Automatic Detection of Vehicular Traffic Elements based on Deep Learning for Advanced Driving Assistance Systems

Laura Cleofas-Sánchez[1], Juan Pablo Francisco Posadas-Durán[2],
Pedro Martínez-Ortiz[2], Gilberto Loyo-Desiderio[2],
Eduardo Alberto Ruvalcaba-Hernández[2], Omar González Brito[1]

[1] Tecnológico de Estudios Superiores de Tianguistenco,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Mexico

laura_cs@test.edu.mx, {jposadasd, pmartinezo}@ipn.mx

Abstract. This paper presents a prototype of an automobile driver assistance system based on YOLOv3. The system detects car types, traffic signs, and traffic lights in real-time and warns the driver accordingly. In the learning phase of the YOLO algorithm, the standard weights are learned first, followed by transfer learning to the objects of interest. The retraining phase uses 2,800 images obtained from the Internet of three countries of the real-life, and the testing phase uses real-time videos of Mexico City roads. In the validation phase, the proposed system achieves 95%, 37%, and 40% performance on the compiled dataset for the detection of road elements. The results obtained are comparable and in some cases better than those reported in previous works. Using a Raspberry Pi 4, the prototype was tested in real-life, generating visual and audible warnings for the driver, with an object recognition rate of 0.4 fps. A mean average precision (mAP) of 53% was reached by the proposed system. The experiments showed that the prototype achieved a poor recognition rate and required high computational processing for object recognition. However, YOLO is a model that can have good performance on low-resource hardware.

Keywords. YOLOv3, automobile detection assistance, object recognition, deep learning.

1 Introduction

Motor vehicle accidents have a significant impact on the mortality rate in the Latin America and Caribbean (LAC) region.

In 2020, more than 100,000 deaths and 5 million people were injured due to car accidents in the region. Worldwide, it is estimated that approximately 1.3 million people die as a result of car accidents per year and these accidents are the leading cause of death for people between the ages of 5 and 29, according to a study carried out by the World Health Organization in the year 2021.

In the same year, Mexico ranked seventh in the world in the number of deaths from traffic accidents [10]. There are several causes that can lead a vehicle to suffer an accident. These causes are called risk factors.

Examples of risk factors include mechanical failure, weather conditions, poor road infrastructure and others. Human error is the risk factor that contributes most to accidents. Taking this into account, it has been suggested that driver assistance systems could reduce the percentage of accidents caused by this risk factor [21].

One active area of research is vehicle automation. This area seeks to develop vehicles that are capable of performing various actions without the need for human intervention, for example, the automatic driving of the vehicle [3].

The Society of Automotive Engineers (SAE) proposes a taxonomy for Driving Automation Systems, which has been adopted as a reference for the development of prototypes.

Fig. 1. Road environment of Mexico City

The taxonomy defines 6 levels to classify the driving automation, ranging from level 0 (driving without automation) to level 5 (fully automated driving) [2]. A key element in the early levels of the taxonomy is the Advanced Driver Assistance System (ADAS), which helps to the driver avoid collisions and maintain control of the vehicle by emitting warning signals to the driver or performing specific actions when necessary.

An ADAS uses various sensors to perceive the environment around the vehicle, including cameras, ultrasonic sensors, radar, and others [11]. However, it is important to consider that most vehicles on the road do not have cameras or powerful on-board electronics to integrate an ADAS system.

This paper describes a prototype ADAS based on a machine learning approach using the You Only Look Once (YOLO) algorithm for object detection. The system detects some of the most important elements in a road environment (cars, traffic signals and traffic lights) and warns the driver of the presence of these objects in real-time.

The main contributions of this work are: 1) to recognize a set of photographs of some road environments in Mexico City, 2) to manually label certain road elements of interest (cars, signs and traffic lights), and 3) to evaluate the use of the YOLO road element recognition algorithm using a Raspberry Pi 4 microcomputer.

The rest of the article is organized as follows. Section 2 presents an overview of the sensors used in ADAS systems, Section 3 describes related works, Section 4 contains description of the proposed method, Section 5 describes the experiments performed and the results obtained, Section 6 contains discussion of the results and conclusions.

2 Overview of ADAS Sensors

Vehicle automation is a filed of research and development that involves different stakeholders. The vehicle manufacturing industry, drivers, and organizations all have an interest in the use of this technology to reduce accidents and improve driver safety [13].

Systems that aid safe driving as part of a vehicle can be classified into two categories: passive systems that prevent injuries to vehicle occupants (airbags, seat belts, etc.) and active systems that control the vehicle to avoid accidents (automatic braking, lane following, etc.) [5].

The latter category includes ADAS systems. ADAS provides additional information from the environment around the vehicle to support the driver and assist in the execution of critical actions. The synchronization of a driver actions and environmental information is critical to the efficient performance of the various ADAS applications [20].

An ADAS system typically includes three essential functions: 1) low latency to enable timely hazard detection and warning, 2) high accuracy to reduce false alarms that distract the driver, and 3) high robustness to handle complex and challenging environments [17].

An ADAS uses various sensors in order to obtain information from its environment and to provide driving assistance. Some of the sensors that have been used in the proposed ADAS architectures are mentioned below.

Digital cameras are used to capture images that are further processed to detect and track objects on the road. Cameras can be monocular (for detection of pedestrians, traffic signs, lanes), stereoscopic (for estimating the proximity to another object, lane keeping), or infrared (for use in dark scenarios) [18, 6].

The LIDAR sensor uses lasers to determine how close the vehicle is to other objects and is able to obtain high-resolution 3D images from a greater distance than cameras.
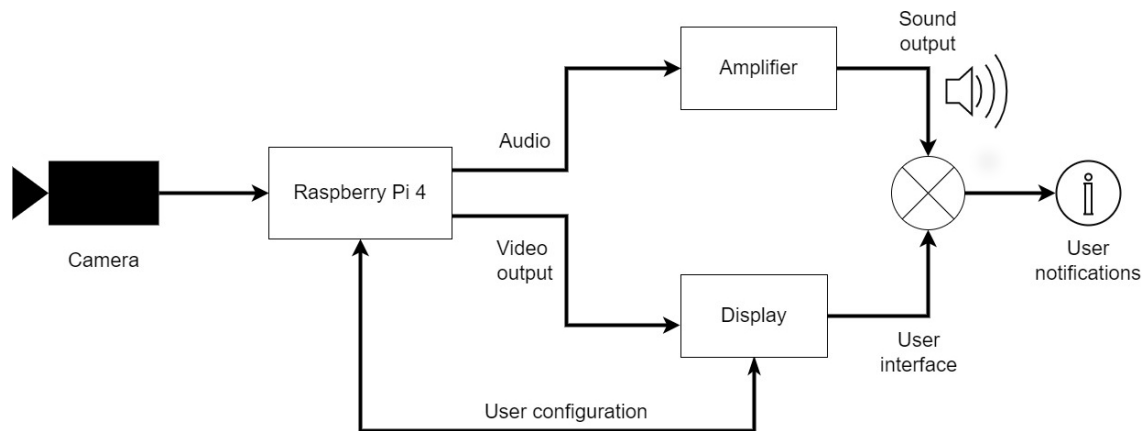
Fig. 2. Assisted driving assistance

LIDAR sensor has been used for object detection, automatic braking and collision avoidance [7]. Radar systems use electromagnetic waves to determine the proximity of objects around the vehicle and the speed at which they are moving.

The detection range offered by radar is greater than that of LIDAR sensors or digital cameras [5]. Some of the applications of radar are blind spot assistance, cross-traffic alert, parking and braking assistance [14, 15]. The ultrasonic sensor or sonar sensor uses sound waves to detect objects close to the vehicle.

An ultrasonic sensor is effective in detecting objects at a short distance from the vehicle. This sensor is used in vehicle parking assistance and near object detection.

3 Vision-based ADAS Related Works

Previous work has proposed various technologies and techniques for environmental sensing and decision making. Computer vision-based ADAS use methods that extract information from images captured by cameras.

The vision-based approach offers the advantage that the devices required for its implementation (cameras and image processing devices) are more affordable compared to technologies such as LIDAR or sonar [9] and has demonstrated efficiency comparable to that obtained by other architectures for specific tasks [6].

Vision-based approach has been used to detect and track obstacles (vehicles, pedestrians, road damage) in front of the vehicle to prevent collisions. The work [6] proposes an ADS that focuses on three actions: lane change detection, collision warning and overtaking vehicle identification.

The proposal uses two monocular cameras to obtain images in both the front and rear view of the vehicle, a digital video recorder (DVR) to store the image sequences and a PC with 4.0 GHz Intel i7 CPU, and Nvidia GTX 1080 GPU for image processing.

Prior to image processing, a heuristic is used to define the adaptive region of interest (ROI) and a CFD-based verification is performed. For overtaking vehicles detection, the CaffeNet [4] is used as the convolutional neural network architecture to identify the objects around the vehicle.

Experiments were conducted on highways and in the city under daylight and night conditions. The article [9] describes a system based on the You Only Look Once (YOLO) model for detecting and marking obstacles. To train the model, video sequences were manually captured while driving on the roads of Tamil Nadu in India.

The videos were collected while driving in the city and on the highway both in the morning and at night. The videos were captured using an 8MP camera connected to a Raspberry Pi and have a format of 640×480 pixels at a rate of 24 frames per second.
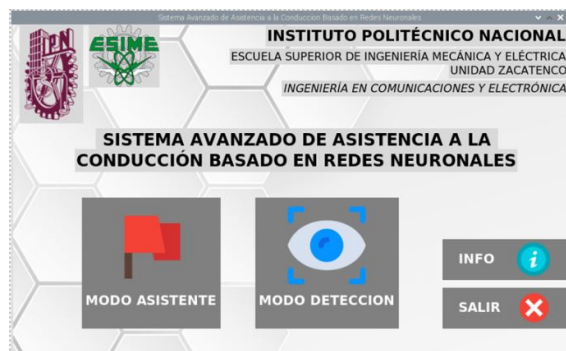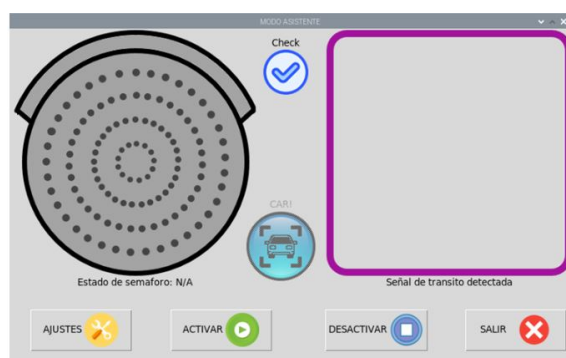
Fig. 3. System called ADAS

Fig. 4. The window of the wizard mode for real-time detection

The resulting images were then manually annotated to identify 2- and 4-wheeled vehicles, pedestrians, animals, speed breakers, road damage and barricades. The system alerts the driver with a buzzer and a visual alert on a mobile application.

The paper evaluates the efficiency of YOLOv3 and YOLOv5 and concludes that YOLOv3 performs better in scenarios where training data is limited and pre-trained weights are not available.

The vision-based approach has also been used for traffic signal and traffic light detection. A prototype for traffic light detection and classification using YOLOv4 is presented in [8]. The prototype additionally alerts the driver if the vehicle does not stop at a red light.

The system was trained using the LISA dataset (images of the streets of California, USA) and tested using images of the streets of Cairo in Egypt.

The proposal uses transfer learning and achieves over 90% in average precision for the three states (green, yellow and red) of the traffic light. The paper [1] describes a deep learning based method for traffic sign detection.

The approach takes an image as input and returns two outputs: the location of the traffic sign in the image and the class to which the traffic sign belongs. A convolutional neural network called Mobile Net is used to perform this task.

To train the network, a dataset of 10,500 images covering 73 types of traffic sign classes was collected from Chinese roads. Testing was performed on a hybrid system consisting of an Intel CPU and an Nvidia GPU with approximately the same performance as an Nvidia AGx module. The proposal achieved an average accuracy of 84.22%.

4 Proposed Method

The ADAS proposal uses a convolutional neural network (CNN) called YOLO-V3, which incorporates the Darknet 53 neural framework. This network received its first training with the pre-training weights downloaded from the repository in [12].

The weights include the recognition of 80 real-world objects that help learn the road environment, such objects correspond to person, bicycle, car, motorbike, aeroplane, bus, train, truck, boat, traffic light, fire hydrant, stop sign, parking meter, bench, bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket, bottle, wine glass, cup, fork, knife, spoon, bowl, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake, chair, sofa, pottedplant, bed, diningtable, toilet, tvmonitor, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink, refrigerator, book, clock, vase, scissors, teddy bear, hair drier, toothbrush.

The CNN's learning was reinforced by a transfer learning process using 2,800 training images of the road environment downloaded from the Internet. Because of the high performance computing required by YOLO, the re-training was done in
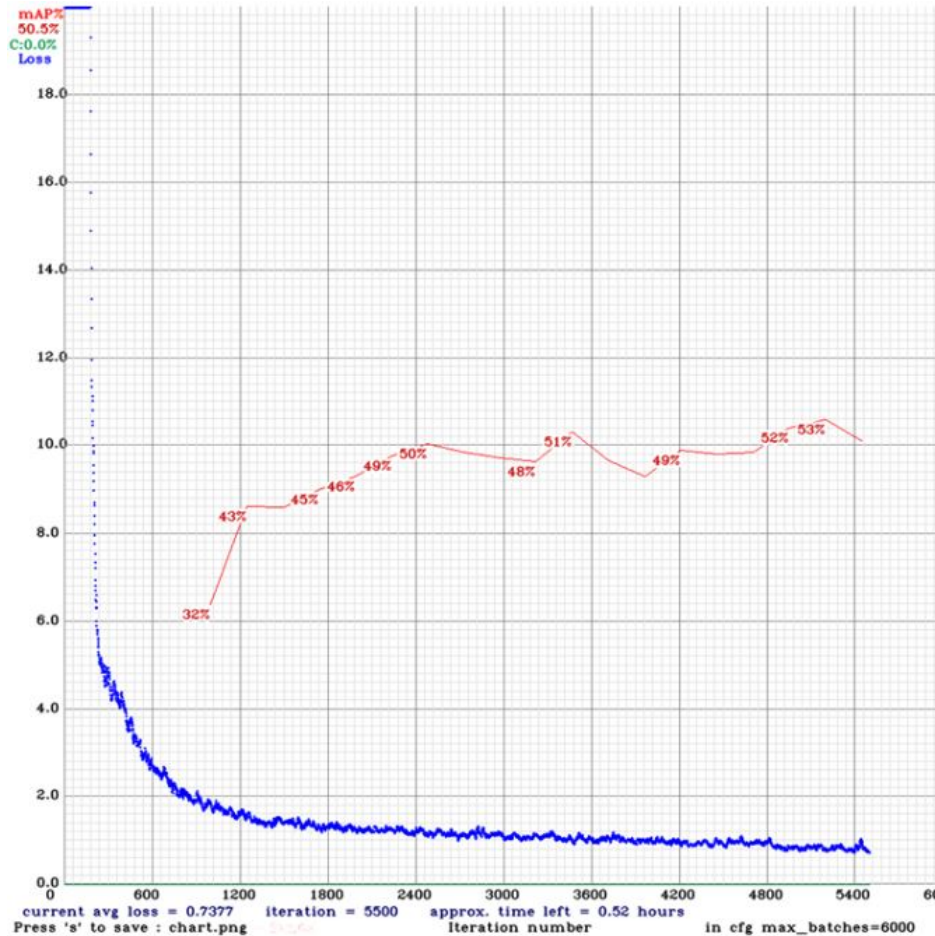
Fig. 5. The loss function versus the mean average precision of each class

Google Colab. For training, 2,800 images of the road environment retrieved from the Internet from different cities around of the world, such as the United States, Chinese, and CDMX of Mexico.

The images included traffic lights (red, yellow and green), traffic signals (preventive, restrictive and informative), and vehicles such as conventional cars, family cars, sports cars, buses, trucks, and trailers.

A tool called Labelimg[1] was used to label each image of the retraining database. In the training phase, 80% of the images were used for retraining and the remaining 20% were used to validate the YOLO learning.

[1]github.com/heartexlabs/labelImg

For testing the model created with YOLO, we tested it with several videos of the road environment of Mexico City (Fig. 1).

For the YOLO configuration, we proceeded to use 6,000 iterations, which indicates obtaining better training of the network that determines a value less than 1 of the total loss function.

Another parameter is the number of batches, which is predefined by default in the network and it should not be lower than the number of labeled images.

Additionally, the number of steps per iteration was equal to the steps obtained by the max batches, which means considering a maximum number of steps between a percentage of 80% and 90%.

The retrained neural network was implemented in the small Raspberry on-board computer. The prototype alerts the driver to the road environment through audio-visual perception.

A webcam connected to the Raspberry via the USB port is used to capture the road environment.

The video of the road environment is processed by the hardware and software system (ADAS) implemented in the Raspberry to identify the images, and then the driver receives an audiovisual notification of his road environment (Fig. 2).

For ADAS detection, the Raspberry Pi executes the YOLO algorithm in real time and displays the detection result in an audiovisual notification. This ADAS system was divided into two phases.

The first phase consists of an interface of a window that displays the ADAS presentation, where the assistant mode performs the detection in the background while the user interface displays a template that contains the elements to be detected in the road environment.

The second phase involved the detection of road elements and audiovisual notifications to prevent the drier (Fig. 3 and Fig. 4).

5 Experiments

When retraining started, the total loss function parameter was high due to the first iterations. However, as the iterations progressed, the loss function decreased, indicating that the network learning was the best.

In the learning process of the YOLO network, the network stores the weights every 100 iterations until the end of the learning process. The average accuracy of the weights was frozen with the best performance during the retraining process.

The total retraining time was about 9 hours at Google Colab. When the retraining is finished, the evaluation of YOLOv3 is done based on the mean average precision and the Intersection over Union (IoU).

In this sense, the value of the mean average precision of each class such as car, traffic signs, and traffic light is 84.92%, 90.76%, and 62.72%, respectively.
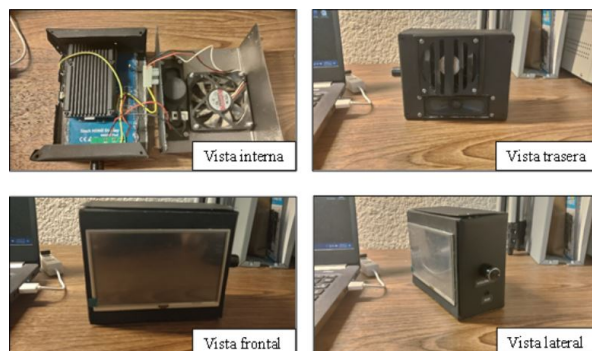

Fig. 6. Assembly of ADAS

Fig. 7. Installation of ADAS in a car

Also, the value of the IoU threshold was 50%. At the end of retraining, YOLO learning is reflected in the plot of the loss function (blue) versus the mean average precision (red) of each class (Fig. 5).

The ADAS was assembled and installed in an automotive system for real-time testing (Fig. 6 and Fig. 7). To control the ADAS, an external computer was remotely connected to the Raspberry.

After that, the YOLOv3 network was implemented to detect the classes using the Tensorflow tools and to acquire the road environment of the CDMX city from the USB camera in real-time.

Although the retraining of YOLO obtained a performance of the detection of objects such as cars, traffic signs, and traffic lights of the 84.92%, 90.76%, and 62.72%, respectively. In the validation phase, the system achieved 95%, 37%, and 40% recognition of traffic signs, cars, and traffic lights, respectively.

Fig. 8. ADAS recognition in real-time

Fig. 9. ADAS recognition in real-time

Considering that the rate of each detection is about 0.4 frames per second. However, it was observed that YOLO hardly detected small images.

Nevertheless, it achieved an mAP of 53% (Fig. 8 and Fig. 9). In the art state, the work [1] uses MobileNet, a convolutional neural network that is developed as a lightweight size model due to its efficiency in saving and combining the output features maps of depthwise convolutions.

The network obtains pointwise convolutions in depthwise separable convolution blocks, which inherently implies the speed (2 s per image) of the network, achieving 84.22% of mAP based on 10,500 images from 73 traffic signs classes.

Conversely, other networks with regular convolution layers do not produce pointwise convolutions. As a result, this type of network consumes more computing time in the convolution layers.

YOLO has an identification time of 0.02 seconds per image, but its performance may be low due to its difficulty recognizing small objects, as well as its difficulty localizing objects close to one another.

In the work of [19], made improvements to the YOLOv2 algorithm, considering it an end-to-end convolutional network involving intermediate convolution layers for obtaining a finer feature map at the top of the layers, and to reduce computational complexity, the network decreases convolutional layers at the top of the layers.

Detection of Chinese traffic signs with a speed of 0.017 seconds per image and a precision of 98%. The work of [16] stated that recognition of traffic signals depends on the network's learning strategy and the real-world environment.

They proposed an end-to-end-deep network with a detection speed of 1.9-1.7x and an accuracy rate of 94 percent. In [9] compares two versions of YOLOv5/YOLOv3 for road environments, including pedestrians, vehicles, animals, speed breakers, and road signs damage.

They implemented the best weights of the networks in an Android Studio device. Based on 5945 images, YOLOv3 achieved 75.5 % and YOLOv5 achieved 72.63%.

6 Discussion and Conclusion

There are two main modes of ADAS: detection and driver assistance. In the first case, the system recognizes objects, while in the second case, the driver assistance provides audiovisual information about the road environment.

To do this, we program in parallel, using threads that perform different functions such as image reading, input model implementation, object detection, and audiovisual outputs for driver assistance in the road environment in real-time.

Although the efficiency and effectiveness of YOLO are good, considering the recognition of 45 frames per second, the best generalization of objects, as well as being a freely available resource

that is constantly updated. Experiments conducted in the present work obtained recognition of 0.4 frames per second from the ADAS system, which is a poor recognition and requires high computational processing for object recognition.

However, YOLO is a model that can have a good performance in low-resource hardware. The recognition process became difficult because the Raspberry Pi 4 generated high temperatures because it does not have a sufficiently acceptable graphics unit, which made it difficult to recognize a large number of objects.

To obtain a better result for the Raspberry Pi, we perform an overclocked it to 2 GHz. This involved the installation of a cooling system and a heat sink to prevent the temperature from exceeding the limit and disabling the system.

Several factors affect object detection, including the lighting conditions, the camera's focus, and the quality of the image. A variety of network structures may also influence research results, such as traditional network models, deep learning models, and improved deep learning models.

According to the state of the art, some works report good results in their experiments [1, 19, 16, 9]. However, experimental results depend on the quality and quantity of the data sets used in the experiments, as well as of the strategy implemented in the data modeling., and also of the rules of learning implemented in the convolution algorithms., as well as of the real-time environment in which the experiments are made.

Nevertheless, they do not test a driving assistance system on a small-board computer that detects traffic signals in real-time. It except in the work [9] where they compare two versions of YOLOv3 and YOLOv5 for road environments recognition, where the models were developed using 5945 images implemented on an Android Studio device, obtaining a performance of 74.5% for YOLOv3 and 72.65% for YOLOv5.

However, it is important to mention that our model was trained with 2,800 images, proving in time real without manipulating an appropriate road environment. Also these works do not consider a complete ADAS system of driver assistance in real-time that detects and displays the result in audiovisual notifications [1, 19, 16, 9].

Nevertheless, our ADAS system considers audiovisual notifications as ADAS driver assistance in real-time.

References

1. **Ayachi, R., Afif, M., Said, Y., Atri, M. (2019).** Traffic signs detection for real-world application of an advanced driving assisting system using deep learning. Neural Processing Letters, Vol. 51, No. 1, pp. 837–851. DOI: 10.1007/s11063-019-10115-8.

2. **Bogdoll, D., Orf, S., Töttel, L., Zöllner, J. M. (2022).** Taxonomy and survey on remote human input systems for driving automation systems. Lecture Notes in Networks and Systems, Springer International Publishing, pp. 94–108. DOI: 10.1007/978-3-030-98015-3_6.

3. **Ha, P., Chen, S., Du, R., Dong, J., Li, Y., Labi, S. (2020).** Vehicle connectivity and automation: A sibling relationship. Frontiers in Built Environment, Vol. 6. DOI: 10.3389/fbuil.2020.590036.

4. **Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T. (2014).** Caffe: Convolutional architecture for fast feature embedding. DOI: 10.48550/ARXIV.1408.5093.

5. **Kumar Kukkala, V., Tunnell, J., Pasricha, S., Bradley, T. (2018).** Advanced driver-assistance systems: A path toward autonomous vehicles. IEEE Consumer Electronics Magazine, Vol. 7, No. 5, pp. 18–25. DOI: 10.1109/mce.2018.2828440.

6. **Lin, H. Y., Dai, J. M., Wu, L. T., Chen, L. Q. (2020).** A vision-based driver assistance system with forward collision and overtaking detection. Sensors, Vol. 20, No. 18, pp. 5139. DOI: 10.3390/s20185139.

7. **Maksymova, I., Greiner, P., Steger, C., Niedermueller, L. C., Druml, N. (2020).** Adaptive MEMS mirror control for reliable automotive driving assistance applications. 23rd Euromicro Conference on Digital System Design (DSD), IEEE. DOI: 10.1109/dsd51259.2020.00080.

8. **Mostafa, M., Ghantous, M. (2022).** A YOLO based approach for traffic light recognition for ADAS systems. 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), IEEE. DOI: 10.1109/miucc55081.2022.9781682.

9. **Neelam Jaikishore, C., Podaturpet Arunkumar, G., Jagannathan Srinath, A., Vamsi, H., Srinivasan, K., Karthik Ramesh, R., Jayaraman, K., Ramachandran, P. (2022).** Implementation of deep learning algorithm on a custom dataset for advanced driver assistance systems applications. Applied Sciences, Vol. 12, No. 18, pp. 8927. DOI: 10.3390/app12188927.

10. **Pan American Health Organization (2019).** Status of road safety in the region of the Americas. Pan American Health Org.

11. **Raviteja, S., Shanmughasundaram, R. (2018).** Advanced driver assitance system (ADAS). Second International Conference on Intelligent Computing and Control Systems, IEEE, pp. 737–740. DOI: 10.1109/iccons.2018.8663146.

12. **Redmon, J. (2016).** Darknet: Open source neural networks in c. pjreddie.com/darknet/.

13. **Ross, H. L. (2021).** Safety for future transport and mobility. Springer.

14. **Sligar, A. P. (2020).** Machine learning-based radar perception for autonomous vehicles using full physics simulation. IEEE Access, Vol. 8, pp. 51470–51476. DOI: 10.1109/access.2020.2977922.

15. **Sun, S., Petropulu, A. P., Poor, H. V. (2020).** MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges. IEEE Signal Processing Magazine, Vol. 37, No. 4, pp. 98–117. DOI: 10.1109/msp.2020.2978507.

16. **Wan, J., Ding, W., Zhu, H., Xia, M., Huang, Z., Tian, L., Zhu, Y., Wang, H. (2020).** An efficient small traffic sign detection method based on YOLOv3. Journal of Signal Processing Systems, Vol. 93, No. 8, pp. 899–911. DOI: 10.1007/s11265-020-01614-2.

17. **Wang, T. (2017).** Phd forum: Real-time lane-vehicle detection for advanced driver assistance on mobile devices. IEEE International Conference on Smart Computing, pp. 1–2. DOI: 10.1109/smartcomp.2017.7947034.

18. **Wijaya, K. T., Bharoto, L. Y., Purwanto, A., Syamsuddin, E. Y. (2020).** Vision-based parking assist system with bird-eye surround vision for reverse bay parking maneuver recommendation. International Electronics Symposium, IEEE, pp. 102–107.

19. **Zhang, J., Huang, M., Jin, X., Li, X. (2017).** A real-time chinese traffic sign detection algorithm based on modified YOLOv2. Algorithms, Vol. 10, No. 4, pp. 127. DOI: 10.3390/a10040127.

20. **Ziebinski, A., Cupek, R., Grzechca, D., Chruszczyk, L. (2017).** Review of advanced driver assistance systems (ADAS). AIP Conference Proceedings, Vol. 1906, No. 1, pp. 120002. DOI: 10.1063/1.5012394.

21. **Zornoza Somolinos, A. (2021).** Vehículos automatizados y seguro obligatorio de automóviles: Estudio de derecho comparado. , pp. 1–273DOI: 10.2307/j.ctv20hctng.