Intelligent Traffic Light System using Deep Reinforcement Learning

RICARDO YAURI[1,2], FRANK SILVA[1], ADEMIR HUACCHO[1], OSCAR LLERENA[3]
[1]Universidad Nacional Mayor de San Marcos,
Lima,
PERU

[2]Facultad de Ingeniería,
Universidad Tecnológica del Perú,
Lima,
PERU

[3]Seoul National University of Science and Technology,
Seoul,
SOUTH KOREA

Abstract: - Currently, population growth in cities results in an increase in urban vehicle traffic. That is why it is necessary to improve the quality of life of citizens based on the improvement of transport control services. To solve this problem, there are solutions, related to the improvement of the road infrastructure by increasing the roads or paths. One of the solutions is using traffic lights that allow traffic regulation automatically with machine learning techniques. That is why the implementation of an intelligent traffic light system with automatic learning by reinforcement is proposed to reduce vehicular and pedestrian traffic. As a result, the use of the YOLOv4 tool allowed us to adequately count cars and people, differentiating them based on size and other characteristics. On the other hand, the position of the camera and its resolution is a key point for counting vehicles by detecting their contour. An improvement in time has been obtained using reinforcement learning, which depends on the number of episodes analyzed and affects the length of training time, where the analysis of 100 episodes takes around 12 hours on a Ryzen 7 computer with a graphics card built-in 2 GB.

Key-Words: -Reinforcement learning, traffic light, deep neural networks, image processing, ESP32, Yolo

Received: November 15, 2022. Revised: July 17, 2023. Accepted: August 15, 2023. Published: September 12, 2023.

1 Introduction

The growth of the world population brings with it that cities grow, making it necessary to keep them interconnected. This implies meeting the needs of citizens such as transportation. Many of them acquire motorized vehicles, affecting urban environments and generating congestion. In addition, due to the increase in drivers, some do not have an efficient vehicle education, so this problem is increasing in urban cities around the world, [1], [2].

In addition, the result of poor installations and traffic increases the possibility of accidents that cause injuries and even deaths in citizens. These accidents currently represent one of the major causes of death worldwide and the numbers are increasing every year, [3], [4].

As economic consequences, traffic is detrimental to both the state and citizens. For example, in 2010 the United States recorded a loss of 115 billion dollars in relation to the loss of time of people stuck in traffic. On the other hand, in South America according to recent studies, Peru was classified within the top 3 cities with the highest, [5], and only in Lima Peru, there are around 45 critical points where vehicular chaos occurs at any time of the day.

To deal with this problem of automatic traffic management, there are different methods, such as improving the road infrastructure by increasing, widening the roads, or increasing the personnel of vehicle flow control. In the case of Peru, these solutions are not effective because there is a great proliferation of informal transport that usually fills the main avenues. One of the methods used is the use of traffic lights that allow traffic regulation, which must be synchronized and in some cases converted into intelligent traffic lights, [6]. Others involve the use of car presence sensors at street crossings and vehicular traffic prediction. This can be achieved through machine learning techniques that consider variables such as: time, the number of pedestrians, and the size of the car, among others, [7], [8]. Some of the machine learning techniques can be integrated into embedded systems with low hardware resources for applications in areas related to vehicular traffic, security, agriculture, and environmental monitoring, [9], [10].

For all the above, this research proposes the following research question: How is it possible to automate traffic light systems to improve traffic in urban centers. Therefore, the objective of the research is to implement an intelligent traffic light system is proposed, with automatic learning by reinforcement, to reduce vehicular and pedestrian traffic.

To develop the objective, vehicle and pedestrian detection algorithms are implemented, with images captured by cameras connected to a processor linked to a data storage platform. In addition, using machine learning in the cloud, remote control, and real-time traffic management are achieved. The specific objectives are: Select an algorithm for image processing and deep learning applied to the recognition of vehicles and pedestrians; Communicate the camera and a control program in Python for light management; and Perform system performance tests considering response times.

This research provides value to know how an intelligent system can be implemented to solve traffic problems in urban areas through automatic traffic lights. In addition, these intelligent traffic lights make it possible to increase the flow of traffic, generating benefits for the population related to saving time and improving the quality of life.

This paper has been divided into the following sections. Related works are shown in section 2. Subsequently, in section 3, the concepts and technologies used in machine learning and deep learning methods to optimize traffic are described. Section 4 shows the system implementation process. The results obtained are described in section 5 and finally, in section 6 the conclusions are mentioned.

2 Literature Review

In the analysis of the authors' papers, several benefits stand out in relation to the implemented systems. On the one hand, the design of intelligent systems for the detection and classification of vehicles through Deep Learning is proposed, with communication through 4G and Ethernet modules, improving the synchronization of traffic lights using image processing tools based on Python and OpenCV. Other approaches focus on reinforcement learning for traffic light control, proving its effectiveness in unbalanced traffic scenarios. These approaches highlight how technology can contribute to traffic control in smart cities, optimizing traffic light management and improving traffic flow.

In some works, the design of a system that records data on the magnitude of traffic and develops an algorithm for the synchronization of traffic lights is proposed, [11], [12]. This intelligent system performs the detection and classification of vehicles using Deep Learning, where it will have a camera that focuses on the streets to measure the flow of vehicular traffic. Communication with the server is done through a 4G module and the Ethernet protocol for communication with the traffic light.

One way to recognize moving cars is through image processing tools based on Python and OpenCV. These allow video processing in real-time as described in, [13], where the use of the "Background Subtractor GMG" is highlighted, which helps to recognize cars with background subtraction and contour detection. The system consists of three stages, where The first performs the configuration and initialization of a video flow camera. Finally, the vehicles are counted using the removed background image.

Another paper, presents the design of an advanced perception and localization system for autonomous driving applications, which includes a high-resolution lidar, a stereo camera, an inertial navigation system and an integrated computer, [14]. The system incorporates perception and localization algorithms to provide real-time information on the location of objects in environments without GPS. A dataset was built under various driving conditions, and the algorithms demonstrated competitive performance and processing times compatible with autonomous driving applications.

Smart traffic lights in smart cities can optimally reduce traffic congestion as described in some research, [15], [16]. In the paper developed in, [15], reinforcement learning is used to train the control agent of a traffic light in an urban mobility simulator. A policy-based deep reinforcement learning method, Proximal Policy Optimization (PPO), is used instead of value-based methods such as Deep Q Network (DQN) and Double DQN (DDQN). As a result, it is shown that an intelligent semaphore can work moderately well in unbalanced traffic scenarios, learning from the optimal policies in these scenarios.

To contribute to traffic control, [17], proposes the development of a portable traffic light enabled by artificial intelligence in the cloud with the ability to work autonomously based on the volume of the car flow. The design involves an ESP32 module for system control and serves as a gateway to the internet.

The current study differs from the available literature by focusing on the specific implementation of vehicle and pedestrian detection algorithms through images captured by cameras. These data are processed by a system that includes a processor connected to a data storage platform. In addition, the use of automatic learning in the cloud is highlighted, which allows remote control and traffic management in real-time. This approach focuses on the optimization and automation of vehicle control, making use of current technologies for greater efficiency in traffic management.

3 Intelligent Traffic Lights and Image Processing

3.1 Smart Traffic Lights

Smart traffic lights have undergone significant evolution over the past century, resulting in a variety of device types. These variations serve a variety of purposes, including vehicular signals, pedestrian signals, audio signals for the visually impaired, and flashing or flashing indicators. However, the appearance of smart traffic lights has revolutionized urban traffic management, [18].

These smart lights have autonomous decision-making capabilities, responding to external factors such as vehicle density and average speed to optimize traffic flow. Smart traffic lights come in various algorithmic implementations, including those that take advantage of radio frequency identification, wireless sensor networks, image processing, and artificial intelligence.

This advanced technology is not limited to the mere regulation of traffic; strives for efficient vehicle control by identifying and organizing congested areas to avoid traffic jams and possible accidents. The application of smart traffic lights has initiated a flourishing field of research, characterized by cutting-edge solutions that integrate machine vision technologies into urban infrastructure. As cities continue to expand, optimizing traffic management through smart systems becomes paramount, [19].

3.2 Image Processing

To achieve efficient image processing, it is essential to consider a few crucial components, which are detailed below. The convergence of these components in image processing opens a range of possibilities for applications in a wide variety of fields, from facial identification in security to medical image analysis, marking an era of significant advances in the understanding and manipulation of visual data.

- Camera. Electronic device that captures and records moving images whose number of frames determines the basic visual quality of the video, [20]. Also, it comes with various extra features like focus, rotation, or other plugins.
- Image processing. Techniques and processes are used to discover characteristics of an image using a computer as the main tool.
- Face detection. It is the process that identifies the region corresponding to a face in an image. Usually, this is a rectangular area for face position and orientation, [21], [22].

3.3 Reinforcement Learning

Machine learning is a branch of computer science that focuses on the analysis and interpretation of patterns, and data structures to learn and make decisions without human intervention, [22], [23]. One of its defining features lies in its ability to process large amounts of information, compensating for human limitations to process such data quickly and efficiently. Within this scope, three fundamental categories emerge: supervised, unsupervised, and reinforcement learning, [24].

Reinforcement learning is formed by an intelligent agent learning to optimize the decision-making process, [25]. For the machine to learn, the agent interacts with the real decision-making process or a simulation of it, observing the environment, making decisions, and observing their effects. If the outcome of the decision is favorable, the agent automatically learns to repeat that decision in the future. On the contrary, if the result is unfavorable, the agent will not make the same decision again, [26], as shown in Fig. 1.

This mechanism endows the agent with a learning process that reflects regulatory functions like those found in living organisms, progressively determining the most appropriate decisions for various scenarios. Deep learning models constitute the "brain" of this agent and embody its learning capacity. Among the spectrum of reinforcement learning methods, Sarsa and Q-learning stand out, [27], [28].

4 Design and Development

The system development process involves analyzing data, collecting traffic videos, and capturing various photos to compare resolution. Technological tools are chosen for the development of the system, including a Python-compatible IDE for the implementation of reinforcement learning algorithms. The system is physically built using hardware modules, implementing object counting and detection. Subsequently, field tests are carried out in places with unobstructed traffic visibility, comparing it with a traditional traffic light configuration. The development process considers the following stages (Fig. 2):

- Analysis and Data Collection. The analysis of the traffic index and the collection of traffic videos are carried out. In addition to this, it takes various photos with different resolutions.

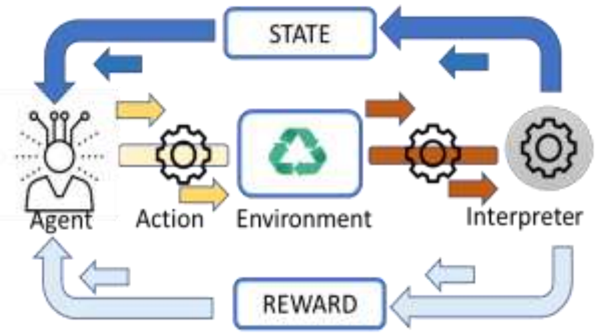


Fig. 1: Scheme of reinforcement learning, [29].

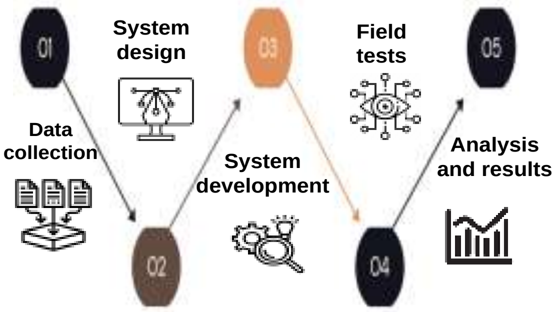


Fig. 2: Development process

- System design. The technological tools for the development of the system are selected, such as the development environment (IDE) compatible with the Python programming language. The software components are also selected to carry out the implementation of deep learning algorithms by reinforcement.
- System development. The system will be physically developed with the hardware modules for the system deployment and the code for object detection and counting using reinforcement learning will be developed.
- Field tests. The solution is deployed in a place where there will be a line of sight for traffic analysis.
- Analysis and Results. The data obtained in the field test is analyzed where the performance will be seen against a traditional traffic light.

4.1 Operation Diagram

The system starts with a video recording that reaches the processing software made based on the framework and libraries of the YOLO tool. In this way, vehicles and pedestrians are detected and counted, and then, through reinforcement learning, change times are obtained with respect to the last capture at the traffic light (Fig. 3).

Based on the previous functions diagram, a series of hardware and software components are used to integrate the trained algorithms in the automatic traffic light system (Fig. 4), made up of cameras, 32-bit hardware modules, and cloud applications.

The components that will be used in the previously defined system are:

- Traffic lights. For the physical design of the traffic light, the cost of implementation and the materials available for manufacturing are considered. 3 luminaires (red, green, and orange) with a diameter of approximately 20 cm are used and are controlled by the Particle Boron electronic card. Thanks to the use of this electronic board, it will be possible to manage the delays of the luminaires through the control of relays.
-

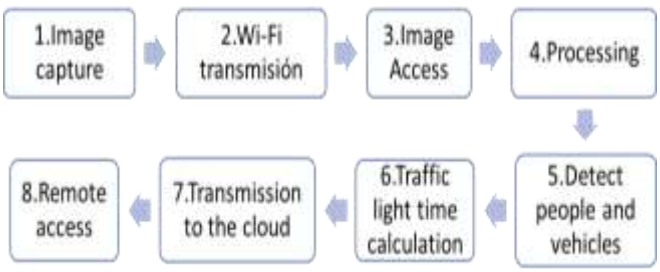


Fig. 3: Block diagram of the system operation.

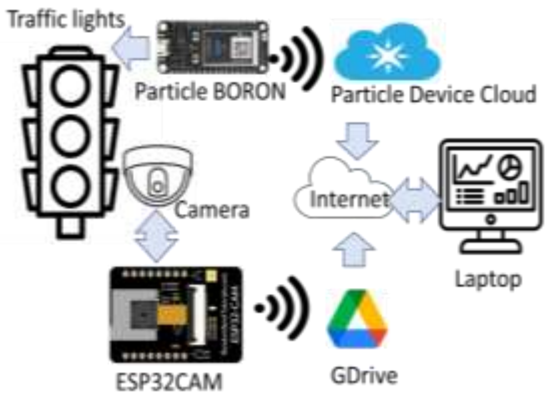


Fig. 4: System hardware and software components

- OV2640 camera. Compatible with the esp32CAM which will give us a resolution of 1600x1200.
- ESP32CAM module. It can access the internet via Wi-Fi, so we can upload the detection results directly to the online storage cloud. To connect this module with web services on the internet we must use the Google script token and the Wi-Fi network credentials (Fig. 5).
- Cloud GDrive Apps Script. The cloud storage platform (Cloud) will be connected to the hardware module to establish communication. To send information from the esp32CAM to the cloud, a Google script is used to manage files from the terminal with Python.

For the transmission of images to Internet services, space repositories in Google Drive are used, for which configuration parameters such as the Google script token and Wi-Fi network credentials are used. This setup is done within the ESP32CAM program, where an infinite loop takes pictures and sends them to the Google Drive cloud (Fig. 5).

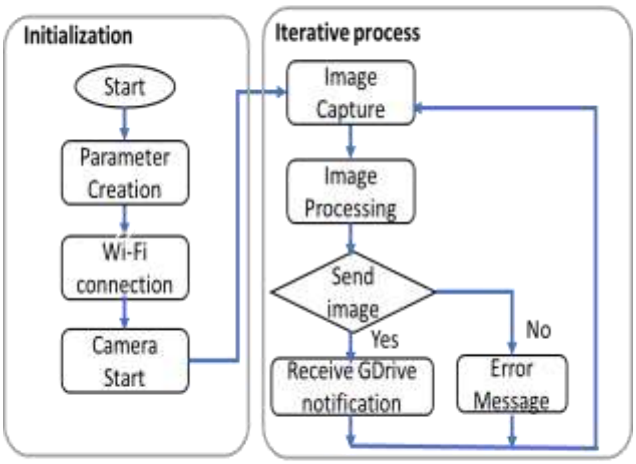


Fig. 5: ESP32CAM Programming Flowchart

4.2 Classification Tool Using Yolo

On the other hand, for the use of YOLO, a series of steps shown in Fig. 6 must be conducted. First, the dependencies and libraries necessary to use the reinforcement learning (RL) techniques are imported, and then the libraries are downloaded. Darknet.

For the counting of objects, it is necessary to introduce a counter that registers the detected elements in a list. Then a function is defined that counts the repeated elements. Thus, the total number of vehicles is obtained.

Fig. 6: YoLo Configuration Diagram

4.3 Vehicular Traffic Simulator

The SUMO traffic simulator is an open-source package, which allows the simulation of various situations and forms of streets for analysis. In our case, it will be used to train the Reinforcement learning algorithm where various situations are evaluated. Additionally, this paper installs the Anaconda tool along with the relevant TensorFlow libraries and GPU device drivers.

5 Results and Discussion

The tests were carried out by evaluating the stage of the car counter by viewing various images and photographs obtained through the deep learning process with training processes of 100 episodes and 25 episodes.

5.1 Yolo V4 Counter

The analysis of results was generated from different positions of the camera. In addition, counting data is obtained for the reinforcement learning process. Fig. 7 shows the detection from the left view, where, due to the low position, it is not able to recognize all the cars, but even so, the result is acceptable. Fig. 8 shows a greater number of detected cars considering the best camera position compared to other positions (Fig. 9). The counting results of objects detected and registered in a dictionary-type file are shown in Fig. 10. This discussion underscores the interplay between model architecture, camera positioning, and object detection outcomes, highlighting the potential for refinement in subsequent iterations.

Fig. 7: Car count (left view)

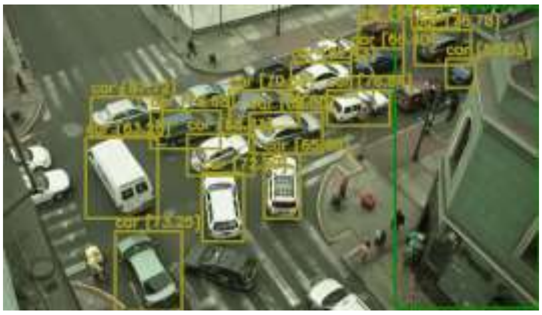
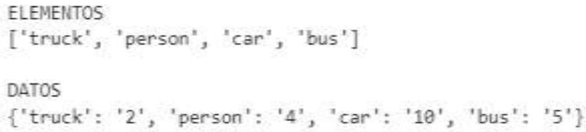


Fig. 8: Car count (top view)

Fig. 9: Car counting (side view)



```
ELEMENTOS
['truck', 'person', 'car', 'bus']

DATOS
{'truck': '2', 'person': '4', 'car': '10', 'bus': '5'}
```

Fig. 10: Object Counting Dictionary

5.2 Learning Process

During the learning process, the agent will start training in the background using the configuration file “training_settings.ni”. In this way, the results are visualized during the training process. Fig. 11 shows the visualization of the simulation using the SUMO-GUI software for each training episode. Upon completion of training, the outcome includes graphs displaying detected objects, an "ini" configuration file containing agent settings, and the trained neural network.

5.3 Waiting Time

The simulation involved 25 episodes, with 1000 cars per episode, represented by yellow arrows. Within each episode, randomly generated vehicles varied in arrival arrangements. As evident in Fig. 12, an accumulated waiting time delay highlights the algorithm's progressive enhancement across episodes, leading to time reduction. The agent stores the average waiting time for vehicles through experience repetition, subsequently reflected in the graph of the average queue duration of vehicles displayed in Fig. 13.

Additionally, the average queue duration of vehicles further underscores the algorithm's success in optimizing traffic conditions. This outcome aligns with the core principle of reinforcement learning, where continuous exposure to real-world scenarios enables the agent to refine its strategies and achieve better results.

6 Conclusions

The YOLOv4 port provides the ability to perform car and people counts by differentiating different types based on size, assigning a weight to each of them, and storing the results in a Python dictionary-type file.

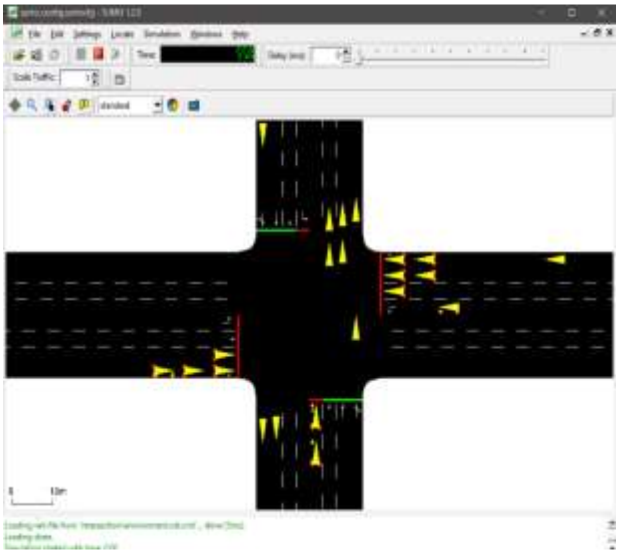


Fig. 11: Capture of training during the episode

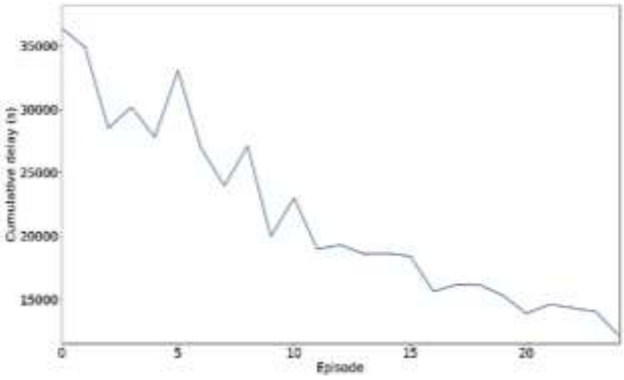


Fig. 12: Accumulated delay or waiting time for 25 episodes

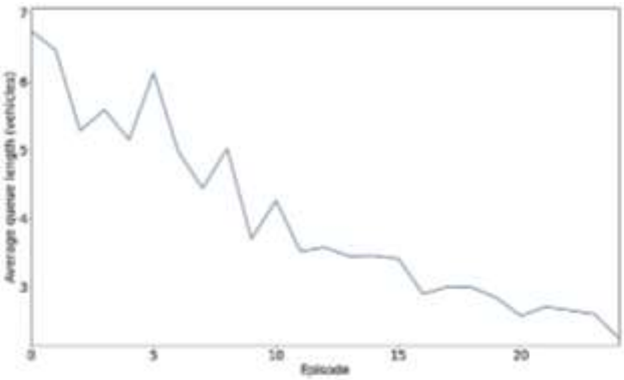


Fig. 13: Average length of queues (vehicles) for 25 episodes

The position of the camera is a key point for the correct counting of the vehicles, which must be at the top, to see the outline of the vehicles for their correct counting. Another critical element is the resolution, where the camera used is acceptable for the system, where the use of a focus lens improves image quality.

An improvement has been obtained in the time used for the RL where this depends on the number of episodes analysed. It is recommended to do several tests of the RL algorithm to improve the results, evaluating how the duration of the training time is affected, since the analysis of 100 episodes takes about 12 hours on a Ryzen 7 computer with 2 GB integrated graphics.

The combination of open-source software and commonly used components allows the simulation to be implemented in a short time, which is a direct advantage. In addition, thanks to existing libraries and standard use, algorithms and data processing are improved.

The limitations of the research cover key aspects such as the need for a precise position of the camera for an exact count and having a relatively long training time of the deep reinforcement learning algorithm, whose scope in its fulfillment was reduced by technological and logistical limitations. These limitations suggest the potential for further improvement in the applicability and efficiency of the system in various settings.

Future directions could optimize the deep reinforcement learning algorithm by further testing to reduce the length of training time. Additionally, you can explore the influence of different hardware configurations, such as cameras, lenses, and graphics cards, on system performance and verify system efficiency and accuracy in potential use cases other than vehicles and people.

References:

[1] Y. Kitamura, M. Hayashi, and E. Yagi, “Traffic problems in Southeast Asia featuring the case of Cambodia’s traffic accidents involving motorcycles,” *IATSS Res.*, vol. 42, no. 4, pp. 163–170, Dec. 2018, doi: 10.1016/J.IATSSR.2018.11.001.

[2] M. Thibenda, D. M. P. Wedagama, and 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,” *Saf. Sci.*, vol. 155, p. 105869, Nov. 2022, doi: 10.1016/J.SSCI.2022.105869.

[3] M. Makhani and N. Bodkhe, “Road Traffic Accidents and their Aftermath: The Victims Perspective,” *Int. J. Med. Toxicol. Leg. Med.*, vol. 25, no. 3–4, pp. 67–74, Jul. 2022, doi: 10.5958/0974-4614.2022.00052.3.

[4] A. M. Ngoc, C. C. Minh, N. T. Nhu, H. Nishiuchi, and N. Huynh, “Influence of the

human development index, motorcycle growth and policy intervention on road traffic fatalities – A case study of Vietnam," *Int. J. Transp. Sci. Technol.*, Sep. 2022, doi: 10.1016/J.IJTST.2022.09.004.

[5] Numbeo, "Traffic Index by Country 2023," 2023. https://www.numbeo.com/traffic/rankings_by_country.jsp (accessed Jan. 24, 2023).

[6] A. Brena, J. Vasquez, M. Silvera, and F. Campos, "Reduction of BRT delays at highway intersections through adaptive traffic lights control," 2022. doi: 10.1109/CONIITI57704.2022.9953668.

[7] Y. Wang, T. Xu, X. Niu, C. Tan, E. Chen, and H. Xiong, "STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control," *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 2228–2242, Jun. 2022, doi: 10.1109/TMC.2020.3033782.

[8] M. A. Basmassi, S. Boudaakat, J. A. Chentoufi, L. Benameur, A. Rebbani, and O. Bouattane, "Evolutionary reinforcement learning multi-agents system for intelligent traffic light control: new approach and case of study," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 5, pp. 5519–5530, Oct. 2022, doi: 10.11591/IJECE.V12I5.PP5519-5530.

[9] R. Yauri, A. Castro, R. Espino, and S. Gamarra, "Implementation of a sensor node for monitoring and classification of physiological signals in an edge computing system," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 28, no. 1, pp. 98–105, Oct. 2022, doi: 10.11591/IJEECS.V28.I1.PP98-105.

[10] R. Yauri and R. Espino, "Edge device for movement pattern classification using neural network algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 1, pp. 229–236, Apr. 2023, doi: 10.11591/IJEECS.V30.I1.PP229-236.

[11] S. P. Yadav, "Vision-based detection, tracking, and classification of vehicles," *IEIE Trans. Smart Process. Comput.*, vol. 9, no. 6, pp. 427–434, Dec. 2020, doi: 10.5573/IEIESPC.2020.9.6.427.

[12] R. Zhu, L. Li, S. Wu, P. Lv, Y. Li, and M. Xu, "Multi-agent broad reinforcement learning for intelligent traffic light control," *Inf. Sci. (Ny).*, vol. 619, pp. 509–525, Jan. 2023, doi: 10.1016/J.INS.2022.11.062.

[13] P. Bailke and S. Divekar, "Real-Time Moving Vehicle Counter System using Opencv and Python," *Int. J. Eng. Appl. Sci. Technol.*, vol. 6, pp. 190–194, 2022, Accessed: Jan. 24, 2023. [Online]. Available: http://www.ijeast.com

[14] X. Dauptain, A. Koné, D. Grolleau, V. Cerezo, M. Gennesseaux, and M. T. Do, "Conception of a High-Level Perception and Localization System for Autonomous Driving," *Sensors*, vol. 22, no. 24, Dec. 2022, doi: 10.3390/S22249661.

[15] Y. Zhu, M. Cai, C. W. Schwarz, J. Li, and S. Xiao, "Intelligent Traffic Light via Policy-based Deep Reinforcement Learning," *Int. J. Intell. Transp. Syst. Res.*, Dec. 2022, doi: 10.1007/S13177-022-00321-5.

[16] J. Liu, S. Qin, Y. Luo, Y. Wang, and S. Yang, "Intelligent Traffic Light Control by Exploring Strategies in an Optimised Space of Deep Q-Learning," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 5960–5970, Jun. 2022, doi: 10.1109/TVT.2022.3160871.

[17] B. Kamasetty, M. Renduchintala, L. L. Shetty, S. Chandarshekar, and R. Shettar, "Design and development of portable smart traffic signaling system with cloud-artificial intelligence enablement," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 1, pp. 116–126, Apr. 2022, doi: 10.11591/IJEECS.V26.I1.PP116-126.

[18] Desmira, M. A. Hamid, N. A. Bakar, M. Nurtanto, and Sunardi, "A smart traffic light using a microcontroller based on the fuzzy logic," *IAES Int. J. Artif. Intell.*, vol. 11, no. 3, pp. 809–818, Sep. 2022, doi: 10.11591/IJAI.V11.I3.PP809-818.

[19] A. Navarro-Espinoza *et al.*, "Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms," *Technol. 2022, Vol. 10, Page 5*, vol. 10, no. 1, p. 5, Jan. 2022, doi: 10.3390/TECHNOLOGIES10010005.

[20] O. Daisuke, "Subscribe to the weekly Japan Media Review newsletter! printable version Camera phones changing the definition of picture-worthy," Tokyo, 2002.

[21] R. Mohammadian Fini, M. Mahlouji, and A. Shahidinejad, "Real-time face detection using circular sliding of the Gabor energy and neural networks," *Signal, Image Video Process.*, vol. 16, no. 4, pp. 1081–1089, Jun. 2022, doi: 10.1007/S11760-021-02057-3.

[22] X. Guan, J. Huang, and T. Tang, "Robot vision application on embedded vision implementation with digital signal processor," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 1, Jan. 2020, doi: 10.1177/1729881419900437.

[23] E. M. Farella, S. Malek, and F. Remondino, "Colorizing the Past: Deep Learning for the Automatic Colorization of Historical Aerial Images," *J. Imaging 2022*, Vol. 8, p. 269, vol. 8, no. 10, p. 269, Oct. 2022, doi: 10.3390/JIMAGING8100269.

[24] V.-R. Dănăilă, S. Avram, and C. Buiu, "The applications of machine learning in HIV neutralizing antibodies research—A systematic review," *Artif. Intell. Med.*, vol. 134, p. 102429, Dec. 2022, doi: 10.1016/j.artmed.2022.102429.

[25] M. Wiering and M. van Otterlo, Eds., "Reinforcement Learning," vol. 12, 2012, doi: 10.1007/978-3-642-27645-3.

[26] L. P. Kaelbling, M. L. Littman, and A. W. Moore, *Reinforcement Learning: A Survey*, vol. 4. Morgan Kaufmann Publishers, 1996. doi: 10.1613/JAIR.301.

[27] D. Zhou, G. Sun, W. Lei, and L. Wu, "Space Noncooperative Object Active Tracking with Deep Reinforcement Learning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, pp. 4902–4916, Dec. 2022, doi: 10.1109/TAES.2022.3211246.

[28] L. Chen, K. Fu, Q. Zhao, and X. Zhao, "A multi-channel and multi-user dynamic spectrum access algorithm based on deep reinforcement learning in Cognitive Vehicular Networks with sensing error," *Phys. Commun.*, vol. 55, Dec. 2022, doi: 10.1016/J.PHYCOM.2022.101926.

[29] J. Orr and A. Dutta, "Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey," *Sensors 2023, Vol. 23, Page 3625*, vol. 23, no. 7, p. 3625, Mar. 2023, doi: 10.3390/S23073625.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

All authors have contributed equally to the creation of this article.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflict of interest to declare.