Identifying and Counting Vehicles in Multiple Lanes by Using a Low-Cost Vehicle-Mounted Sensor for Intelligent Traffic Management Systems

Elnaz Namazi[1[0000-0002-6503-9315]], Jingyue Li[1[0000-0002-7958-391X]],

Rudolf Mester[1[0000-0002-6932-0606]], and Chaoru Lu[2]

[1] Department of Computer Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
{elnaz.namazi, jingyue.li, rudolf.mester}@ntnu.no

[2] Department of Civil and Environmental Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway
chaoru.lu@ntnu.no

Abstract. There is evidence that accessing online traffic data is a key factor to facilitate intelligent traffic management, especially at intersections. With the advent of autonomous vehicles (AVs), new options for collecting such data appear. To date, much research has been performed on machine learning to provide safe motion planning and to control modern vehicles such as AVs. However, few studies have considered using the sensing features of these types of vehicles to collect traffic information of the surrounding environment. In this study, we developed new algorithms to improve a traffic management system when the traffic is a mixture of human-driven vehicles (HDVs) and modern vehicles with different levels of autonomy. The goal is to utilize the sensing ability of modern vehicles to collect traffic data. As many modern vehicles are equipped with vehicle-mounted sensors by default, they can use them to collect traffic data. Our algorithms can detect vehicles, identify their type, determine the lane they are in, and count the number of detected vehicles per lane by considering multi-lane scenarios. To evaluate our proposed approach, we used a vehicle-mounted monocular camera. The experimental work presented here provides one of the first investigations to extract real traffic data from multiple lanes using a vehicle-mounted camera. The results indicate that the algorithms can identify the detected vehicle's type in the studied scenarios with an accuracy of 95.21%. The accuracy of identifying the lane the detected vehicle is in is determined by two proposed approaches, which have accuracies of 91.01% and 91.73%.

Keywords: Lane Detection, Multiple Lanes, Vehicle Detection, Intelligent Traffic Management, Vehicle-Mounted Monocular Camera.

1 Introduction

There is a growing body of literature that recognizes the importance of collecting traffic data in intelligent traffic management systems. Developments in machine learning techniques and sensors' capabilities have led to proposing various approaches for collecting different types of traffic data (e.g., [1]). These data can be used to manage traffic safely and efficiently, especially at intersections [2]. When focusing on intersection management systems, detecting vehicles' types [3], identifying the lanes they are in, and counting the number of vehicles per lane are vital to provide a global view of the intersection to manage the traffic with high performance.

Previous research on collecting traffic data has mostly used stationary sensors, which are affected by the brightness and weather condition, besides having high installation and maintenance costs. Moreover, equipping all streets with these types of sensors can be costly. The main contribution of our research is taking advantage of the sensing capabilities of modern vehicles, e.g., AVs, which are equipped with various types of sensors, to collect data of the surrounding vehicles to manage traffic. Moreover, this idea is reachable in pure AVs traffic and mixed traffic (a combination of HDVs and AVs), as managing mixed traffic is one of the most important issues for the near future, since changing all vehicles to autonomous versions will be a time-consuming process. Even after this period, traffic might include HDVs as well, because some people enjoy driving. Another contribution of this research is proposing an approach which is generalizable with various levels of vehicle autonomy. Therefore, we used a vehicle-mounted monocular camera, which is one of the cheapest sensors, so there is a high probability that most modern vehicles will be equipped with one. Moreover, by using the camera vision, we are able to record video from multiple lanes. Therefore, we used the camera data to analyze the surrounding traffic. Our developed algorithms are able to detect and classify vehicles in multiple lanes, detect the lanes next to the equipped vehicle, determine the location of the detected vehicles, and count the number of vehicles in each lane. By accessing this information and sharing it with traffic management systems, these systems would have a better global view of the environment and would be able to make better traffic management decisions, especially at intersections.

Our proposed algorithms attempt to answer two research questions:

- **RQ1.** How can we enhance the accuracy of detected vehicles' types based on existing object detection algorithms?
- **RQ2.** How can we identify the lane the detected vehicle is in on multi-lane streets to estimate the number of vehicles in each lane?

The remainder of the paper proceeds as follows. The next section summarizes related works. Chapter 3 explains the research methodology used in this study. The implementation to answer the proposed RQs is described in chapter 4. Chapter 5 presents the experimental results on real traffic data. The last chapter discusses the findings and concludes.

2 State of the Art

In the past few years, a considerable amount of literature has been published on vehicle detection, lane detection, lane-keeping, and tracking for driver assistant systems (e.g., [4]).

Target detection algorithms can be classified into three categories [5]. The first category is the digital image processing approach, such as the frame difference (FD) approach. The second one is a machine learning approach, which is usually based on an AdaBoost classifier or support vector machine (SVM). The last category is based on deep learning approaches. The proposed algorithms in this group are based on convolution neural networks (CNN), Fast-RCNN, Faster-RCNN, YOLO (You Only Look Once), etc. [5].

To improve the object detection performance, Tian et al. [5] proposed a hybrid method, which combined the FD method and YOLO. The results show that this approach can improve the bounding boxes' precision. Moreover, they introduced a model to estimate the distance and speed of the targets based on video from a stationary monocular camera in real time. To detect and track objects and estimate distance and motion in real time, Chen et al. [4] proposed an approach based on deep learning. First, they compared YOLOv3 with a single shot detector (SSD). Second, their object distance estimation was developed based on the Monodepth algorithm. Third, they proposed a new method to analyze object behavior based on SSD. To validate the proposed methodology, they used real traffic from a city center and a railway.

Moreover, different methodologies have been proposed for lane detection. Hillel et al. classified the purpose of lane understanding into lane departure warning, adaptive cruise control (ACC), lane keeping, lane centering, lane change assist, turn assist, fully autonomous driving for paved roads, and fully autonomous driving for cross-country trips [6]. Lane boundary tracking generally includes three major steps [7]. The first step is lane marking detection. In this step, various types of sensors, such as a camera (e.g., [8]), lidar (e.g., [9]), radar, GPS (e.g., [10]), and a line sensor camera (e.g., [11]), can be used. The second step is lane boundary estimation, which includes position, object type, lane information, and vehicle information. The last step is lane boundary tracking. In this step, different filtering approaches such as a Kalman filter, extended Kalman filter, unscented Kalman filter, and particle filter are used [7].

Jo et al. [12] proposed a new method to build an accurate lane-level road map based on a stereo camera, GPS, and in-vehicle sensors. The lane map generation process includes two main steps. The first step is pre-processing, which includes global optimization, ego-motion estimation, and lane detection. The second step includes coordination conversion, clustering, and polyline fitting. Jia et al. [13] proposed a sequential monocular road detection algorithm. The algorithm is classified into sequential road modeling, probabilistic segmentation, and boundary refinement. The current image, previous image, and previous road maps are the input to this process, and the current road map is its output. The multi-lane detection approach is proposed by Chao et al. based on the deep convolutional neural network. The full connected network (FCN) is applied to the captured image by the monocular camera to extract the lane boundary feature. On the image, perspective transform, Hough transform, and the least square method are applied for the lane fitting [14].

Cao et al. [15] proposed a lane detection algorithm that considered dynamic environments and complex road conditions. It is based on the superposition threshold algorithm and the random sample consensus (RANSAC) algorithm. Another approach proposed color-based segmentation for lane detection; it used global convolution networks (GCN), residual-based boundary refinement, and Adam optimization [16]. Yuan et al. introduced a new approach to segmentation and lane detection [17]. It was based on a normal map, an adaptive threshold segmentation method, denoising operations, Hough transform, and the vanishing point.

3 Research Methodology

3.1 Research strategy

A case study approach was chosen to evaluate the effectiveness of the proposed algorithms with real traffic in an urban area. A vehicle-mounted monocular camera was driven on a predefined path in Trondheim, Norway. For the purpose of data analysis, the recorded video was divided into smaller scenarios. Five scenarios were selected by considering the situation coverage and the research scope. The studied scenarios are presented in Table 1.

Table 1. Scenarios.

Scenarios	Description	Total frames
S1	Includes streets with 4 lanes and 3 lanes (1 left and 2 right).	994
S2	Includes streets with 4 lanes and 1 reserved lane in the center, 1 four-way intersection with a red traffic light and 2 traffic lights at two-way intersections.	533
S3	Includes a 4-lane street, 1 red traffic light at a four-way intersection, 1 green traffic light at a four-way intersection, and 1 red traffic light at a two-way intersection.	2249
S4	Includes a 4-lane street with a guardrail in the center, 1 green traffic light at a curved four-way intersection, 1 red traffic light at a curved intersection, and 1 red traffic light at a four-way intersection.	1819
S5	Includes 4-lane and 2-lane streets and 1 red traffic light at a three-way intersection.	2278

3.2 Data collection

To test our proposed algorithms with real traffic, we decided to record our own footage. Therefore, we equipped a vehicle with a front-facing GoPro Hero 7 camera [18]. The video resolution and frame rate were 1920 × 1080 and 30 frames per second (FPS), respectively. The GPS information includes latitude, longitude, altitude, speed, and a UTC stamp.

The equipped vehicle was driven along the predefined path in Trondheim, Norway, between 9 and 10 a.m. on a typical workday. In this experiment, we focused on city traffic with various numbers of lanes, intersections, and traffic lights.

The recorded video was split into small scenarios to be analyzable, and one frame was analyzed in every 30. The experiments were run using a desktop computer with an Intel Core i7-4770k CPU 3.40 GHz × 8 and Intel Haswell Desktop graphics.

The data telegram is defined as follows:

- Type of the detected vehicles
- Location of the detected vehicles on the multi-lane streets
- Number of vehicles in each lane

4 Implementation

By extending existing vehicle detection and lane detection algorithms, the proposed method is able to extract the information of the traffic surrounding the camera-mounted vehicle. Several existing algorithms and libraries have been widely applied for vehicle detection and classification, such as YOLO ([24, 25]), PyTorch [19], and OpenCV [20]. Since YOLO is able to run in real-time vehicle detection and classification based on the global context in the image and a single network evaluation [27], it has the potential to provide traffic information to help with real-time traffic management systems [5]. In order to detect lanes, the results of comparing three different edge detection algorithms—Sobel edge detection, Canny edge detection, and Prewitt edge detection—show that Canny edge detection is able to detect the required lanes with less noise than the other two [3]. Therefore, in this paper, we used Canny edge detection [21] and progressive probabilistic Hough transform [22, 23] to deal with lane detection.

The major goal of this paper is to propose a method which can provide lane-based traffic information by extracting data from video via a vehicle-mounted monocular camera. In our last paper [3], we proved that a vehicle-mounted monocular camera can collect traffic data, such as the speed and distance of the detected vehicles. However, traffic management systems need more detailed information on each lane. In this paper, we focused on localizing the detected vehicles in each lane.

4.1 RQ1. Vehicle type detection

As we mentioned before, we used YOLO to do vehicle detection and classification. YOLO was originally trained on the COCO dataset, which includes 80 object categories, such as car, cat, umbrella, cell phone, etc. Therefore, the accuracy of the model is not good enough to extract real-world traffic data [3]. Since the traffic management only requires traffic objects, a pre-trained weight on the KITTI dataset was used to train YOLO to enhance its accuracy in classifying traffic objects. The KITTI dataset focuses on traffic objects and contains eight categories named car, van, truck, pedestrian, person_sitting, cyclist, tram, and misc. [28]. The proposed system architecture is shown in Fig. 1. As shown in Fig. 1, the input of the system is the recorded videos from real-world traffic, as described in section 3.2. The algorithm is based on YOLO trained on the KITTI dataset. Moreover, the output of the system is the processed videos. In these videos, bounding boxes are drawn around the detected vehicles, and the types of detected vehicles are identified. Moreover, lane markers are detected and highlighted. This information is recorded in JSON files for further analysis.

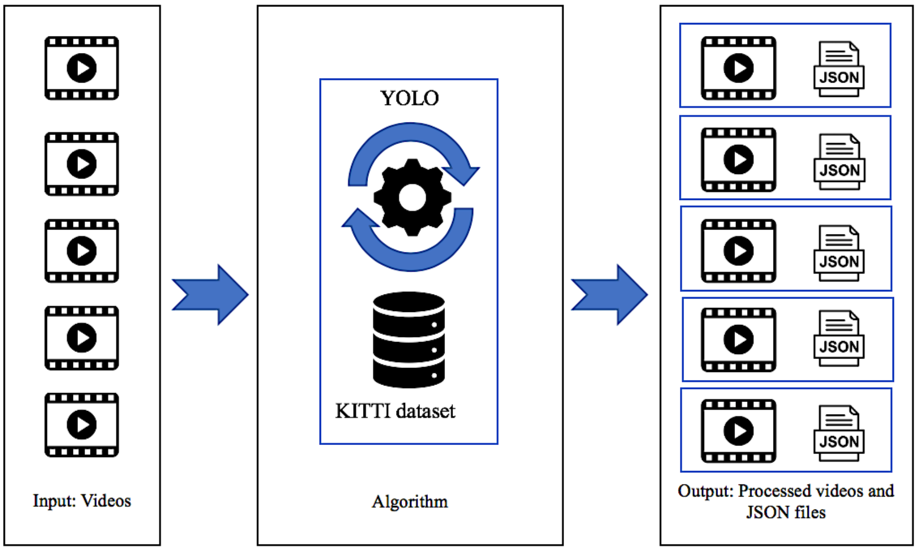


Fig. 1. System architecture.

4.2 RQ2. Extracting traffic data

To answer this research question, we followed three steps, as shown in Fig. 2. The first step was to identify the nearby lanes on both sides of the equipped vehicle. To do this we converted the extracted frames to grayscale to reduce the processing time. To remove the noise, frames were blurred. After that, as we mentioned in section 4, we used Canny edge detection [21], and the regions of interest (RoI) [29] to reduce the computation time. Moreover, progressive probabilistic Hough transform [22, 23] is applied to detect lines. After that, lines were drawn on top of the frames, which are shown in green in Fig. 3 and Fig. 4. The parametrization for the detected lines is based on the starting point (x_1, y_1) and ending point (x_2, y_2) of the line in the defined RoI.

The second step is detecting vehicles and dividing them into three groups. To do this, based on the distance between a central point on the bottom side of the bounding boxes around the detected vehicles and detected lanes, we classified vehicles into three groups, named left, middle, and right. To classify the vehicles, we followed these rules: If the vehicles were driven in the same lane as the equipped vehicle, we classified them as middle; if they were to the left side of that vehicle, we classified them as left; and others were classified as right. The conditions to make these decisions are shown in Table 2. This table includes three figures, in which green lines are the detected lanes on both sides of the equipped vehicle; they are named the left line (LL) and right line (RL). Bounding boxes around the detected vehicle are shown as a red rectangle. The central point on the bottom side of the bounding box is named “central point” (CP). Blue arrows represent the conditions, which are called left of the left line (LoL), left of the right line (LoR), right of the left line (RoL), and right of the right line (RoR).

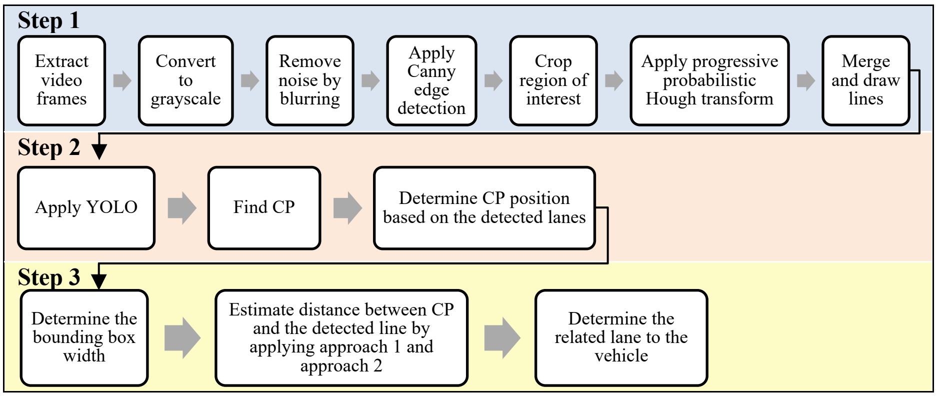


Fig. 2. Lane detection, object detection, and location estimation.

In the third step, we identify the location of each vehicle in multiple lanes. The idea is based on the assumption that vehicle size is less than the lane width. So, the vehicle location is identified based on the distance between the CP in the bounding box around the detected vehicle and the detected lane which that vehicle is in. The distance is measured by two proposed approaches as follows.

Approach 1.
In the first approach, we estimate the location of the detected vehicle based on the shortest path between the CP and the related lane. The shortest distance between a point and a line which is defined by two points, is presented in equation (1) [30]. The distance (D_i) of the point CP on the bounding box around the vehicle i, which is expressed by $(x_{vi,0}, y_{vi,0})$ from the line which passes through two points, $P_1:=(x_1,y_1)$ and $P_2:=(x_2,y_2)$, is as follows:

$$\text{distance}\left((P_1, P_2), \left(x_{vi,0}, y_{vi,0}\right)\right) = \frac{\left|(x_2-x_1)\left(y_1-y_{vi,0}\right)-\left(x_1-x_{vi,0}\right)(y_2-y_1)\right|}{\sqrt{(x_2-x_1)^2+\left(y_2-y_1\right)^2}} \quad (1)$$

This approach is presented in Fig. 3. In this figure, similar to Table 2, green lines are the detected lanes on both sides of the equipped vehicle, called LL and RL. Red rectangles are bounding boxes around the detected vehicle. CP represents the central point on the bottom side of the bounding box. Blue arrow which is called D_i, shows the shortest distance between a CP on vehicle i and a related line. W_{vi} shows the width of the vehicle i.

Approach 2.
In this approach, we propose a solution to estimate the vehicle distance (d_i) to the related line in the horizontal direction, as shown in Fig. 4. Other variables are named as in Fig. 3.

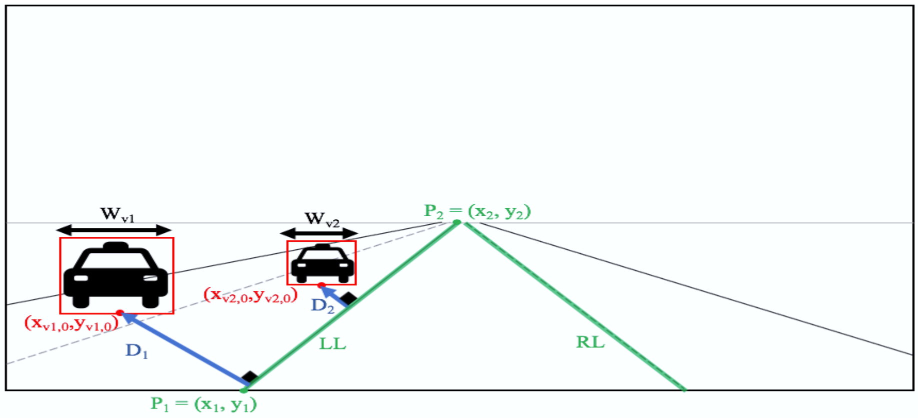


Fig. 3. The first approach to estimating the lane the detected vehicle is in.

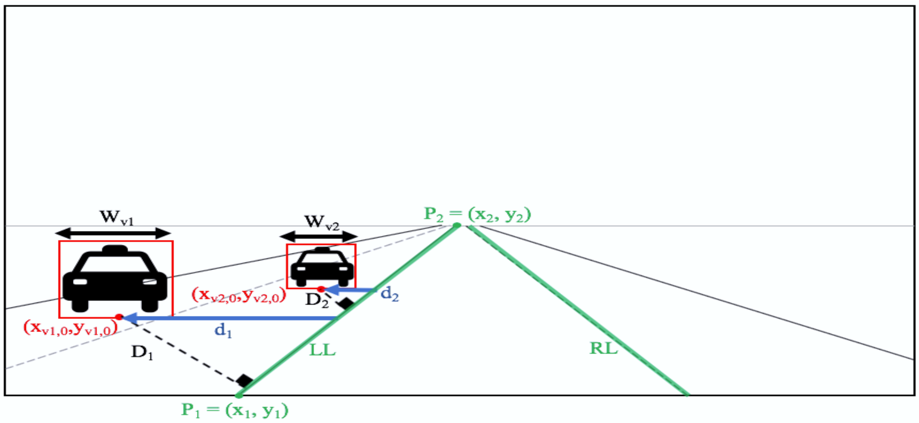
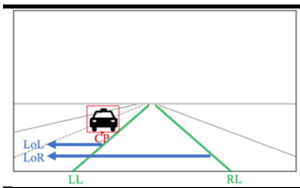
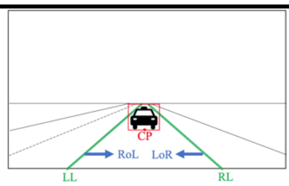
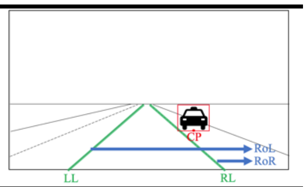


Fig. 4. The second approach to estimating the lane the detected vehicle is in.

Table 2. Dividing vehicles into three main groups, left, middle, and right.

		
If the CP is located on the left side of the left line and on the left side of the right line, then the vehicle is on the left side.	If the CP is located on the right side of the left line and on the left side of the right line, then the vehicle is in the middle.	If the CP is located on the right side of the left line and on the right side of the right line, then the vehicle is on the right side.

To measure d_i, our proposed approach consists of the following steps.

1- Measuring the slope of the related line (jL, j:=L or R), which passes through two points, P_1 and P_2 [31].

$$\text{Slope}_{jL} = \frac{(y_2 - y_1)}{(x_2 - x_1)} \tag{2}$$

2- Converting the line's slope to an angle in degrees [31].

$$jL_{degree} = \arctan\left(\text{Slope}_{jL}\right) \tag{3}$$

3- Estimating d_i by using triangulation formulas, as shown in Fig. 5.

Based on Euclidean parallelism [26],

$$L \parallel d_i \quad \Rightarrow \quad \beta = \propto = \gamma = N_{degree} \tag{4}$$

$$D_i \perp N \quad \Rightarrow \quad \lambda = 90° \tag{5}$$

By considering the triangle rules [32],

$$\theta + \lambda + \gamma = 180° \;\Rightarrow\; \theta = 180° - 90° - \gamma \quad \Rightarrow \quad \theta = 90° - N_{degree} \tag{6}$$

Based on the trigonometric ratios, the hypotenuse (d_i) is calculated by the following formula [33]:

$$d_i = \frac{D_i}{\cos(\theta)} \quad \Rightarrow \quad d_i = \frac{D_i}{\cos\left(90° - N_{degree}\right)} \tag{7}$$

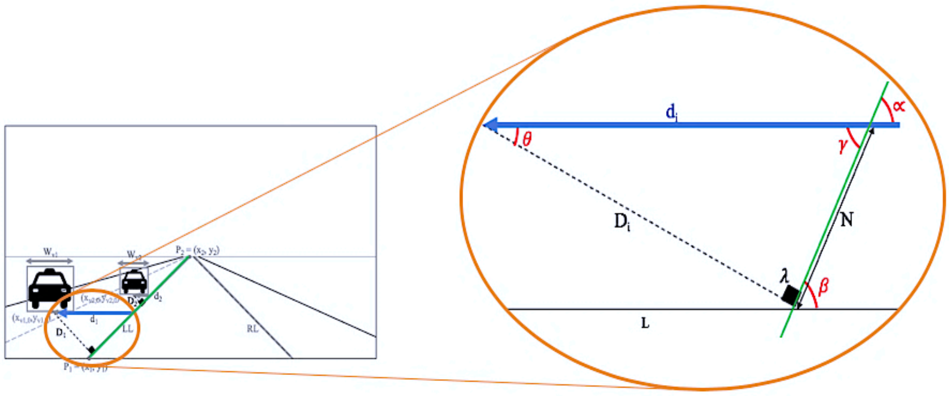


Fig. 5. Identifying the vehicle's lane by the second approach.

Finally, as the last step, the location of the vehicle is estimated by considering the distance and vehicle size, as shown in Table 3, in which Distance_i is the distance calculated for a vehicle i by following approach 1 and approach 2, and S_{vi} is the size of vehicle i.

Table 3. Conditions for finding the detected vehicle's location in multiple lanes.

Condition	Output
$0 < Distance_i < S_{vi}$	1st lane on the left/right
$S_{vi} < Distance_i < 2 \times S_{vi}$	2nd lane on the left/right
$2 \times S_{vi} < Distance_i < 3 \times S_{vi}$	3rd lane on the left/right
$(n-1) \times S_{vi} < Distance_i < n \times S_{vi}$	nth lane on the left/right

We ran our algorithms on predefined scenarios and extracted the frames. Out of every 30 frames, we analyzed one frame manually as a ground truth. In this study, the scenarios include 7873 frames in total, and we analyzed 262 frames. Then, the outputs of the algorithms were compared with the manually extracted data.

5 Results

The purpose of the first experiment was to determine the accuracy of the improved algorithms in identifying the type of the detected vehicles. Table 4 illustrates our results. It is apparent from this table that the accuracy of identifying the detected vehicles' type is higher than 90.74% for all lanes in the studied scenarios.

Table 4. Vehicle type detection in the predefined scenarios.

		2nd lane on the left	1st lane on the left	Middle	Right	Total
Scenario 1	Manual	5.00	19.00	22.00	2.00	48.00
	System	5.00	19.00	22.00	0.00	46.00
	Correct (%)	100.00	100.00	100.00	0.00	95.83
	Error (%)	0.00	0.00	0.00	100.00	4.17
Scenario 2	Manual	0.00	12.00	9.00	0.00	21.00
	System	0.00	12.00	9.00	0.00	21.00
	Correct (%)	100.00	100.00	100.00	100.00	100.00
	Error (%)	0.00	0.00	0.00	0.00	0.00
Scenario 3	Manual	0.00	70.00	24.00	1.00	95.00
	System	0.00	68.00	24.00	1.00	93.00
	Correct (%)	100.00	97.14	100.00	100.00	97.89
	Error (%)	0.00	2.86	0.00	0.00	2.11
Scenario 4	Manual	1.00	2.00	44.00	7.00	54.00
	System	1.00	1.00	42.00	5.00	49.00
	Correct (%)	100.00	50.00	95.45	71.43	90.74
	Error (%)	0.00	50.00	4.55	28.57	9.26
Scenario 5	Manual	10.00	11.00	5.00	48.00	74.00
	System	9.00	10.00	5.00	45.00	69.00
	Correct (%)	90.00	90.91	100.00	93.75	93.24
	Error (%)	10.00	9.09	0.00	6.25	6.76

In the second experiment, we analyzed the identified location of the detected vehicles in each lane. The results obtained from the selected scenarios are shown in Table 5. App 1 and App 2 indicate approach 1 and approach 2, respectively. The results obtained from the experiments show that the accuracy of vehicle location identification is between 71.43% and 90.54% with the first approach, and between 71.43% and 94.59% with the second approach, for all lanes.

Table 5. Vehicle location detection in the predefined scenarios.

Scenarios	Outputs	2nd lane on the left		1st lane on the left		Middle		Right		Total	
		App 1	App 2	App 1	App 2	App 1	App 2	App 1	App 2	App 1	App 2
Scenario 1	Correct (%)	80.00	100.0	94.74	84.21	86.36	86.36	100.0	100.0	89.58	87.50
	Error (%)	20.00	0.00	5.26	15.79	13.64	13.64	0.00	0.00	10.42	12.50
Scenario 2	Correct (%)	100.0	100.0	75.00	75.00	66.67	66.67	100.0	100.0	71.43	71.43
	Error (%)	0.00	0.00	25.00	25.00	33.33	33.33	0.00	0.00	28.57	28.57
Scenario 3	Correct (%)	100.0	100.0	80.00	80.00	95.83	95.83	100.0	100.0	84.21	84.21
	Error (%)	0.00	0.00	20.00	20.00	4.17	4.17	0.00	0.00	15.79	15.79
Scenario 4	Correct (%)	100.0	100.0	100.0	100.0	90.91	90.91	71.43	71.43	88.89	88.89
	Error (%)	0.00	0.00	0.00	0.00	9.09	9.09	28.57	28.57	11.11	11.11
Scenario 5	Correct (%)	70.00	100.0	72.73	72.73	100.0	100.0	97.92	97.92	90.54	94.59
	Error (%)	30.00	0.00	27.27	27.27	0.00	0.00	2.08	2.08	9.46	5.41

In total, the accuracy of the vehicle type detection and location identification for the vehicles with the correct type detection in all scenarios when considering all lanes is shown in Table 6. As this table shows, the accuracy of the second approach for estimating the lanes the detected vehicles are in is higher than that of the first approach.

Table 6. Total accuracy for all scenarios.

	Type identification	Localization based on App 1	Localization based on App 2
Correct (%)	95.21	91.01	91.73
Error (%)	4.79	8.99	8.27

6 Discussion and Conclusion

The main objective of this research was to study modern vehicles' sensing abilities for collecting traffic data to improve traffic management systems. To achieve this objective, we have developed a system and done experiments with real traffic data. Some of the prior studies that have noted the importance of collecting traffic data used stationary sensors to achieve this goal (e.g., [5]). As using stationary sensors are costly to equip all streets, we have used a vehicle-mounted sensor, as modern vehicles are equipped with various types of sensors, which are powerful and free resources to use.

As we have mentioned, modern vehicles are equipped with various types of sensors, but we decided to use a monocular camera to make our solution more feasible in the real world. Due to lidars are more expensive than cameras, the possibility of equipping all vehicles with a lidar is low, which will limit the generalizability of the proposed approach in reality. Therefore, we decided to use a monocular camera, which is cheap and likely to be mounted on most modern vehicles. Moreover, the camera's field of view gives us the possibility to collect data from multiple lanes to provide a better understanding of the traffic situation.

Our proposed algorithms are a combination of a deep learning algorithm called YOLO, which was trained on the KITTI dataset to detect vehicles and identify their type, and image processing approach to provide robust vehicle location estimation for multiple lanes. Although most of the existing papers in this scope have focused on lane detection (e.g., [15]) or object detection (e.g., [5]), we have combined both methodologies to extract more data types. In reviewing the literature, we found that more recent studies have been limited to lane detection and tracking for driver assistance systems (e.g., [17]). No approaches were found on the dependency between vehicle detection and the related lane, as it is vital for traffic management systems, especially at intersections, to access the traffic volume per lane.

One of the most significant findings from our proposed algorithms is that a vehicle-mounted monocular camera is able to extract traffic data, such as the detected vehicles' type, what lanes they are in, and the number of detected vehicle in each lane. Our experiments on real traffic data with five scenarios confirmed that our algorithms can identify the detected vehicles' type with an accuracy higher than 90.74%. The accuracy of vehicle location identification for all lanes with the first and second approaches is between 71.43% and 90.54%, and between 71.43% and 94.59%, respectively. The observed low accuracy of the second scenario can be explained by the fact that the lane marks on the right side almost vanished, which had a direct effect on the accuracy of the vehicle location detection. Moreover, the accuracy of identifying the lane the detected vehicle with the correct determined type was in by considering the total lanes was 91.01% for the first approach, and 91.73% for the second approach. Although this study was limited by driving an equipped vehicle in the middle lane, the findings prove that this idea would be feasible in reality. However, further experimentation to consider various scenarios is recommended. Moreover, as our proposed algorithms are based on object detection and lane detection algorithms, therefore, by enhancing the accuracy of the object detection and lane detection algorithms, the performance of our proposed algorithm would be enhanced. Our future work will improve the performance and accuracy of our approach further.

References

1. Lamouik, I., Yahyaouy, A., and Sabri, M.A.: Smart multi-agent traffic coordinator for autonomous vehicles at intersections: Book Smart multi-agent traffic coordinator for autonomous vehicles at intersections. IEEE, pp. 1-6 (2017).
2. Namazi, E., Li, J., and Lu, C.: Intelligent intersection management systems considering autonomous vehicles: A systematic literature review. IEEE Access, pp. 91946-91965 (2019).
3. Namazi, E., Holthe-Berg, R.N., Lofsberg, C. S., and Li, J.: Using vehicle-mounted camera to collect information for managing mixed traffic. 15th International Conference on Signal-Image Technology & Internet-Based Systems, pp. 222-230 (2019).
4. Chen, Z., Khemmar, R., Decoux, B., Atahouet, A., and Ertaud, J.-Y.: Real time object detection, tracking, and distance and motion estimation based on deep learning: Application to smart mobility. IEEE, pp. 1-6 (2019).
5. Tian, S., Yu, H., Yang, Z., Jing, X., Zhang, Z., Shi, M., and Wang, Y.: An improved target detection and traffic parameter calculation method based on YOLO with a monocular camera. CICTP, pp. 5696-5708 (2019).
6. Hillel, A.B., Lerner, R., Levi, D., and Raz, G.: Recent progress in road and lane detection: A survey. Machine Vision and Applications, pp. 727-745 (2014).
7. Keatmanee, C., Jakborvornphan, S., Potiwanna, C., San-Uml, W., and Dailey, M.N.: Vision-based lane keeping—A survey. IEEE, pp. 1-6 (2018).
8. Andrade, D.C., Bueno, F., Franco, F.R., Silva, R.A., Neme, J.H.Z., Margraf, E., Omoto, W.T., Farinelli, F.A., Tusset, A.M., and Okida, S.: A novel strategy for road lane detection and tracking based on a vehicle's forward monocular camera. IEEE Transactions on Intelligent Transportation Systems, pp. 1497-1507 (2018).
9. von Reyher, A., Joos, A., and Winner, H.: A lidar-based approach for near range lane detection. IEEE, pp. 147-152 (2005).
10. Goldbeck, J., Hürtgen, B., Ernst, S., and Kelch, L.: Lane following combining vision and DGPS. Image and Vision Computing, pp. 425-433 (2000).
11. Narita, Y., Katahara, S., and Aoki, M.: Lateral position detection using side looking line sensor cameras. IEEE, pp. 271-275 (2003).
12. Jo, Y., Han, S.-J., Lee, D., Min, K., and Choi, J.: An autonomous lane-level road map building using low-cost sensors. In Eleventh International Conference on Machine Vision. (ICMV 2018), vol. 11041 (2019).
13. Jia, B., Chen, J., Zhang, K., and Wang, Q.: Sequential monocular road detection by fusing appearance and geometric information. IEEE/ASME Transactions on Mechatronics, 24 (2), pp. 633-643 (2019).
14. Chao, F., Yu-Pei, S., and Ya-Jie, J.: Multi-lane detection based on deep convolutional neural network. IEEE Access, pp. 150833-150841 (2019).
15. Cao, J., Song, C., Song, S., Xiao, F., and Peng, S.: Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments. Sensors, pp. 3166 (2019).

16.Zhang, W., and Mahale, T.: End to end video segmentation for driving: Lane detection for autonomous car. arXiv preprint (2018).
17.Yuan, C., Chen, H., Liu, J., Zhu, D., and Xu, Y.: Robust lane detection for complicated road environment based on normal map. IEEE Access, pp. 49679-49689 (2018).
18. GoPro hero7 black, https://gopro.com/en/us/shop/cameras/hero7-black/CHDHX-701-master.html, last accessed 2020.
19.Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A.: Automatic differentiation in PyTorch. 31st Conference on Neural Information Processing Systems (NIPS 2017) (2017).
20.Bradski, G.: The opencv library. Software Tools, pp. 120-125 (2000).
21.Ding, L., and Goshtasby, A.: On the Canny edge detector. Pattern Recognition, pp. 721-725 (2001).
22.Galamhos, C., Matas, J., and Kittler, J.: Progressive probabilistic Hough transform for line detection. IEEE, pp. 554-560 (1999).
23.Matas, J., Galambos, C., and Kittler, J.: Robust detection of lines using the progressive probabilistic hough transform. Computer Vision and Image Understanding, pp. 119-137 (2000).
24.Redmon, J., Divvala, S., Girshick, R., and Farhadi, A.: You only look once: Unified, real-time object detection. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788 (2016).
25.Redmon, J., and Farhadi, A.: YOLOv3: An incremental improvement. arXiv preprint (2018).
26.Geometry, https://en.wikipedia.org/wiki/Parallel, last accessed 2019.
27.YOLO, https://pjreddie.com/darknet/yolo/, last accessed 2019.
28.Use YOLOv3 PyTorch to train KITTI, https://github.com/packyan/PyTorch-YOLOv3-kitti, last accessed 2019.
29.Deng, G., and Wu, Y.: Double lane line edge detection method based on constraint conditions Hough transform. IEEE, pp. 107-110 (2018).
30. Point-Line Distance--2-Dimensional, https://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html, last accessed 2020.
31. Slope, https://en.wikipedia.org/wiki/Slope, last accessed 2019.
32.Sum of angles of a triangle, https://en.wikipedia.org/wiki/Sum_of_angles_of_a_triangle, last accessed 2019.
33.Trigonometry, https://en.wikipedia.org/wiki/Trigonometry, last accessed 2019.