or unexpected fails. As mentioned before, this is because of the optimal hyperparameter settings and the well-performing techniques used in each program method.

However, the methods still did produce errors in false-positives and false-negatives. False-positives are instances

where the algorithm detects an object that is not actually present in the image, such as a shadow or a reflection. False-negatives are instances where the algorithm fails to detect an object that is actually present in the image, such as a small or occluded object. There were some specific instances where methods did not have a high performance and produced false-negatives and false-positives.

In Monocular 3D object detection and the Deep Learning Network, for instance, when objects in images were too small, too far, or too occluded to be detected by the camera, false-negatives occurred. When the objects had complex poses or orientations that were not aligned with the camera such as a pedestrian leaning towards the ground or a cyclist tilted at an angle, false negatives also happened. In the Triangulation Learning Network, when objects were moving too fast or unpredictably for the camera to track (for example, swerving cars and cyclists speeding across the screen), false-negatives occurred. Across all three methods, when objects had similar appearances or shape to the background such as bench next to a car in the distance, false-positives occurred.

Each method has advantages but also some key drawbacks. Deep Learning and Geometry, despite combining deep neural networks with geometric reasoning for an innovative detection design, has limitations that affect its performance and applicability including:

1. DLG relies on accurate depth estimation from monocular images, which is a challenging and ill-posed problem. Errors in depth estimation can propagate to the 3D bounding box prediction and reduce the accuracy of the detection.

2. DLG assumes that the objects are rigid and have a fixed shape, which limits its ability to handle deformable or articulated objects, such as humans or animals. Moreover, DLG does not account for occlusion or truncation of the objects, which can affect the visibility and geometry of the objects.

3. DLG requires a large amount of annotated data to train the deep neural networks, which is costly and time-consuming to obtain.

DLG is not a universal solution for 3D object detection, and further research is needed to address these challenges and improve the robustness and generalization of the method.

Monocular 3D object detection is widely available and cost-effective, but it also suffers from a lack of accurate depth information. Therefore, it has to rely on additional cues, such as geometry constraints or pseudo point clouds to infer the 3D properties of objects. These cues are often noisy, incomplete or domain-specific, which limit the generalization and robustness of the method. This contributes to the following drawbacks:

1. The method is sensitive to occlusion, truncation, and scale variation of objects in the image, which affect the estimation of depth and orientation.

2. Mono3D has to deal with the ambiguity and uncertainty of the 3D reconstruction problem from a single view, which may lead to multiple plausible solutions and inaccurate predictions.

3. Mono3D is computationally intensive and requires multiple stages or modules to perform feature extraction, 2D detection, 3D projection, and refinement. This reduces the efficiency and scalability of the method.

Future research should focus on developing a more robust, efficient, and scalable monocular method that can overcome these limitations.

Triangulation Learning Network outperforms Deep Learning and Geometry and Monocular 3D Object Detection. It can effectively exploit the inherent structure of 3D objects and scenes from multiple views to improve the accuracy and robustness of the 3D detection. TLNet consists of two modules - a triangulation module (that generates 3D proposals by triangulating 2D keypoints), and a refinement module (that adjusts the proposals using semantic features) – which allows it to handle occlusion, truncation, and varying viewpoints better than DLG and Monocular Object Detection which rely on a single-view geometry or depth estimation network that is prone to errors and ambiguities. TLNet also reduces the computational cost and memory consumption by processing only a subset of views instead of all available ones. Finally, it does not require any explicit 3D representation or reconstruction of the scene, which simplifies the implementation.

TLNet has some relatively small limitations that may affect its performance and applicability in real-world scenarios:

1. TLNet relies on a fixed number of anchor boxes to represent the 3D objects, which may not be able to capture the diversity and complexity of real-world objects.

2. TLNet assumes that the camera parameters are known and fixed, which may not always hold in dynamic environments where the camera pose and intrinsic parameters change over time. This in some cases may hinder the TLNet in detecting 3D objects that are truncated by image boundaries resulting in possible detection flaws for the vehicle.

Despite these drawbacks, the Triangulation Learning Network achieves state-of-the-art results for 3D object detection and has high effectiveness and efficiency.

Conclusion

Triangulation Learning Network has been shown to outperform Deep Learning Geometry and Monocular 3D Object Detec-

tion and achieve high accuracy and robustness. It can handle occlusions and partial views better, as it can exploit the complementary information from multiple views. It can generalize well to new scenarios and domains, as it can learn and adapt transferable features from diverse viewpoints. These advantages make TLNet a powerful tool for autonomous driving technology. It can enhance the safety, efficiency, and reliability of the perception system by reducing the errors and uncertainties in object detection, which are critical for planning and control. It can also enable new applications and functionalities for autonomous driving, such as semantic segmentation, scene understanding, and 3D reconstruction, which require high-quality object detection as a prerequisite.

References

1 B. Deng *et al.*, *Research – Revisiting 3D Object Detection from an Egocentric Perspective*, Waymo technical report, 2019.

2 A. Geiger, P. Lenz and R. Urtasun, 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.

3 A. Mousavian *et al.*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

4 Z. Qin, J. Wang and Y. Lu, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

5 X. Chen *et al.*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.