Advancing Traffic Management: A Real-Time CNN-Based Multi-Vehicles Speed Estimation and License Plate Recognition.

No Author Given

No Institute Given

Abstract. The growing complexity of modern traffic systems calls for robust, real-time solutions for vehicle monitoring and enforcement. This paper presents an integrated framework for multi-vehicle speed estimation and license plate recognition (LPR), combining the strengths of three state-of-the-art technologies: YOLOv11 for object detection, SORT for multi-object tracking, and EasyOCR for optical character recognition. The system detects vehicles and license plates in real-time video streams, tracks their movement, estimates vehicle speed using pixel displacement and temporal data, and recognizes license plate characters with high accuracy. Experimental results show that YOLOv11 achieves superior performance compared to YOLOv8 and YOLOv9, with a precision of 94.5%, recall of 75.9%, F1-score of 84.2%, and mAP@0.5 of 85.6%. The model demonstrates robust detection even under challenging conditions such as occlusion, Perspective Distortion, and plate angle variation. The results validate the model's potential for integration into Algerian national traffic management systems and future smart city infrastructure.

Keywords: CNN · YOLOv11 · traffic · speed estimation · license plate recognition.

1 Introduction

The exponential increase in global vehicular traffic—fueled by rapid urbanization, population growth, and industrialization—has placed unprecedented pressure on transportation infrastructure. This surge has amplified the demand for intelligent traffic management systems (ITMS) capable of improving road safety, minimizing traffic congestion, enforcing regulatory compliance, and facilitating data-driven urban planning [1] [2]. Among the critical capabilities of these systems, vehicle speed estimation and automatic license plate recognition (ALPR) stand out due to their broad applicability. These functionalities support a diverse array of applications, including automated speed limit enforcement, electronic toll collection, restricted-area access control, surveillance of criminal activity, parking management, and traffic violation detection [3].

Historically, traffic monitoring systems have relied on hardware-intensive solutions, such as radar guns, inductive loop sensors, and specialized ANPR cameras, which, although effective, suffer from multiple limitations. These include high installation and maintenance costs, limited adaptability to diverse environments, and inability to scale without significant infrastructural investment. Moreover, traditional systems often underperform in dynamic real-world conditions, such as varying lighting, weather changes, heavy traffic density, and occlusion, all of which are commonplace in modern urban settings [4] [5].

In response, computer vision technologies, empowered by deep learning, have revolutionized traffic monitoring by enabling data-driven, sensor-independent, and camera-based solutions. Specifically, Convolutional Neural Networks (CNNs) have become central to visual perception tasks, offering state-of-the-art performance in object classification, detection, and recognition across various domains. Within this framework, the YOLO (You Only Look Once) series has emerged as one of the most dominant families of object detectors, widely recognized for its speed-accuracy trade-off, real-time processing capability, and compact architecture that supports edge deployment. These traits make YOLO particularly well-suited for real-time traffic monitoring applications such as vehicle and license plate detection, even in constrained environments [6] [7].

Building on these advancements, this work proposes a real-time, integrated system that combines three cutting-edge technologies: YOLOv11 for robust vehicle and license plate detection, SORT (Simple Online and Real-time Tracking) for identity-preserving multi-object tracking, and EasyOCR for high-accuracy license plate character recognition. YOLOv11 introduces transformer-based backbones, spatial attention mechanisms, and adaptive detection heads, which improve its ability to detect small and partially occluded objects like license plates under challenging conditions [8]. SORT offers lightweight yet effective tracking by leveraging Kalman filters and the Hungarian algorithm, ensuring reliable speed estimation through consistent object identity tracking [9] . Finally, EasyOCR, a deep learning-based OCR framework, enables end-to-end recognition of license plate text from varied fonts, angles, and lighting conditions, eliminating the need for traditional character segmentation or heuristic post-processing [10].

Unlike previous systems that focus on single-task objectives or depend on manual calibration and rigid sensor infrastructure, our approach provides a unified, fully-automated, and scalable pipeline for performing vehicle tracking, speed estimation, and license plate recognition from simple video feeds. This system is tailored for real-world deployments, including smart cities, urban intersections, and national highways, offering low-latency processing with modest hardware requirements, thereby democratizing access to intelligent traffic analytics. The primary contributions of this paper are as follows:

- A novel multi-vehicle speed estimation and license plate recognition combination system based on YOLOv11 and Optical Character Recognition (OCR).
- Introduction of a robust method for detecting vehicle license plates in challenging conditions such as varying lighting.
- A scalable solution for intelligent transportation systems providing reliable and efficient processing for real-time traffic management.

The structure of this paper is organized as follows: Section 2 introduces the related work. Section 3 introduces the proposed system, providing a detailed description of the YOLOv11-based vehicle license plate detection and speed estimation. Section 4 offers a comprehensive analysis of the evaluation metrics, followed by an in-depth discussion of the experimental results. Finally, Section 5 concludes the paper, summarizing key findings and suggesting directions for future work.

2 Related Work

2.1 Object detection

Real-time vehicle detection, tracking, and license plate recognition have become essential components of intelligent transportation systems and traffic management solutions. Recent advances in object detection based on deep learning have significantly improved the accuracy and efficiency of such applications. In the realm of object detection, the YOLO series (You Only Look Once) has set benchmarks for speed and precision [7]. Studies highlight its adaptability and superior performance, particularly in detecting small or complex objects like vehicles and license plates [11] [12].

2.2 Vehicle tracking and speed estimation

For tracking, SORT (Simple Online and Real-Time Tracking) offers a computationally efficient solution for associating detections across frames. Its seamless integration with YOLO-based systems has proven effective in handling multi-vehicle scenarios and maintaining tracking accuracy in high-speed applications that require real-time performance [13]. Extensions involving Kalman filters have further enhanced its performance, providing smooth trajectories critical for downstream tasks such as speed estimation [14]. Our decision to use SORT is based on its superior suitability for real-time applications compared to more complex tracking algorithms often employed in surveillance systems. While those models may offer marginal gains in accuracy, they typically involve higher computational costs and latency, which are unsuitable for scenarios requiring immediate response. SORT, by contrast, achieves a strong balance between speed and accuracy, ensuring efficient tracking without sacrificing performance in fast-paced traffic environments. Accurate vehicle speed estimation relies on a combination of robust detection, precise tracking, and effective measurement methodologies. Traditional techniques, such as optical flow, often falter under dynamic or adverse conditions. In contrast, frameworks integrating YOLO, SORT, and scene calibration techniques have demonstrated improved reliability, enabling precise pixel-to-meter conversions and trajectory analysis for real-world speed estimation [15].

2.3 License plate recognition

typically involves three main tasks: detecting the license plate within the image or video, segmenting its characters, and recognizing the segmented characters.

Systems have benefited from YOLO's advances in small-size object detection. The improved YOLOv11 backbone and neck architectures address long-standing challenges such as plate angle variations and poor lighting [8]. When paired with OCR methods [16], these systems achieve high recognition accuracy, further supporting applications in traffic monitoring, law enforcement, and intelligent transportation. By integrating YOLOv11 for detection, SORT for tracking, and advanced OCR for recognition, modern vehicle analysis frameworks deliver a robust pipeline for real-time processing. These advances bridge critical gaps in speed, accuracy, and adaptability, making them indispensable for addressing the complex demands of intelligent transportation systems.

3 Proposed Framework

The proposed model processes video input sequentially to enable real-time vehicle analysis as depicted in Fig.1. It performs the detection of vehicles and license plates in each frame, producing bounding boxes and confidence scores for vehicles and license plates using YOLOv11. A tracking algorithm (SORT) is then used to assign unique (Identification) IDs to the detected vehicles and track their trajectories over consecutive frames. Using these trajectories, the speed of vehicles is estimated, and the license plates of speeding vehicles are recognized using OCR. The results, including vehicle IDs, bounding boxes, license plate texts, and speeds, are saved in a structured CSV file.

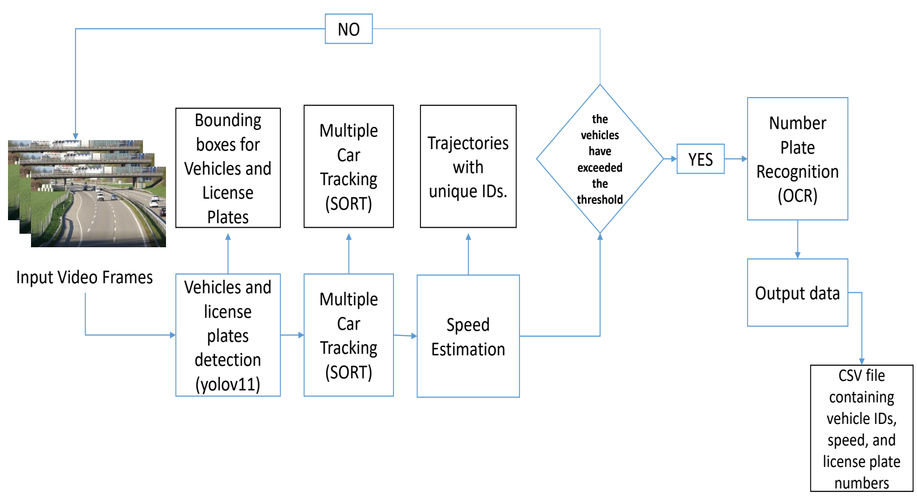


Fig. 1: Overview of the proposed method for vehicle speed estimation and license plate recognition.

3.1 Vehicle and License Plate Detection

CNN-based YOLOv11 is chosen for vehicle and license plate detection as it integrates structural innovations in object detection, ensuring both real-time efficiency and high accuracy. The model architecture consists of three main components:

- **Backbone Network:** Uses a transformer-based structure to capture long-range dependencies and improve small and occluded object detection, such as a license plate. The C3k2 module improve feature partitioning while reducing computational load.
- **Neck Network:** Employs Spatial Pyramid Pooling Fast (SPPF) module and the Cross-Stage Partial with Spatial Attention (C2PSA) module for multiscale feature aggregation and selective spatial attention, improving the accuracy of feature extraction .
- **Head Network:** Features a dynamically adaptive detection head for optimized resource allocation in complex scenes. It also incorporates an Non-Maximum Suppression (NMS) replacement algorithm and a dual-label assignment mechanism to enhance accuracy in dense target scenarios while minimizing inference latency for real-time application.

These innovations make YOLOv11 a highly efficient and precise object detection framework. For a more detailed understanding of the architecture, refer to [8].

3.2 Vehicle Tracking

After detection, we use Online and Real-Time Tracking (SORT) [9], an efficient and real-time algorithm well-suited for vehicle tracking in applications such as speed estimation and plate recognition. SORT leverages Kalman filtering to predict vehicle positions and velocities, and the Hungarian algorithm to associate new detections with existing tracks on an intersection over union (IoU) cost matrix, as shown in Equation 1.

$$IoU(i,j) = \frac{|B_i \cap B_j|}{|B_i \cup B_j|} \tag{1}$$

where B_i is the detected bounding box and B_j is the predicted bounding box. Each vehicle is assigned a unique ID, which is maintained throughout its trajectory, ensuring consistent identification across frames. This is essential for correlating vehicle detections with license plate recognition and accurately estimating vehicle speed. With its simplicity, low computational cost, and real-time performance, SORT perfectly meets the requirements of our system for vehicle detection, speed estimation, and license plate recognition.

3.3 Speed Estimation

After detecting and tracking a moving vehicle, the model needs to estimate its speed. To achieve this, a custom function named estimate speed has been developed, see Eq. (6).
The model relies on two fixed positions of the vehicle: Location 1= (x_1, y_1) and Location 2= (x_2, y_2), which represent the centers of the lower edge of the vehicle's boundary box on two consecutive frames. Using these two distinct positions, the function calculates the vehicle's speed by measuring the Euclidean distance between them, as demonstrated in Eq. (2):

$$d_{pixel} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

This distance, initially expressed in pixels, is converted into meters using a predefined constant, "pixels per meter" (ppm) formula as in Eq. (3).

$$d_{\text{meters}} = \frac{\text{dpixel}}{\text{ppm}} \tag{3}$$

The speed is then estimated on the basis of the frame rate and temporal conversion factors and it is calculated as distance divided by time. The time interval between two frames is constant and is encapsulated in the term $time_constant$, which combines the frame rate (15 FPS) and a conversion to kilometers per hour (kph). The formula is :

$$\text{Speed (kph)} = \frac{d_{\text{meters}}}{\text{time\_constant}} \tag{4}$$

where:

$$\text{time\_constant} = FPS \times kph \tag{5}$$

By substituting d_{meters} in the speed formula, we get:

$$\text{Speed (kph)} = \frac{\text{dpixel}}{\text{ppm} \times \text{time\_constant}} \tag{6}$$

License Plate Recognition After estimating vehicle speeds, the system identifies speeding vehicles and recognizes their license plates using EasyOCR [10], we chose EasyOCR for its user-friendly implementation and high accuracy. EasyOCR converts scanned documents into text using machine learning. Detects characters, refines boundaries, removes redundancies, and corrects distortions. Deep learning models (ResNet, LSTM) extract features and decoding methods, including greedy decoding and beam search, to identify the most likely characters. Post-processing ensures accuracy with confidence scoring and error correction, producing the final recognized text. License plates are detected with the help of a Haar Cascade XML file. The recognized license plate data, along with vehicle speeds and bounding box details, are recorded in a CSV file to support enforcement actions and improve traceability in Intelligent Transportation Systems.

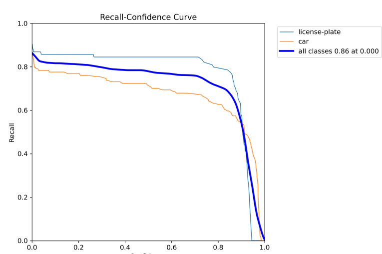
4 Result and discussion

4.1 Datasets

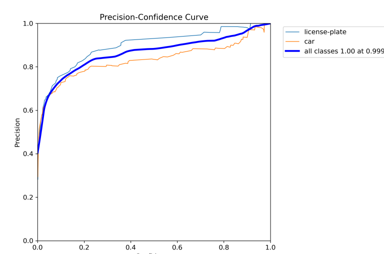
The YOLOv11 model was trained on a license plate and vehicle data set created using Roboflow [17], this data set consists of 1,084 annotated images for license plate detection in YOLOv11 format. The data set was divided into training sets (90%), validation sets (6%), and test sets (3%).

Before training, the images were pre-processed, including auto-orientation (EXIF stripping) and resizing to 416×416 pixels. In addition, data augmentation was applied, generating four versions of each source image with random exposure adjustments (-25% to +25%) and Gaussian blur (0–2 pixels). These enhancements aimed to simulate real-world conditions and improve the robustness of the model.

Since there is no publicly available dataset to simultaneously evaluate license plate detection, vehicle detection, and speed estimation, the final testing was carried out on video obtained from Pexels [18]. This video was used to evaluate the performance of the model in the real world in all three tasks.



(a) Recall-confidence curve for YOLOv11



(b) Precision-confidence curve for YOLOv11

Fig. 2: Confidence curves illustrating the recall and precision variation with confidence thresholds for the detected classes "car" and "license-plate.

4.2 Experimental Setup

The training was conducted on free-tier Google Colab GPU, specifically an NVIDIA T4 with 16GB of VRAM. The system also included an Intel Xeon CPU (2 vCPUs) and 12GB of RAM.

4.3 Performance Metrics

In this subsection, we evaluate the performance of our model in detecting license plates and cars using various metrics. Specifically, we consider Precision, Recall, and mAP-50, which provide insights into the accuracy and effectiveness of our predictions. The definitions of these metrics are presented below:

Precision measures a model's capability to detect only relevant objects, representing the proportion of correctly identified positive instances [19], see Equation 7.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

Recall reflects a model's ability to detect all relevant cases, including all ground-truth bounding boxes. It represents the percentage of correctly identified positives out of all actual ground truths [19], see Equation 8.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

Where TP are True Positives, FP are False Positives, FN are False Negatives, and TN are True Negatives.

F1 Score harmonic mean of precision and recall, measuring a model's balance between false positives and false negatives. It is defined as Equation.9:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

mAP-50 Mean Average Precision (mAP) quantifies the accuracy of object detections across all classes, using an IoU threshold of 0.5. [19], see Equation 10.

$$\text{mAP-50} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{10}$$

AP_i represent the AP for the ith class, while N denotes the total number of evaluated classes.

4.4 Experimental results

The recall confidence curve (Figure 2a) shows that recall decreases as confidence increases, with the "license-plate" class maintaining higher recall than "car," and overall recall peaking at 0.91. In contrast, the precision-confidence curve (Figure 2b) indicates that precision improves with confidence, reaching 1.00 at 0.985, with the "license plate" class again outperforming "car." YOLOv11 balances precision and recall well, though the "car" class shows lower performance in both metrics. Lower confidence thresholds improve recall but introduce false positives, whereas higher thresholds enhance precision at the cost of recall, highlighting the trade-off in selecting an optimal confidence level.

Table 1 presents a comprehensive comparison of YOLOv8, YOLOv9, and YOLOv11 across four key object detection metrics: Precision, Recall, mAP@0.5

(mean Average Precision at IoU 0.5), and F1-score. Among these models, YOLOv11 demonstrates superior performance in all evaluated metrics, establishing itself as the most robust and accurate choice for real-world detection tasks.
YOLOv11 achieves the highest precision (0.94476), significantly reducing false positives, and the highest recall (0.75856), indicating enhanced detection capability for relevant objects. This balance results in the best F1-score (0.8416), reflecting optimal harmony between precision and recall. Additionally, YOLOv11 leads in mAP@0.5 (0.85602), outperforming both YOLOv8 (0.85355) and YOLOv9 (0.83437), further validating its accuracy in bounding box predictions. While YOLOv9 maintains strong precision (0.93428), its lower recall (0.72637) limits its overall effectiveness compared to YOLOv11.
The high recall of YOLOv11 suggests particularly strong performance in detecting the 'license-plate' class, improving system reliability for vehicle identification. Meanwhile, its exceptional precision enhances false positive filtering, ensuring accurate vehicle detection and, consequently, more reliable speed estimation. These results solidify YOLOv11 as the optimal backbone model for the proposed pipeline, excelling in both detection accuracy and robustness.

Table 1: Performance comparison of YOLOv8, YOLOv9, and YOLOv11 across key detection metrics.

Version	**Precision**	**Recall**	**mAP-50**	**F1-score**
YOLOv8	0.91719	0.74209	0.85355	0.8204
YOLOv9	0.93428	0.72637	0.83437	0.8172
YOLOv11	**0.94476**	**0.75856**	**0.85602**	**0.8416**

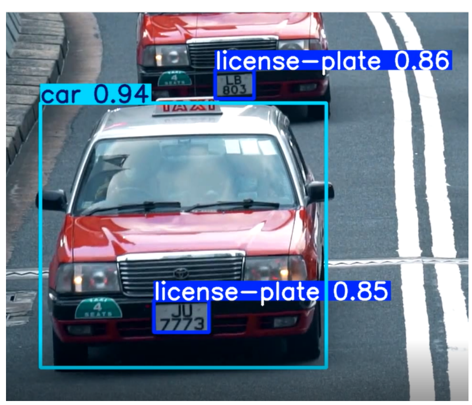


Fig. 3: Visualization of YOLOv11-based vehicle and license plate detection with SORT tracking.

Figure 3 illustrates the effectiveness of our proposed model, which combines YOLOv11 for object detection and SORT for real-time tracking. The system accurately identifies and tracks multiple vehicles and their corresponding license plates across consecutive frames. This demonstrates its capability for robust multi-object tracking in complex traffic environments. In Figure 4, the model's performance is further emphasized through its ability to detect and highlight speeding vehicles, estimate their speed with high precision, and recognize license plate numbers using EasyOCR. Notably, the model maintains strong detection and recognition accuracy even under challenging conditions such as occlusion, perspective distortion, and variations in plate orientation. These results underscore the reliability and adaptability of the system for real-world traffic surveillance applications, where dynamic scenes and environmental variability are common

Fig. 4: Speed estimation and license plate recognition using YOLOv11, SORT, and EasyOCR.

5 Conclusion

The rapid increase in the number of vehicles on the road presents major challenges for traffic management and enforcement, highlighting the need for efficient monitoring solutions. This paper addresses these challenges by introducing a system capable of detecting speeding vehicles and recognizing license plates that support real-time traffic regulation.

The proposed model leverages state-of-the-art deep learning techniques, combining YOLOv11 for vehicle detection, SORT for object tracking, and EasyOCR for license plate recognition. The model enables transportation authorities to identify and respond to traffic violations effectively.

Our results demonstrate strong performance of the model in multi-vehicle speed estimation and license plate recognition. As a next step, this work opens pathways for several impactful future developments. First, we plan to construct and release a publicly available dataset of Algerian vehicle license plates to support regional benchmarking and accelerate research in LPR under Arabic and Francophone conditions. This addresses a critical gap, as most existing datasets are limited to Western license plate formats. Second, we aim to collaborate with national authorities to integrate our system into the Algerian traffic management infrastructure, enabling large-scale deployment for speed enforcement, traffic analysis, and public safety monitoring.

References

1. Dhar, Prashengit, et al. "A system design for license plate recognition by using edge detection and convolution neural network." 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2). IEEE, 2018.
2. Quraishi, Abdul Awal, Farid Feyzi, and Asadollah Shahbahrami. "Detection and recognition of vehicle licence plates using deep learning in challenging conditions: a systematic review." International Journal of Intelligent Systems Technologies and Applications 22.2 (2024): 105-150.
3. Khan, Muhammad Gufran, et al. "A novel deep learning based anpr pipeline for vehicle access control." IEEE Access 10 (2022): 64172-64184.
4. Luvizon, Diogo C., Bogdan T. Nassu, and Rodrigo Minetto. "Vehicle speed estimation by license plate detection and tracking." 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014.
5. Buch, Norbert, Sergio A. Velastin, and James Orwell. "A review of computer vision techniques for the analysis of urban traffic." *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, 2011, pp. 920–939.
6. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934*, 2020.
7. Redmon, J. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
8. Khanam, Rahima, and Muhammad Hussain. "YOLOv11: An overview of the key architectural enhancements." arXiv preprint arXiv:2410.17725 (2024).
9. Bewley, Alex, et al. "Simple online and realtime tracking." 2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016, pp. 3464–3468.
10. JaidedAI. *EasyOCR* [Software]. GitHub, 2023. Available: `https://github.com/JaidedAI/EasyOCR`.
11. Wang, Ao, et al. "Yolov10: Real-time end-to-end object detection." arXiv preprint arXiv:2405.14458 (2024).
12. Amin, Arslan, et al. Next-Generation License Plate Detection and Recognition System Using YOLOv8. 2023 IEEE 20th International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET). IEEE, 2023.
13. Chopade, Rohan, et al. "Automatic Number Plate Recognition: A Deep Dive into YOLOv8 and ResNet-50 Integration." 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS). IEEE, 2024.

14. Rais, Asif Hummam, and Rinaldi Munir. "Vehicle speed estimation using YOLO, Kalman filter, and frame sampling." 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA). IEEE, 2021.
15. Rodríguez-Rangel, Héctor, et al. Analysis of statistical and artificial intelligence algorithms for real-time speed estimation based on vehicle detection with YOLO. Applied Sciences 12.6 (2022): 2907.
16. Baviskar, Dipali, Shweta Choudhary, Riya Danve, Ketki Kinkar, and Shivraj Patil. "Auto Number Plate Recognition." 2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST). IEEE, 2022, pp. 1–6.
17. Roboflow.https://universe.roboflow.com/samrat-sahoo/license-plates-f8vsn/dataset/5
18. Pexels.https://www.pexels.com/fr-fr/
19. Padilla, Rafael, Sergio L. Netto, and Eduardo A. B. Da Silva. "A survey on performance metrics for object-detection algorithms." 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). IEEE, 2020, pp. 237–242.