JOURNAL OF MECHANICAL ENGINEERING AND SCIENCES
ISSN: 2289-4659 e-ISSN: 2231-8380
VOLUME 18, ISSUE 1, 2024, 9949 – 9958
DOI: https://doi.org/10.15282/jmes.18.1.2024.11.0786

REVIEW ARTICLE

Investigation of collision estimation with vehicle and pedestrian using CARLA simulation software

M. S. Beg[1], I. M. Yusri[1,2]*

[1] Faculty of Mechanical and Automotive Engineering, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia
Phone: +6094246234; Fax: +609424222
[2] Centre for Automotive Engineering, Universiti Malaysia Pahang Al-Sultan Abdullah, Pekan, 26600, Malaysia

ABSTRACT - The effectiveness of object detection systems in diverse driving environments is crucial in the growing field of automotive safety. The increasing frequency of traffic accidents, especially at busy intersections with heavy traffic and limited visibility, highlights the pressing requirement for advanced vehicle detection systems. Prior to implementing the real-time experiment, it is advisable first to conduct a simulation in order to gain a deeper understanding of the practical implementation in real-time scenarios. On the other hand, this approach has the potential to reduce both time and cost significantly. The system utilized a software-based solution by implementing the CARLA simulator. This study aims to analyze vehicle detection at T-junctions, cross-junctions, and roundabouts using image data obtained from the CARLA platform. Subsequent analysis differentiates between vehicles and non-vehicle objects in the dataset. The model concludes by proposing Python-based integrative solutions to enhance object detection systems for diverse roads and atmospheric situations. The significance of this study is evaluating the probability of accidents by tracking key factors like vehicle speed, distance, and density on various road types. In future research, it will be essential to investigate how different weather conditions, including rain, haze, and low-light scenarios, affect the sensor performance, specifically LiDAR sensors. Advanced machine learning techniques are proposed to evaluate the effectiveness of the vehicle detection system in collecting key parameters like vehicle count, speed, and distance in junction and roundabout scenarios. These findings have important implications for the advancement of more efficient, context-aware detection systems in the automotive sector.

ARTICLE HISTORY

Received	:	24th Oct. 2023
Revised	:	17th Jan. 2024
Accepted	:	24th Jan. 2024
Published	:	30th Mar. 2024

KEYWORDS
Object detection
Safety driving
Carla simulation
Collision analysis
Vehicle's speed

1. INTRODUCTION

The focus on vehicle safety has intensified due to rising consumer demand and stricter legislation. As a result, developers and manufacturers are compelled to enhance features like vehicle detection, blind-spot monitoring, and Advanced Driver Assistance Systems (ADAS) [1-3]. The key to these advancements lies in the development of high-performance, accurate radar systems, which not only reduce the likelihood of accidents but also benefit all road users, thereby setting a new standard for automotive safety [4-6]. The automotive sector holds significant importance for national development, necessitating revisions in various areas, including legislation, manufacturing, import-export activities, and research and development, with a particular focus on traffic safety [7]. Traditional real-world simulations for testing sensor accuracy and effectiveness are both costly and hazardous [8]. Simulation software offers a safer and more cost-effective alternative, replicating real-world scenarios without endangering drivers or passengers and minimising expenses such as fuel and energy [9,10]. For instance, while distance radar sensors face challenges in severe weather conditions like snow, rain, and haze, infrared radar sensors can detect heat generated from surrounding vehicles without being affected by these conditions [11-13].

However, real-world testing of such sensors is relatively inexpensive. Simulation software like Car Learning to Act (CARLA) Simulator provides an alternative for developing and testing radar systems designed to detect vehicles in a driver's blind spot [14]. The CARLA software requires specialised programming to precisely identify and distinguish vehicles from other objects in various traffic situations, including junctions and straight sections. CARLA is a freely available simulator designed for researching autonomous driving. It offers a genuine urban setting to assess the capabilities of self-driving vehicles in different scenarios. This software facilitates the creation of algorithms for detecting objects, simulating traffic behaviour, and interacting with the environment. The extensive simulation capabilities enable researchers and engineers to construct, train, and verify autonomous driving systems in a controlled yet realistic virtual environment [15,16]. This study aims to develop a vehicle-detecting system and investigate the momentum parameter for avoiding on-road collisions.

In a recent study, Wang et al. [17] addressed the high costs and limitations of existing car radar sensor systems, particularly their poor performance in challenging weather conditions like snow, rain, and haze, as well as at the drak night. To tackle these issues, Wang developed a cost-effective optical add-on radar sensor system suitable for standard

*CORRESPONDING AUTHOR | I. M. Yusri | ✉ yusri890@yahoo.com

cars. The research involved using an SUV outfitted with millimetre-wave radar and a thermal imaging camera on the front bumper. Utilising a machine learning model called LightGBM, the system achieved a remarkable 95.8 % target accuracy. Furthermore, Wang introduced four new Haar-like templates, which led to a 2.9 % improvement in vehicle detection accuracy.

Utilising virtual simulation platforms can significantly reduce research costs in the field of autonomous driving. According to a study by Gao et al. [18], based on actual vehicles, it not only proves to be expensive but also presents limitations in feasibility. The study highlights that real-world testing environments are unable to simulate dangerous situations, a challenge that can be easily overcome using the CARLA simulator. The researchers employed that simulator for image acquisition and used Python scripts to capture images in specific formats. When comparing the performance of three algorithmic models, YOLOv4 emerged as the most efficient, boasting the highest accuracy with a mean average precision (mAP) of 65.96 % and a frames per second (FPS) rate of 16.31. That was followed by the CenterNet algorithm, which had an mAP of 54.83 % and an FPS of 12.25. The Faster-RCNN algorithm lagged, achieving an mAP of 52.67 % and an FPS of 7.43.

In a noteworthy study conducted by Cuenca et al. [19], deep learning techniques were employed to address the complexities of autonomous vehicle navigation, particularly in roundabouts. Utilising the CARLA simulation environment, the researchers developed and trained a Q-learning model to navigate two distinct scenarios: roundabouts with and without traffic. The roundabout in the simulation featured three exits, and the model's performance was rigorously evaluated after 100 training iterations. The assessment involved running 100 trajectory examples until a collision occurred, providing valuable insights into the challenges and solutions for autonomous vehicles in complex manoeuvres. In a recent study conducted by Sumbal Malik, the focus was on enhancing safety measures in vehicle driving through software simulation. Malik emphasised that human errors, including identification and judgement mistakes, are responsible for approximately 92 % of road accidents. The study employed Car Learning to Act (CARLA), a free and open-source simulator designed for autonomous city driving. CARLA was developed to facilitate the training, development, and validation of various autonomous driving technologies, including perception and control systems. Malik's research involved a comparative analysis of multiple simulators to identify the most effective tool for autonomous driving simulation. The study concluded that CARLA is highly versatile, catering to a wide range of advanced driver assistance system use cases, from developing perception algorithms to mastering driving techniques [20].

Another study by Stevic et al. [21] highlighted the challenges associated with autonomous driving, particularly those arising from the unpredictability of the physical world. They emphasised that unpredictable circumstances could lead to accidents, making it crucial to develop robust testing mechanisms. To address that, the researchers advocate for the use of virtual simulators as an essential part of the development process. They employed the CARLA simulation method to create various environmental models, which can be constructed using high-resolution LiDARs, cameras, or annotated virtual maps. The study also utilised RViz for visualising LiDAR scans, joint movements, and RGB camera images generated by CARLA. Additionally, Jang et al. [15] discuss the complexities involved in developing camera-based perception systems for autonomous vehicles. The authors highlight the need for a vast dataset of training images, which are usually collected and labelled through a labour-intensive and error-prone manual process. The challenge lies in acquiring a diverse range of real-world driving images, as it is not feasible to artificially simulate unexpected scenarios or varying weather and lighting conditions. To address that, the study employed a CARLA client to build a simulated environment using a pre-defined map. The research utilises the Darknet framework and the YOLO object detector for both the training and inference phases. Then evaluated the scalability, accuracy, and training performance of their system, dubbed CarFree. The experimental setup includes a forward-facing camera mounted on an ego vehicle, featuring a 90° field of view and a resolution of 960 × 540 pixels. The simulation also mimics varying illumination conditions by altering the sun's azimuth and altitude and allowing for changes in weather parameters like cloudiness, precipitation, and windiness.

Meanwhile, a model developed by Perez-Gil et al. [22] focused on the integration of high-precision sensors like LIDAR and cameras in autonomous vehicles. The study investigated the use of Deep Reinforcement Learning (DRL) algorithms for autonomous vehicle navigation. Specifically, the Deep Q-Network (DQN) and the Deep Deterministic Policy Gradient (DDPG) methods were examined. The study aimed to compare the performance of these algorithms in both the training and validation stages, providing insights into their general applicability. The result is that, by using an open-source simulation in the CARLA simulator environment, the end-to-end model was trained and evaluated for usability and security. Simulators may access information about the automobile, such as speed, steering angle, throttle settings, and brake positions, as well as information about the surrounding area, such as lane markings and traffic signs. CARLA provides further details on urban communities with various layouts. Other simulations, like TORCS and Udacity, aren't intended for driving in cities. Lacking are intersections, lane regulations, and other complexity, like distinguishing between urban and rural driving. The researcher gathered data that required autonomous cars to avoid obstacles, collisions with cars, and pedestrians.

This study relates to software development to create a radar system by using coding and simulation to detect surrounding vehicles in several road types and conditions, such as T-junction, cross-junction and roundabout.

2. METHODS AND MATERIALS

In this study, the CARLA simulator was used for collecting data. It is open-source software that can be used on Windows. The system required a minimum GPU of 6 GB and more than 20 GB of disc space. Python is the main scripting language in CARLA. Primarily setup PythonAPI and pip installation for the library required. Figure 1 shows the flow chart of the system. The systematic process started with the installation and setup of the CARLA simulator. It includes the steps for environment configuration, where specific scenarios and parameters within the simulator are defined. This setup phase is crucial for ensuring that the simulated conditions accurately reflect the intended real-world scenarios for data collection.

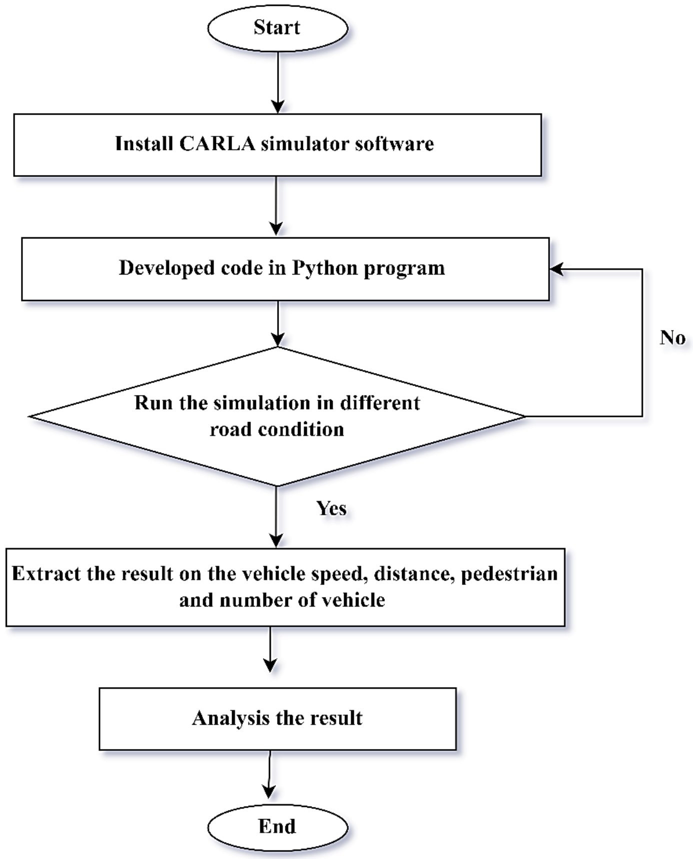
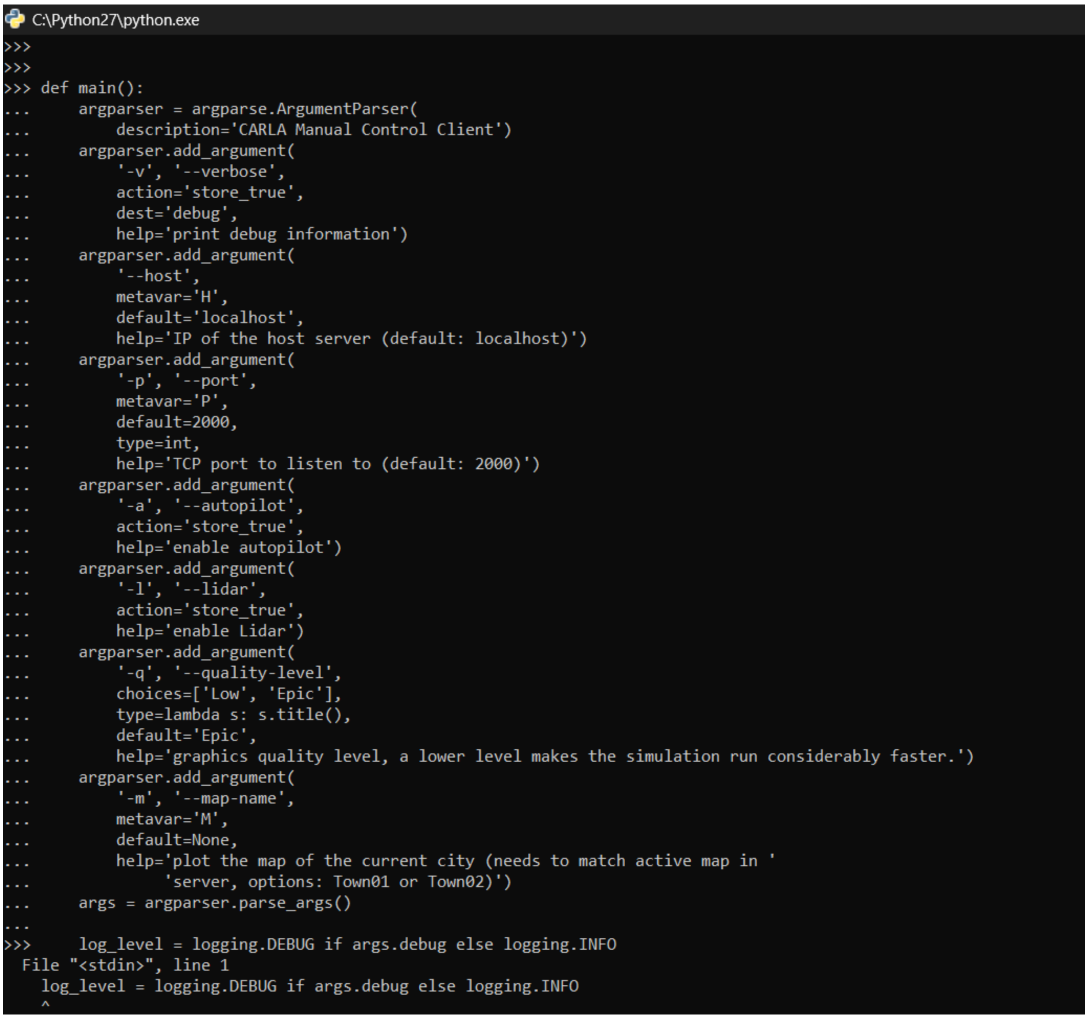


Figure 1. Flow-chart of the system

Figure 2 shows the Python coding for the system. The file path "C:\Python27\pythone.exe" raises two significant points of interest. Located in a Windows directory, it ostensibly points to a Python executable file. However, the folder "Python27" suggests the use of Python 2.7, an outdated version that no longer receives official support, posing potential security risks. Additionally, the filename "pythone.exe" deviates from the standard "python.exe," which could either be a typo or a custom naming convention. This anomaly warrants caution, urging users to verify the file's legitimacy and consider updating to a current, secure version of Python.

Figure 3 shows the schematic of collision particles or objects for derivation of conservation momentum. According to Newton's third law, when an object *A* produces a force on another object *B*, object *B* responds by exerting a force that is equal in size but directed in the opposite direction. Newton derived the law of conservation of momentum using this concept. This formula can be used in results and discussion to investigate the momentum after a collision. Consider two colliding particles *A* and *B* whose masses are m_1 and m_2, with initial and final velocities as u_1 and v_1 of *A* and u_2 and v_2 of *B*. The time of contact between two particles is given as time, t.



```
C:\Python27\python.exe
>>>
>>>
>>> def main():
...     argparser = argparse.ArgumentParser(
...         description='CARLA Manual Control Client')
...     argparser.add_argument(
...         '-v', '--verbose',
...         action='store_true',
...         dest='debug',
...         help='print debug information')
...     argparser.add_argument(
...         '--host',
...         metavar='H',
...         default='localhost',
...         help='IP of the host server (default: localhost)')
...     argparser.add_argument(
...         '-p', '--port',
...         metavar='P',
...         default=2000,
...         type=int,
...         help='TCP port to listen to (default: 2000)')
...     argparser.add_argument(
...         '-a', '--autopilot',
...         action='store_true',
...         help='enable autopilot')
...     argparser.add_argument(
...         '-l', '--lidar',
...         action='store_true',
...         help='enable Lidar')
...     argparser.add_argument(
...         '-q', '--quality-level',
...         choices=['Low', 'Epic'],
...         type=lambda s: s.title(),
...         default='Epic',
...         help='graphics quality level, a lower level makes the simulation run considerably faster.')
...     argparser.add_argument(
...         '-m', '--map-name',
...         metavar='M',
...         default=None,
...         help='plot the map of the current city (needs to match active map in '
...              'server, options: Town01 or Town02)')
...     args = argparser.parse_args()
...
>>>     log_level = logging.DEBUG if args.debug else logging.INFO
  File "<stdin>", line 1
    log_level = logging.DEBUG if args.debug else logging.INFO
    ^
```

Figure 2. Example of coding in Python programming for setting up the system

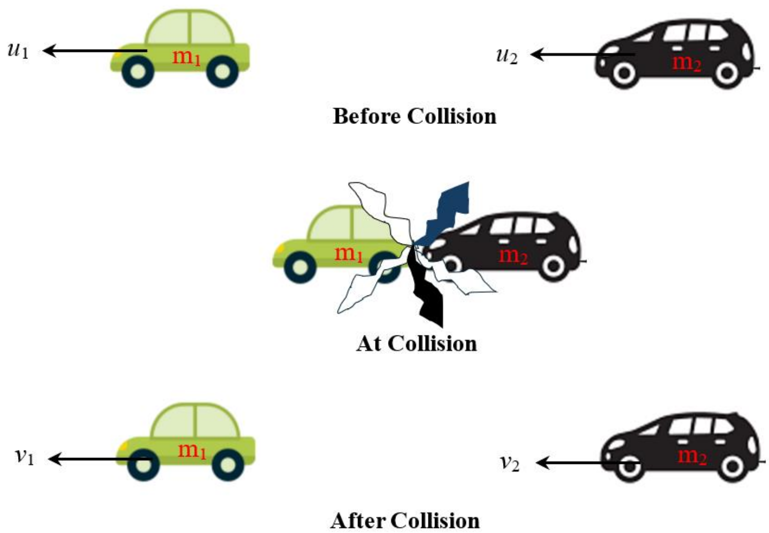


Figure 3. Schematic of collision particles for derivation of conservation momentum

Change in momentum of object of *A*,

$$\text{A} = m_1(v_1 - u_1) \tag{1}$$

Change in momentum of object of *B*,

$$\text{B} = m_2(v_2 - u_2) \tag{2}$$

From the third law of motion,

$$F_{BA} = -F_{AB} \tag{3}$$

where,

$$F_{BA} = m_2 \times a_2 = \frac{m_2(v_2 - u_2)}{t} \tag{4}$$

$$F_{AB} = m_1 \times a_1 = \frac{m_1(v_1 - u_1)}{t} \tag{5}$$

Thus,

$$\frac{m_2(v_2-u_2)}{t} = \frac{-m_1(v_1-u_1)}{t}$$

$$m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2 \tag{6}$$

The momentum during a collision can be calculated by the momentum Eq. (7) as below:

$$P = mv \tag{7}$$

where, P is the momentum (in kg.m/s), m = mass (in kg) and v = velocity (in m/s)

and the law of conservation of momentum,

$$m_1u_1 + m_2u_2 = m_3v_3 \tag{8}$$

3. RESULTS AND DISCUSSION

This section presents the results and analysis in the CARLA simulation software. The aim of this simulation is to investigate the vehicle speed, distance, pedestrian presence, and traffic pattern during driving in several road conditions. It consists of different road conditions like T-junctions, cross-junctions, and roundabouts. Python coding relates to CARLA simulation and is connected to test driving environments. The result is observed based on the vehicle's safe distance from incoming hazards, such as vehicles and pedestrians that are crossing the road, and will be warned by a red frame when the situation is unsafe. Besides that, the accuracy of object detection is one of the goals, such as making a red frame at an unsafe distance. The comparison of driving simulations in safe and unsafe modes will be included in this part. In addition, the simulation system will assist the drivers in acknowledging safe driving and reducing the possibility of accidents.

3.1 Vehicle Simulation on Different Conditions

This part shows the before and after application of the object detection system. The first simulation are conducted without the presence of a detection system. This is a basic CARLA simulation that comes with it when it’s installed in Figure 4. The driving conditions without warning will increase the accident probability since there are no safety features to assist the driver in recognising and detecting vehicles in front during junction situations. In real-world driving, there are many unexpected situations that may happen, such as when a front vehicle is accidently reversed or incoming cars from other junctions oversteer and enter the wrong lane.

Medenwald et al. [16] utilised the CARLA simulator not for observing coronary heart disease among adults with high cardiovascular risk factors but for a different purpose related to autonomous driving research. Three years later, Javier et al. [14] employed CARLA to delve into vehicle motion control and object detection, drawing insights from the Smart Elderly Car project. Their work provided detailed analysis and advancements in these areas. In 2022, Jang et al. [15] revolutionised object detection dataset generation, traditionally a costly task limited to large entities, through their CarFree system. The system integrated with the CARLA simulator and created accurate 2D bounding boxes for vehicles and pedestrians in driving images. Their findings show that CarFree efficiently produces realistic driving images with corresponding ground truths.

In this study, the object detection system has been applied, and it shows the situation analysis when object detection is in range. When the object is detected, it will display a red frame. The simulated vehicle (a black Ford Mustang) is at a safe distance from the front object when driving out of the red frame. Unsafe distance is when the simulated vehicle is in a red frame. Figure 4 below shows the comparison between safe and unsafe distances towards in-front vehicles. Safe distance means when a driver stops behind a vehicle, he or she can at least see the tyre of the front vehicle. Good distance may be advantageous to the driver when unexpected situations happen and to avoid accidents.

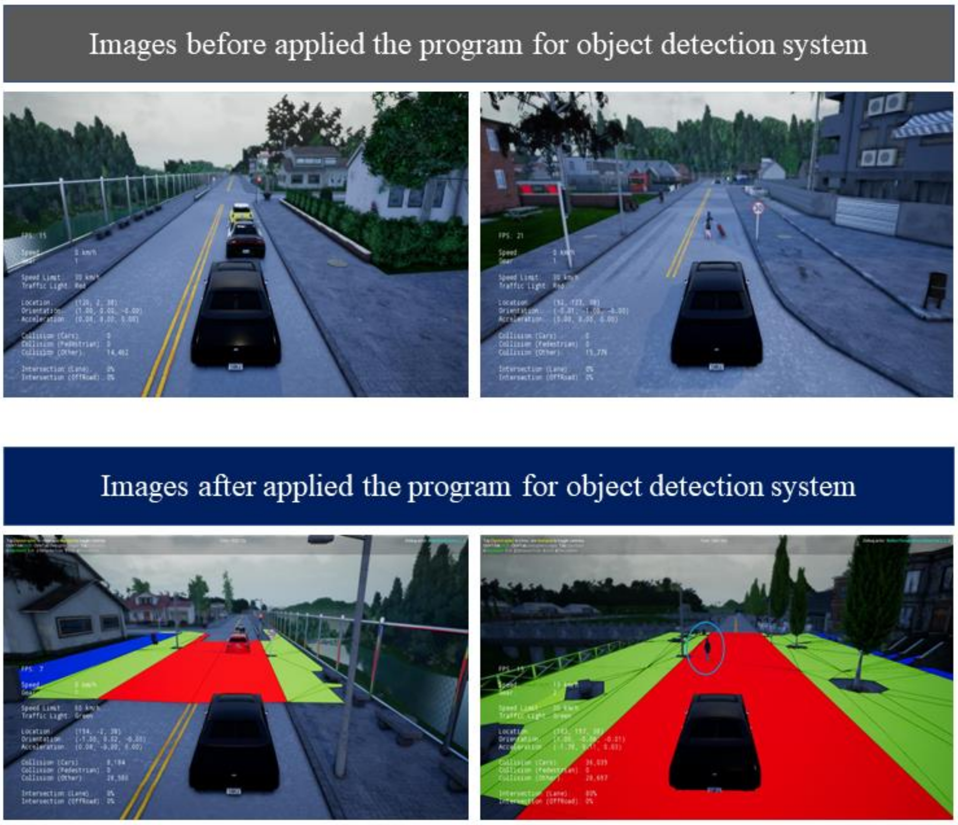
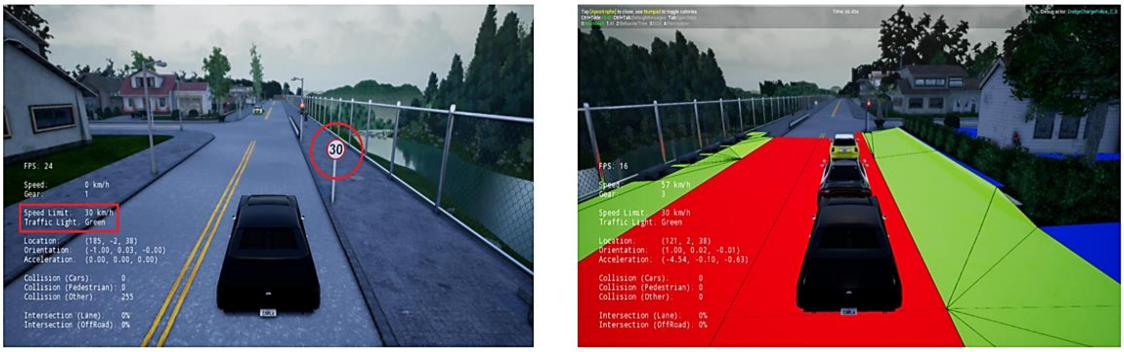


Figure 4. Before and after images of applying the program

3.2 Data Analysis of Simulation

With the simulation software, the traffic conditions can be simulated according to the laws or breaking the laws without worrying about dangerous situations for drivers or other road users compared to the real-life simulations in Figure 5. In the CARLA simulation, several traffic laws will be tested to ensure a safe driving experience, such as speed limits and distances between vehicles. At the T-junction in the CARLA simulation, the speed limit sign is 30 km/h. When simulated at high speed and not obeying the traffic laws, it is found that the vehicle is difficult to stop and enter the unsafe distance with the in-front vehicle, thus leading to an accident situation with a high momentum value.

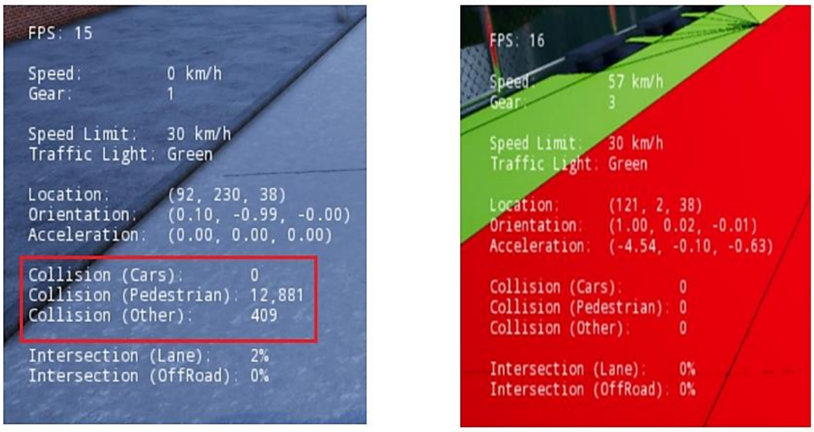


(a) Speed limit point

(b) High speed driving

Figure 5. Driving analysis for speed limit point and high speed in CARLA

Figure 6 shows the details of the vehicle's speed, gear, fps, and location for both speed limit points and high-speed driving conditions. Figure 6(a) shows the vehicle speed was 0 km/h and the speed limit was 30 km/h. The vehicle was stopped at that point. Figure 6(b) shows the vehicle in driving condition with a speed of 57 km/h, and it has crossed the speed limit. This data is crucial to analysing the driving conditions for calculating safe distances to avoid collisions.



(a) Speed limit point (b) High speed driving

Figure 6. Data taken at the speed limit point and high-speed driving

A collision with an in-front vehicle at a speed of 57 km/h instead of 30 km/h according to the speed limit led to 30,599 kg/m/s of momentum.

Figure 8. Schematic of collision simulation

In this simulation, the investigation focused on how speed affects collision momentum, with a focus on two iconic vehicles: the first-generation Ford Mustang and the Dodge Charger Police edition. These vehicles possess distinct masses, with the Mustang weighing 1109 kg and the Charger tipping the scales at 2014 kg. When vehicles adhered to a speed limit of 30 km/h, the collision aftermath resulted in a combined velocity of 30 km/h, generating a specific collision momentum. This scenario provides valuable insights into the consequences of obeying speed limits for road safety. Conversely, when the speed exceeded the limit, reaching 57 km/h, the outcome differed significantly. At this higher speed, the post-collision velocity of the wrecked cars was substantially greater, leading to a noticeably higher collision momentum. This comparison underscores the substantial impact of speed on collision dynamics and highlights the increased risks associated with speeding. These findings highlight the critical importance of adhering to speed limits not only for road safety but also for minimising the severity of accidents. By obeying speed regulations, safer roads and reduced collision-related risks can be achieved.

Table 1 presents the results of five simulations, detailing the speed in both kilometres per hour (km/h) and metres per second (m/s), as well as the corresponding momentum in kilogrammes per second (kg.m/s). From the data, it is evident that as the speed increases, the momentum also shows a significant rise. In the first simulation, a speed of 15 km/h (or 4.17 m/s) resulted in a momentum of 8038 kg/s. In the second simulation, a doubling of the speed to 30 km/h (8.33 m/s) led to only a modest increase in momentum to 9244.1 kg/s. However, as the simulations progressed, the rate of increase in momentum became more pronounced. For instance, in the fourth simulation, a speed of 79 km/h (21.94 m/s) produced a momentum of 43121 kg/s, which is nearly 2.5 times the momentum observed in the third simulation at a speed of 57 km/h (15.83 m/s). The trend continued in the fifth simulation, where a speed of 105 km/h (29.17 m/s) resulted in a momentum of 57403 kg/s.

Table 1. Tabulate result for vehicle-to-vehicle collision simulation

Simulation No.	Speed (km/h)	Speed (m/s)	Momentum (kg.m/s)
1	15	4.17	8038
2	30	8.33	9244.1
3	57	15.83	17551.3
4	79	21.94	43121
5	105	29.17	57403

Another simulation was also conducted with pedestrians. When the simulated vehicle was driving at a high speed (57 km/h) instead of obeying the speed limit (30 km/h), it led to late braking when the pedestrian was crossing the road and an accident occurred. When simulating by following the speed limit, the vehicle can stop at a safe distance, thus no accident occurred.

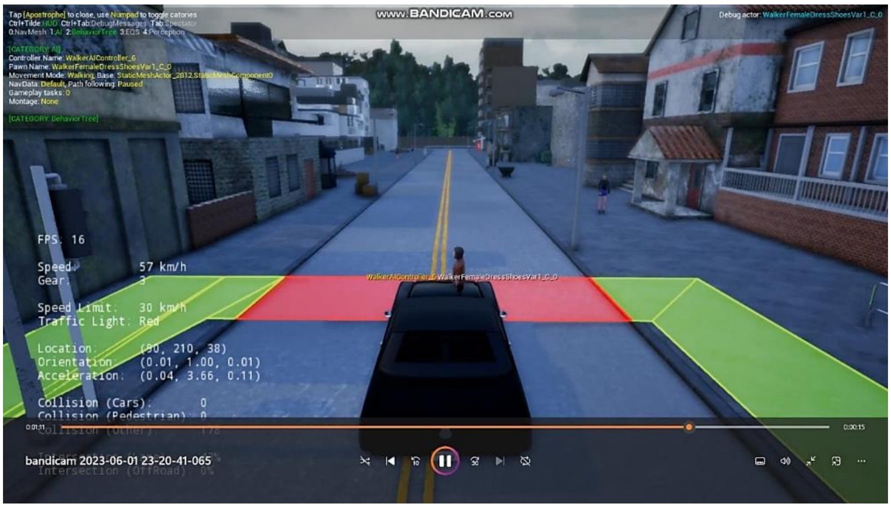


Figure 9. Collision with crossing pedestrian

In Figure 10, the outcomes of five distinct simulations are presented, offering a comprehensive analysis of vehicular speeds expressed in both kilometres per hour (km/h) and metres per second (m/s), along with their corresponding momentum measured in kilogrammes per second (kg/s). A clear trend is observed across these simulations: as the vehicle's speed escalates from the first simulation (24 km/h or 6.67 m/s) to the fifth simulation (90 km/h or 25 m/s), there is a concurrent and significant increase in momentum, rising from 7419 kg/s in the first instance to 18716 kg/s in the last.

This trend vividly illustrates the direct and proportional relationship between speed and momentum, as the data shows. Specifically, the progression from the first to the fifth simulation highlights how momentum, a fundamental physics concept that is the product of mass and velocity, responds to changes in speed. This correlation is critical in understanding the dynamics of vehicular motion, particularly in the context of autonomous driving, where accurate prediction of momentum based on speed is essential for ensuring safety and efficiency.

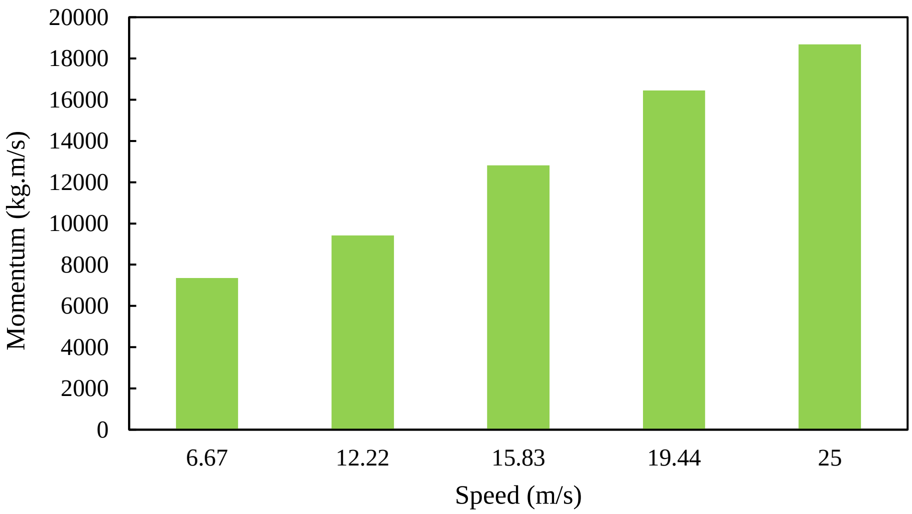


Figure 10. Momentum against speed result for vehicle with pedestrian collision simulation

The data suggests a non-linear relationship between speed and momentum. While the initial increase in speed led to a relatively smaller increment in momentum, the subsequent increases in speed resulted in more substantial growth in momentum. This observation may be attributed to various factors that could influence momentum, such as the mass of the object in motion or external forces acting on it. Further investigations and experiments might be necessary to delve deeper into the underlying causes of this observed trend.

4. CONCLUSIONS

In summary, the stated objectives have been effectively achieved, and all aspects of the project have been executed with accuracy. By employing the designated settings within the HUD of the CARLA simulator programme, it is possible to ascertain the velocity of the simulated vehicle precisely. In addition, the system successfully utilises Python programming and object detection techniques to accurately detect and emphasise the presence of other vehicles and pedestrians. In the context of scenarios occurring at T-junctions and straight roadways, vehicles and pedestrians that have been specifically identified are visually demarcated by a conspicuous red frame. In the course of this investigation, multiple elements that contribute to accidents have been uncovered, with a specific focus on simulated scenarios, particularly those occurring near T-junctions. Research has indicated that surpassing the designated speed limit is associated with an extended distance required to come to a complete stop, hence increasing the likelihood of crashes with preceding vehicles in situations where a sufficient stopping distance cannot be attained. Furthermore, within urban driving environments, a considerable number of pedestrians engage in road crossing activities, and a failure to prioritise attention towards driving can substantially increase the likelihood of accidents.

5. ACKNOWLEDGMENTS

The authors would like to thank the Ministry of Higher Education for providing financial support under Fundamental Research Grant Scheme (FRGS), No. FRGS/1/2021/TK02/UMP/02/1 (University reference RDU210153) and Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) for laboratory facilities as well as additional financial support under Internal Research grant RDU1903101.

6. REFERENCES

[1] N. Singh, G. Ji, “Computer vision assisted, real time blind spot detection based collision warning system for two wheelers,” in *5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2021, pp. 1179–1184.

[2] K. Gildea, D. Hall, C. Simms, “Configurations of underreported cyclist-motorised vehicle and single cyclist collisions: Analysis of a self-reported survey,” *Accident Analysis & Prevention*, vol. 159, p. 106264, 2021.

[3] H. Li, N. Bamminger, Z. F. Magosi, C. Feichtinger, Y. Zhao, T. Mihalj, et al., “The effect of rainfall and illumination on automotive sensors detection performance,” *Sustainability*, vol. 15, no. 9, p. 7260, 2023.

[4] F. Uysal, S. Sanka, “Mitigation of automotive radar interference,” in *IEEE Radar Conference (RadarConf2018)*, Oklahoma City, USA, 2018, pp. 405–410.

[5] M. Thibenda, D. M. P. Wedagama, D. Dissanayake, “Drivers’ attitudes to road safety in the south east asian cities of Jakarta and Hanoi: Socio-economic and demographic characterisation by multiple correspondence analysis,” *Safety Science*, vol. 155, p. 105869, 2022.

[6] S. Cafiso, G. Pappalardo, “Safety effectiveness and performance of lane support systems for driving assistance and automation – Experimental test and logistic regression for rare events,” *Accident Analysis & Prevention*, vol. 148, p. 105791, 2020.

[7] M. Schlögl, R. Stütz, G. Laaha, M. Melcher, “A comparison of statistical learning methods for deriving determining factors of accident occurrence from an imbalanced high resolution dataset,” *Accident Analysis & Prevention*, vol. 127, pp. 134–149, 2019.

[8] Z. F. Magosi, C. Wellershaus, V. R. Tihanyi, P. Luley, A. Eichberger, “Evaluation methodology for physical radar perception sensor models based on on-road measurements for the testing and validation of automated driving,” *Energies*, vol. 15, no. 7, p. 2545, 2022.

[9] S. Chen, Y. Chen, S. Zhang, N. Zheng, “A novel integrated simulation and testing platform for self-driving cars with hardware in the loop,” *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 425–436, 2019.

[10] T. H. B. Huy, H. Truong Dinh, D. Ngoc Vo, D. Kim, “Real-time energy scheduling for home energy management systems with an energy storage system and electric vehicle based on a supervised-learning-based strategy,” *Energy Conversion and Management*, vol. 292, p. 117340, 2023.

[11] S. Das, S. K. Mishra, “A machine learning approach for collision avoidance and path planning of mobile robot under dense and cluttered environments,” *Compututer and Electrical Eng*ineering, vol. 103, p. 108376, 2022.

[12] J. Zhou, B. Zhang, X. Yuan, C. Lian, L. Ji, Q. Zhang, et al., “YOLO-CIR: The network based on YOLO and ConvNeXt for infrared object detection,” *Infrared Physic & Technology*, vol. 131, p. 104703, 2023.

[13] Z. Liu, Y. He, C. Wang, R. Song, “Analysis of the influence of foggy weather environment on the detection effect of machine vision obstacles,” *Sensors*, vol. 20, no. 2, p. 349, 2020.

[14] J. del Egido Sierra, A. Díaz, L. M. Bergasa, R. Barea, M. E. López, “Autonomous Vehicle Control in CARLA Challenge,” *Transportation Research Procedia*, vol. 58, pp. 69–74, 2021.

[15] J. Jang, H. Lee, J. C. Kim, "Carfree: Hassle-free object detection dataset generation using carla autonomous driving simulator," *Applied Sci*ences, vol. 12, no. 1, p. 281, 2022.

[16] D. Medenwald, A. Kluttig, M. E. Lacruz, J. Schumann, "Serum dietary fatty acids and coronary heart disease risk – A nested case-control-study within the CARLA cohort," *Nutrition, Metabolism and Cardiovascular Diseases*, vol. 29, no. 2, pp. 152–158, 2019.

[17] Z. Wang, J. Zhan, Y. Li, Z. Zhong, Z. Cao, "A new scheme of vehicle detection for severe weather based on multi-sensor fusion," *Measurement*, vol. 191, p. 110737, 2022.

[18] W. Gao, J. Tang, T. Wang, "An object detection research method based on CARLA simulation," *Journal of Physics: Conference Series*, vol. 1948, p. 012163, 2021.

[19] L. G. Cuenca, E. Puertas, J. F. Andrés, N. Aliane, "Autonomous driving in roundabout maneuvers using reinforcement learning with q-learning," *Electronics*, vol. 8, no. 12, p. 1536, 2019.

[20] S. Malik, M. A. Khan, H. El-Sayed, "CARLA: Car learning to act - An inside out," *Procedia Computer Sci*ence, vol. 198, pp. 742–749, 2021.

[21] S. Stević, M. Krunić, M Marko Dragojević, N. Kaprocki, "Development of ADAS perception applications in ROS and 'Software-In-the-Loop' validation with CARLA simulator," *Telfor Journal*, vol. 12, no. 1, pp. 40–45, 2020.

[22] Ó. Pérez-Gil, R. Barea, E. López-Guillén, L. M. Bergasa, C. Gómez-Huélamo, R. Gutiérrez, et al., "Deep reinforcement learning based control for autonomous vehicles in CARLA," *Multimedia Tools and Applications*, vol. 81, pp. 3553–3576, 2022.