INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING
ISSN: 2229-8649 e-ISSN: 2180-1606
VOL. 20, ISSUE 3, 2023, 10649 – 10658
DOI: https://doi.org/10.15282/ijame.20.3.2023.08.0822

RESEARCH ARTICLE

Investigation of Object Detection and Identification at Different Lighting Conditions for Autonomous Vehicle Application

N. Abdul Razak*, N.A.A. Sabri, J. Johari, F. Ahmat Ruslan, M. Md. Kamal and M.A. Aziz

School of Electrical Engineering, Universiti Teknologi MARA, 40450 Selangor, Malaysia

ABSTRACT - Ensuring the safety of autonomous vehicles requires effective detection and tracking of surrounding objects. This paper proposes the design and development of a driverless transportation system module focused on identifying obstacles around vehicles. By integrating computer vision with deep learning, the system presents a reliable and cost-effective solution for autonomous driving. Utilizing Raspberry Pi 4B and a USB webcam, a compact hardware setup is created for seamless implementation in autonomous vehicles. The algorithm presented in this study enables the detection, classification, and tracking of both moving and stationary objects, including cars, buses, trucks, people, and motorcycles. TensorFlow Lite, a deep-learning network, is employed for efficient object detection and classification. Leveraging Python as the primary programming language, known for its high-level object-oriented features and integrated semantics, the algorithm is tailored for web and application development. Experimental results demonstrate the system's capability to concurrently detect and identify multiple local objects with an accuracy ranging from 50% to 80% in day and night conditions. These findings underscore the potential of deep learning in advancing autonomous vehicle technology.

ARTICLE HISTORY

Received	:	21st Jan 2022
Revised	:	23rd June 2023
Accepted	:	07th Aug 2023
Published	:	01st Oct 2023

KEYWORDS

Autonomous vehicle;
TensorFlow;
Object identification;
Deep learning

1.0 INTRODUCTION

In the past few years, there has been a major increase in research interest in the autonomous vehicle, which is an automobile platform capable of sensing and reacting to its immediate environment to navigate roadways without human interaction. Perception is the process of detecting and classifying objects in the environment [1]. It generally includes several subtasks such as object detection [2], object classification [3], three-dimensional (3D) positioning [4], and Simultaneous Localization and Mapping (SLAM) [5]. The collected data from these sensors may be gathered and integrated to help the automobile locate itself and track objects in its environment, allowing it to navigate from one location to another successfully. As a result, Autonomous Vehicles (AV) provide mobility to a greater spectrum of individuals, provide passengers with additional free time during their journey [6], reduce emissions of carbon dioxide (CO_2), ease traffic congestion, and, most significantly, result in better road safety. Many aspects of active safety systems are currently implemented in automobiles, such as lane-keeping assistant, blind spot detection, collision avoidance system, and adaptive cruise control that can form the base for an autonomous vehicles framework [2]. Object detection is one of the main perception issues that autonomous vehicles' Advanced Driver Assistance Systems (ADAS) encounter. These vehicles' multi-modal sensors give important data that can be utilized with popular deep learning-based object detectors. Self-driving vehicle perception systems, on the other hand, must be accurate and robust enough to operate safely in challenging circumstances such as heavy urban traffic, worst weather conditions, unmapped roads, or locations with intermittent connectivity [7].

However, many aspects of this technology must improve before it can entirely replace humans. This research aims to improve the accuracy of two-dimensional (2D) object detection in images captured by autonomous vehicle onboard cameras. The purpose of this detection task is to identify objects from different classes and return the spatial location of each instance via a bounding box [8]. The main tendency in current object identification literature is to develop increasingly complex architectures to outperform the general-purpose Common Objects in Context (COCO) benchmark [9-11]. It is suggested that several changes be made to the popular Faster Regions with Convolutional Neural Networks (Faster R-CNN) detection framework and adapted to the self-driving car situation. The original model's performance is improved by optimizing the anchor generation procedure and modifying the learning process to increase the accuracy across minority instances. The scale of objects in this context has a strong correlation with their position in the image due to the perspective projection of onboard cameras in vehicles. This means that producing small-scale anchors is inappropriate in areas where objects tend to be quite large and vice versa [12]. To solve the anchor mismatch problem, researchers propose separating the photos into numerous sections and optimizing each separately [13]. Huval et al. [14] discussed that deep learning was tested empirically on real-life driving data in real-time for two projects which are lane detection and vehicle detection. This research concludes that by operating at the frame rate required for a real-time system, Convolutional Neural Networks (CNN) can provide a suitable solution for the following objectives.

***CORRESPONDING AUTHOR | Noorfadzli Abdul Razak** | ✉ noorfadzli@uitm.edu.my

Aryal and Baine [15] propose an approach that uses a state-of-the-art deep-learning network called You Only Look Once (YOLO) paired with data from a laser scanner to detect and identify objects and estimate their positions around the car. The Oriented FAST and Rotated BRIEF (ORB) feature descriptor is used to match the same object from one image frame to the next. Using an Extended Kalman Filter (EKF), this data was combined with measurements from a paired Inertial Navigation System (INS) and Global Positioning System (GPS). The resultant solution aids in the localization of both the car and the objects in its surrounding, allowing it to travel the highways on its own safely. The YOLO neural network was chosen due to its speed and resource consumption [16-17]. As training inputs, a data set of images of objects was generated and placed into the YOLO network. Singha and Bhowmik et al. [18] look into the possibility of using simple techniques to build an object detector that excels in the metrics provided by our autonomous driving environment, where we want to train a detector that can fit in limited memory, predict in real-time, and has acceptable accuracy for use in a self-driving platform. SimpleNet is the name of the resultant detector.

In another research, Sarda et al. [19] utilized the state-of-the-art algorithm YOLO to detect and classify objects on the road with the help of bounding boxes. A customized YOLO model was developed and trained using open images dataset. The outcomes show the model capable of detecting objects on the road and captioning each of them to its respective type. However, correct training via more images is needed to minimize errors that occur. Naghavi and Pourreza [20] propose a real-time detection and classification of on-road objects via a single deep convolutional neural network—the KITTI Road dataset to train the model to recognize various on-road objects. The result from the experiment did prove the model perform better than YOLO and SSD300, which was justified via Mean Average Precision (mAP). Ciberlin et al. [21] presented an approach for object detection and tracking using a front-view camera that utilized the Viola-Jones algorithm and YOLOv3. In this research, the Viola-Jones algorithm was used for creating object detectors to detect vehicles, pedestrians, traffic lights, and traffic signs in a video sequence. Then the object detectors were compared with a YOLOv3 object detection model. A comparison of the performance was performed in terms of accuracy and processing speed. The result shows YOLOv3 model performs better than Viola-Jones-based detectors. Gluhaković et al. [22] developed a method for vehicle detection and added with alerting system for a potential collision. A Robot Operating System (ROS) was employed, and the method comprises two parts. The first part is a model developed based on the YOLO v2 algorithm to perform object detection, which were cars, vans, trucks, and buses. It was trained via images gained generic web database and traffic video. The second part is for distance assessment which contains two nodes applied in the Carla simulator and real-world distance assessment. The outcomes demonstrate the method works in real time. However, improvement is needed so that the method can perform well for high distance assessment and in conditions when it is night and rain. Studies conducted by [23-25] have also emphasized the significance of considering the visibility of object detection in relation to weather conditions and the environment.

With this recent development and established techniques for AV application, it becomes our motivation to develop a prototype AV based on a 12-seater buggy that can be used in a controlled environment place. In this research, the controlled environment is interpreted as a geofenced area where the road lane is prescribed for the vehicle to move at low speed and low traffic. This prototype development intends to be used as a shuttle service so that it becomes a solution for places with heavy traffic and limited parking spaces. Sensors such as Light Detection and Ranging (LIDAR), GPS, Laser Range Finder, (LRF), ultrasonic, and camera are included on the vehicle. At present, the vehicle can operate safely; the speed is limited to 25 km/h. At this speed, it features sensors that can work effectively to steer and prevent collisions between other objects or vehicles. In autonomous driving, accuracy in identifying a moving object is crucial. The model of the object or environment can be identified more accurately by integrating object classification from various sensor detections [26]. The size and speed of the moving object are two critical parameters in maximizing accuracy. Hence, this research discusses the accuracy of object identification in AVs such as cars, motorcycles, pedestrians, and others.

2.0 METHODOLOGY

2.1 Hardware System Setup

The main intention is to allow the system to identify and classify the obstacles for an autonomous vehicle. Hence, to enable the system to do this, Figure 1 depicts a hardware block diagram that was set up for the system to execute the function. From the figure, it can be identified that the system comprises five main components, namely a camera, Raspberry Pi controller board, data storage card, liquid crystal display (LCD) screen, and direct current (DC) power supply. In this system, the camera used is a 12-Mega-pixel type that is utilized to capture real-time images and video of the target area. It is connected to the controller board via a Universal Serial Bus (USB) 3.0 port.

Meanwhile, the controller board used for the system is Raspberry Pi 4B, which is equipped with an 8GB LPDDR4-3200 Synchronous Dynamic Random-Access Memory (SDRAM). This board operates using Broadcom BCM2711 Security Operation Center (SOC) as the main processor, and it consists of a Quad-core ARM Cortex-A72 Central Processing Unit (CPU) with ARM v8 operating at 1.5 GHz. It becomes a target controller in this research because it is compact, affordable, consumes less energy, and has a powerful embedded processor that is suitable for the project application. As for the board to execute the Operating System (OS) along with the data storage function, a 32GB micro–Secure Digital (micro-SD) card is employed. The card was installed on the micro-SD port that is accessible on the board. An LCD screen is utilized as Human Machine Interface (HMI) to observe the OS operation of the board and data in the storage card. This screen was connected to the board via a High-Definition Multimedia Interface (HDMI) port. Lastly, a

DC power supply from a power bank that produces 5V with a current of 2.5A was used to power up all the components, as stated above. This power supply is used because it is sufficient to run the experiment for a short period. However, for later autonomous vehicle applications, the power supply from the car's battery was used to support long-term operation. The power supply was connected to the controller board via a USB cable Type C, and later this power was distributed to the other components to activate them. A clear view of how the interfacing of all the components above was carried out is demonstrated in Figure 2.

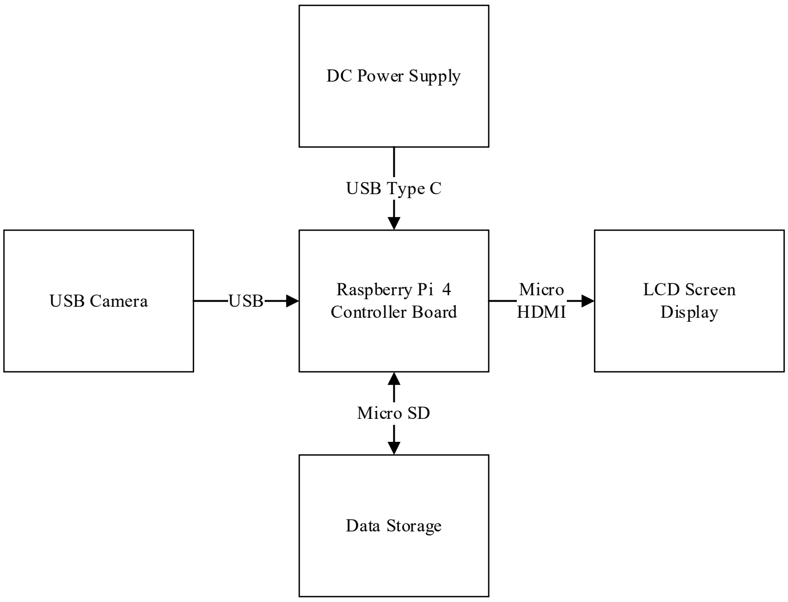


Figure 1. Block diagram of the hardware system

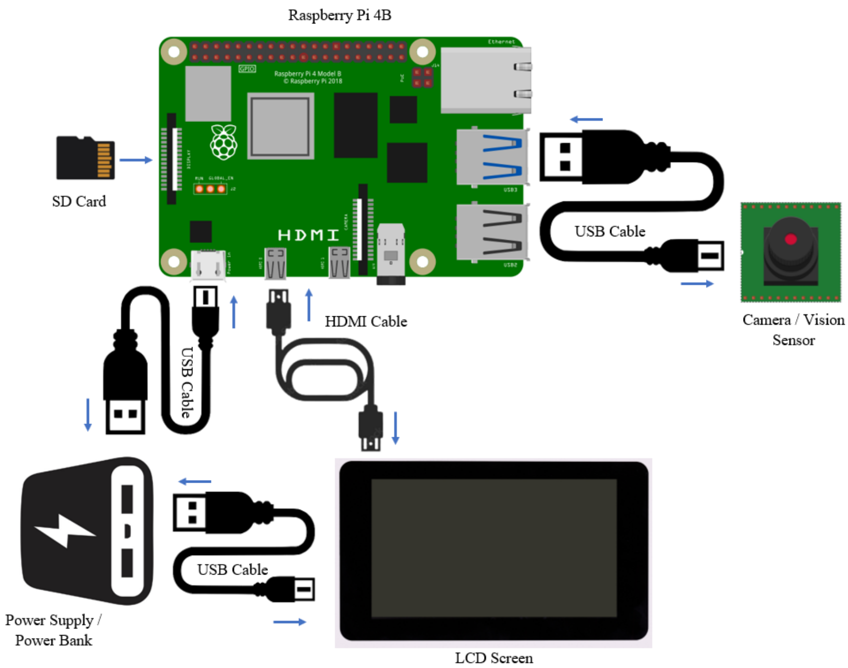


Figure 2. Interfacing diagram of the hardware system

2.2 Software Design Setup

To enable the hardware especially the Raspberry Pi controller (to execute the object detection and identification function), several software such as Raspbian Operating System (OS) and Python Integrated Development Environment (IDE) were installed in it. The Raspbian OS was used as the operation manager that enable the Raspberry Pi to run application programs and control its peripherals [27]. As for Python IDE, it was utilized as a programming platform to develop a customize Python-based program to execute the designed function. Additionally, in the developed program, Open CV and TensorFlow Lite libraries were also integrated. Both libraries are needed to execute image processing and machine learning, which is necessary for this application. To give an idea of how this software along with the libraries operate for specific functions, a block diagram in Figure 3 is presented.

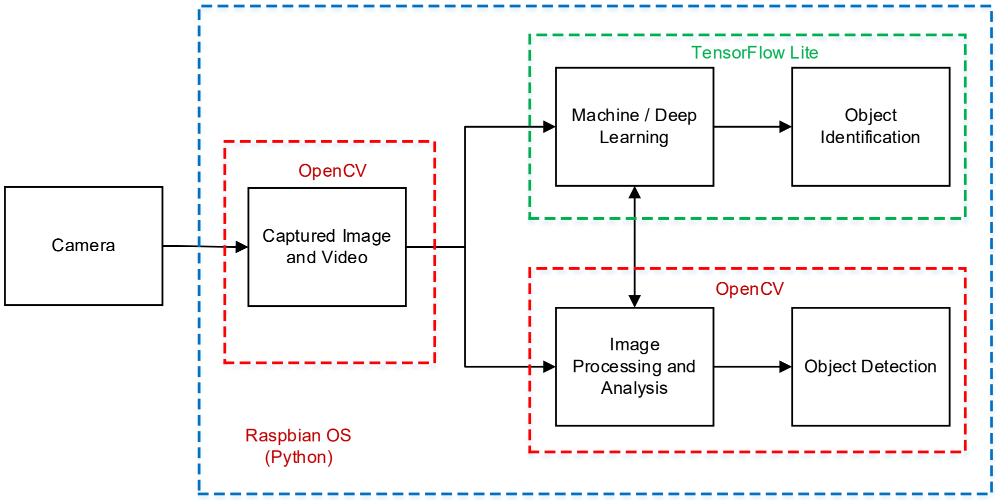


Figure 3. Block diagram of the object identifications system

According to Figure 3, once the board is activated, the Raspbian OS runs the customized Python program, which includes the OpenCV library to capture and collect raw images and video of the object on the road. Next, the images and video data were processed and analyzed to remove the foreground for object detection. These data were segmented, and then all the segmented data were structured in the form of a tensor. The composite tensor was then further processed to optimize the bounding box and classification of unlabelled data. Subsequently, Tensorflow Lite was used by deep learning to do the data learning for a person, motorcycle, car, bus, and truck. Then, the output of the moving object detection will be displayed [28]. Meanwhile, on how to enable the designed program to have the capability to perform the detection and identification task, the flow chart in Figure 4 details the process.

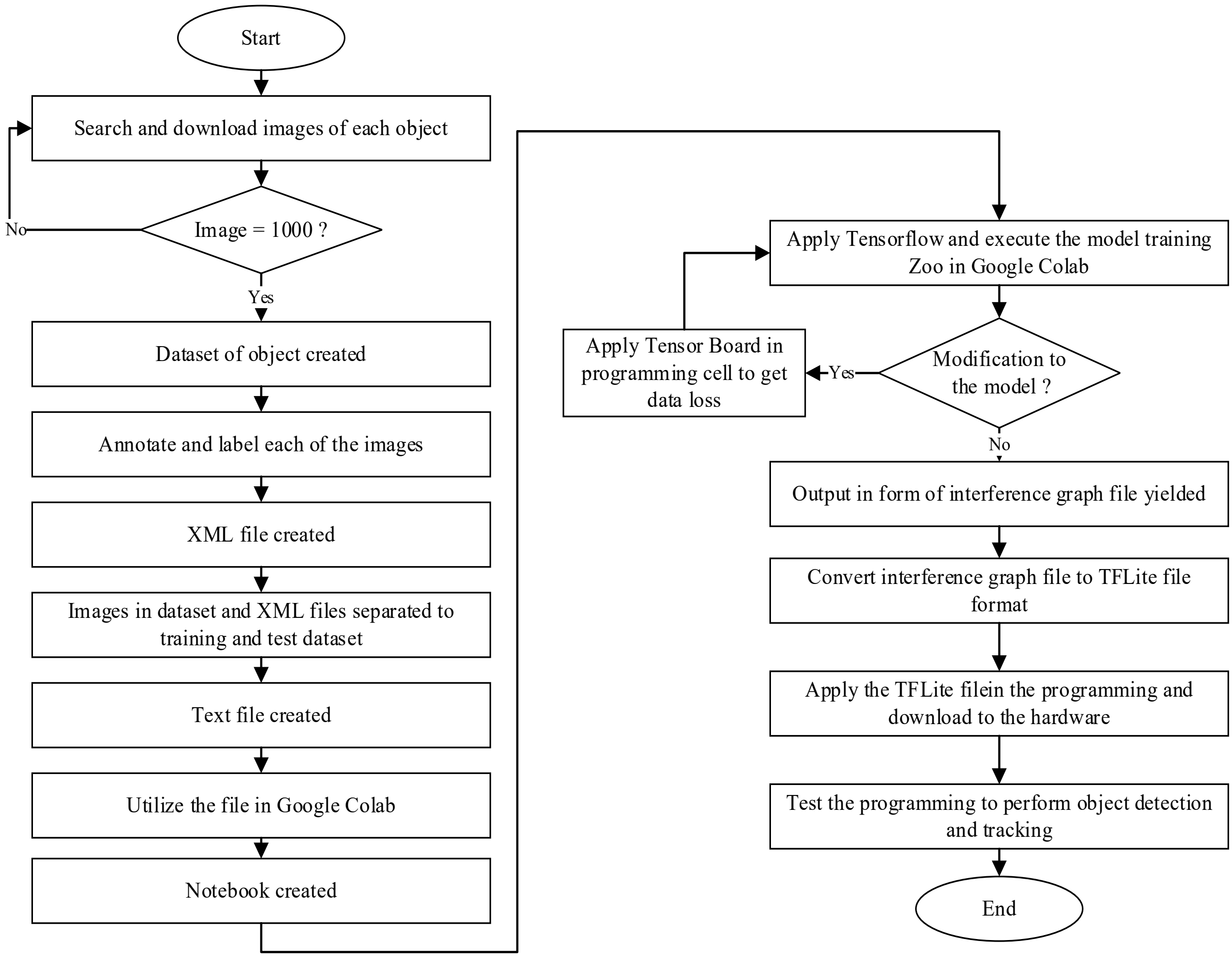
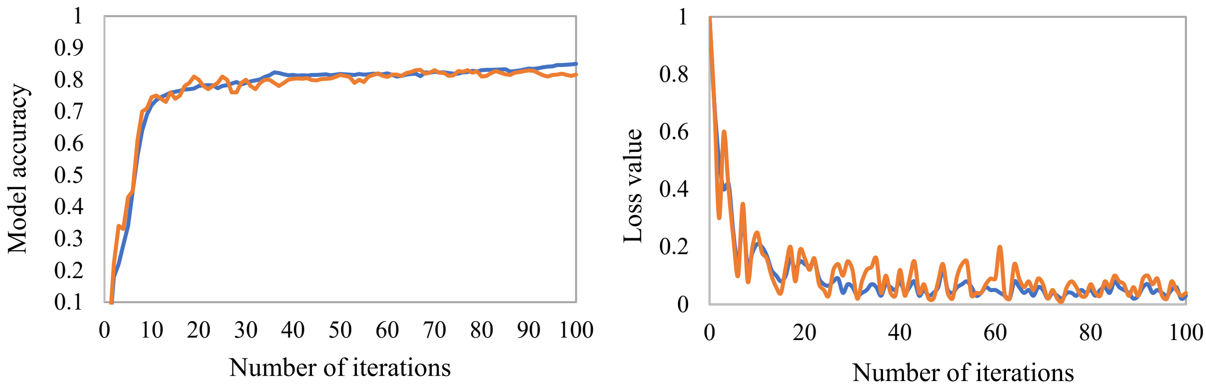


Figure 4. Process to enable the algorithm to have the capability to perform object detection and tracking

At initial, the images for each object which included cars, trucks, motorcycles, buses, and people, were gathered to perform this experiment. Technically, a higher number of images used to create the data set will result in the trained model having an optimal performance for object detection and identification. However, the selection of the image numbers must be carefully determined. This is not to burden the controller with high computational efforts because it becomes slow and time-consuming. Therefore, for this experiment, it was considered sufficient to utilize 1000 images for each object to create one data set. These object images at day and night were gathered from the internet source and also added with images captured manually via a digital camera. After that, the images were annotated, where for this purpose LabelImg software was employed. Next, all of the objects must be labelled one by one to create an eXtensible Markup Language (XML) file. Then, all of the images in the dataset and the XML file will be separated into two files, namely a train image dataset and a test image dataset. The train image dataset comprises 80% of the images and the balance, which is 20%, was meant for the test image dataset. The images were trained using the training file, while the image was tested using the test file. Subsequently, a text file, namely labelmap.txt file was then generated, which included a list of the objects that were utilized to be detected by this system. Afterwards, the Google Colab was utilized to create a notebook environment and Model Zoo was used to train the dataset of annotated images. Figure 5(a) demonstrates the training accuracy, signified by a blue line and the testing accuracy, which is represented by the red line. Meanwhile, Figure 5(b) shows the loss for training and testing data set for the model to be established.



(a) (b)

Figure 5. (a) Model training and testing accuracy, and (b) model training and testing loss

Overall, the accuracy attained by the model was 82.2% and at that moment, the research did accept the model's performance. Furthermore, the comparable development between training and validation loss indicates a non-overfitted training procedure. Next, the output file from the model which is in a TensorFlow format needs to be converted into TFLite format. This format is required since the TensorFlow Lite model was used in the Raspberry Pi 4B. Other than this format will result in the model cannot work in the controller since heavy computing is not suitable for edge computers like Raspberry Pi that only have limited computing resources [29]. The model Zoo used in TensorFlow Lite was SSD. In order to perform object detection, Python programming was used for video processing detection. All the output files from TensorFlow were executed in this program. The model functionality was then initially tested via an offline method before being applied for real-time application. In the method, a sample of recorded video attained from an internet source was used. Then, it was programmed to run in the controller and simultaneously, the model was set to execute the designed task. The outcomes of the testing are shown in Figure 6. It demonstrates the model's functionality to perform the task where the detected and identified object including the accuracy were displayed.

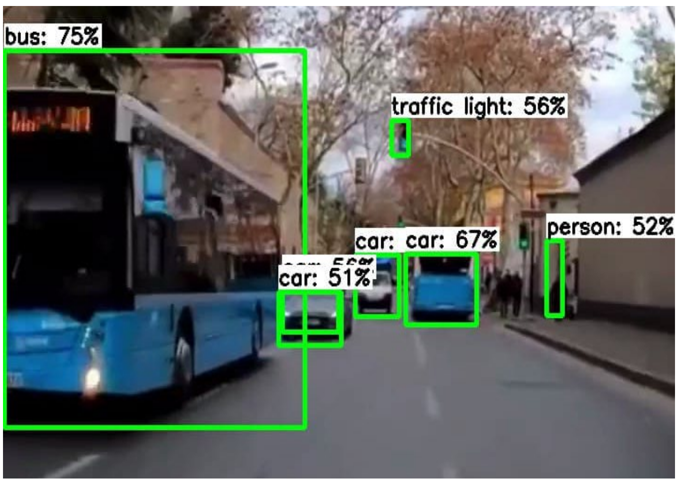
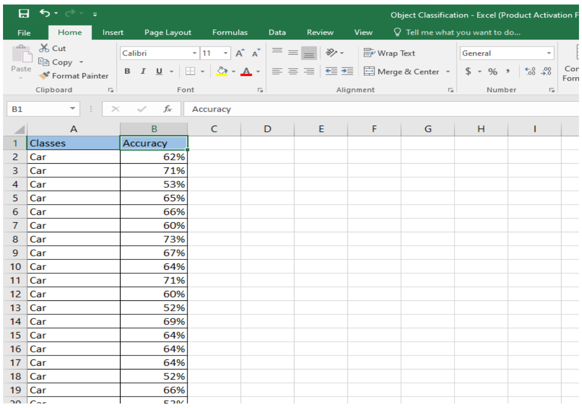


Figure 6. Accuracy of object detection

The objects were detected in real-time, and simultaneously the identification process was executed to distinguish whether the object is a person, motorcycle, car, bus, or truck. Moreover, this program was programmed to display the accuracy of the function to detect and identify the object. Correspondingly, the accuracy data was recorded in Microsoft Excel format as shown in Figure 7. In this figure, the designed system successfully recorded the data when performing the detection and identification. The recorded data represents the respective accuracy for different types of classes such as car, bus, truck, motorcycle, and person. Additionally, this function did work either during the day or night. However, during the experiment, there are lots of identification accuracy data that will be captured for the respective type of objects. Hence, a simplified approach must be used to the data so that the performance of the system can be justified. For this research, it was set that the mean of the accuracy by each of the objects identified by the system will be employed. Equation 1 was utilized to obtain the result and correspondingly, it is used to justify the system performance.

$$\bar{X} = \sum \frac{X_n}{n} \tag{1}$$

where $\bar{X}$ is mean, $\sum X$ is the sum of accuracy by classes, and n is the number of the object detected by classes.



Classes	Accuracy
Car	62%
Car	71%
Car	53%
Car	65%
Car	66%
Car	60%
Car	73%
Car	67%
Car	64%
Car	71%
Car	60%
Car	52%
Car	69%
Car	64%
Car	64%
Car	64%
Car	52%
Car	66%

Figure 7. Sample of recorded object identifications in Excel format

2.3 Experiment Setup

To test the capability of the system to perform object detection and identification in real-time applications, the hardware shown in Figure 2 was installed in a car. Figure 8 depicts the hardware setup done to a car in an area located at our house nearby to the experiment place. A camera was installed to a suction cup mount and then fitted to the car windshield. The camera was linked via a USB port to the Raspberry Pi controller that is placed on the car dashboard. At the dashboard seat, there was also present of a 7-inches touchscreen that connected to the controller via HDMI cable. The screen is used to display the process of detection performed by the system. A wireless keyboard and mouse were also there and links to the controller via another USB port. They were used to control the program operation and make the necessary adjustment to it when necessary. Meanwhile, to power up the devices as mentioned above, a 5V DC power bank with 20000 mAh capacity was used.

Figure 8. The hardware setup in a car

3.0 EXPERIMENTAL RESULTS

This section presents the results obtained using this designed system. All five types of objects were detected with accuracy between 50% to 80% during the day and night. There were 600 data recorded via the system to make this analysis which is 300 during the day and another 300 during the night. Indeed, as mentioned above, our target for this research is for the model can be applied in a controlled environment where the place has low traffic and the vehicles move at low speeds. However, to test the performance of the model, it was decided to utilize a place with normal traffic where the presence of the targeted object to be detected and identified on the road can be acquired. By doing this, realistic numbers of the objects detected and identified can be attained to analyze the performance of the model. Indeed, the experiment can be carried out in any place. However, to facilitate the testing process and easy access from the setup area, the experiment was conducted in Jalan Persiaran Pandan 1, Pandan Jaya, Kuala Lumpur, Malaysia. Figure 9 depicts the result of the experiment for object detection conducted during the day at 11 am for 1 hour.

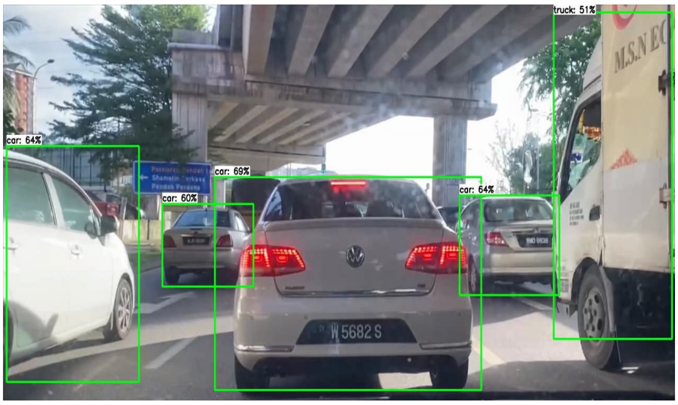


Figure 9. Object detection during daytime

Meanwhile, in Figure 10, the number of each type of object detected by the system during the experiment, including their respective mean of accuracy, is shown. According to the figure, the maximum mean accuracy between each type of object was the car, with 70.1%. This is because the number of cars detected during the day was higher than the other objects. There will be multiple cars in parallel at the same time, which will affect the track counting accuracy. The mean accuracies of motorcycle and person were 61.1% and 65.3%, respectively. In the recorded video, the motorcycle and person were small objects and were easily blocked by other large vehicles such as buses, cars, and trucks. Bus and truck were low inaccuracy because the vehicles could not fully fit in the bounding boxes resulting in low accuracy. Actually, the boxes were set fixed to a certain size that is acceptable for the model to detect large-size vehicles such bus and trucks. If this setup is not done, then the model will detect buildings, houses, bridge columns or beams along with other large objects and then identifies them as a bus or truck. Consequently, this condition will cause the accuracy becomes worse. As for object detection during the night, the result of the experiment when the system performs object detection is shown in Figure 11. The experiment was carried out at 8.30 pm for one hour on the same road as tested during the day experiment. In the meantime, the number of each type of object detected by the program during the experiment is shown in Figure 12. The figure also displays their respective mean of accuracy. Referring to the figure, the maximum mean of accuracy between each type of object was the cars, buses, and trucks, with 61.0%, 59.9%, and 62.2% mean, respectively. Due to the over brightness of the objects, it was difficult to detect the object at night because of the light reflections from vehicles. The mean accuracy of motorcycles and people was low at 56.4% and 54.5% due to low appearance, and it was hard to detect a smaller moving object at night.

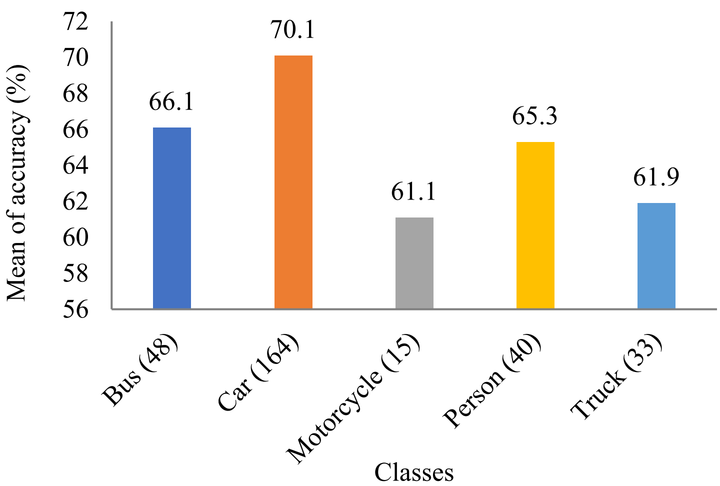


Figure 10. Accuracy performance according to the type of object when the experiment was performed during daytime

Figure 11. Object detection at night

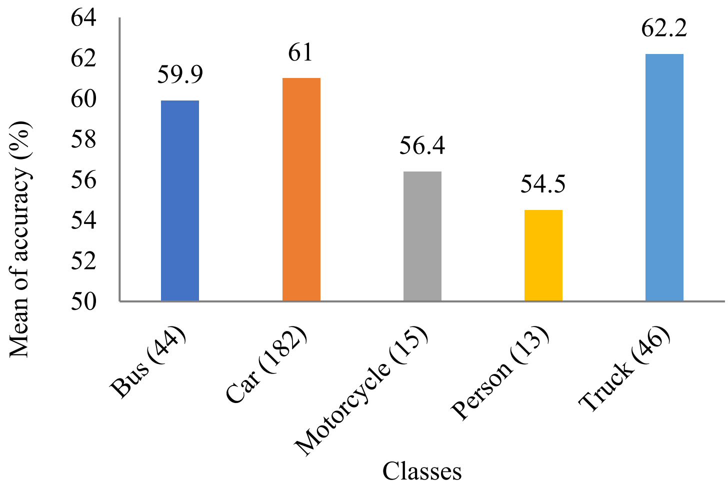


Figure 12. Accuracy performance according to the type of object when the experiment was performed during the night

From the results discussed above, the mean accuracy during the day was slightly greater than the accuracy at night. This is due to the low resolution, low brightness, and low appearance of objects during the night. This is known from Figure 12 where the mean accuracy for all classes at night was at low accuracy between 54.5% to 62.2%. Meanwhile, because the videos of objects detected were clear during the daytime, the mean accuracy for all classes was significantly higher, as indicated in Figure 10. Besides, during the experiment performed, it was identified the distance between the object and the camera becomes one of the factors that must be considered to achieve high accuracy in this system. This is proven when the distance between the objects and the camera is 1 meter to 10 meters; it results in the accuracy of detection and identification being around 60% and above. However, as the distance increases, the accuracy begins to be less and continues to decrease in proportion to the increase in distance.

4.0 DISCUSSIONS

The objective of this study was to propose an approach for object identification and classification on roadways, specifically tailored for autonomous driving, by employing TensorFlow Lite, a deep learning network. The focus was on designing and developing a crucial component of a driverless transportation system that enables the detection and identification of objects both during daytime and nighttime, utilizing video footage from a camera. A comprehensive experimental study using TensorFlow Lite was conducted to assess the performance of the proposed approach. The results revealed some interesting findings, particularly regarding the varying accuracy of object detection between different types of road users, such as vehicles and pedestrians, during daytime and nighttime conditions. Notably, the accuracy was generally higher during the day compared to the night. This discrepancy can be attributed to poor visibility at night, especially in pitch-dark environments, where recorded images of objects were not as clear as those captured during the daytime.

To overcome challenges posed by reduced visibility in such conditions, the study suggests the implementation of Far-Infrared (FIR) cameras. FIR cameras are known to offer robust object detection regardless of weather conditions, as the longer spectrum wavelength reduces the impact of adverse environments [30]. However, the higher cost associated with FIR cameras has limited their usage in moving object detection studies during the night, particularly under different atmospheric conditions using thermal images. Additionally, employing cameras with Autofocus (AF) mode could enhance object tracking by compensating for reduced accuracy caused by varying distances between the camera and the objects of interest. One of the key observations made during the research was the relatively low speed of 0.8 Frames per Second (FPS) achieved in the Raspberry Pi 4B when utilizing TensorFlow Lite. To address this limitation, the study proposes utilizing the Coral USB accelerator, which could significantly increase the FPS and improve the overall efficiency of the application.

In the quest for further improvements, the research recommends exploring alternative deep learning network approaches, such as YOLO, and expanding the dataset with additional training images to achieve even higher accuracy, potentially reaching 100%. Moreover, incorporating and experimenting with various loss functions during dataset training could lead to more precise and accurate findings. The proposed approach using TensorFlow Lite showcases the promising potential for object detection and classification in autonomous driving scenarios. To enhance the system's performance, the utilization of specialized cameras, like Far-Infrared (FIR) cameras, and employing advanced hardware accelerators, such as the Coral USB accelerator, are viable solutions. The continuous exploration of different deep learning approaches and the expansion of training datasets will contribute to the further refinement of the system's accuracy and efficiency in real-world applications.

5.0 CONCLUSIONS

In conclusion, this paper presented a comprehensive approach for identifying and classifying objects on roadways, specifically tailored for autonomous driving, utilizing the power of TensorFlow Lite, a deep learning network. The developed system successfully demonstrated its capability to detect and identify objects during both daytime and nighttime conditions, providing valuable insights into the performance of object detection in varying visibility scenarios. The experimental study using TensorFlow Lite revealed interesting findings regarding the accuracy of object detection among different road users under different lighting conditions. It was observed that accuracy was generally higher during the day, while challenges arose during nighttime due to reduced visibility in pitch-dark environments. The integration of FIR cameras emerged as a viable solution to enhance object detection in adverse weather conditions, as the longer spectrum wavelength reduced the impact of a challenging environment. Furthermore, the implementation of cameras with AF mode showcased potential in mitigating reduced accuracy caused by varying distances between the camera and the objects of interest. The study also identified a speed limitation of 0.8 FPS on the Raspberry Pi 4B while employing TensorFlow Lite. To address this, the introduction of the Coral USB accelerator was proposed to increase the FPS and optimize the system's efficiency significantly. Looking ahead, the paper highlighted areas for further improvement, including exploring alternative deep learning network approaches like YOLO and expanding the training dataset to achieve higher accuracy, potentially reaching 100%. Additionally, experimenting with various loss functions during dataset training holds promise for generating more precise and accurate outcomes.

6.0 ACKNOWLEDGEMENT

This research was supported by a fund from Universiti Teknologi MARA (UiTM) Selangor Branch, 600-UiTMSEL (PI. 5/4)(001/2021). The authors would like to thank Innovative Electromobility (ITEM) Research Lab, College of Engineering, UiTM Shah Alam, Selangor, UiTM Selangor Branch and ACEV Systems Sdn. Bhd. for providing the financial support, equipment, and place to conduct this research.

7.0 REFERENCES

[1] M. Y. Ismail, M.S. Beg, M.F. Jamlos, W.H. Azmi, N.H. Badrulhisam, and Omar I. Awad, "Potential and limitation of Internet of Things (IOT) application in the automotive industry: An overview," *International Journal of Automotive and Mechanical Engineering,* vol. 19, no. 3, pp. 9939–9949, Oct. 2022.

[2] B. Xiao, J. Guo and Z. He, "Real-time object detection algorithm of autonomous vehicles based on improved YOLOv5s," In 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), 2021, pp. 1-6, 2021.

[3] A. Juyal, S. Sharma and P. Matta, "Object classification using artificial intelligence techniques in autonomous vehicles," In 2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP), 2023, pp. 1-6.

[4] Z. Xiang and U. Ozguner, "A 3D positioning system for off-road autonomous vehicles," IEEE Proceedings. Intelligent Vehicles Symposium 2005, pp. 130-135.

[5] K. Yilmaz, B. Suslu, S. Roychowdhury and L. S. Muppirisetty, "AV-SLAM: Autonomous vehicle SLAM with gravity direction initialization," In 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 8093-8100, 2021.

[6] E. Wigley and G. Rose, "Who's behind the wheel? Visioning the future users and urban contexts of connected and autonomous vehicle technologies," *Geografiska Annaler, Series B: Human Geography*, vol. 102, no. 2, pp. 155–171, 2020.

[7] M. Carranza-García, P. Lara-Benítez, J. García-Gutiérrez, and J. C. Riquelme, "Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance," *Neurocomputing*, vol. 449, pp. 229–244, 2021.

[8] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen *et al.*, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[9] T. Y. Lin, M.Maire, S. Belongie, J. Hays, P. Perona, *et al.*, “Microsoft COCO: Common objects in context,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. 5, pp. 740–755, 2014.
[10] A. Z. Syed Sahil, M.S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A survey of modern deep learning-based object detection models,” *Digital Signal Processing,* vol. 126, p. 103514, 2022.
[11] M. Carranza-Garcia, P. Lara-Benítez, J. García-Gutiérrez, J.C. Riquelme, “Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance,” *Neurocomputing,* vol.449, pp. 229–244, 2021.
[12] M. R. S. Mohd, S.H. Herman, and Z. Sharif, “Enhancing image segmentation in thermal infrared image processing for faulty detection on broadcasting equipment,” *Journal of Electrical and Electronic Systems Research*, vol. 13, pp. 56–63, 2018.
[13] N. Jain and Sangeeta Mittal, “Review of computational techniques for modelling eco-safe driving behavior,” *International Journal of Automotive and Mechanical Engineering*, vol. 20, no. 2, pp. 10422–10440, Jul. 2023.
[14] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, *et al.*, “An empirical evaluation of deep learning on highway driving,” *ArXiv*, vol. abs/1504.01716, pp. 1–7, 2015,
[15] M. Aryal and N. Baine, “Detection, classification, and tracking of objects for autonomous vehicles,” In ION 2019 International Technical Meeting Proceedings, 2019, pp. 870–883.
[16] H. Huang, Z. Liu, T. Chen, X. Hu, Q. Zhang, and X. Xiong, “Design space exploration for YOLO neural network accelerator,” *Electronics*, vol. 9, no. 11, p. 1921, Nov. 2020.
[17] J. Jiang, X. Fu, R. Qin, X. Wang, and Z. Ma, “High-speed lightweight ship detection algorithm based on YOLO-V4 for three-channels RGB SAR image,” *Remote Sensing*, vol. 13, no. 10, p. 1909, May 2021
[18] A. Singha and M. K. Bhowmik, “Moving object detection in night time: A survey,” In Proceedings of 2nd International Conference on Innovations in Electronics, Signal Processing and Communication, 2019, pp. 44–49.
[19] A. Sarda, S. Dixit and A. Bhan, “Object detection for autonomous driving using YOLO [You Only Look Once] algorithm,” In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 1370-1374.
[20] S. H. Naghavi and H. Pourreza, “Real-time object detection and classification for autonomous driving,” In 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE), 2018, pp. 274-279.
[21] J. Ciberlin, R. Grbic, N. Teslić and M. Pilipović, “Object detection and object tracking in front of the vehicle using front view camera,” In 2019 Zooming Innovation in Consumer Technologies Conference (ZINC), 2019, pp. 27-32.
[22] M. Gluhaković, M. Herceg, M. Popovic and J. Kovačević, “Vehicle detection in the autonomous vehicle environment for potential collision warning,” In 2020 Zooming Innovation in Consumer Technologies Conference (ZINC), 2020, pp. 178-183.
[23] J.M. Muhammad, C. Buerkle, J. Jarquin, M. Opitz, F. Oboril, *et al.*, “Robustness of object detectors in degrading weather conditions,” In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, USA, 2021, pp. 2719–2724.
[24] F. Reway, A. Hoffmann, D. Wachtel, W. Huber, A. Knoll and E. Ribeiro, "Test method for measuring the simulation-to-reality gap of camera-based object detection algorithms for autonomous driving," In 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020, pp. 1249-1256.
[25] M. Azam, S.A. Hassan, O.C. Puan, S.F. Azhari, and R.U. Faiz, “Performance of autonomous vehicles in mixed traffic under different demand conditions,” *International Journal of Automotive and Mechanical Engineering,* vol. 19, no. 4, pp. 10050–10062, Dec. 2022.
[26] Y. Xiaoyu and M. Marin, “A study on recent developments and issues with obstacle detection systems for automated vehicles,” *MDPI Journal - Sustainability*, vol. 12, no. 8, p. 3281, 2020.
[27] S. Karthikeyan, R.A. Raj, M.V. Cruz, L. Chen, J.L.A. Vishal, and V. S. Rohith, "A systematic analysis on Raspberry Pi prototyping: Uses, challenges, benefits, and drawbacks," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14397-14417, 2023
[28] R. S. Dheekonda, S. Panda, M. N. Khan, M. Hasan, and S. Anwar, “Object detection from a vehicle using deep learning network and future integration with multi-sensor fusion algorithm,” *SAE Technical Papers*, vol. 2017-March, no. March, 2017.
[29] P. Maolanon and K. Sukvichai, “Development of a wearable household objects finder and localizer device using CNNs on Raspberry Pi 3,” In 2018 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (IEEE WIECON-ECE 2018), 2018, pp. 25–28.
[30] Zuhair, MS Az, A. Widiyanto, and S. Nugroho, “Comparison of tensorflow and tensorflow lite for object detection on Raspberry Pi 4,” In AIP Conference Proceedings, vol. 2706, no. 1, p. 020129, 2023.