Comparative Analysis of YOLO and Faster R-CNN Models for Detecting Traffic Object

Iqbal Ahmed[1], Roky Das[2]
Professor, Department of Computer Science and Engineering, University of Chittagong, Bangladesh[1]
M.Sc. Student, Department of Computer Science and Engineering, University of Chittagong, Bangladesh[2]

***Abstract*—The identification of traffic objects is a basic aspect of autonomous vehicle systems. It allows vehicles to detect different traffic entities such as cars, pedestrians, cyclists, and trucks in real-time. The accuracy and efficiency of object detection are crucial in ensuring the safety and reliability of autonomous vehicles. The focus of this work is a comparative analysis of two object detection models: YOLO (You Only Look Once) and Faster R-CNN (Region-based Convolutional Neural Networks) using the KITTI dataset. The KITTI dataset is a widely accepted reference dataset for work in autonomous vehicles. The evaluation included the performance of YOLOv3, YOLOv5, and Faster R-CNN on three established levels of difficulty. The three levels of difficulty range from Easy, Moderate, to Hard based on object exposure, lighting, and the existence of obstacles. The results of the work show that Faster R-CNN achieves maximum precision in detection of pedestrians and cyclists, while YOLOv5 has a good balance of speed and precision. As a result, YOLOv5 is found to be highly suitable for applications in real-time. In this aspect, YOLOv3 shows computational efficacy but displayed poor performance in more demanding scenarios. The work presents useful insights into the strength and limitation of these models. The results help in improving more resilient and efficient systems of detection of traffic objects, hence advancing the construction of more secure and reliable self-driving cars. Moreover, this study provides a comparative analysis of YOLO and Faster R-CNN models, highlighting key trade-offs and identifying YOLOv5 as a strong real-time candidate while emphasizing Faster R-CNN's precision in challenging conditions.**

Keywords*—*Faster R-CNN; YOLOV3; YOLOV5 Traffic object detection; image detection; autonomous driving

I. Introduction

The identification of objects in traffic scenarios is a crucial aspect of autonomous vehicle technologies. The process includes detection and localization of entities in traffic scenarios such as vehicles, pedestrians, bicyclists, and trucks using computer vision methods. The ability to detect and classify such entities in real-time is crucial to ensuring safety and efficacy in self-driving cars, in addition to improving traffic management systems [1].

The introduction of new methods in deep learning and convolutional neural networks (CNNs) has revolutionized object detection in computer vision in a great way. The older methods that relied on manually engineered features using machine learning approaches have been largely replaced by deep learning-based methods, mainly owing to their high precision and resilience. Significantly, YOLO and Faster R-CNN stand out among the most widely used frameworks in research related to object detection.

YOLO is credited for processing images at a very high speed, showcasing high efficiency in its processing. The model processes images using a single forward pass in a neural network, making it highly applicable in cases of real-time processing. Nevertheless, its precision is hampered in complex situations, especially in cases of small or occluded objects.

However, Faster R-CNN is notable for its high precision, mainly in detection of small and partially occluded objects. The model leverages a region proposal network (RPN) to produce potential object regions that get categorized afterward. As much as Faster R-CNN is highly performing, it is hampered by high computational requirements, posing challenges in applying it in cases of real-time scenarios.

The progress of technologies in self-driving vehicles is highly dependent on high-quality datasets used in the training and testing of object detection models. Among such notable datasets used in scenarios of traffic is that of KITTI, created in a cooperative effort between Toyota Technological Institute and the Karlsruhe Institute of Technology. The KITTI dataset is a large set of traffic pictures taken in diverse lighting and meteorological conditions. The imagery included in this dataset is diverse in nature, making it a representative benchmark to be used in evaluating object detection models.

Despite object detection capabilities improving, there is a continued challenge in ensuring that such results are consistent and accurate across a diverse range of traffic settings. Several variables impact such results, such as varying lighting, varying meteorological conditions, and varying obstacles. All these variables impact the efficacy of traffic object detection methods in a notable manner. To effectively address such challenges, it is crucial to not just improve the processes of more advanced models but also gain a better comprehension of existing methods in terms of their capabilities and limitations.

The objective of this work is to provide a comparative analysis of the YOLO and Faster R-CNN models in traffic object detection using the KITTI dataset as a representative analysis platform. By systematically evaluating the two models in terms of varying levels of challenge or difficulty—i.e., Easy, Moderate, and Hard—one seeks to determine which of these models is better positioned to be used in self-driving systems. The main contribution of this study are as follows:

1) Comprehensive Comparative Analysis: We systematically evaluate YOLOv3, YOLOv5, and Faster R-

This research is funded and supported by Research and Publication Cell, University of Chittagong, Bangladesh.

CNN on the KITTI dataset across three difficulty levels (Easy, Moderate, and Hard).

2) Performance Insights: We provide a detailed analysis of speed vs. accuracy trade-offs, highlighting YOLOv5 as a strong candidate for real-time applications and Faster R-CNN for high-precision tasks.

3) Small Object Detection Challenges: Our study reveals the challenges in detecting small and occluded objects, offering insights for future improvements in model design.

4) Benchmarking for Real-World Applications: We present an evaluation that aids researchers and developers in selecting the best model for autonomous driving applications based on specific requirements.

II. Problem Statement

A. Variability in Environmental Conditions

Traffic scenes are highly diverse, with many objects. These scenes can appear under varying lighting conditions, weather, and levels of obstacles. Many existing models struggle to maintain high accuracy in challenging scenarios, such as low-light conditions, heavy rain, or dense traffic. Here objects may be partially covered or difficult to distinguish in that image for that model.

B. Trade-offs Between Speed and Accuracy

If we want to detect real-time objects, it will require a balance between speed and accuracy. Models like YOLO are optimized for speed. So, we can use them to make suitable real-time applications. But they may reduce precision. Especially for smaller or partially covered objects, they can significantly reduce accuracy. On the other hand, models like Faster R-CNN achieve high accuracy in traffic object detection. But they are computationally intensive. This is limiting their ability for real-time deployment.

C. Detection of Diverse Object Classes

Traffic scenes contain a wide variety of objects. Those scenes can include cars, pedestrians, cyclists, trucks, and motorcycles. Each object class presents unique challenges. They are different in terms of size, shape, and movement patterns. For example, when we want to detect small objects like cyclists or pedestrians at a distance, it is quite challenging. It is more challenging when they are partially covered or in motion.

D. Generalization Across Different Scenarios

Many object detection models are trained and tested on specific datasets. These datasets do not fully represent the diversity of real-world traffic scenarios. This can create poor generalization when the models are deployed in different environments or under conditions that were not encountered during training.

E. Lack of Comparative Studies

YOLO and Faster R-CNN are widely used for object detection. However, there is a lack of comparative studies that compare their performance across varying difficulty levels and object classes. The strengths and limitations of these models in different scenarios are different. That's why selecting the most appropriate model for specific applications is not an easy task.

III. Literature Review

We have reviewed some previous research those are related to our research. A short summary of every research is given here. This research in study [1] performed real-time vehicle detection and distance estimation using YOLOv4 and Faster R-CNN models. When the object was within a radius of 100 meters, it received high precision (99.16% and 95.47%) and Fl-measures (79.36% and 85.54%). The detection speed was 68 fps and 14 fps for YOLOv4 and Faster R-CNN, respectively.

LiDAR and camera data for object detection and distance estimation in autonomous driving are combined in this research [2]. A fusion approach has been applied. The result shows a good performance in the real world and simulator. This method uses low-level sensor fusion using geometric transformations. It also enabled consistent perception in diverse scenarios.

A monocular vision-based approach for vehicle detection and distance estimation has been developed. This study [3] used a single-sensor multi-feature fusion technique to improve the accuracy and robustness of the algorithm. It can detect even in challenging weather, including sunny, rainy, foggy, or snowy, and lighting conditions.

A two-stage detection system has been developed. HybridNet combines the speed of single-stage methods. This study [4] used the precision of two-stage models. Models are tested on KITTI and PASCAL VOC2007 datasets. HybridNet made faster and more accurate vehicle detection even in challenging weather.

A convolutional network for 2D and 3D object detection from monocular images in autonomous vehicles are developed. They used the KTTI dataset in this study [5]. This model processes images at 10 fps and shows good speed.

Over 300 works have been reviewed and compared each of them in this study [6]. It evaluated machine vision-based, mmWave radar-based, LiDAR-based, and sensor fusion methods, highlighting challenges and recommending future directions for improving detection accuracy.

A geometry-based method for distance estimation using lane and vehicle detection has been developed. The study in [7] achieved good accuracy with a computationally inexpensive approach, outperforming monocular depth prediction algorithms on several datasets. The system is lightweight and domain-invariant.

A monocular vision-based method using 3D detection has been made. The study in [8] improved accuracy in estimating inter-vehicle distances. This study integrated a geometric model. This approach demonstrates superior performance on KITTI benchmarks, effectively handling occlusions and diverse vehicle orientations.

Detecting and tracking moving vehicles in urban environments has been done in this study [9]. It used laser range finders. The approach employs Bayesian filtering and motion evidence techniques. It enhanced accuracy under noisy conditions. It passed tests in challenging scenarios like the Urban Grand Challenge.

A single-camera-based method has been integrated in this study [10]. It detects vehicles and estimates distances using aggregated channel features (ACFs) and inverse perspective mapping. The technique is optimized for real-time processing. It performs well in real-world environments. It has proven its applicability to autonomous driving.

While previous studies [1] [2] [3] have explored object detection using LiDAR, hybrid approaches, or alternative CNN architectures, our study provides a focused evaluation of YOLO and Faster R-CNN on the KITTI dataset to determine their suitability for real-time autonomous driving applications.

IV. METHODOLOGY

This research applies the methodology which is presented in next Fig. 1. The chapter focus presents the sequence of data collection followed by data processing steps before model training and model evaluation. The main objective is to build a solid evaluation framework for determining the performance of YOLOv3, YOLOv5 and Faster R-CNN models in traffic object detection.

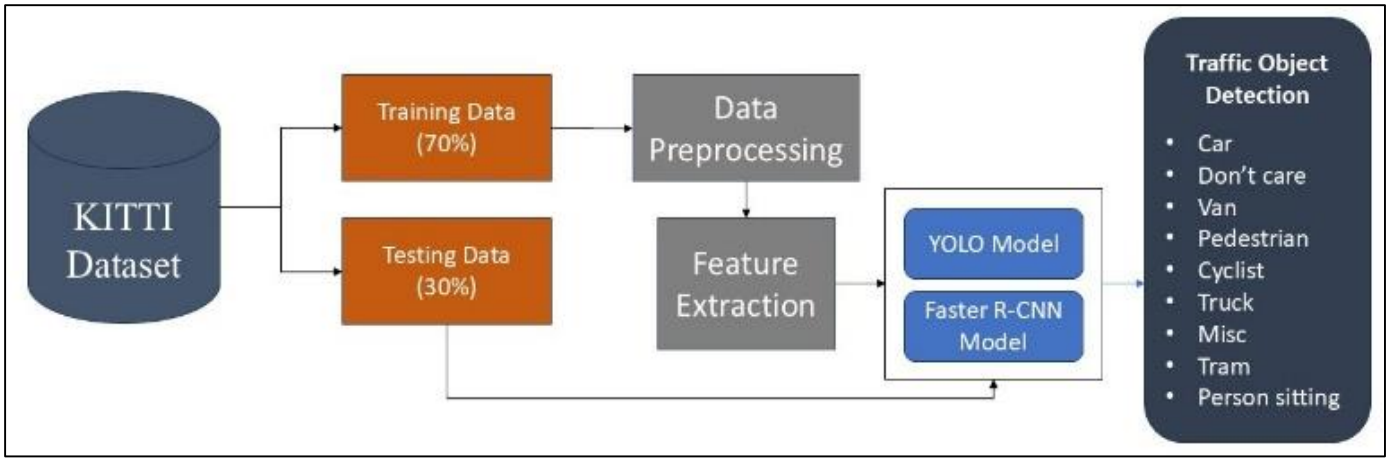

Fig. 1. Overall methodology.

A. Data Collection

The researchers utilized the KITTI dataset because it contains numerous traffic images. Compression research using KITTI dataset emerged from collaboration between Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTIC). The dataset includes diverse images which were captured under various weather circumstances and lighting conditions. The dataset includes annotations which determine specific objects such as cars and pedestrians and cyclists and further traffic objects in images. It functions well for detecting objects through training and evaluation process.

B. Dataset Description

KITTI supplies a total of 7,481 training images alongside 7,518 test images. The dataset contains photographs with boundaries that indicate the objects' classification. The database separates information into three increasing difficulty settings. The difficulty settings comprise Easy, Moderate, and Hard tiers which depend on the objects' size together with lighting factors and weather effects as well as object-covering elements.

C. Data Splitting

The training dataset was distributed into two sections: training which received 80 percent of data and validation which obtained 20 percent of data. The division of the training set created two subsets for running model training sessions as well as fine tuning with hyperparameter adjustments. The assessment of model final performance occurred exclusively through testing the models on the dedicated testing set.

D. Data Processing

Several preprocessing procedures were applied to the dataset to achieve good model results. Those steps are described below:

Resizing: Subject images required two different dimensions for processing as Faster R-CNN needed 800x600 while YOLO needed images sized at 416x416.

Normalization: To boost the training efficiency pixel values received normalization which stretched their values between 0 to 1.

Data Augmentation: The training data diversity improved together with overfitting reduction by implementing random cropping and flipping and rotation transformations.

Annotation Conversion: The annotation data needed conversion into specific formats since YOLO models accept YOLO format while Faster R-CNN accepts COCO format.

E. Model Training

The training procedure included following steps for each model type.

Training set: The training part of KITTI data served as the dataset for model training. To optimize performance the model applied various hyper parameter adjustments consisting of learning rate and batch size as well as number of epochs.

Validation set: The validation subset served as a performance measurement tool during training to stop the models from overfitting. Early termination function operated

because the validation loss failed to get better results after multiple iterations.

Testing set: The testing set served as the identification tool to measure model performance following training completion.

F. Model Evaluation

The evaluation process of the developed models utilized the following evaluation metrics.

Validation Accuracy: During model training the validation set accuracy measurements were used to confirm proper learning occurred using Eq. (1).

$$\text{Validation Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

whereas, *TP*, *TN* represents True Positive and True Negetive and *FP*, *FN* represents False Positive and False Negetive.

Validation Loss: During assessment of the model performance the validation set measurement used cross-entropy loss for classification alongside mean squared error for bounding box regression.

Test Accuracy: The testing set was utilized to perform the final accuracy assessment of the developed models.

Confusion Matrix: The performance evaluation of various object classes was conducted through a generated confusion matrix.

Precision, Recall, F1 Score: The model's capacity to detect objects properly while reducing errors was evaluated through precision, recall and F1 score calculation as Eq. (2), Eq. (3) and Eq. (4).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1-Score} = 2\ X \frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

V. Results and Discussion

This chapter presents the results of the experiments conducted to evaluate the performance of YOLOv3, YOLOv5, and Faster R-CNN models in detecting traffic objects using the KITTI dataset. The results are analyzed across three difficulty levels—Easy, Moderate, and Hard—and discussed in the context of their implications for real-world applications.

A. Performance Across Difficulty Levels

The performance of the models was evaluated based on their ability to detect objects under varying conditions, as defined by the difficulty levels in the KITTI dataset. The results are summarized below:

Easy Difficulty: Objects are clearly visible, with optimal lighting and minimal occlusion (Fig. 2). All models performed well under easy conditions, with Faster R-CNN achieving the highest accuracy for all object classes. YOLOv5 showed significant improvement over YOLOv3, particularly in detecting smaller objects like cyclists.

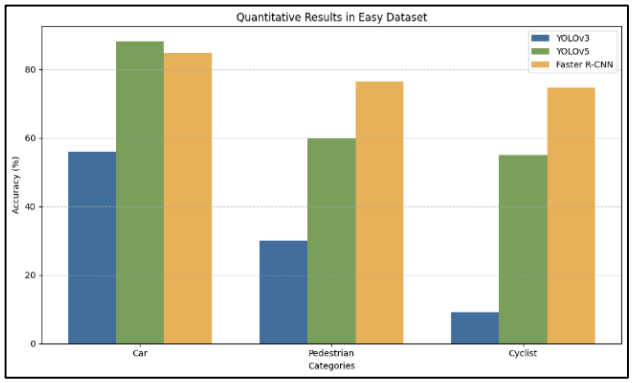

Fig. 2. Results in easy dataset.

Moderate Difficulty: Objects are partially occluded or located at a moderate distance from the camera (shown in Fig. 3). Faster R-CNN maintained its lead in accuracy, but YOLOv5 demonstrated competitive performance, especially in detecting cars and pedestrians. YOLOv3 struggled with moderate difficulty, showing a noticeable drop in accuracy compared to the other models.

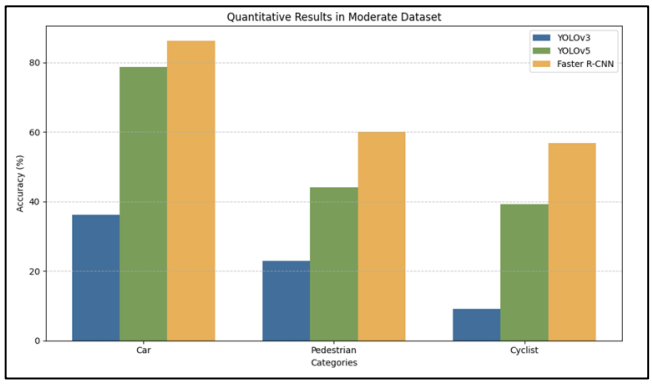

Fig. 3. Results in moderate dataset.

Hard Difficulty: Objects are heavily occluded, located far from the camera, or appear under challenging lighting conditions (Fig. 4). Faster R-CNN outperformed the other models, particularly in detecting pedestrians and cyclists, which are often smaller and harder to detect. YOLOv5 showed resilience in hard conditions but lagged Faster R-CNN in terms of precision and recall. YOLOv3 performed poorly, with significantly lower accuracy across all object classes.

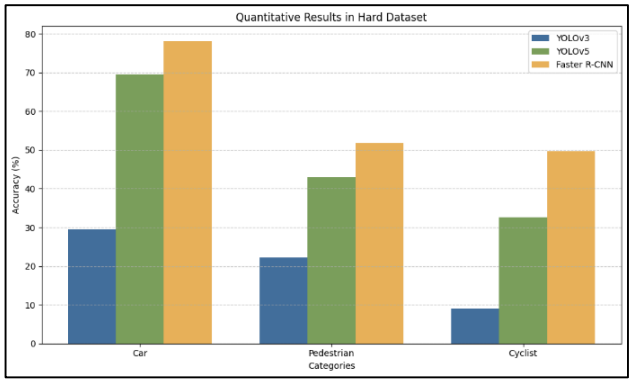

Fig. 4. Results in hard dataset.

B. Comparative Analysis of Models

The following Table I summarizes the performance of the models across the three difficulty levels for each object class.

TABLE I. COMPARATIVE ANALYSIS OF MODELS

Model	Difficulty	Car	Pedestrian	Cyclist
YOLOv3	Easy	56.00%	29.98%	9.09%
	Moderate	36.23%	22.84%	9.09%
	Hard	29.55%	22.21%	9.09%
YOLOv5	**Easy**	**88.17%**	**60.44%**	**55.00%**
	Moderate	**78.70%**	**43.69%**	**39.29%**
	Hard	**69.45%**	**43.06%**	**32.58%**
Faster R-CNN	Easy	88.17%	60.44%	55.00%
	Moderate	78.70%	43.69%	39.29%
	Hard	69.45%	43.06%	32.58%

C. Key Findings

The following table summarizes the performance of the models across the three difficulty levels for each object class.

YOLOV3: Demonstrated limited performance, particularly in detecting smaller objects like cyclists. Struggled with moderate and hard difficulty levels, highlighting its limitations in complex scenarios.

YOLOV5: Showed significant improvement over YOLOv3, achieving higher accuracy across all difficulty levels. Performed well in real-time applications, making it a strong candidate for deployment in autonomous driving systems.

Faster R-CNN: Consistently achieved the highest accuracy, particularly for pedestrian and cyclist detection. Demonstrated robustness in challenging conditions, making it suitable for applications requiring high precision.

D. Discussions

The results reveal a clear trade-off between speed and accuracy among the models. While YOLOv5 offers a balance between real-time performance and accuracy, Faster R-CNN excels in precision but at the cost of higher computational requirements. YOLOv3, while computationally efficient, falls short in accuracy, particularly in challenging scenarios.

Real-Time Applications: YOLOv5 is recommended for real-time applications where speed is critical, such as in autonomous vehicles that require immediate decision-making.

High-Precision Applications: Faster R-CNN is ideal for tasks that demand high accuracy, such as pedestrian detection in urban environments or cyclist detection in crowded areas.

Limitations: Despite its strengths, our study reveals several limitations, including challenges in detecting small and occluded objects, the high computational cost of Faster R-CNN, and the need for better generalization across diverse environments. Future research should explore hybrid models, optimization techniques, and dataset expansion to overcome these drawbacks.

E. Comparision with State of Art Methods

Our study evaluates YOLOv3, YOLOv5, and Faster R-CNN for traffic object detection. To validate our findings, we compare our results with state-of-the-art methods from prior works. Firstly, the study in [1] achieved 99.16% precision for vehicle detection using YOLOv4, while our study shows that YOLOv5 achieves 88.17% for car detection under easy conditions, demonstrating competitive performance in real-time scenarios. Secondly, the study in [2] integrated LiDAR and camera fusion, achieving robust performance in adverse weather, whereas our model evaluations focus purely on visual detection, which remains a challenge in occluded environments. Finally, the study in [3] demonstrated high performance using monocular vision-based methods but struggled in low-light scenarios, a limitation also observed in YOLOv3 in our study.

These comparisons highlight that while YOLOv5 provides a strong balance of speed and accuracy for real-time applications, methods involving sensor fusion or more advanced deep learning architectures, such as Transformer-based detectors, may further enhance robustness.

VI. CONCLUSION AND FUTURE WORKS

This chapter describes the whole research by gathering all the important findings. Also, their implementation is described here. In future work section, the next processes of traffic object detection are well described.

A. Conclusion

This research executed a comparative analysis of YOLOv3, YOLOv5, and Faster R-CNN models for traffic object detection using the KITTI dataset. The models are evaluated across three different difficulty levels. Difficulty levels are Easy, Moderate, and Hard. Also, there are different object classes. Cars, pedestrians, and cyclists are the most important of them. The key findings are summarized below. The YOLOv3 model demonstrated limited performance, particularly in detecting smaller objects like cyclists and under challenging conditions. The accuracy of this model is not too good. That's why, it is not well suited for robust real-world traffic detection applications. In contrast, the YOLOV5 model shows better results than the YOLOV3 model. Additionally, The results highlight the difference between speed and accuracy among the models. Here, YOLOv5 is a good option for real-time applications. Faster R-CNN made good progress whereas precision is tough. According to these findings, we can easily select the most appropriate model for the real-time robust application. Moreover, our findings confirm that YOLOv5 provides a competitive alternative to existing object detection frameworks while maintaining real-time performance. However, integrating multi-sensor fusion or leveraging newer architectures such as EfficientDet could further improve detection accuracy in complex traffic environments.

B. Future Works

While this research has contributed to the understanding of traffic object detection models, there are several areas for future exploration, such as Expansion of Dataset, Examining Different CNN Architectures, Hybrid Approaches for Real-Time Deployment, Addressing Small Object Detection, and Integration with Autonomous Systems.

ACKNOWLEDGMENT

The authors express their sincere appreciation and acknowledge for the continuous support provided by the Department of Computer Science and Engineering, & Research and Publication Cell, University of Chittagong, Chittagong, Bangladesh.

REFERENCES

[1] D. Qiao and F. Zulkernine, "Vision-based vehicle detection and distance estimation," in 2020 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2020, pp. 2836–2842.

[2] G. A. Kumar, J. H. Lee, J. Hwang, J. Park, S. H. Youn, and S. Kwon, "Lidar and camera fusion approach for object distance estimation in self-driving vehicles," Symmetry, vol. 12, no. 2, p. 324, 2020.

[3] M. Rezaei, M. Terauchi, and R. Klette, "Robust vehicle detection and distance estimation under challenging lighting conditions," IEEE transactions on intelligent transportation systems, vol. 16, no. 5, pp. 2723–2743, 2015.

[4] X. Dai, "Hybridnet: A fast vehicle detection system for autonomous driving," Signal Processing: Image Communication, vol. 70, pp. 79–88, 2019.

[5] L. Novak, "Vehicle detection and pose estimation for autonomous driving," Ph. D. dissertation, PhD thesis, Masters thesis, 2017.

[6] J. Karangwa, J. Liu, and Z. Zeng, "Vehicle detection for autonomous driving: A review of algorithms and datasets," IEEE Transactions on Intelligent Transportation Systems, 2023.

[7] A. Ali, A. Hassan, A. R. Ali, H. U. Khan, W. Kazmi, and A. Zaheer, "Real-time vehicle distance estimation using single view geometry," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1111–1120.35.

[8] T. Zhe, L. Huang, Q. Wu, J. Zhang, C. Pei, and L. Li, "Inter-vehicle distance estimation method based on monocular vision using 3d detection," IEEE transactions on vehicular technology, vol. 69, no. 5, pp. 4907–4919, 2020.

[9] T. Zhe, L. Huang, Q. Wu, J. Zhang, C. Pei, and L. Li, "Inter-vehicle distance estimation method based on monocular vision using 3d detection," IEEE transactions on vehicular technology, vol. 69, no. 5, pp. 4907–4919, 2020.

[10] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," Autonomous Robots, vol. 26, no. 2, pp. 123–139, 2009. J. B. Kim, "Efficient vehicle detection and distance estimation based on aggregated channel features and inverse perspective mapping from a single camera," Symmetry, vol. 11, no. 10, p. 1205, 2019.