ISSN 2007-9737

Vehicle Counting System Using Image Processing with Pre-trained Machine Learning Models

Antonio David Ruiz-Diaz-Medina, Nazario Luis Ayala-Frasnelli*,
Rodrigo Javier Martínez-Meza, Angel Gustavo Heimann-Fernández

Universidad Nacional de Canindeyú,
Facultad de Ciencias y Tecnología,
Paraguay

{davidruizdiaz, nazarioayala, rodrigo.martinez, angelheimann}@facitec.edu.py

Abstract. Understanding tourist flow dynamics is crucial for decision-making authorities in strategic tourism policy planning, infrastructure development, and natural resource management. This data enables informed decision-making to develop a sustainable tourism industry, promoting economic benefits while minimizing negative impacts on destinations. In this context, the city of Saltos del Guairá, Paraguay, whose economy is linked to the flow of tourists from Brazil, faces similar challenges due to the lack of a data collection mechanism to analyze incoming tourist flow. To address this challenge, the work proposes the use of a vehicle counting system based on image processing with pre-trained machine learning models for analyzing tourism flows in Saltos del Guairá. The methodology to achieve the expected results was divided into four stages. The first phase involved the evaluation of two machine learning models to select the one with the best performance in vehicle detection and tracking according to the requirements of this work. The second stage was the analysis and design of the system, with the aim of obtaining an architecture that allows for the interoperability and scalability of the system. In the third stage, the designed system was developed, using the selected machine learning model and the designs obtained in the previous phases. Finally, integration tests and the commissioning of the entire system were carried out. The proposed solution is valid for the problem, as it allows monitoring the flow of vehicles entering the city.

Keywords. Vehicle counting, tourism, object tracking, SORT, YoloV7, Jetson Nano.

1 Introduction

Understanding tourist flow dynamics is crucial for decision-making authorities in strategic tourism policy planning, infrastructure development, and natural resource management. This data enables informed decision-making to develop a sustainable tourism industry, promoting economic benefits while minimizing negative impacts on destinations.

However, the primary challenge lies in collecting data that helps analyze and understand the behavior and flow of tourists in the area [22].

Similarly, given the importance of the aforementioned points, the city of Saltos del Guairá, Paraguay, whose economy partially depends on the flow of tourists from Brazil [2], faces the same difficulties due to the lack of a mechanism to obtain data for analyzing the tourist flow entering the city.

Additionally, there is a challenge in addressing the need for a local company, "MP Sales S.A.", to maintain a historical record of the number of vehicles entering its premises. This is to study the flow of its customers and implement better marketing strategies to attract more customers during periods of low tourist activity.

There are several approaches to addressing the described issue, as observed in Benkhard [3, 12]. One of these approaches is monitoring vehicular traffic in cities.

A review of the methods utilized for vehicle traffic monitoring is given in this respect by the research done by Jain et al. [11], with image processing

Table 1. Machine learning model selection criteria

ID	Criteria	Variable
C1	Free license	License type
C2	Must be optimized for the Jetson Nano™ without any further adjustments	Ease of implementation
C3	Must allow smooth video processing, higher frames per second (FPS)	Inference speed
C4	Must detect and identify as many vehicles as possible	Accuracy and precision

using computer vision techniques standing out as a particularly useful approach.

Thórhallsdóttir et al. in [21] have proposed a method for analyzing the number and distribution of tourists in Iceland. Their data collection strategy relies primarily on magnetic radar-based vehicle counters. However, this method may present some additional challenges, such as the limited range of the radars for vehicle detection and the need for periodic radar calibration to ensure their correct operation.

At the national level, in Paraguay, efforts have been made for the implementation of traffic monitoring systems, although these have been mainly focused on road safety aspects [10].

Therefore, considering all of the above, the main objective of this work is the design and implementation of a vehicle counting system using image processing by means of pre-trained machine learning (ML) models for the analysis of tourist flow in the city of Saltos del Guairá.

The proposal seeks to provide an affordable solution that can be adapted to the needs of the city and its road infrastructure.

2 Methodology

This work focuses on the design and implementation of a specific technological system, in this case, a vehicle counting system. In addition, existing knowledge and methods are applied to solve this problem. Therefore, this work can be classified as an applied technology research. The processes followed are detailed as follows:

2.1 ML model comparison

The purpose of this phase is to identify the most appropriate pre-trained ML model for vehicle detection in the context of the developed project. The selected model will be implemented on a Jetson Nano™ [15]. This device is ideal for Edge Computing and Internet of Things (IoT) projects. It offers a Linux-based development environment, ideal for implementing AI solutions on embedded devices with limited resources.

Two pre-trained object detection models were compared: TrafficCamNet [16], based on DetectNet_v2 [19], and YOLOv7-tiny, a compact and efficient version of YOLOv7 [23, 24]. Object tracking algorithms were integrated into both models.

Initially, trials were conducted with the TrafficCamNet model, implemented through the use of the Jetson Inference library, which makes it easy to build and train AI models in Jetson based environments [8]. This library incorporates, by default, an object tracking algorithm using the IoU (Intersection over Union) metric [6]. But tests have also been carried out by integrating the SORT tracking algorithm (Simple Online and Realtime Tracking) [5].

Similar tests was conducted using an optimized model based on YOLOv7-tiny but modified for execution using TensorRT [14]. This model was also integrated with the SORT tracking algorithm.

In order to determine the model to be used, the selection criteria to be considered were set out in the Table 1. In order to determine the performance of the models and meet the criteria C3 and C4, tests will be carried out using two separate pre-recorded videos.

Table 2. Technical specifications of the prototype components

Components	Technical information
Jetson Nano™	**GPU:** NVIDIA Maxwell™ architecture with 128 CUDA® cores **CPU:** Quad-core ARM® Cortex®-A57 MPCore processor **Memoria RAM:** 4 GB 64-bit LPDDR4 1600MHz - 25.6 GB/s **Performance:** 472 GFLOPs for running modern AI algorithms
High Quality Arducam Camera	**Resolution:** 12.3 megapixels **Optical Size:** 1/2,3” **Focal length:** 6mm
Jetson Nano™ cooling fan	**Working voltage:** 5V **Connector:** 4 pins **Speed control:** PWM
Power supply	**Input voltage:** CA 100 120/200 240V 50 / 60HZ **Output power:** DC 5V 5A With high efficiency DC wave

2.2 Analysis and design

The system is composed of two main modules: the vehicle counter, developed and deployed at the Jetson Nano™, and a web application for the visualization and disclosure of data collected by the counter.

The functional requirements for the counter were defined based on the detection data that can be obtained from the selected ML model. The timestamp and location data were also recorded. Moreover, the requirements for the web application were established taking into account the data submitted by the counter, as well as the data transformations necessary for the graphical visualization of the number of vehicles counted.

The following artifacts were used to design the system:

— Data dictionary, which establishes the data structure and data types managed in the system.

— Component diagram, which describes the components and modules of each application.

— General architecture diagram, providing a general overview of the entire system.

2.3 Prototype building

The development process of the system was carried out using a methodology based on functionality-driven Development, as proposed in the work of Ruiz Diaz et al. [18]. This software development approach focuses on iterative delivery of functional features and includes several subprocesses, such as:

— Creation of repositories for the project.

— Definition of tasks for the implementation of functionalities.

— General tracking of the project through the review and integration of functionalities.

For the development of the vehicle counter, it was implemented within the Jetson Nano™ operating system itself, using the Python 3 programming language. As for the web application, the Laravel framework version 10 with PHP version 8 was used. In both cases, the data was stored using SQLite 3.

In parallel, the design and manufacture of the housing for the electronic components for the vehicle counter were carried out. The components are composed of a Jetson Nano™ minicomputer, a compatible camera, and others. Table 2 details the technical specifications of these components.

The manufacturing process of the casing involved modeling a 3D design of a structure to contain the equipment presented in Table 2. For this purpose, the open-source 3D modeling software FreeCAD version 0.21 was used [20]. The dimensions and technical specifications of the electronic components were considered in this process. The designed 3d model is available at `https://github.com/FACITEC-INV/prototipoCarcaza`.

Once the 3D model of the casing for the camera was obtained, it was printed on a 3D printer. The material used for printing was acrylonitrile butadiene styrene (ABS) filament, whose physical and thermal characteristics are suitable for this purpose [1].

2.4 Testing and deployment

During this stage, tests were conducted on both the vehicle counter and the web application. The constructed vehicle counting prototype will be installed at the main entrance of ”MP Sales S.A.”. At the selected location, it is possible to monitor the main access route for foreign tourists to the city of Saltos del Guairá.

The tests performed on the counter consisted of verifying the correct operation of the counting system, the integrity and persistence of the detection data, and the correct sending of the data to the web application server.

To carry out the tests on the counter, the Jetson Nano™ device was accessed remotely using VNC connections. It should be noted that for headless connections (without a physical screen), the use of an HDMI connection emulation device is necessary if you want to obtain an adequate screen resolution that can be adjusted.

For their part, the tests carried out on the web application included functionality tests of the exposed APIs and verification of the correct display of the data sent by the counter within the development environment.

Finally, the system deployment consisted of installing the vehicle counter and bringing the web application online.

3 Results

3.1 Evaluation of ML Models

As can be verified in the available documentation, both models meet the criterion of having free licenses. On the one hand, both the jetson-inference library project and the TrafficCamNet model are licensed under the MIT license type [8, 15]. This type of license allows free use of the software, even allowing modification and redistribution [17]. On the other hand, the YoloV7-tiny model is licensed under the GPL-3.0 license [24]. This type of open source licensing guarantees users the freedom to use, modify and distribute the software [9].

As for the optimization of the models for their use in the development environment used in this project, it can also be mentioned that both TrafficCamNet, implemented with Jetson Inference, and YoloV7-tiny, implemented with TensorRT, meet the criterion.

The jetson-inference library allows implementing various ML models in a simple and optimized way on Jenson devices, including the Jetson Nano™, as specified in the documentation [8]. As can be seen in Fig. 1, in the tests conducted with this library, it was possible to correctly implement object detection with the TrafficCamNet model.

As mentioned by Nguyen et al. [13] with the use of TensorRT, it is possible to significantly improve the inference speeds of models such as YoloV7-tiny on equipment with limited computational resources, such as that used in this work. In Fig. 2, the detection tests performed with YoloV7-tiny for object detection on the Jetson Nano™ used in this project can be observed.

In the tests carried out, it was observed that both models achieved promising results in terms of inference speed, taking into account the resource restrictions of the Jetson Nano™. The TrafficCamNet model achieved an image processing rate of 18 FPS, while the YOLOv7-tiny model, implemented with TensorRT, reached a speed of 15 FPS. It should be noted that these values were obtained with the already integrated tracking algorithms. These results indicate that

Table 3. Counting test results: percentage of vehicles counted

Models	Video 1	Video 2
TrafficCamNet with jetson-inference tracking algorithm	57%	10%
TrafficCamNet with SORT tracking algorithm	57%	90%
YoloV7-tiny with SORT tracking algorithm	100%	90%

Fig. 1. Object detection and tracking test with TrafficCamNet

both models are capable of processing videos in real time with an acceptable level of efficiency.

As can be seen in Table 3, in the results obtained for vehicle counting, it can be observed that the YoloV7-tiny model integrated with the SORT detection algorithm obtained the best performance. In terms of detection algorithms, even in the tests performed with the TrafficCamNet model, the SORT algorithm obtained better results.

In the tests carried out with the default tracking algorithm of jetson-inference, the vehicle count was very irregular. Problems with the correct tracking of objects can be observed. As can be seen in Table 3, in the test with video 1, 57% of the vehicles were counted, but with video 2, only 10% could be counted.

It is important to note that, as mentioned in [4], a good part of the accuracy in object tracking depends on the performance of object detection models. This may indicate that the YoloV7-tiny model implemented with TensorRT is the best option for object detection in the context of this work, however, more exhaustive studies on the subject are necessary.

Table 4. Data dictionary for vehicle counting application

	Column	Data Type	Description
PK	id	int	Identification number
	id_zona	varchar(10)	Identification of the detection location
	clase	varchar(10)	Type of object detected [car|truck]
	fecha	timestamp	Timestamp of the detection

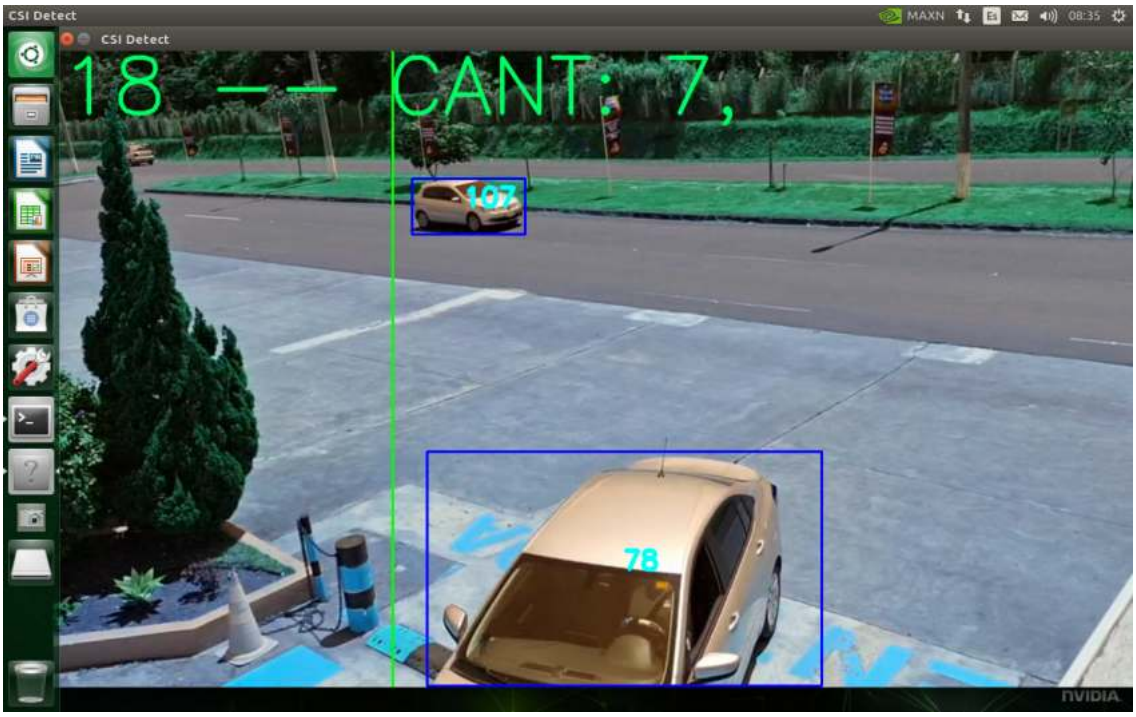

Fig. 2. Object detection and tracking test with YoloV7-tiny

Based on the results obtained, it was decided to use the YoloV7-tiny model, implemented with TensorRT, for the development of the vehicle counting project proposed in this work.

3.2 System Design

Once the machine learning model to be used for counting vehicles detected by the system was defined, the system was designed. First, the data that needed to be stored to record the vehicle count was identified. This task was carried out using a data dictionary, as indicated in the previous section and illustrated in the Table 3.

This data dictionary makes it possible to understand what the main data structure will be that will be shared between the vehicle counter and the web application.

The representation of the system modules was essential to understand each part of the system and its interrelationships. The result of this analysis for the web application can be observed in Fig. 3.

Basically, the web application implements a REST API, which is the input interface for data sent by the counter and for requests from internal modules, as well as for external systems. This

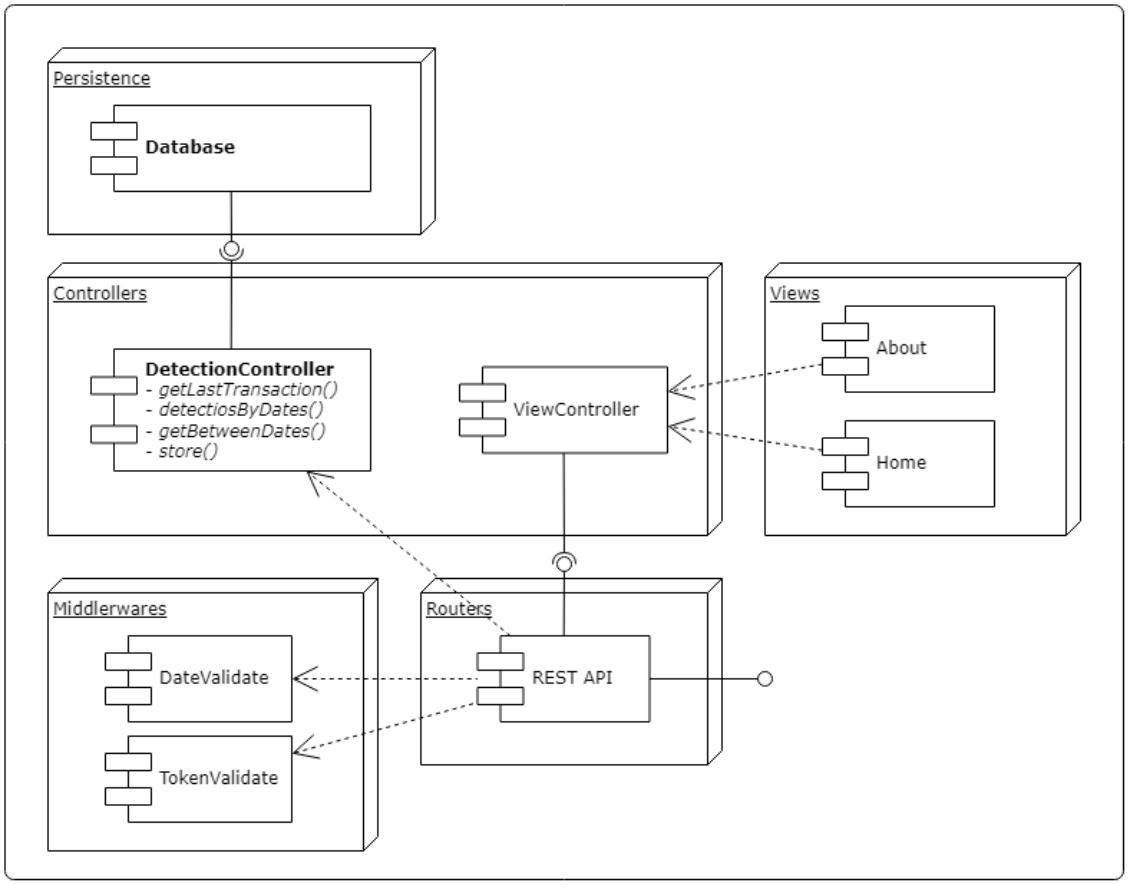

Fig. 3. Web application component diagram

approach contributes to the interoperability and integral functionality of the system as a whole [7].

Fig. 4 presents the outcome of the vehicle counter design. The diagram shows the interaction of the components of the vehicle counting system, which will run on the Jetson Nano™ device. The system is divided into three main components: vehicle detection and tracking, vehicle counting, and sending data to the server. Data is sent to the server periodically (e.g., every 30 minutes), for which the local database records are consulted and the results are sent to the server through REST API requests.

An overview of the system design is provided in Fig. 5, which illustrates the entire system's architecture, its physical and logical components, and the flow of the data through those components.

The illustration distinguishes the different parts that make up the entire system. On the left side are the vehicle counter components, which include the camera, the Jetson Nano™ minicomputer, the application that uses the selected ML model for detection, tracking, and counting, as well as data transmission and its respective database. Similarly, on the right side of the illustration are the components corresponding to the web application.

3.3 Prototype building

Regarding the prototype development, two results are presented. On the one hand, the results of the software development process and, on the other hand, the design and printing of the prototype of

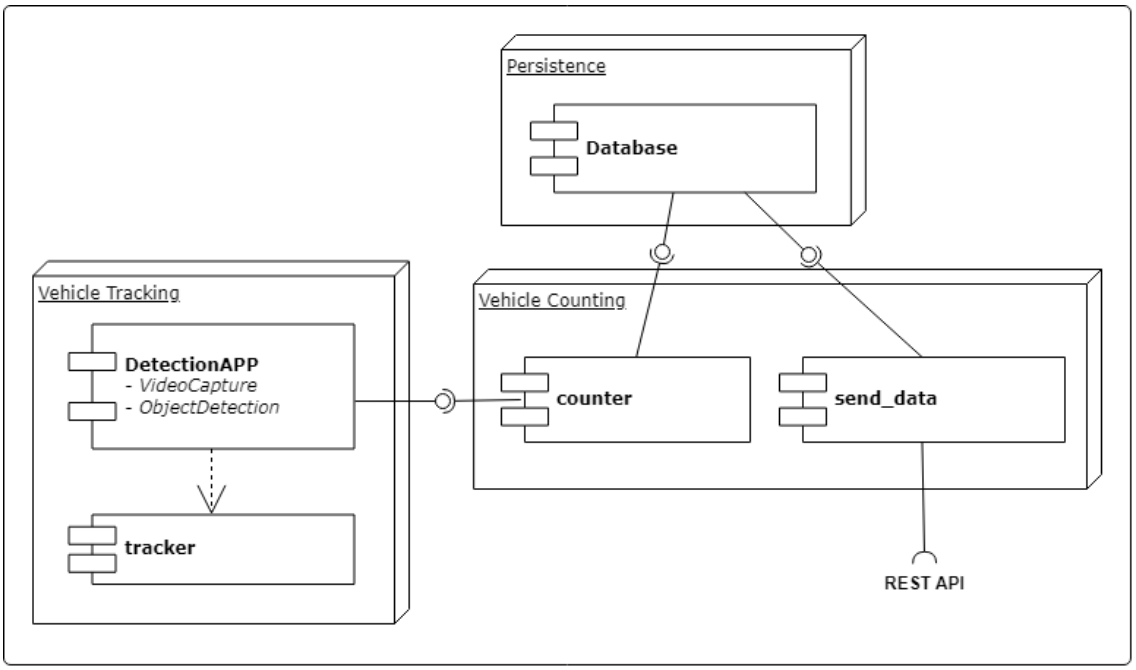

Fig. 4. Vehicle counter component diagram

the housing for the electronic components for the vehicle counter.

The result of the development process, already described above, includes two main components: the software for the vehicle counter and the web application. The repositories with the source code can be found at the following links: for the vehicle counter at `https://github.com/FACITEC-INV/jetson-yolov7`, and for the web application at `https://github.com/FACITEC-INV/vcounter_vista`.

Figure 6 presents the prototype's 3D design. One can see that it has several components, including the camera housing. The camera housing measures 280 millimeters in length, 110 millimeters in height, and 110 millimeters in width.

Once assembled, the design proved to be adequate to contain all the electronic components necessary for the operation of the vehicle counter system.

3.4 Testing and deployment

With the printed housing structure, the prototype was assembled and installed. Fig. 7, shows an image of the vehicle counting equipment installed at the entrance of "MP Sales S.A.".

During the tests, the correct operation of the counting of vehicles circulating on the road was verified. In the same way, the collected data is sent periodically to the server. This data collected on the server is publicly available for access through a web application.

Finally, Fig. 8 shows the interface of the web application for querying the collected counting data. As can be seen in the illustration, this data can be viewed through a web application that is available for consultation, both in total value format and in a graph that allows visualizing the variations in vehicle flow over time. The application is available at the link `https://appmapy.facitec.edu.py/`.

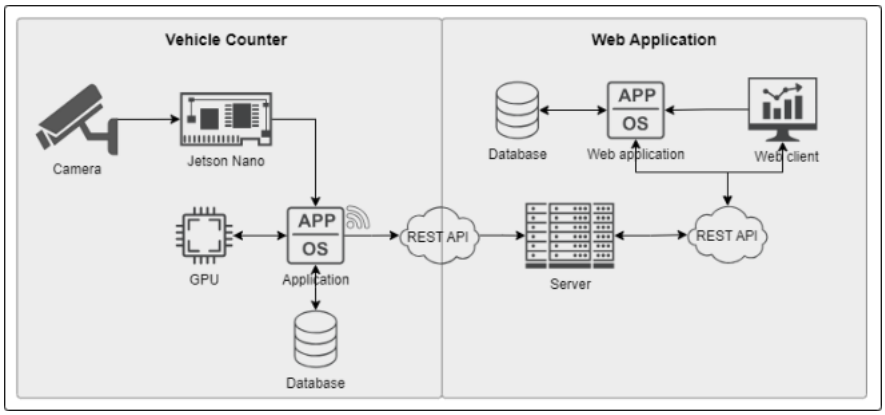

Fig. 5. Overall architecture of the vehicle counting system

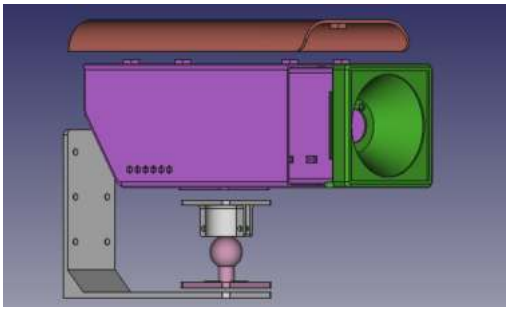

Fig. 6. Housing prototype design

Fig. 7. Field tests: vehicle counting prototype installed

4 Conclusions

In this work, a vehicle counting system based on machine learning models has been developed to analyze tourist flow in the city of Saltos del Guairá. Therefore, the system has the potential to address the needs of the local company ”MP Sales S.A.”. The system consists of two main components: a vehicle counter implemented on a Jetson Nano™, and a web application for visualizing and analyzing the collected data.

Based on the results obtained in the vehicle counting tests, the YoloV7-tiny model with SORT tracking algorithm was selected for the implementation of the counting system. This technology was the one that showed the best results among the evaluated models. It should be noted that more tests are required, since the TrafficCamNet model also showed good results when used with the SORT algorithm.

The choice of the ML model depends on the needs of the project. In other contexts with different needs, other, more suitable models can be considered. One way to increase the precision of models is by either retraining on them or developing specialized ones tailored for Edge Computing settings. Such spikes can include TensorRT among many others.

The model to be selected depends on the needs of the project and, therefore, in other case studies, other more suitable models can be considered. In

Fig. 8. Vehicle counting data query web application

order to improve the accuracy of the models, it is possible to retrain the models or train their own models, considering optimization for this type of Edge Computing environment, using tools such as TensorRT for this purpose.

The analysis and design of the system components, as well as the housing for the electronic components of the vehicle counter, enabled the development of two software applications that work together through an API. This API facilitates the interoperability and scalability of the system, allowing for the incorporation of new counting points for the collection of spatially distributed data.

It can be considered that the developed system offers a viable and affordable solution to the city of Saltos del Guairá, facilitating the monitoring of the flow of tourists entering the city. In addition, it serves as a basis for future improvements and adaptations of the system in other localities.

It is relevant to mention that the system developed could also serve the needs of the company “MP Sales S.A.”. This is because it also allows for the detection of vehicles entering the premises.

References

1. **Alhallak, L. M., Tirkes, S., Tayfun, U. (2020).** Mechanical, Thermal, Melt-Flow and Morphological Characterizations of Bentonite-Filled ABS Copolymer. Rapid Prototyping Journal, Vol. 26, No. 7, pp. 1305–1312. DOI: 10.1108/rpj-12-2019-0321.
2. **Azevedo da Rocha, A. P., Ferrari, M. (2020).** Fronteira e Interações Transfronteiriças Entre Brasil e Paraguai no Segmento de Guaíra e Salto del Guairá. Geografia em Questão, Vol. 13, No. 2. DOI: 10.48075/geoq.v13i2.24740.
3. **Benkhard, B. (2018).** Determination of Tourist Flow Patterns in a Low Mountain Study Area. Tourism & Management Studies, Vol. 14, No. 3, pp. 19–31. DOI: 10.18089/tms.2018.14302.
4. **Bewley, A. (2023).** SORT. GitHub. URL: https://github.com/abewley/sort.
5. **Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B. (2016).** Simple Online and Realtime Tracking. IEEE International Conference on

Image Processing (ICIP), pp. 3464–3468. DOI: 10.1109/icip.2016.7533003.

6. **Bochinski, E., Eiselein, V., Sikora, T. (2017).** High-Speed Tracking-by-Detection Without Using Image Information. 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. DOI: 10.1109/avss.2017.8078516.
7. **Flaishans, J., Galvin, M., Ignatius, A., Kuan, C., Laniak, G., Wolfe, K., Purucker, T. (2016).** Environmental Models as a Service: Enabling Interoperability through RESTful Endpoints and API Documentation. URL: https://scholarsarchive.byu.edu/iemssconference/2016/Stream-A/18.
8. **Franklin, D. (2024).** Jetson-Inference: Deploying Deep Learning. GitHub. URL: https://github.com/dusty-nv/jetson-inference.
9. **Free Software Foundation (2007).** The GNU General Public License V3.0. URL: https://www.gnu.org/licenses/gpl-3.0.html.
10. **Fretes, A. (2018).** Sistema Avanzado de Control de Tráfico de la ciudad avanza en la instalación de nuevas cámaras de monitoreo. Municipalidad de Asunción. URL: https://www.asuncion.gov.py/transito/sistema-avanzado-control-trafico-la-ciudad-avanza-la-instalacion-nuevas-camaras-monitoreo.
11. **Jain, N. K., Saini, R. K., Mittal, P. (2018).** A Review on Traffic Monitoring System Techniques. In Soft Computing: Theories and Applications. Springer Singapore, pp. 569–577. DOI: 10.1007/978-981-13-0589-4_53.
12. **Mohamad-Toha, M. A., Ismail, H. N. (2015).** A Heritage Tourism and Tourist Flow Pattern: A Perspective on Traditional versus Modern Technologies in Tracking the Tourists. International Journal of Built Environment and Sustainability, Vol. 2, No. 2, pp. 1–8. DOI: 10.11113/ijbes.v2.n2.61.
13. **Nguyen, H.-V., Bae, J.-H., Lee, Y.-E., Lee, H.-S., Kwon, K.-R. (2022).** Comparison of Pre-Trained YOLO Models on Steel Surface Defects Detector Based on Transfer Learning with GPU-Based Embedded Devices. Sensors, Vol. 22, No. 24, pp. 1–13. DOI: 10.3390/s22249926.
14. **NVIDIA (2022).** TensorRT. NVIDIA Developer, URL: https://developer.nvidia.com/tensorrt.
15. **NVIDIA (2024).** Kit para desarrollador jetson nano. URL: https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/product-development/.
16. **NVIDIA (2024).** TrafficCamNet. NVIDIA NGC Catalog. URL: https://catalog.ngc.nvidia.com/models.
17. **Open Source Initiative (2023).** The MIT license. URL: https://opensource.org/license/mit.
18. **Ruiz-Diaz-Medina, A. D., Ayala-Frasnelli, N. L., Heimann-Fernández, A. G., Cáceres-Urdapilleta, F. A., Golin-Galeano, C. C. A., López-Villalba, A. (2021).** Flujo de trabajo para pequeños equipos de desarrollo utilizando FDD y GitHub. Tecnología, Diseño e Innovación, Vol. 7, No. 1, pp. 31–48. URL: https://www.unae.edu.py/ojs/index.php/facat/article/view/334.
19. **Tao, A., Barker, J., Sarathy, S. (2022).** DetectNet: Deep Neural Network for Object Detection in DIGITS. NVIDIA Technical Blog. URL: https://developer.nvidia.com/blog/detectnet-deep-neural-network-object-detection-digits/.
20. **The FreeCAD Project Community (2024).** Freecad (version 0.21). URL: https://www.freecadweb.org/.
21. **Thórhallsdóttir, G., Ólafsson, R., Jóhannesson, G. T. (2021).** A Methodology of Estimating Visitor Numbers at an Icelandic Destination Using a Vehicle Counter and a Radar. Journal of Outdoor Recreation and Tourism, Vol. 35, pp. 1–9. DOI: 10.1016/j.jort.2021.100378.
22. **Vu, H. Q., Li, G., Law, R., Ye, B. H. (2015).** Exploring the Travel Behaviors of Inbound Tourists to Hong Kong Using Geo-tagged Photos. Tourism Management, Vol. 46,

pp. 222–232. DOI: 10.1016/j.tourman.2014.07.003.

23. **Wang, C., Bochkovskiy, A., Liao, H. (2022).** YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Teal-Time Object Detectors. URL: http://arxiv.org/abs/2207.02696.

24. **Wong, K.-Y. (2023).** YOLOv7: Implementation of Paper. GitHub. URL: https://github.com/WongKinYiu/yolov7.

Article received on 20/06/2024; accepted on 13/02/2025.
**Corresponding author is Nazario Luis Ayala Frasnelli.*