

Using Vehicle-Mounted Camera to Collect Information for Managing Mixed Traffic

Elnaz Namazi

Department of Computer Science
Norwegian University of Science and Technology
(NTNU)
Trondheim, Norway
elnaz.namazi@ntnu.no

Rein Nisja Holthe-Berg

Department of Computer Science
Norwegian University of Science and Technology
(NTNU)
Trondheim, Norway
reinhb1994@gmail.com

Christoffer Skar Lofsberg

Department of Computer Science
Norwegian University of Science and Technology
(NTNU)
Trondheim, Norway
chris.lofsberg@gmail.com

Jingyue Li

Department of Computer Science
Norwegian University of Science and Technology
(NTNU)
Trondheim, Norway
jingyue.li@ntnu.no

Abstract—With increasingly rapid advances in the field of producing modern and autonomous vehicles, the need for intelligent traffic management systems, which take advantage of the vehicle’s abilities to sense and communicate, has increased. A considerable amount of literature has been published on managing traffic that includes only autonomous vehicles. However, changing all vehicles to autonomous versions is a long-term process. In the near future, traffic will be a mixture of human-driven and autonomous vehicles. To date, few studies have investigated mixed traffic in intelligent management systems. The main objective of this research is to study the possibility of using a vehicle-mounted camera to sense and collect the required traffic data of the surrounding vehicles in mixed traffic. To achieve this, a vehicle with a monocular camera is used to collect image information for detecting and counting the vehicles in different lanes and estimating their distance and speed on the defined route. The results indicate that our proposed image processing algorithms can acquire the information needed for intelligent traffic management systems.

Keywords—*intelligent traffic management; autonomous vehicle; image processing; vehicle detection; speed estimation; distance estimation*

I. INTRODUCTION

Recent developments in autonomous vehicle (AV) technology have heightened the need for intelligent traffic management systems that are suitable for AVs. Managing AV traffic has been studied by many researchers. They used AVs to collect and share information based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications (e.g., [1], [2], [3], [4], [5], [6], and [7]). Despite the importance of managing AVs, changing all vehicles to autonomous versions will take time. Thus, we believe that traffic will be a

mixture of human-driven vehicles (HDVs) and AVs for a long period. However, little attention has been paid to the mixed traffic that we will face in the coming years [8].

Intersections play a critical role in enhancing the efficiency and safety of traffic. Intelligent intersection management systems are introduced to manage traffic by using traffic data. There are various approaches to collecting traffic data. If traffic comprises purely AVs, AVs’ sensors and V2X communication are used for collecting and sharing traffic data. However, for HDVs, such technologies are not applicable. Therefore, streets and intersections are equipped with various sensors such as inductive loop detectors and stationary cameras for collecting traffic information. Using these sensors can improve a traffic management system through accessing traffic data. However, weather conditions and brightness affect data quality. Furthermore, equipping all intersections and streets with these sensors would be costly.

To deal with these challenges of advanced cars or AVs, the idea of our study is to use vehicles’ sensors to collect traffic data of AVs and HDVs. As the main focus of this work is to study the effect of a vehicle sensing and collecting traffic data, and we did not have access to an AV in our experiments, we used a vehicle-mounted camera to achieve our goals. This study seeks to identify how to use sensors, especially the mounted monocular camera, to collect the required data from multiple lanes. We limited our focus to the necessary data for managing traffic, especially at intersections. We developed algorithms to analyze the video data collected from a camera. We collected and analyzed real traffic information from a route in Trondheim to evaluate our algorithms.

Our data collection and analyses focus on answering the following research questions:

- RQ1: How can we identify the number and type of vehicles in front of a vehicle and in the nearby lanes using image data captured by a vehicle-mounted monocular camera?
- RQ2: What is the most accurate combination of width and height when calculating the distance of vehicles in front and in nearby lanes using image data captured by a vehicle-mounted monocular camera?
- RQ3: How can the speed of the detected vehicles be estimated by using image data captured by a vehicle-mounted monocular camera?

The experimental work presented here provides one of the first investigations on how to use a vehicle-mounted camera's ability to collect traffic data to improve traffic management systems at intersections by considering mixed traffic. The results indicate that autonomous vehicles can provide the required mixed traffic data such as number and type of vehicles, their distance, and their speed. However, more studies are needed to improve the accuracy of the outputs.

The remainder of this paper is organized as follows: Section II provides a brief overview of studies related to intersection management methodologies. Section III explains the research objective and approach. Section IV presents the implementation and evaluation of our proposed approach and algorithms. Section V discusses the advantages and remaining challenges of this study. Section VI concludes.

II. BACKGROUND

Managing intersections plays a critical role in improving the performance and safety of traffic management systems. Developments in software, hardware, networks, and communications and the introduction of AVs have led to the use of intelligent systems to manage traffic. Therefore, more studies recognize the importance of an intelligent traffic management system that includes AV traffic. Many different methodologies have been presented. Most of the current literature has paid particular attention to using rule-based (e.g., [9] and [10]), optimization (e.g., [11], [12], and [13]), and hybrid (e.g., [14] and [15]) methodologies [8]. Moreover, to enhance the smartness of intersection management systems, artificial intelligence (AI) techniques can be applied (e.g., [16], [17], and [18]). Most of the AI-based intersection management research focuses on two issues: One is about decision-making and predicting the traffic situation based on the traffic data, which are collected by stationary sensors at the intersections. Another is to use AVs' AI capabilities to sense, collect, and share information about themselves in purely AV traffic. Our recent literature review [8] showed that in intelligent intersection management systems from 2008 to 2019, only 3.8% of the papers used AI to achieve their goals. However, AI-based traffic management systems have the potential to enhance traffic performance by improving the data collection process and predicting traffic features.

During the last few years, there has been a growing body of literature on using AI in object detection algorithms. Girshick et al. [19] proposed a region-based convolutional network (ConvNet) method (R-CNN) that used a deep ConvNet to classify the object proposals. It is a combination

of region proposal and CNN. As the training phase is a multi-stage pipeline, it is costly in terms of space and time. Moreover, it is slow at detecting objects [20]. To mitigate the limitations of R-CNN, the fast region-based convolutional network method (Fast R-CNN) was presented by Girshick [20]. It used a deep ConvNet to classify object proposals efficiently. It tried to improve the training and testing speed and detection accuracy. Fast R-CNN was developed with Python and C++. The experiments indicate that Fast R-CNN trains the deep VGG16 network 19 times faster than R-CNN. Also, test time is 213 times shorter. In addition, it is more accurate.

Faster R-CNN was presented by Ren et al. to detect objects by considering region proposal networks (RPNs) [21]. Mask R-CNN, which is the extended version of Faster R-CNN, was proposed by He et al. [22]. It detects objects in an image efficiently and generates a high-quality segmentation mask simultaneously. Training with Mask R-CNN is simple and adds just a small overhead to Faster R-CNN.

Redmon presented You Only Look Once (YOLO) [23], which is a real-time object detection algorithm. In contrast with previous studies, YOLO is based on a regression problem rather than classification. It uses a single neural network for the detection pipeline. Moreover, classes and bounding boxes are predicted in one run of the algorithm for the whole image.

A number of researchers have considered using various proposed algorithms to detect vehicles, inter-vehicle distance, and vehicle speed. For instance, to detect vehicles, Godha [24] proposed an algorithm using a mounted camera in real-time that could send a warning to the driver. This system was developed in MATLAB as a driver assistant system. Asvadi et al. [25] proposed a real-time and multimodal vehicle detection system. It uses YOLO [23], [26] as a deep ConvNet object detection framework. Moreover, it is based on fusing the data collected by a color camera and 3D-LIDAR. The KITTI object detection dataset is used in the experiments phase. Caltagirone et al. [27] developed a fusion fully convolutional neural network (FCN) for road detection. It uses KITTI as a dataset and LiDAR and camera fusion.

In addition, some studies were done that focused on detection of the inter-vehicle distance. For example, Huang et al. [28] proposed a driver assistant system to detect vehicles and estimate the inter-vehicle distance. This system uses a camera as a sensor and includes image processing, information collection, vanishing point detection, road region segmentation, and estimation of the inter-vehicle distance. Lee [29] presented a method for estimating the inter-vehicle distance using a blackbox camera. The idea is to estimate the distance based on the lane width for the detected vehicle. Chadwick et al. [30] proposed an approach using radar and a camera to estimate the vehicle distance. Moreover, an automatic process was introduced for training and labeling the new dataset from multiple cameras. It used YOLO [23] as an object detector and KITTI as a dataset. Furthermore, several studies consider determining the vehicles' speed. For instance, Gerát et al. [31] used Gaussian mixture models, density-based spatial clustering of applications with noise (DBSCAN), a Kalman filter, and the optical flow method to detect vehicle speed using a stationary camera. Moazzam et al. [32]

proposed a new approach to determine vehicle speed based on video captured by a stationary camera. They used the QMUL dataset [33] for this experiment.

III. METHODOLOGY

A. Research objective

An intelligent intersection management system tries to improve traffic flow performance by accessing traffic data. In this study, we have tried to collect the number and type of vehicles, their distance, and their speed in mixed traffic by using a mounted monocular camera installed on a vehicle. We limited our focus to analyzing the monocular camera rather than more expensive sensors, e.g., radar or LiDAR, because we observe that many advanced vehicles have cameras installed by default and not many vehicles will have radar or LiDAR installed in the future. We believe that considering the data from a camera only will make our system more applicable. The objective of this work is to study the possibility of using a mounted monocular camera to collect mixed traffic data from multiple lanes by considering the effect of camera movement.

B. Research approach

This study is exploratory and normative in nature, since vehicle-mounted cameras have not been used to collect vehicle data to manage mixed traffic in existing studies, and new algorithms are developed. In this study, we followed Pfeffers et al.'s Design Science Research Process [34].

IV. IMPLEMENTATION AND EVALUATION

To answer the research questions, we decided to construct a system based on various state-of-the-art algorithms. The system was developed in Python and used popular frameworks that have well-documented outcomes in various projects. We proposed a system that is a combination of PyTorch [35] for implementation of the pre-trained version of YOLO [23], [36], [37] and OpenCV [38], Canny edge detection [39], and progressive probabilistic Hough transformation [40], [41] for lane detection while driving.

The data were collected from a vehicle equipped with a front-facing camera. We used a GoPro Hero 7 camera [42], since it is able to record GPS data as well. The video resolution was 1920×1080 , the frame rate was set to 30 frames per second (FPS), and the GoPro had built-in video stabilization. Every 55 ms, the GPS sensor registered information including latitude, longitude, altitude, speed, and a coordinated universal time (UTC) stamp.

To collect data and to evaluate our data analysis algorithms, the route driven was defined by considering the coverage of various road types. For instance, a motorway with multiple lanes, city traffic with traffic lights, buses, and pedestrians, road sections with tunnels or roundabouts, and other mixed traffic were considered. The recording took place between 9 and 10 a.m. on a typical workday. The recorded video was split into manageable sequences. Moreover, the GPS data were extracted to a JSON file by an online tool [43]; then, the GPS file related to the video sequences was split.

In our evaluation, the system ran at around 10–15 FPS on a medium- to a high-end desktop computer with an Intel i7-7700k CPU and NVIDIA GTX 1080ti GPU. This gave a processing time of 60–100 ms per frame. Considering that the videos were captured at 30 FPS, this meant that the system performed at roughly half the speed of the videos themselves.

In the following, we will describe the approaches used to answer each research question and the outcomes.

A. RQ1. Detect number and type of vehicles in nearby lanes

We followed two steps to estimate the vehicles' positions and count them in each lane, namely, vehicle detection and lane detection. The first step was vehicle detection, which was done based on the existing object detection implementation called YOLO [23], [36], [37]. We chose to use YOLO because it is a real-time object detection algorithm. The selected implementation was trained on the COCO dataset [44]. We adapted YOLO to make it fit with our objectives. In the second step, image processing techniques were used to detect the lanes on the road. To achieve this goal, we experimented and compared various edge detection methods: Sobel edge detection [45], Canny edge detection [39], and Prewitt edge detection [46]. As Fig. 1 shows, Sobel has too much noise, and Prewitt is able to recognize only a few edges. Canny showed a good number of lane edges without much noise. Therefore, we decided to use Canny edge detection. In addition, to find the continuous lines, we decided to use progressive probabilistic Hough transform [40], [41], which provided great results for a small computing power cost. To further reduce the computation time, the system uses the grayscale image and regions of interest (ROIs) approach [47].

The steps applied in the lane detection algorithm are shown in Fig. 2. The output of Canny edge detection, cropping of the image, and progressive probabilistic Hough transform for lane detection are displayed in Fig. 3.

To answer RQ1, the findings of the vehicle detection and lane detection were merged and processed to yield information about detected vehicles and their relative positions. The output of the algorithm can be seen in Fig. 4. It shows the lanes and objects by using bounding boxes. It represents the number, speed, and distance of the detected bicycles, buses, vans, motorbikes, trucks, and cars in multiple lanes in front of the equipped vehicle with a monocular camera.

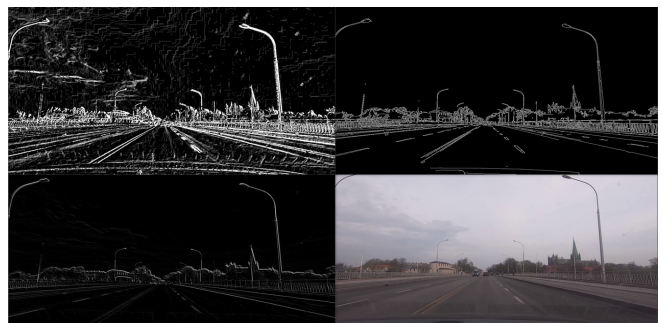


Figure 1. Comparison of different edge detection algorithms. Top left: Sobel edge detection; top right: Canny edge detection; bottom left: Prewitt edge detection; bottom right: the original image

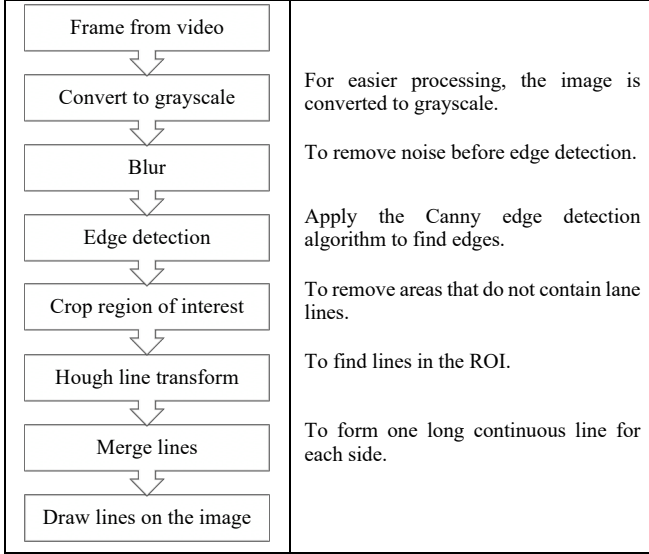


Figure 2. Steps of the lane detection algorithm

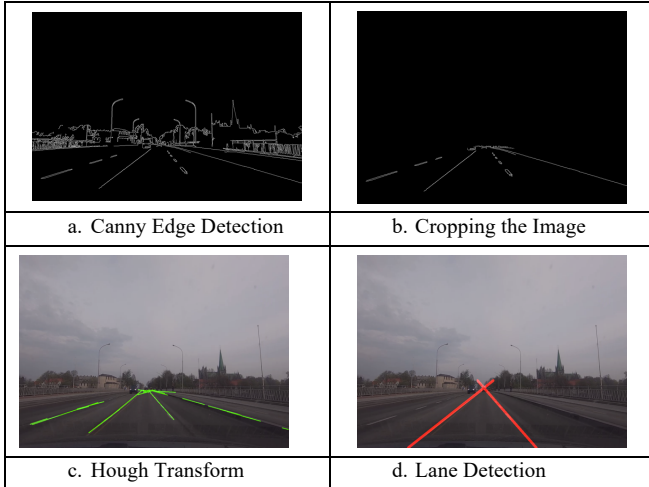


Figure 3. Outcomes of applying Canny edge detection, cropping the image, Hough transform, and lane detection

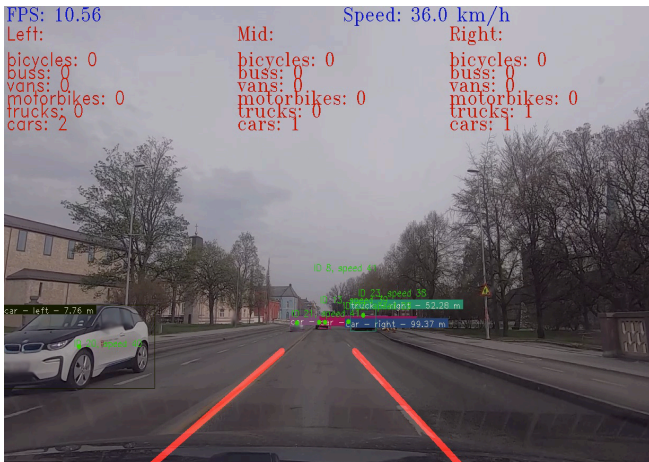


Figure 4. Lanes and vehicles detection on the road

To evaluate the proposed algorithm, three scenarios are selected with various durations and locations.

- S1. City traffic – Elgeseter Street, Trondheim
 - Includes several traffic light intersections, buses, and pedestrians
 - Video duration is equal to 4 minutes
 - 72 readings, giving a reading approximately every 3.5 seconds
- S2. Mixed traffic – Lade, Trondheim
 - Includes normal to heavy traffic, multiple traffic light intersections, crossing traffic, and surrounding parking lots
 - Video duration is equal to 3.5 minutes
 - 78 readings, giving a reading every 2.7 seconds
- S3. Mixed traffic – Tempe to Lerkendal, Trondheim
 - Includes normal to heavy traffic, and the lanes were separated by a central reservation with a medium-high fence
 - Video duration is equal to 1.5 minutes
 - 27 readings, giving a reading every 3.5 seconds

The evaluation of the algorithm is based on the comparison of the outputs with manually counted results. It is evaluated on two measures:

Measure 1: Overall ability to detect and count objects, not respecting the vehicle type.

Measure 2: Number of times vehicles were counted correctly and incorrectly in different lanes, respecting vehicle type.

The results obtained from the evaluation process of the proposed algorithm are presented in Tables I, II, III, and IV. The outputs show that the total error rate in S1, S2, and S3 is 1.0%–10.6% for measure 1. The total wrong on average for the proposed scenarios in measure 2 is 34.4%–46.3%. These findings show that the proposed algorithm is able to detect and count vehicles with high accuracy without considering their locations and types, but it is still not accurate if it focuses on identifying the type of the vehicle and its position in the lane.

B. RQ2. Using a camera to estimate the distance

We proposed a novel approach based on the pinhole camera geometry for calculating the distance of the vehicles in front in the same lane as the camera and in the left, right, and opposite lanes [48]. The pinhole camera is defined as equation 1, where d is the distance to the object, F_c is the focal length of the camera, H_o is the real height of the object, and h_o is the height of the image.

$$d = F_c \times \frac{H_o}{h_o} \quad (1)$$

We used a combination of height and width to estimate the vehicle's size and enhance the accuracy of the estimated distance. The values used for the calculations, based on

approximate sizes of vehicles, are presented in Table V. Moreover, the distance estimation algorithm is shown in Fig. 5.

TABLE I. THE OUTPUT OF SCENARIO 1 - BASED ON MEASURE 1

S1	Lane			Total
	Left	Mid	Right	
Manual	99	49	50	198
System	85	51	60	196
Error	14.1%	4.1%	20.0%	1.0%

TABLE II. THE OUTPUT OF SCENARIO 2 - BASED ON MEASURE 1

S2	Lane			Total
	Left	Mid	Right	
Manual	265	56	159	480
System	228	70	131	429
Error	14.0%	25.0%	17.6%	10.6%

TABLE III. THE OUTPUT OF SCENARIO 3 - BASED ON MEASURE 1

S3	Lane			Total
	Left	Mid	Right	
Manual	35	39	24	98
System	40	21	31	92
Error	14.3%	46.2%	29.2%	6.1%

TABLE IV. THE OUTPUT BASED ON MEASURE 2

Scenarios	Measure 2			Measure 2 (Total corrects on average)
	Counted too many in average	Counted too few in average	Total wrongs calculation on average	
S1	19.7%	14.7%	34.4%	65.6%
S2	24.3%	22.0%	46.3%	53.7%
S3	24.7%	10.0%	34.7%	65.3%

TABLE V. THE APPROXIMATE SIZES OF VEHICLES

Vehicle Type	Width	Height
Bus	2.4 m	4.0 m
Car	1.8 m	1.6 m
Motorbike/Bicycle	1.0 m	1.0 m
Truck	2.4 m	4.0 m
Van	1.9 m	2.5 m

As we did not have the ground truth of vehicle distance in our collected videos, we recorded new videos. The goal was to find the most accurate ratio of the vehicles' heights and widths. These videos captured a stationary vehicle at different distances and different angles. Then, we used a laser to measure the ground truth distance to stationary vehicles. We experimented with different ratios of height and width, and the results were compared with the ground truth from the laser. The average error with varying ratios of height and width is shown in Fig. 6. Based on the experiment results, the best ratio is 85% of the height and 15% of the width, which is affected by the detected vehicle angle.

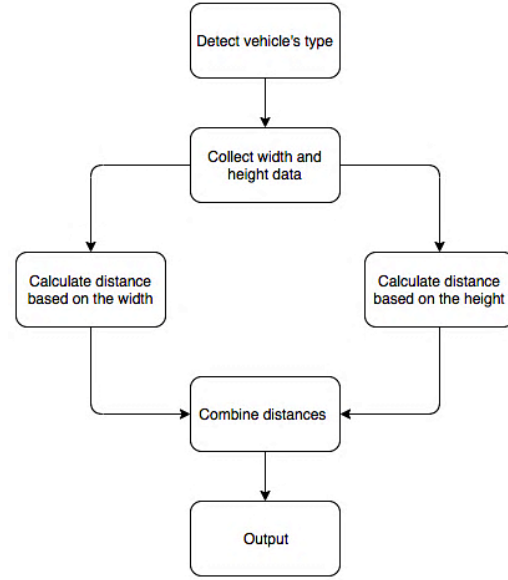


Figure 5. Distance estimation algorithm

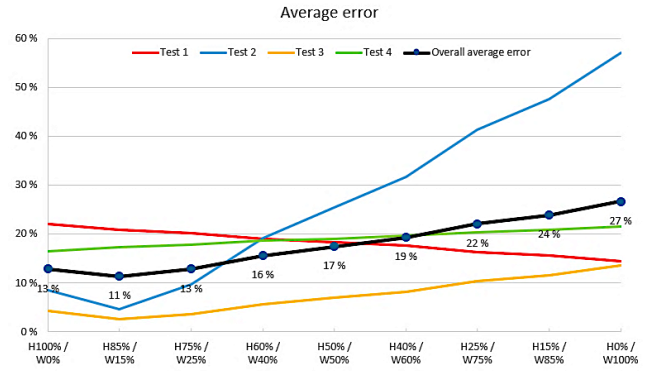


Figure 6. Average error with varying ratios of height and width

C. RQ3. Estimating the speed

Two steps were followed to answer the third research question: tracking the object in different frames and estimating the speed.

- How can we track vehicles between multiple frames? As we used a vehicle-mounted monocular camera, in a given period, the same vehicle could be viewed in the collected video. Then, the centroid of the bounding boxes identified by YOLO object detection and the Euclidean distance between a vehicle's centroids in different frames were used to track the same vehicle.

- How can we estimate the speed (v)? Based on the physics concepts, distance traveled (Δd) over time (Δt) is needed. The formula is shown in equation 2.

$$v = \frac{\Delta d}{\Delta t} \quad (2)$$

Referring to the study of Chai and Wong [49], to calculate the speed, we used the known frame rate of the camera. As the FPS in this work is 30, to estimate the speed, the average change in distance over the last 30 frames was used to find the change per second. Moreover, the camera's speed based on GPS data was considered in the estimation process.

The proposed algorithms for object tracking between multiple frames and speed estimation are shown in Fig. 7 and Fig. 8, respectively.

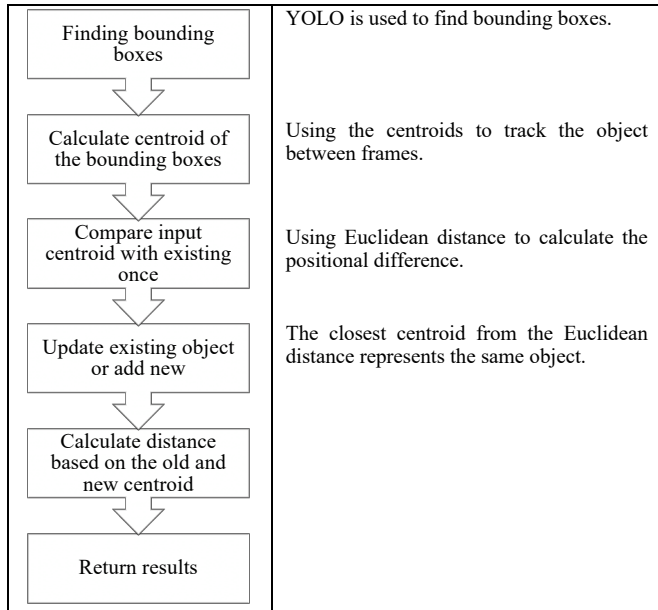


Figure 7. Steps for tracking objects

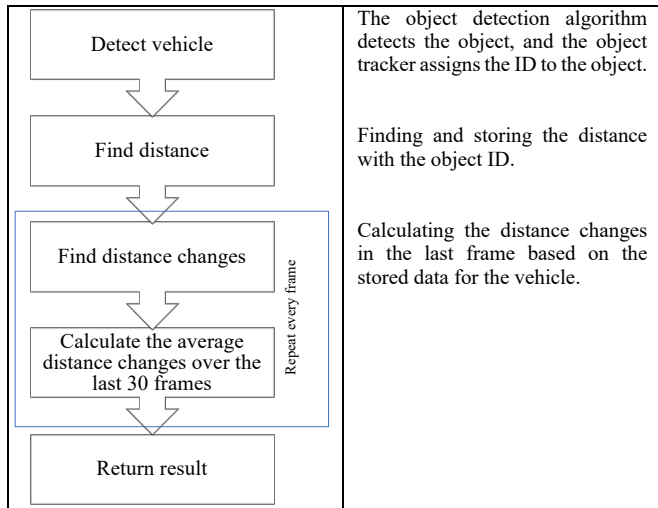


Figure 8. Speed estimation algorithm

To evaluate the algorithms, as we did not have equipment for measuring the true speed of the vehicles, we analyzed the three video sequences manually based on the estimated distance in RQ2 as the traveling distance (Δd) for each selected frame by considering the distance moved by the camera. The results are shown in Fig. 9, which shows the true

speed, based on the calculation described in the test procedure, and the speed estimated by the system. The average difference of the 75 total manual readings across all the sequences was 2.09 m/s, and the maximum difference was 10.64 m/s.

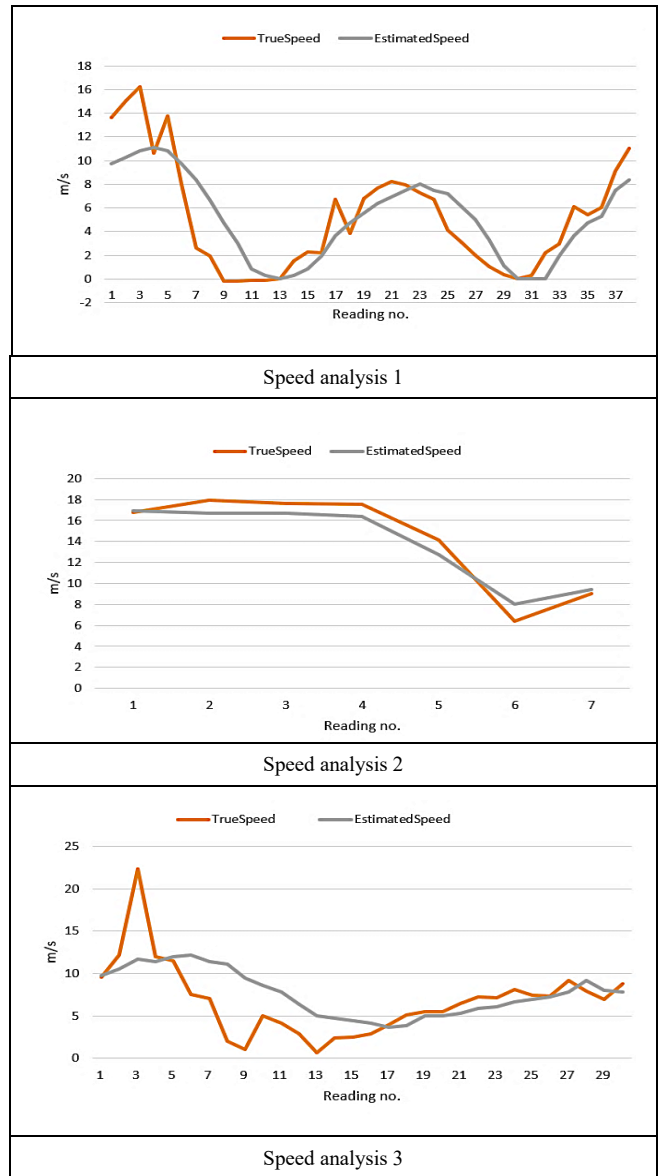


Figure 9. Speed analysis

V. DISCUSSION

This study set out with the aim of assessing the feasibility of using vehicle-mounted sensors to sense the surrounding traffic to collect and share traffic data rather than using stationary sensors on the road. Prior studies focused on collecting traffic data that can be classified into two main groups. One group of papers used stationary sensors [31], [32]. While these methods might be effective in collecting the required data, they count AVs as HDVs without considering AVs' possibilities. Moreover, this methodology is not compatible with using a vehicle-mounted camera, as it does

not consider the effect of the sensor's movement on the quality of detection. The other group of studies used vehicle-mounted sensors, which are the main focus of this study. Various autonomous and modern vehicles are equipped with different types of sensors. As there is a strong relationship between the type of sensors and a vehicle's price, some auto manufacturers might use a limited number of sensors to mitigate the vehicle's cost. Therefore, it is vital to find a method that is usable for all types of vehicle-mounted sensors. As the camera is the most common sensor, we tried to extract all the data from video. However, some studies focused on sensors' fusion, which is suitable for vehicles equipped with various types of sensors, such as LiDAR and RADAR (e.g., [25] and [27]). Although it might be effective in the correctness of detection, it is costly as well and might not be practically useful for all vehicles.

Moreover, unlike many other studies (e.g., [27], [25], and [32]), in this research, we equipped a vehicle with a camera and drove it on a defined route that contained many different scenarios to collect real traffic data.

In addition, one of the main goals of collecting traffic data is to improve the performance of intelligent traffic management systems. Therefore, considering the data type required by the traffic management system was a key point in developing our algorithms. To the best of our knowledge, no other paper has determined the number, type, distance, and speed of vehicles at the same time. Most of the papers have tried to extract one data type (e.g., [24], [25], [27], [30], [29], [31], and [32]), which is not enough for managing traffic safely and efficiently.

As a part of our research to determine the number and type of vehicles in each lane, we proposed algorithms for object detection and lane detection. The current study found that overall object detection with YOLO worked quite well. The results show that the average accuracy of vehicle detection is 92.4% in the proposed system. One unanticipated finding was the low accuracy of vehicle classification, which affected the accuracy of the position based on lanes. Almost 60% of the errors were caused by the detector misclassifying vehicles. A possible explanation for this might be that the network was trained on the COCO dataset, which contains 80 different objects, and not only objects related to traffic [44]. Hence, it could conceivably be hypothesized that using a specialized dataset for traffic objects could contribute to improving the classification accuracy, thus lowering the error rate and boosting the accuracy of the system. Moreover, the generated boundary boxed by YOLO may be unstable between frames. This result may be explained by the fact that YOLO's boundary detection approach leads to unstable detections. Non-maximum suppression could be used to fix these multiple detections [23].

Another important finding was the ability of the proposed system to detect lanes and separate vehicles in different lanes. In our evaluation of the system, this worked well on straight roads when the lane markings were clear and easily visible. However, overall lane detection results were not satisfactory with curved lanes. Edge detection with Canny edge worked as anticipated. However, sometimes the edge detector detected curbs as a lane edge. Contrary to expectations, the second-largest source of error was wrongly identified lanes. It would

be hard to address some issues with lane detection, such as that faraway lanes are difficult to detect and distinguish, and bad or non-existent lane markings create some difficulties in the detection process. We suspect that, when choosing the lane detection part of the algorithm, progressive probabilistic Hough transform [40], [41] might not be the best choice. The method proposed by Kim [50] of tracking left and right lane markings separately and utilizing an ANN that was trained to detect lines could be used to increase the accuracy. This would also have the added benefit of being able to detect lanes with curves. However, using an ANN might increase the resource demand and would also require training. The use of standardized lane sizes in the algorithm could be another future enhancement. The Norwegian Public Roads Administration [51] handbook includes different standard sizes of lanes and markings. This could ensure that different lane detections were not bigger than a set threshold. The other observed limitation of the system related to lane detection is that the lane detected on the left side was often the lane with opposing traffic, which caused noise in the results.

Another interesting finding of our study concerned estimating the vehicle distance based on the object size. We used the pinhole model and proposed an efficient ratio of object width and length to estimate the vehicle distance. Our evaluations show that combining the height and width of the detected object with a ratio of 85% and 15%, which gave the lowest amount of error with 11% on average. The unexpected finding with this idea was that camera movement and the varying sizes of the generated bounding boxes affected distance estimation. This result may be explained by the fact that images with only 2D information from a camera were used for distance estimation. More advanced equipment such as LiDAR, which generates 3D information, might be more accurate. The other limitation is caused by assuming a fixed true value for vehicle size. Moreover, the accuracy of the distance estimation could be affected by wrongly classified vehicles. Furthermore, in this study, we proved that it is feasible to estimate vehicle speed using a moving monocular camera. Our study found that the speed estimation worked as expected for vehicles in front of the camera that were driving in the same direction as the camera, with a mean difference of 2.09 m/s. However, this experiment was not accurate enough for vehicles going the opposite direction as the camera. A possible explanation for these results may be the lack of adequate time to capture the vehicles driving in the opposite direction, which is needed to calculate the speed accurately. Movement of the camera caused some errors in estimating the speed. Additionally, the estimation was not that accurate for faraway vehicles, since their determined centroid points vanished. On the other hand, as speed was calculated based on distance, any error in distance estimation had a negative effect on the correctness of the estimated speed, and a significant error in estimated speed occurred if the estimated distance suddenly spiked or varied between consecutive frames.

VI. CONCLUSION AND FUTURE WORK

This study set out to use vehicle-mounted monocular camera technology to collect the traffic data from multiple lanes required for managing traffic intelligently and

efficiently. We tried to achieve this objective by answering three research questions. First, we defined a system based on object detection algorithms and computer vision methods. Experiments on the recorded images from a predefined route in Trondheim confirmed that the proposed approach worked well for object and lane detection in that specific situation. However, more studies are needed to enhance the accuracy of the outputs and generalize the system to various situations. The second aim of this study was to investigate the effects of considering both the length and the width of the detected object in estimating the distance. The results of this investigation show that combination height and width with the ratio of 85% and 15% worked best. The third purpose of the current study was to estimate the speed of the nearby vehicles based on their distance changes over time. In general, this study proved the possibility of collecting traffic data from a camera, which is useful for managing mixed traffic. Our future work will focus on improving the performance of the proposed algorithm to minimize the error rate in real traffic. To achieve this, we will extend the system to be able to work in a broader environment and include more lanes. In addition, we will try to collect more traffic data types, which is required for traffic management systems, considering the state of the art. We will also improve the accuracy of the proposed approach by improving the object detection algorithm and using a training dataset specific to traffic. In addition, we could improve the lane detection approaches by considering the lane width standards.

REFERENCES

- [1] F. Yan, M. Dridi, and A. E. Moudni, "Autonomous vehicle sequencing algorithm at isolated intersections," in 12th Int. IEEE Conf. Intell. Transp. Syst., 2009, pp. 1–6.
- [2] G. R. d. Campos, P. Falcone, and J. Sjöberg, "Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing," in 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC 2013), 2013, pp. 1456–1461.
- [3] S. Adams and M. J. Rutherford, "Towards decentralized waypoint negotiation," in 2012 AAAI Workshop, Toronto, ON, vol. WS-12-10, pp. 2–5.
- [4] Y. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Decentralized optimal control for connected automated vehicles at intersections including left and right turns," in 2017 IEEE 56th Annu. Conf. Decision and Control (CDC), 2017, pp. 4428–4433.
- [5] L. Makarem and D. Gillet, "Fluent coordination of autonomous vehicles at intersections," in 2012 IEEE Int. Conf. Syst., Man, Cybern. (SMC), 2012, pp. 2557–2562.
- [6] R. Krajewski, P. Themann, and L. Eckstein, "Decoupled cooperative trajectory optimization for connected highly automated vehicles at urban intersections," in 2016 IEEE Intell. Veh. Symp. (IV), pp. 741–746.
- [7] J. Ding, H. Xu, J. Hu, and Y. Zhang, "Centralized cooperative intersection control under automated vehicle environment," in 2017 IEEE Intell. Veh. Symp. (IV), pp. 972–977.
- [8] E. Namazi, J. Li, and C. Lu, "Intelligent intersection management systems considering autonomous vehicles: A systematic literature review," IEEE Access, vol. 7, pp. 91946–91965, 2019.
- [9] M. Vasirani and S. Ossowski, "A computational market for distributed control of urban road traffic systems," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, pp. 313–321, 2011.
- [10] C. Wuthishuwong and A. Traechtler, "Consensus-based local information coordination for the networked control of the autonomous intersection management," Complex Intell. Syst., vol. 3, no. 1, pp. 17–32, Mar 2017.
- [11] S. A. Fayazi, A. Vahidi, and A. Luckow, "Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP," in 2017 American Control Conf. (ACC), pp. 4920–4925.
- [12] J. Wu, A. Abbas-Turki, and A. E. Moudni, "Intersection traffic control by a novel scheduling model," in 2009 IEEE/INFORS Int. Conf. Service Operations, Logistics, Inform., pp. 329–334.
- [13] J. Wu, A. Abbas-Turki, and A. E. Moudni, "Discrete methods for urban intersection traffic controlling," in VTC Spring 2009 – IEEE 69th Veh. Techn. Conf., pp. 1–5.
- [14] F. Ze-hua, L. Han-bo, H. Wei, and Y. Tian, "District control strategy for vehicle collision avoidance based on hybrid automaton model," in Proc. 11th World Congr. Intell. Control Autom., 2014, pp. 5378–5383.
- [15] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 5, pp. 1908–1916, 2019.
- [16] Y. Wu, H. Chen, and F. Zhu, "DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles," Transp. Res. Part C: Emerg. Technol, vol. 103, pp. 246–260, 2019.
- [17] Y. Mo, M. Wang, T. Zhang, and Q. Zhang, "Autonomous cooperative vehicle coordination at road intersections," in 2018 IEEE Int. Conf. Commun. Syst. (ICCS), pp. 192–197.
- [18] N. Aloufi and A. Chatterjee, "Autonomous vehicle scheduling at intersections based on production line technique," in 2018 IEEE 88th Veh. Techn. Conf. (VTC-Fall), pp. 1–5.
- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conf. Comput. Vision Pattern Recognition, 2014, pp. 580–587.
- [20] R. Girshick, "Fast r-cnn," in Proc. IEEE Int. Conf. Comput. Vision, 2015, pp. 1440–1448.
- [21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances Neural Inf. Process. Syst., 2015, pp. 91–99.
- [22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in Proc. IEEE Int. Conf. Comput. Vision, 2017, pp. 2961–2969.
- [23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vision Pattern Recognition, 2016, pp. 779–788.
- [24] S. Godha, "On-road obstacle detection system for driver assistance," Asia Pacific J. Engineering Science and Technology, vol. 3, no. 1, pp. 16–21, 2017.
- [25] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3D-LIDAR and color camera data," Pattern Recognition Lett., vol. 115, pp. 20–29, 2018.
- [26] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proc. IEEE Conf. Comput. Vision Pattern Recognition, 2017, pp. 7263–7271.
- [27] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR-camera fusion for road detection using fully convolutional neural networks," Robot. Auton. Syst., vol. 111, pp. 125–131, 2019.
- [28] D.-Y. Huang, C.-H. Chen, T.-Y. Chen, W.-C. Hu, and K.-W. Feng, "Vehicle detection and inter-vehicle distance estimation using single-lens video camera on urban/suburb roads," J. Visual Commun. Image Representation, vol. 46, pp. 250–259, 2017.
- [29] J. Lee, "Intervehicle distance estimation through camera images," J. Electron. Imag., vol. 27, no. 6, p. 063001, 2018.
- [30] S. Chadwick, W. Madder, and P. Newman, "Distant vehicle detection using radar and vision," arXiv preprint, arXiv:1901.10951, 2019.
- [31] J. Gerát, D. Sopiak, M. Oravec, and J. Pavlovicová, "Vehicle speed detection from camera stream using image processing methods," in 2017 Int. Symp. ELMAR, pp. 201–204: IEEE.

- [32] M. G. Moazzam, M. R. Haque, and M. S. Uddin, "Image-based vehicle speed estimation," *J. Comput. Commun.*, vol. 7, no. 6, pp. 1–5, 2019.
- [33] QMUL Junction Dataset, Chinese University of Hong Kong, Feb. 2018. [Online]. Available: http://personal.ie.cuhk.edu.hk/~ccloy/downloads_qmul_junction.html
- [34] K. Pfeffers, T. Tuunanen, C. Gengler, M. Rossi, W. Hui, V. Virtanen, and J. Bragge, "The design science research process: A model for producing and presenting information systems research," in *Proc. 1st Int. Conf. Design Sci. Research Inf. Syst. Techn. (DESRIST 2006)*, Claremont, CA, USA, pp. 83–106.
- [35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 31st Conf. on Neural Information Processing Systems (NIPS), 2017.
- [36] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint, arXiv:1804.02767, 2018.
- [37] Ayooshkathuria/pytorch-yolo-v3. [Online]. Available: <https://github.com/ayoooshkathuria/pytorch-yolo-v3>
- [38] G. Bradski, "The opencv library," *Dr Dobb's J. Software Tools*, vol. 25, pp. 120–125, 2000.
- [39] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognition*, vol. 34, no. 3, pp. 721–725, 2001.
- [40] C. Galambos, J. Matas, and J. Kittler, "Progressive probabilistic Hough transform for line detection," in *Proc. 1999 IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition (Cat. No PR00149)*, vol. 1, pp. 554–560: IEEE.
- [41] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Comput. Vision Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.
- [42] "Gopro hero 7 black." [Online]. Available: <https://shop.gopro.com/EMEA/cameras/hero7-black/CHDHX-701-master.html>.
- [43] "Gopro telemetry extractor." [Online]. Available: <https://tailorandwayne.com/gopro-telemetry-extractor/#!>.
- [44] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," in *European Conf. Comput. Vision*, 2014, pp. 740–755: Springer.
- [45] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," *Int. J. Comput. Sci. Manag. Res.*, vol. 2, no. 2, pp. 1578–1583, 2013.
- [46] A. Seif, M. M. Salut, and M. N. Marsono, "A hardware architecture of Prewitt edge detection," in *2010 IEEE Conf. Sustain. Utilization and Develop. Eng. Technol.*, pp. 99–101.
- [47] G. Deng and Y. Wu, "Double lane line edge detection method based on constraint conditions Hough transform," in *2018 17th Int. Symp. Distrib. Comput. Appl. Business Eng. Sci. (DCABES)*, pp. 107–110: IEEE.
- [48] K.-Y. Park and S.-Y. Hwang, "Robust range estimation with a monocular camera for vision-based forward collision warning system," *Scientific World J.*, vol. 2014, 2014.
- [49] C. Chai and Y. D. Wong, "Automatic vehicle classification and tracking method for vehicle movements at signalized intersections," in *2013 IEEE Intell. Veh. Symp. (IV)*, pp. 624–629.
- [50] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, pp. 16–26, 2008.
- [51] Vegdirektoratet, "Vegoppmerking," 2015. [Online]. Available: <https://www.vegvesen.no/attachment/69741>