CAP4CITY
Co-funded by the Erasmus+ Programme of the European Union

A DEEP LEARNING TOOL TO CLASSIFY VEHICLES IN REAL TIME

Mauricio Solar, UTFSM
Humberto Farias, UTFSM
Daniel Ortiz, UTFSM

Abstract

The problem of vehicular congestion is transversal to all the cities worldwide. To deal with it, it can be done from the construction of new roads or the management of these as in the case of exclusive roads for public transport. In the latter case, information is necessary. Currently, in Chile this collection process is manual, so it does not respond to current information needs. The use of deep learning models for automatic counting has shown outstanding results, but based on a scheme for sending video streaming to remote servers. The present proposal aims to not depend on fiber optic connectivity. It is based on a network of modules based on Edge Artificial Intelligence (Edge AI) in order to build a scalable, real-time data capture platform.

Keywords

Deep learning, Smart City, Smart Mobility

1. Introduction

The Serena-Coquimbo conurbation has been systematically identified as one of the places with the best quality of life in Chile. This denomination is affected by the explosive increase in the vehicle fleet and its effect on vehicle congestion. According to figures from the Instituto Nacional de Estadísticas (INE), there were already 100,531 vehicles (motorized and non-motorized) in the conurbation by 2012. In 5 years, said fleet increased by 22%, reaching a total of 128,931 vehicles in 2017. Another aspect that increases the complexity of the problem is the increase since 2018 by 30% in the sale of new vehicles in the region. According to the figures of the Chilean National Automotive Association (ANAC), in 2019 the commercialization of new vehicles raising the annual record 12,083.

This significant and steady increase has meant a superlative increase in traffic congestion. A traditional approach to trying to solve this problem is public works, that is, the construction of new streets. According to figures from the Report on Critical Infrastructure for Development 2018-2027 of the Cámara Chilena de la Construcción (CChC) in the conurbation there is an urban road deficit that exceeds US$ 3,500 million. Although new roads have been inaugurated, the problem of congestion still persists. This shows that another type of infrastructure is also required, in this case digital. It is necessary to have quantitative data that allow, for example, to plan axes only for public locomotion.

2. Context/Problem

The regional transport authority has two offices that require data to plan and execute measures to aim to reduce travel times in the region: the Unidad Operativa de Control de Tránsito (UOCT) and the Secretaría de Planificación y Transporte (SECTRA). In both cases, studies are carried out manually (human vehicle counting). One of the main information inputs on the Impact Study on the Urban Transport System (EISTU), this instrument is prepared by mandate of the General Urban Planning and Construction Ordinance, DFL N ° 850 (MOP) of 1997 and DS N ° 83 of 1985, of the Ministry of Transport and Telecommunications. This report is built based on this manual count and is done every 5 years. It has an approximate cost of US$ 414,421.

Cities are complex systems and in constant transformation, for this reason the information in the EISTU report loses value over time. This is evidenced by the dynamism of internal migration from Chile, which according to the 2017 census, the Coquimbo region leads this phenomenon. This impacts on a transformation of the city in terms of the construction of new houses and apartments. According to the statistics of the CChC, until 2018 the offer of new real estate projects was growing by 5%. Additionally, it is projected that this will continue in this line due to migration factors, and because in 2019 there was already a housing deficit of 14%.

These data show that the current information gathering system and its temporality are obsolete in relation to the needs of the conurbation.

Given the antecedents raised, it is concluded the need to have a system that allows to contribute to the solution to vehicular congestion based on the effective and real-time quantification of private cars, collective locomotion, cyclists and pedestrians. This system is transcendental for the optimization of travel times within the conurbation. An autonomous, scalable and low-cost Intelligent Mobility Platform (Smart City) called Numera is proposed, which works in real time based on Edge AI.

3. Dilemma

As indicated the proposal is based on Edge AI. This is different from the current solutions available in multiple cities in the world (Bhattacharya, Somayaji, Gadekallu, Alazab, and Maddikunta, 2020). The main justification for choosing this approach is connectivity and linked to this the scalability of the platform. Currently there are a total of 159 signalized intersections, where 27 of them have a digital camera that fulfills the role of visualizing the traffic flow. These images are received by the regional Traffic Control Operational Units (UOCT) and analyzed manually. There is no automatic system for quantifying the flow in real time and our platform aims to provide the UOCT with this new feature.

Traditionally, this would imply that the platform could only work 27 points. Due to this, there is the connectivity to send the streaming video to a central processing server either on-premise or cloud. Given these limitations, the present API proposal brings the GPU processing to the phenomenon, that is, the use of Nvidia Jetson class devices. In this approach, the processing of streaming video is locally done and a data stream is sent that summarizes the phenomenon to be observed. In the case of Numera these data correspond to: type of vehicle, license plate, observation timestamp, and location of the observation module.

The software and hardware of the Numera architecture is divided into two sections: The Edge AI and the data processing for storage and visualization (Cloud Computing). Steps (01), (02), (03) and part of (04) are part of the analytics module. Steps (04) - in part, (05) and (06) correspond to processes executed in Amazon Web Service (AWS).

4. Case development

The capture and analysis module works based on an Nvidia Jetson Xavier AGX card. This card features a 512-core Volta GPU with Tensor Cores. One 8-Core ARM v8.2 64-Bit CPU, 8MB L2 + 4MB L3. A 32 GB 256-Bit LPDDR4x internal memory — 137 GB/s. In terms of connectivity, it has ports: PCIe X16, RJ45, USB-C, Camera connector ((16x)CSI-2 Lanes), among others. The module also includes the incorporation of 4G connectivity for sending information tothe AWS cloud.When considering Edge AI projects, hardware isolation conditions are a determining factor in module enclosure design. The Numera video capture and analysis module is composedof 3 components: (1) Enclosure Nvidia, (2) Enclosure Cameraand (3) Camera-Nvidia connection cable. For cases (1) and (2) the industrial standard IP67 was followed. Digit 6 corresponds to a protection level against dust, which in this case is maximum. The second digit describes the level of protection against liquids, the maximum is also available where the module could even be submerged in 1 meter of water for 30 minutes.

The video capture and analysis process in the analysis module consists of a series of stages: (1) Collect the data,(2) Decode the data, (3) pre-process the data, (4) AI: inference and tracking, (5) register the data. All these stages are developed using Nvidia's DeepStream as the only framework. DeepStream is comprised of a suite of streaming and analytics tools for building AI-powered applications. It allows to receive data from the USB/CSI camera or Streaming RTSP. On this input Deepstream offers two alternatives: The creation of newmodels using its SDK or importing models developed in Tensorflow or

Pytorch. Finally, DeepStream supports the development of applications in different programming languages such as C, C++ and Python.

Going into detail about the stages of the Numera pipeline from the reception of the videos to the registration of the event we have: Stage (1) Collect the data uses the uri decodebin plugin which obtains one or more data streams from an URI address that can correspond to either a video file or an information flow through the RTSP protocol. The video source (s) must be decoded by the (2) Decode the data thread through the decodebin plugin which decompresses the sources, sending the frames to the later stages of the process. In stage (3) pre-process the data uses the nvstreammux component which unifies all data sources generating a batch of frames, which allows scaling the images of each frame to optimize the performance of the process. Another important function of this stage is to add the metadata to the NvDsBatchMeta that stores information such as the ID and resolution of the frame, to be used by other threads of the application. In stage (4), AI: inference and tracking, the nvinfer and nvtracker plugins are used respectively, which allow to identify and track the objects that have been detected. Finally, through step (5) register the data, the vehicles that have been detected are recorded in a data log with csv format, recording their class (car, motorcycle, bus, truck), date and time when the event was generated. After this, this csv is sent to AWS where it is processed using ElasticStack and deployed in a kibana Dashboard.

The Transfer Learning Toolkit (TLT), whose objective is to offer a series of pre-trained deep learning models applied to computer vision. These models are divided into two categories according to their level of specialization. The Purpose-Built Pre-trained models (PBP) correspond to highly specialized models in tasks such as people counting, car detection, face detection, among other tasks. The other category is the Customizable Model Architecture (CMA) corresponding to generic computer vision models for tasks such as classification, object detection and instance segmentation. Both types of models are available in two modes: pruned and unpruned. The pruned version has gone through a compression process aimed at efficient deployment on Edge AI devices using DeepStream. In the case of the unpruned version, it is oriented to the training process using TLT plus user data. It is seen that a version has been implemented for the objective of automatic vehicle counting of the project Numera in both cases. In the case of PBP, the TrafficCamNet Model Card was used. This model is specialized in detecting, tracking and counting 4-class (cars, two wheelers, persons, and road signs) object on 960x544 RGB images to detect. It is based on a network of the type Nvidia DetectNetV2 detector with ResNet18 (He, Zhang, Ren and Sun, 2016) as a feature extractor. In the case of the CMA model used it was an architecture based on YOLOV3 (Farhadi and Redmon, 2018).

A crucial aspect to consider for the relevance of the results is to count only the vehicles that are detected by the observed phenomenon. That is, the vehicles to circulate and count them once. For this reason, vehicles that are parked for example should not be counted. For this, the digital Gantry was included, which corresponds to the counting area within the video streaming. The digital Gantry is established by four parameters defined in the application configuration file. In this, two points of the image are defined (area_position_top and area_position_left) that position the counting area in its upper left corner and the width and length parameters (area_position_width and area_position_height) define the size of the box. Figure 1 shows how the Digital Gantry counts only the vehicles that pass through the left lane of the lane, registering the event only when the position of the detected object is within this box and corresponds to the established interest classes.

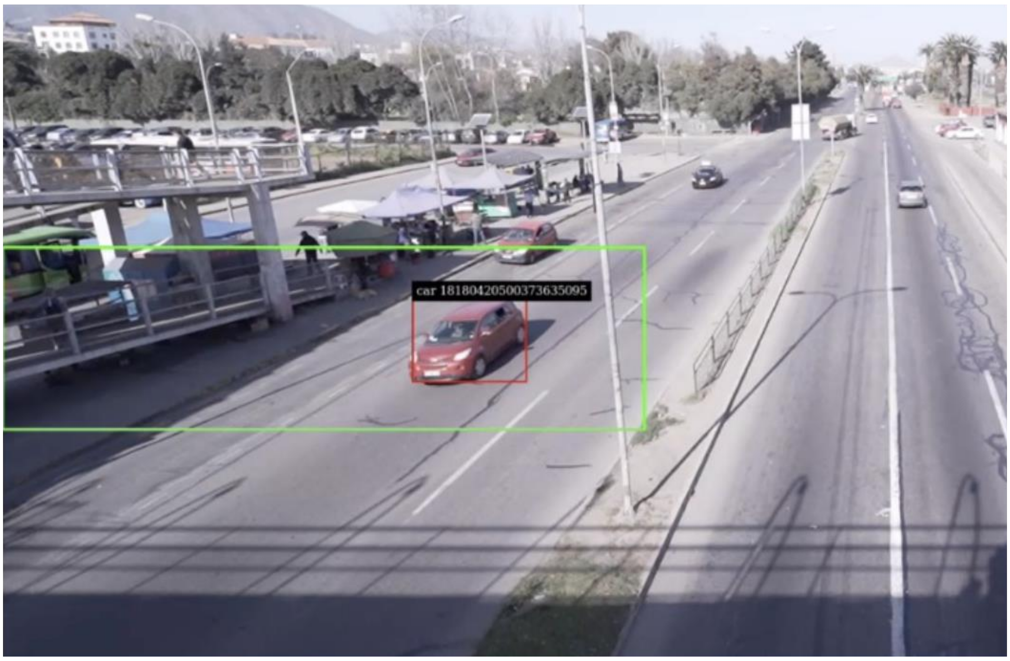

Fig. 1: The green box corresponds to the Digital Gantry. This is the exclusive area of video streaming where you count the vehicles.

In general terms, the accuracy of 83.9% was achieved, which is declared by Nvidia for the TrafficCamNet model. But there is a problem related to the height of the camera from which the streaming video is captured. The TrafficCamNet specifications indicate that it works correctly on videos captured at a height of 6 to 7 meters. According to what is observed in the UOCT cameras, they are twice that height, which is why the model fails to detect objects smaller than 20x20 pixels. In the case of YOLOV3 there is the same height problem and the inference times are not adequate for the objectives of the platform. Notwithstanding this, the mAP of 57.9% is reached.

Figure 2 shows the input of the model and the result of the application of the platform's analysis pipeline. It can be seen that the counter of detected objects, the green box corresponds to the digital portico or counting area.

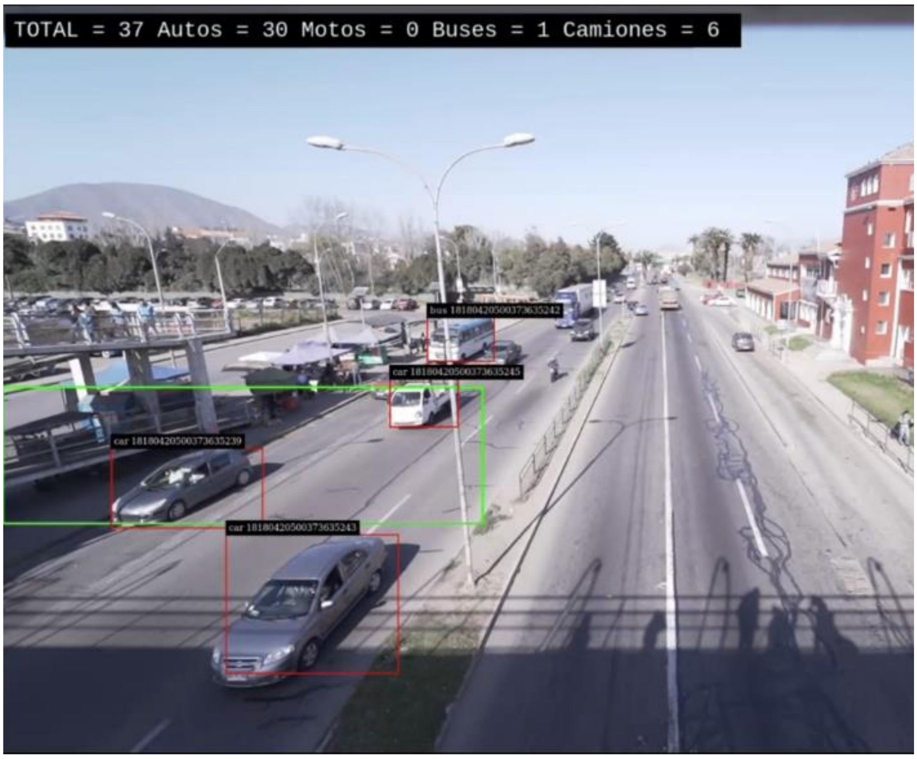
TOTAL = 37 Autos = 30 Motos = 0 Buses = 1 Camiones = 6
bus 181804205003736352242
car 181804205003736352245
car 181804205003736352239
car 181804205003736352243

Fig. 2: Results of the platform pipeline. The green box is the counting zone. The red boxes are the red ones are the location bounding box of the counted vehicles. In the upper part the number of registered vehicles is registered.

5. Closing the case

The current advancement of the platform shows that the incorporation of specialized deep learning models in vehicle counting will contribute to providing information on vehicle flow in time. In addition to generating a historical knowledge base for the application of optimization models. The above is only viable in terms of scalability in countries like Chile, with the use of Edge AI. The growth of these platforms cannot depend on fiber optic connectivity. The next stages of the project are focused on two scenarios: re-training the models with images of the current height of the cameras. Or on the other hand, incorporate a digital zoom to the cameras.

6. Attachments or Appendixes

Nothing to add.

REFERENCES

Bhattacharya, S., Somayaji, S. R. K., Gadekallu, T. R., Alazab, M., Maddikunta, P. K. R. (2020). A review on deep learning for future smart cities. Internet Technology Letters, e187.

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Farhadi, A., Redmon, J. (2018). Yolov3: An incremental improvement. Computer Vision and Pattern Recognition.