Traffic Condition Estimation at the Edge using Deep Learning

Fabrizio De Vita, Orlando Marco Belcore,
Antonio Polimeni, Francesco Longo, Dario Bruneo, Massimo Di Gangi

Department of Engineering, University of Messina, Italy, {fdevita, obelcore, antpolimeni flongo, dbruneo, mdigangi}@unime.it

***Abstract*—The technology and innovations introduced by information systems have revolutionized how public institutions use their infrastructures and manage their assets. In such a context, Information Computer Technologies (ICT) represent a powerful tool through which it is possible to ensure the transition between the previous management and the new needs in the context of Smart mobility and, in general, Smart City. This paper presents a remote monitoring system built at the port terminal of Tremestieri (located in the city of Messina, Italy) that exploits deep learning to perform an Automatic Number Plate Recognition (ANPR) of vehicles transiting the area. Using a set of cameras installed at the terminal, this system allows to compute synthetic traffic indicators that can be used to infer the traffic conditions and the overall operability of the infrastructure in a non-invasive manner, highlighting daily peak hours and trends in the use of the terminal, also achieving an estimation of congestion phases.**

***Index Terms*—Smart City, Smart Port, Edge computing, Cloud computing, Data management, ANPR**

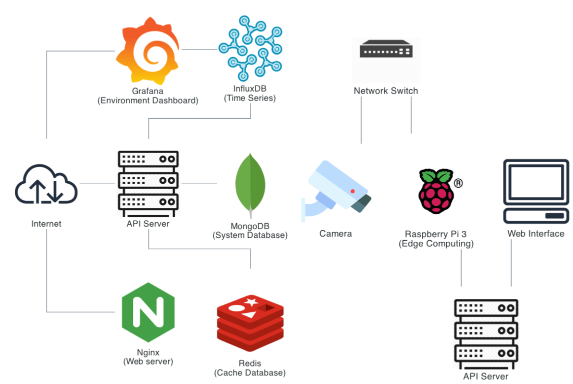
I. Introduction

Over the past decade, the pervasive deployment of Internet of Things (IoT) devices and advances in ICT have paved the way for the realization of large-scale Smart City scenarios [1]. Modern Smart Cities are very complex systems where human beings play a central role and are supported while carrying out their daily activities. Such a complexity derives from the multitude of heterogeneous services and application contexts that should be implemented, managed, and organized in a seamless way for the final user. Hence, the IoT acts as an enabling technology giving the access to the physical world, and context awareness abilities by "sensing" the surrounding environment [2]. In such a context, Cloud and Edge paradigms represent two core components at the base of the Smart City infrastructure: the former providing the computing power and storage functionalities, the latter ensuring ubiquitous access to services and better resource management [3].

In a Smart City context dominated by devices that exhibit dual interaction with the physical and cyber worlds, Artificial Intelligence (AI) is fundamental to enable a "reasoning" process, thus making these devices "active" entities equipped with an intelligence that can be exploited to produce better support and make autonomous decisions.

In this paper, we present the results derived from a research activity conducted with the "Autorità di Sistema Portuale dello Stretto" consisting in the design and implementation of a Smart recognition system in a Ro-Pax terminal for the evaluation of the arrivals at the infrastructure. To this aim, we adopted a deep learning approach based on a Convolutional Neural Networks (CNNs) model to implement an Automatic Number Plate Recognition (ANPR) algorithm running at the Edge that keeps track of vehicles passing through a monitored area. This data is then passed to a Cloud based processing system, where by means of statistical analysis tools, it is able to compute traffic parameters such as the number of vehicles, the time headway, and the speed, thus obtaining information about traffic conditions in the terminal area [4]. Thanks to this collaboration, we were able to deploy the system in a real scenario and use it as a case study for the evaluation of the proposed solution.

II. Proposed Architecture



(a) Cloud system architecture. (b) Camera communication scheme.

Fig. 1: Proposed architecture scheme.

The Cloud-based processing system is able to perform statistical analysis on the gathered data and provide the results to managers in order to monitor the conditions of the whole system.

Fig. 1a depicts the overall Cloud system architecture we implemented. It is structured into several separated components, namely: the web server, the API server, an InfluxDB instance, a MongoDB instance, a Redis instance, and a Grafana instance.

With respect to the last one, we made it accessible from the Internet so that the user can have access to a dashboard showing almost in real-time a set of information like the vehicle queuing time. With regards to the web server, it provides the static files for the web application (also known as the front-end Graphical User Interface) accessible from the Internet. The API server plays a central role in our infrastructure as it provides the back-end functionalities and performs the statistical analysis. This server is also responsible for providing the access to all the platform functionalities by exposing a set of REST API interfaces. The API server is directly connected to the databases storing users and vehicles information together with the platform configurations.

Moreover, to increase the performance of the system, the Redis instance is used as a cache database, to quickly provide the data necessary for calculations. In particular, this cache database is hosted near the API server services to reduce as much as possible the latency in data provisioning.

The interaction between the sensors and the Cloud-based processing system happens through the Edge computing device. Here, in fact, the cameras and the Edge device (which is represented by a Raspberry Pi 3) are able to communicate with the API server to update the system information and push data to the databases.

For sake of simplicity and without loss of generality, Fig. 1b depicts the communication scheme for a single camera. Raspberry plays a central role, acting as a bridge to interact with the cameras. It runs also the ANPR algorithm that will be discussed in Section III and, moreover, it is able to communicate the data to the back-end through the API server.

III. Application

The ANPR problem has been tackled as a supervised object detection approach, so we started collecting the video stream captured by the cameras to build our own dataset. Such a choice has been made to obtain the best recognition results by training the algorithm on video frames captured directly from the operational setting, allowing the learning of those features that best describe our application context. After this step, we labeled the frames by defining the Regions Of Interest (ROIs) to be detected (i.e., the vehicle and its number plate) and applied a data augmentation process.

Among the models available in the literature, we selected a lighter version of the state of the art You Only Look Once (YOLO) architecture [5] named Tiny-YOLO, which is particularly suitable to be executed on embedded devices while maintaining a very good level of performance.

In the first step the model takes as input the video frame captured by cameras and generates two bounding-boxes containing the ROIs, namely: the vehicle and the corresponding number plate. The first one, it is used for statistical purposes, as well as to count the number of vehicles passing through the monitored ares. The second ROI relative to the number plate is extracted from the video frame image and passed to Tesseract, an Optical Character Recognition (OCR) tool capable of detecting and extracting text from images.

IV. Experimental Results

On-field data were collected during a pilot survey between October 2020 and March 2021 and from September up to November 2021. During calibration analyses, the sampling was tested with the twofold objective of comparing the total amount of detected vehicles and the quality of the data achieved by the ANPR.

In Table I are reported the flow rate condition at the beginning of the spill-back; the ship supply in the next 45 minutes; the number of vehicles platooning that represents the number of vehicles arrived during the unstable flow condition; the elapsed time to come back to the free flow conditions.

TABLE I: Queuing Phenomena

Day	Start (hh:mm)	λ_{range} (veich/min)	Ship Supply (units)	Platoon (units)	Elapsed Time (hh:mm)
28/09	13:44	1.10 – 1.30	2	45	24:40
29/09	16:59	0.95 – 1.30	3	55	01:05:27
05/10	13:12	0.95 – 1.11	3	20	18:10
—	13:57	0.52 – 0.75	2	9	05:02
—	15:21	0.31 – 0.53	2	58	33:30
—	17:18	0.51 – 0.72	2	39	39:56
12/10	17:24	0.90 – 0.95	3	56	01:09:10
03/11	15:44	1.04 – 1.20	3	72	44:04
16/11	15:28	0.55 – 1.12	3	41	29:40
—	17:38	0.67 - 1.00	2	57	29:55

The examination revealed that, during peak hours, delays in transit operations (to have access to boarding operations) on average do not exceed 30 minutes. Although the observed phenomenon may occupy a larger time window, this only manifests that the system does not have to allow the incoming vehicle to be loaded onto the first departing ship. However, the number of units queuing in the platoon is limited. It is worth noting that the port terminal offers, on an hourly basis, a load capacity ranging from 650 to more than 1300 linear meters, corresponding to about 50 - 80 trailers. During the day, small platoons waiting at the checkpoint were registered. However, as also reported in the Table I, queuing phenomena under 15 vehicles do not lead to an actual performance decay for the terminal. Such local phenomena can be reconnected with the interference manoeuvres in the loading/unloading terminal area and a slight variation in the shipping companies scheduling.

The obtained results demonstrate the effectiveness of the proposed approach in detecting queuing phenomena and the overall traffic conditions of the terminal. Moreover, such information resulted fundamental for the infrastructure managers to provide a better service to visitors.

References

[1] P. Bellini and et al., "Iot-enabled smart cities: A review of concepts, frameworks and key technologies," *Applied Sciences*, vol. 12, no. 3, 2022.

[2] D. Bruneo and et al., "An iot service ecosystem for smart cities: The smartme project," *Internet of Things*, vol. 5, pp. 12–33, 2019.

[3] F. De Vita and et al., "Using deep reinforcement learning for application relocation in multi-access edge computing," *IEEE Communications Standards Magazine*, vol. 3, no. 3, pp. 71–78, 2019.

[4] R. D. Salvo and et al., "Modeling users' performance: Predictive analytics in an iot cloud monitoring system," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12054 LNCS, p. 149 – 158, 2020.

[5] J. Redmon and et al., "You only look once: Unified, real-time object detection," 2015.