Smart Cities Mobility Monitoring through Automatic License Plate Recognition and Vehicle Discrimination

M. Spanu, M. Bertolusso, G. Bingöl, L. Serreli, C.G. Castangia, M. Anedda, *IEEE, Member*, M. Fadda, *IEEE, Member*, M. Farina, and D.D. Giusto *IEEE, Senior Member*

2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) | 978-1-6654-4908-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/BMSB53066.2021.9547163

***Abstract*—This work deals with vehicular monitoring within smart cities through Automatic License Plate Recognition (ALPR) techniques and vehicle discrimination object detection (YOLO), in order to obtain timely statistical data. The combined use of the two techniques allows to obtain much more refined data than a simple vehicle counter. Moreover, the collected data undergoes a process of anonymization in accordance with the European regulations for the protection of personal data (GDPR). The use of convolutional neural networks (CNN) made it possible to obtain vehicle tracking statistics, returning daily, weekly, monthly and yearly habits with the ultimate goal of allowing a monitoring and control of the city traffic conditions. The results obtained showed a high accuracy in the classification of vehicles and a wide range of statistics concerning the occurrences of each vehicle within the area of interest.**

***Index Terms*—Automatic License Plate Detection, Image Processing, Smart City, Vehicle Discrimination, Deep Learning**

I. Introduction

Nowadays, many traffic applications take advantage of Automatic License Plate Recognition (ALPR) [1] thanks to advances in deep learning and parallel processing that day by day increase the reliability in the field of computer vision.The advantages can be recognized in multiple areas such as object detection and recognition or Optical Character Recognition (OCR) [2] that deploy the use of deep Convolutional Neural Networks (CNN) [3] for automatic features learning needed for feature detection or classification from raw data. This breakthrough has been made possible thanks to the evolution of cloud services able to manage huge amounts of data with ever-increasing real-time applications.

The ALPR systems are based on various methodologies, and are considered a challenging issue under different aspects, such as high vehicle speed, uneven vehicle license plate, foreign vehicle license plate, particular lighting conditions, and scene angles. Therefore, these conditions can affect the overall recognition accuracy. Another technology used in traffic monitoring is object detection. YOLO is frequently used to detect disparate objects [4]. It consists of 24 convolutional layers followed by 2 fully connected layers. Last 4 convolutional layers followed by 2 fully connected layers are added to train the network. The above mentioned technology has various limits if introduced in this type of application, especially if the vehicle transits several times.

In [5] a new method for ALPR in complex scene is proposed. It consists of two stages: in the first one YOLOv2 is used for car detection, while the other, YOLOv2 is used for license plate detection.

It is stated that character recognition helps to identify the characters in the license plate image and to convert them into a text string [6]. Like most license plate recognition algorithms, the “single method” applies to character recognition. This process takes place through the training of a neural network which consists of making the network itself learn to recognize characters by loading the extracted and labeled features of interest.

Different tools in the monitoring phase concern Wi-Fi sniffing and video camera systems for the classification of vehicles, people, bicycles, and so on, can be used. Real-time monitoring allows citizens to know in advance the road conditions with a high level of reliability, overcoming the limitations of a single technological system that offers imprecise monitoring [7].

The Artificial Neural Network (ANN) [8] model is used to classify characters and it can be seen as a three-tier structure. The input layer sends the data to the next layers and the hidden level (i.e., intermediate level) receives the input signal and converts them into an output. Finally, an output level processes the input data coming from the hidden level [9].

The ALPR system in [10], involves the following steps: 1) vehicle image acquisition and pre-processing, 2) license plate extraction, 3) character segmentation, 4) character recognition.

Achieving higher robustness is a difficult topic common to all ALPR-related studies. This difficulty stems from data acquisition performed, where conditions are not always optimal. In order to improve this aspect, the solution adopted by Shohei Yonetsu et al. in [11], was to train the CNNs and tune them to be robust under different conditions (e.g., background, illumination, camera type) using a two-stage approach for segmentation and recognition. This approach employs some tricks such as inverted license plates and upside-down characters, to obtain larger datasets and obtaining a 93.53% recognition rate. In [12], ALPR comprises four features: 1) the cascaded structure is useful to both speed and accuracy, 2) the adoption of a CNN, 3) the use of the information on the license plate can improve the recognition performance, 4) the weight-sharing

M. Spanu, M. Bertolusso, G. Bingöl, L. Serreli, C.G. Castangia, M. Anedda, and D.D. Giusto are with the Dept. of Electrical and Electronic Engineering - DIEE/UdR CNIT, University of Cagliari, Cagliari, Italy, {m.spnau13, m.bertolusso, g.bingol, l.serreli2, c.castangia1, }@studenti.unica.it, {matteo.anedda, m.farina, ddgiusto}@.unica.it. M. Fadda is with the University of Sassari, Sassari, Italy, mfadda1@uniss.it.

TABLE I
OPENCV CAMERA CONFIGURATION

FEATURES	VALUES
Firmware	Python
Resolution	2304×1296
Focus	Auto focus
Distances	0 to 30 meters
Steps	5 meters
Angle of inclination	30 degree

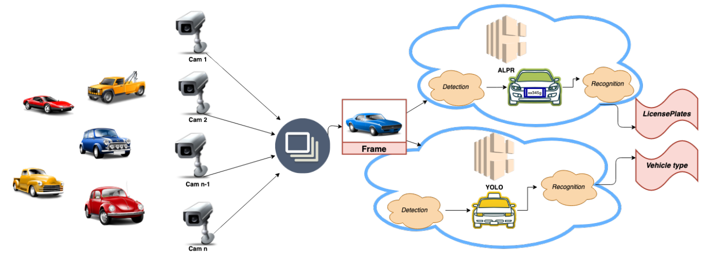

Fig. 1. System Block Scheme

character classifier solves the problem of lack of training images in small-scale data sets.

The proposed solution overcomes the ALPR and YOLO limitations through a match of these two system. It provides both the license plates and vehicle types discrimination through traffic monitoring in the urban area by providing a multi-class solution. The developed model shows the significant steps in the following key points: 1) scenario analysis to detect traffic density and/or congestion, 2) license plate recognition, 3) vehicle type recognition, 4) algorithm development for mobility monitoring and control, towards a sustainable model for smart cities.

The paper is structured as follows: key requirements for ALPR, vehicle discrimination and the design of the proposed system are presented in Section II. The real scenarios considered to evaluate the feasibility of the system are described in Section III. The obtained results are discussed in Section IV. Finally, the conclusions are drawn in Section V.

II. PROPOSAL SOLUTION

Compared to the solutions adopted in the literature, the proposed scheme aims at employing different functionalities of neural networks in an innovative way with the goal of offering solutions for monitoring and controlling vehicular and pedestrian flows. The proposed solution can be summarized as follows.

A. *Hardware setup*

The developed software makes use of OpenCV [13] set according to the parameters listed in Table I.

In addition, character recognition with a low error rate requires a certain image resolution and it has been achieved through preliminary tests that allowed the optimal setting of optical parameters. This setup phase was implemented in a controlled environment. This procedure has a fundamental importance because even with the same camera, there could be constructive variations that determine different settings of the optics, with consequences on zoom and focus.

B. *Vehicle identification and classification*

The proposed system controls traffic and detects vehicles in a smart city scenario with a very high accuracy. The system was designed to consider the various types of vehicles circulating on a two-way street. Under these conditions, the ALPR recognizes a vehicle's license plate but it is not able to determine the vehicle types without license plates (e.g. bicycle, electric scooters, etc.) and shadowing factors. YOLO provides object identification and it evaluates and observes the accuracy rate of the coincidences. Nevertheless, this type of approach placed in a vehicle monitoring context clearly shows limitations on the identification of occurrences as it is only an object detection algorithm. To increase the accuracy the scenarios have been appropriately studied. The crossroads examined have provided with the ideal scenarios in which the cars were obliged to respect the rules of the road can move only one at a time, solving in part if not totally, the problem of acquiring a greater number of frames, required to obtain greater accuracy.

Finally, combining these two approach, shortcomings were eliminated, and a higher accuracy was achieved. The proposed architecture is depicted in Fig. 1.

Unlike other methods that use classifiers for object detection in scenes, object detection by YOLO as a regression problem to spatially separated bounding boxes and associated class probabilities, using a single neural network for the prediction of bounding boxes and associated class probabilities in a single evaluation, resulting in a system due to its unified architecture capable of processing from 45 to over 300 images per scene depending on the system and hardware capabilities available [14]. Moreover, the latest versions of YOLO (e.g. YOLO9000) are capable of detecting and predicting more than 9000 classes of objects in real time. [15].

C. *Data anonymization and transmission*

According to the European privacy policies [16], data collected by video cameras can not disseminated without a process of anonymization. It is necessary to encrypt the information before it is sent to a storage center for further processing. The acquired images are processed and encrypted with the Secure Hash Method 512 (SHA-512) algorithm [17] before being stored on any media. Finally, such information is sent via LTE connection to the Cloud database for further processing. SHA-512 is a hashing algorithm that converts text of arbitrary length into a 512-bit fixed-size string (64 bytes). A cryptographic hash, often known as a "digest", is a form of

978-1-6654-4908-3/21/$31.00 ©2021 IEEE

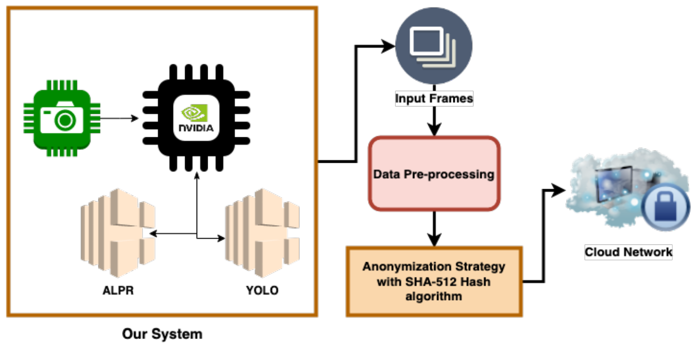

Fig. 2. Privacy anonymization in video/image using SHA-512 hash algorithm based on ALPR.

text or data file signature. SHA-512 generates a roughly unique 512-bit signature for a text. The hash algorithm includes some steps: 1) input formatting, 2) hash buffer initialization, 3) message processing and 4) output. These depicts the general operation of the SHA structure model, which is carried out as follows:

- **Input formatting**: the message is padded with sufficient zeros to create the message size divisible by the block size. This is called formatted input.
- **Hash buffer initialization:** the method operates by using the previous block's result to process each block of 1024 bits from the message [18]. Since each intermediate result must be used in the processing of the next block, it needs to be temporarily stored somewhere for later use. The implementation of this step is done through the hash buffer, which also stores the final hash digest of the entire processing step as the last of these outputs.
- **Message processing:** as a result of the padding process, the message is split into uniform size blocks. This step consists of many rounds and additions operations. Then, the Initial Vector (IV) is utilized for the initial round of the first block of the formatted input.
- **Output:** the hash value is obtained from the results of calculating the last block after processing all previous messages.

The Fig. 2 shows the ALPR-based video/image acquisition and anonymization process with the SHA-512 algorithm performed using Algorithm 1. YOLO is used for vehicle classification. The periodic transmission of encapsulated data, must take into account that hardware installations can be carried out in the most disparate sites of interest. The solutions that generally offer coverage next to almost the entire territory are those linked to cellular networks. The proposed system is equipped with 4G LTE module directly connected to the data processing module. Therefore, the data were transmitted to the cloud using the 4G data connection.

D. Mobility monitoring Algorithm

The mobility monitoring algorithm acquires images from the camera system, through a splitting technique obtains several frames that undergo parallel processing. The first analysis concerns license plate recognition while the second analysis performs a classification of the vehicle type. The license plate data undergoes a process of irreversible anonymization so that it cannot be traced back to the vehicle and therefore to the physical person who owns the vehicle. The anonymized information is aggregated to the vehicle type, together with other statistical data such as camera ID, time, date, pointing direction and road of interest. The aggregated data is routed through the 4G LTE network to the cloud for subsequent statistical traffic monitoring processing.

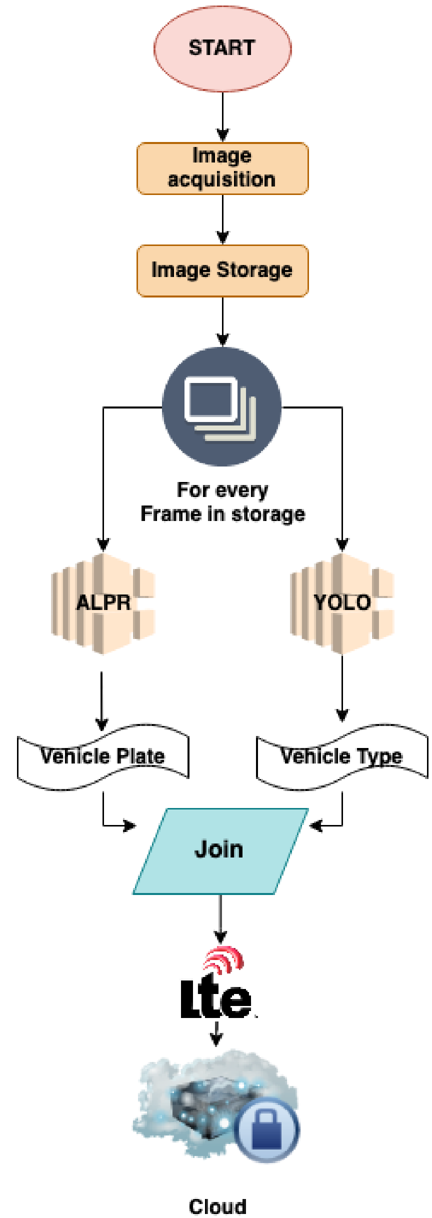

Fig. 3. Flow chart of the data acquisition process.

The ALPR and YOLO systems returned a plate and a shape, respectively. The algorithm starts with the ALPR system reporting when a new license plate has been detected. The system reads the records where files containing more than one

978-1-6654-4908-3/21/$31.00 ©2021 IEEE

```
Algorithm 1: Mobility monitoring
Data: A video is acquired by the system which, by
      acquiring a single frame per iteration, is able to
      classify license plates and vehicle types.
Result: License plate and vehicle type predicted
        results.
Parameters
TS = timestamp
nC = Cam number
n = Number of itarations;
Fi = i-Frame acquired;
CFi = Pre-processed i-Frame;
ALPR = License Plate NN algorithm
YOLO = Vehicle type NN algorithm
Rlp = Predict of ALPR algorithm
Rvt = Predict of YOLO algorithm
TRUE = plate or vehicle type found
FALSE = plate or vehicle type no found
RPS = Result Predicted String
End Parameters
begin
    for i =1 to n do
        Input  : Read i-Frame Fi
        Input  : Pre-processing apply to Fi;
        Input  : CFi = pre-processing(Fi)
        Input  : ALPR NN loading AND YOLO NN
                 loading;
        Input  : Rlp = ALPR(CFi) AND Rvt =
                 YOLO(CFi)
        while (Rlp == TRUE) do
            if (Rvt == TRUE) then
                Input  : Rlp = Anonymization(Rlp);
                Input  : RPS =
                         strcat('TS','nC','Rlp','Rvt')
                Output: send(RPS) json file to Cloud
                        by encrypted protocol;
            end
            else
                Input  : Rlp = Anonymization(Rlp);
                Input  : RPS = strcat('TS','nC','Rlp')
                Output: send(RPS) json file to Cloud
                        by encrypted protocol;
            end
        end
        if (Rvt == TRUE) then
            Input  : RPS = strcat('TS','nC','Rvt')
            Output: send(RPS) json file to Cloud by
                    encrypted protol;
        end
        else
            Output: No plate or vehicle found;
        end
    end
end
```

new record are stored. The flowchart is depicted in Fig. 3.

E. Data management

Among the numerous data processing operations, in the very first analysis it is possible to carry out a vehicle count, together with the extraction of statistics by vehicle type. The main core of the proposed system consists in the identification of the directions and habits undertaken by drivers during their routes.

Moreover, it is possible to estimate travel times and road occupations. The level of vehicle concentration is determined based on the number of vehicles attributable within a certain spatial location. The classifier returns 4 classes based on the number of vehicles relative to the road occupancy defined as the percentage of space on a road occupied by vehicles. Based on this classification we are able to determine if we are in a condition of:

- **Free street:** it occurs when the concentration of vehicles is such that there is less than 15% road occupancy;
- **Fluid traffic:** is the condition of moderate traffic with road occupation higher than 15% and less than 50%, where vehicles are characterized by different speeds with the possibility of overtaking without causing or incurring in slowdowns;
- **High traffic**: this is the situation where road occupancy is higher than 50% and less than 80%, it's difficult to overtake and speed is maximum 50 Km/h;
- **Heavy traffic**: last, it occurs when road occupancy is higher than 80% of the road. In this case, overtaking is quite impossible and speed is maximum 25 km/h.

III. SCENARIOS

A first scenario analyzed was the monitoring of a road intersection. Two cameras have been placed with different angles, applying the above mentioned technology, ALPR and YOLO, respectively. The goal is to perform vehicle monitoring during the day in different traffic and light conditions. The cameras were placed in opposite directions to ensure that two different directions were identified. The first camera identified vehicles exiting the city center, the second monitored the flow entering the city. This configuration allows to identify the traffic flow with relative directions, quantify the traffic and understand people's attitude in crowdsensing. The video stream was pre-processed in real-time by Gifran cameras that have different levels of speed of acquisition for real time elaboration. Thereafter, frames created by cameras go to the NVIDIA®JETSON ™, for the next processing in the CNN located inside. Subsequently, the result was sent to the cloud inclusive of the respective metadata on vehicular acquisitions. The cloud is equipped with a neural network that performs flow identification and road occupancy concentration tasks.

The hardware configuration shown in Fig. 4 represents the hardware/software acquisition system used in the testing phase. It is composed of different hardware modules that interpret the data acquired through the neural networks, returning the real time traffic modeling. More in detail, the various components used can be summarized as follows:

978-1-6654-4908-3/21/$31.00 ©2021 IEEE

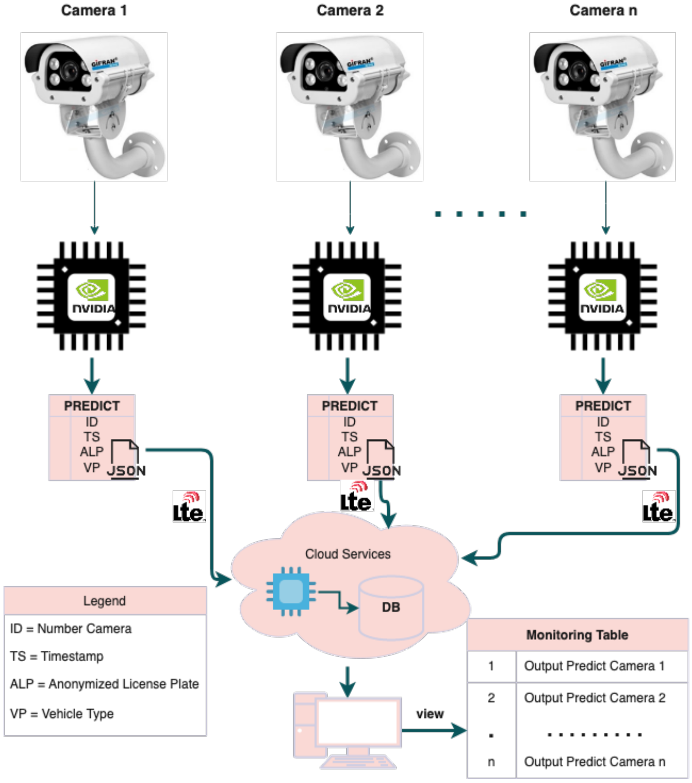

Fig. 4. Mobility monitoring hardware configuration

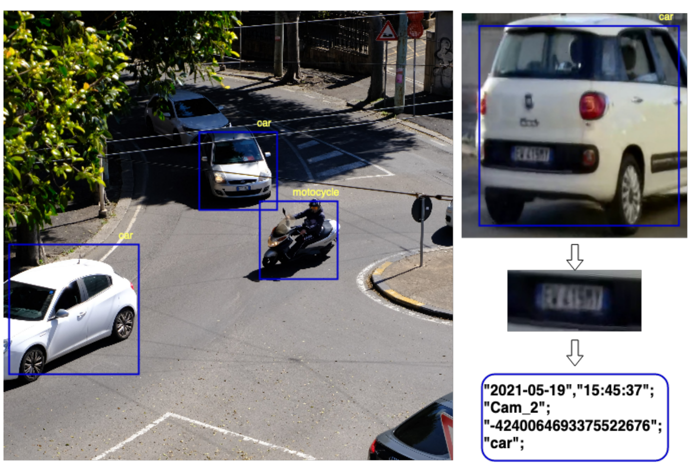

Fig. 5. Example of vehicle frame acquisition

- **Camera:** Gifran cameras were employed for the image capture equipped with a native functionalities for camera settings such as zoom, focus and the lens opening. Moreover, the camera can be set on four different speed model, up to a maximum speed of 120 Km/h for lossless acquisition;
- **NVidia module:** Nvidia Jetson AGX is the heart of the system, used in the pre-processing step, for the plate and vehicle type detection. This module sends to the cloud the results of the neural network in json file format;
- **Lte 4G module:** SIM-7600E represents the LTE module integrated on the system for the communication protocols to the cloud;
- **Cloud services:** Cloud services are used to store inside the cloud database, the data received from the system. The cloud is also responsible for post-processing the data which will subsequently be made available to the end user;
- **Monitoring table:** it consists in the set of information necessary to detect traffic intensity made available to citizens and competent authorities for real-time traffic monitoring and control..

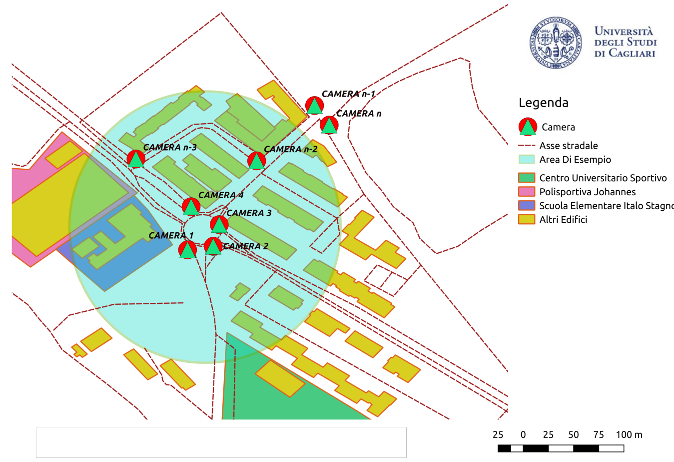

Fig. 6. The positioning of the cameras in a real scenario: a high traffic density crossroad involving Is Mirrionis Street, Cadello Street, Campania Street.

IV. Results

The proposed approach is able to efficiently identify a large number of object types, including any variety of present vehicles in the considered circumstance. An example of vehicle frame acquisition system is shown in Fig. 5.The Data Structure (1) provides a sample calculated via CNN managing the dataset to provide statistical information.

$$< UniqueID, TimeStamp, CamNumber, VehicleType > \tag{1}$$

Tests were performed in an urban environment, as shown in Fig. 6. The traffic circle has been equipped to identify the traffic of 4 arterial roads. the positioning of the cameras identifies the occurrences and directions of the vehicles. The system also provides daily, weekly, monthly statistics on possible redundant travel by the same vehicles. As reported in the Table II, data were verified to validate the reliability of the proposed system and also to prove if it was properly working. The k-fold Cross-Validation method has been used to estimate the skill of the proposed machine learning model on a limited data sample. Eq. (2) shows the estimated accuracy of our scenarios

$$Accuracy = \frac{TP + TN}{TN + FP + FN + TP} \tag{2}$$

978-1-6654-4908-3/21/$31.00 ©2021 IEEE

TABLE II
VEHICLES TYPE DISCRIMINATION

		Predicted Data			Sum
		Bus	*Cars*	*Trucks*	
Real data	*Bus*	656	0	14	670
	Cars	6	2690	9	2705
	Trucks	8	13	143	164
Sum		670	2703	166	3542

TABLE III
PERFORMANCE COMPARISON IN CONTROLLED SCENARIO

	Time[s]	Accuracy
Haar Cascade **[19]**	0.3-0.4	75%
SSD **[19]**	11-14	70%
YOLO **[19]**	1.0-1.8	86%
Mask R-CNN **[19]**	2.4-3.0	91%
Our System	**3.4-3.8**	**95.7%**

where TP is the True Positive, TN represents the True Negative, FP is the False Positive, and FN is the False Negative. The Producer Accuracy in a multi-classification scenario has been defined in Eq. (3) as the ratio between the number of values correctly classified in i-*th* class and the total number of values belonging to the i-*th* class.

$$ProducerAccuracy = \frac{CorrectPredictedValuesXClass}{TotalValuesXClass} \tag{3}$$

Table III shows the performance of the system compared with that of other algorithms whose data comes from [19], evaluated in terms of execution time and accuracy of the algorithm.

V. CONCLUSIONS

The experimental results revealed that the proposed solution allows to reach an higher accuracy through a combined use of ALPR and YOLO. ALPR and YOLO work in parallel with the same frames acquired and pre-processed by different video cameras. The results of the elaboration of the two different systems are processed to generate a single information sent to the Cloud. The complete system has been compared with the YOLO system and ALPR holistic system. Through the adoption of these technologies, the proposed system is able to detect traffic flows through the data aggregation, and analysis of information taken from the cameras and processed in the cloud. This enable to detect flows in real-time and after a suitable training phase to predict the vehicular flows on a seasonal, weekly, daily basis in different time slots. Our approach is designed not only to evaluate traffic conditions or their forecasts but also to detect possible car accidents or problems in the road system, based on unexpected events. In addition, through communication platforms, it will be feasible to generate alert messages to divert car flows intelligently without congesting the vehicular flows within the city, avoiding unnecessary traffic accidents that result in unnecessary emissions of pollutants, thus improving the air quality of our smart cities, and global mobility.

ACKNOWLEDGMENT

The research activities described in this paper have been conducted within the RD project "Cagliari2020" partially funded by the Italian University and Research Ministry (grant# MIUR_PON04a2_00381).

REFERENCES

[1] D. S. C. Patel and A. Patel, "Automatic number plate recognition system (anpr)," vol. 69, no. 9, pp. 22–23, 2013.

[2] N. Islam, Z. Islam, and N. Noor, "A survey on optical character recognition system," *CoRR*, vol. abs/1710.05703, 2017.

[3] Z. Li, W. Yang, S. Peng, and F. Liu, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[5] S. Yonetsu, Y. Iwamoto, and Y. W. Chen, "Two-stage yolov2 for accurate license-plate detection in complex scenes," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–4, 2019.

[6] C.-N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Transactions on intelligent transportation systems*, vol. 9, no. 3, pp. 377–391, 2008.

[7] M. Bertolusso, M. Spanu, M. Anedda, M. Fadda, and D. Giusto, "Vehicular and pedestrian traffic monitoring systemin smart city scenarios," in *IEEE World Forum on Internet of Things*, pp. 1–5, 2021.

[8] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[9] S. Kumari, L. Gupta, and P. Gupta, "Automatic license plate recognition using opencv and neural network," *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 5, no. 3, pp. 114–118, 2017.

[10] S. N. Hashmi, K. Kumar, S. Khandelwal, D. Lochan, and S. Mittal, "Real time license plate recognition from video streams using deep learning," *International Journal of Information Retrieval Research (IJIRR)*, vol. 9, no. 1, pp. 65–87, 2019.

[11] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the yolo detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, 2018.

[12] Y. Wang, Z.-P. Bian, Y. Zhou, and L.-P. Chau, "Rethinking and designing a high-performing automatic license plate recognition approach," *arXiv preprint arXiv:2011.14936*, 2020.

[13] Itseez, "Open source computer vision library." https://github.com/itseez/opencv, 2015.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[15] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017.

[16] G. D. P. Regulation, "Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016," *Official Journal of the European Union. Available at: http://ec. europa. eu/justice/data-protection/reform/files/regulation_oj_en. pdf (accessed 20 September 2017)*, 2016.

[17] M. Sumagita and I. Riadi, "Analysis of secure hash algorithm (sha) 512 for encryption process on web based application," *International Journal of Cyber-Security and Digital Forensics*, vol. 7, pp. 373–381, 2018.

[18] N. Sklavos, "Book review: Stallings, w. cryptography and network security: Principles and practice," *Information Security Journal: A Global Perspective*, vol. 23, no. 1-2, pp. 49–50, 2014.

[19] D. Impedovo, F. Balducci, V. Dentamaro, and G. Pirlo, "Vehicular traffic congestion classification by visual features and deep learning approaches: a comparison," *Sensors*, vol. 19, no. 23, p. 5213, 2019.

978-1-6654-4908-3/21/$31.00 ©2021 IEEE