

Where to park? Architecture and implementation of an empty parking lot, automatic recognition system

(¿Dónde estacionarse? Arquitectura e implementación de un sistema automático para detección de espacios libres de parqueo)

Héctor Ávalos,¹ Estevan Gómez,¹ Diego Guzmán,¹ Diego Ordóñez-Camacho¹, Jéssica Román,¹ Oswaldo Taipe¹

Abstract

The traffic congestion present in practically every city has, among its increasing factors, the unavailability of enough parking spaces. A typical driver invests a considerable part of the total trip time looking for a free space where to park his vehicle; in many cases, this leads to delays and the consequent discomfort due to the undesired consequences of the generated tardiness. For this problem it is possible to find partial palliative solutions, minimizing the time spent searching for parking by applying Internet of Things techniques, oriented to smart cities and buildings. This research has been focused on finding an appropriate computer architecture that will allow the implementation of a distributed system, which, thanks to the use of computer vision and machine learning techniques, detects free parking spaces inside a parking lot, and provides real time information to the driver, allowing him to go as directly as possible to a vacant parking space.

Keywords

Intelligent parking; Distributed system; Free parking slots; Computer vision, Machine Learning.

Resumen

La congestión vehicular en las ciudades tiene varios factores que la acrecientan, uno de ellos es la falta de disponibilidad de suficientes plazas de parqueo. Un conductor típico invierte una considerable parte del tiempo total del trayecto en buscar un espacio libre para estacionar su vehículo; esto deviene en muchos casos en retrasos involuntarios, seguidos del malestar ocasionado por la impuntualidad generada. Con el fin de encontrar soluciones paliativas parciales, que reduzcan el tiempo dedicado a la búsqueda de estacionamiento, se plantea aplicar técnicas del Internet de las Cosas orientadas a ciudades y edificios inteligentes. En esta investigación el foco ha sido hallar una arquitectura informática apropiada, que permita implementar un sistema distribuido, el cual mediante técnicas de visión computarizada y aprendizaje automático, detecte las plazas de estacionamiento libres al interior de un parqueadero, y brinde al conductor información en tiempo real que le permita dirigirse directamente a un lugar de parqueo vacante.

Palabras clave

Parqueaderos inteligentes; sistema distribuido; plazas de parqueo libres; visión computarizada; aprendizaje automático.

1. Introduction

Fluent mobility lies among the main objectives of smart cities. Developing intelligent parking systems, then, is one of the prominent aspects to reach this goal (Mainetti, Patrono, Stefanizzi, & Vergallo, 2015). Searching for a free parking space is a common problem in cities, both in streets and in public or private parking lots. There are reports pointing out that more than 30% of the traffic in large cities is made up of drivers looking for a parking place (IBM, 2011); moreover, other reports show that on average 25 % of the travel time in urban areas is wasted looking for a free parking slot and that the average time to find it is around 8 minutes (Margreiter, Orfanou, &

1 Universidad UTE, Quito, Ecuador ({havalos, estevan.gomez, ggda103722, dordonez, rlj103429, thow103795}@ute.edu.ec)

Mayer, 2017). Only this search burns around of one million oil barrels worldwide daily (Das, Dash, & Mishra, 2018).

Although the aforementioned data refer to large cities, the problem does not disappear in smaller cities; moreover, even inside private, institutional parking lots, this problem has unintended implications. Take the case of universities, where it is paramount to minimize the time spent searching for a parking space, to avoid delays on the courses starting time. Not finding space on a parking lot row could imply leaving it in reverse, maneuvering on a narrow space, with pedestrians circulating, which increases the risk of accidents.

This problem can be now alleviated thanks to the advent of the Internet of Things (IoT). The IoT, in conjunction with mobile applications, machine learning and Big Data, favors the development of alternative systems that can inform the driver, in real time, about the availability of parking spaces near their location (Salpietro, Bedogni, Felice, & Bononi, 2015).

1.2. Related work

Only in the last three years, several investigations have been interested in this area and have produced a considerable theoretical framework deepening into the subject. Next up we present some of the most relevant work, dividing them into those mostly related to the use of computer vision techniques in parking lots, and those focused on the implementation of specific systems for this purpose.

1.2.1. On computer vision techniques for the detection of free parking spaces

Shih and Tsai (2014) implement a system based on images taken with high-resolution wide angle (fish eye) cameras. They use Hough filters and subtraction of the bottom for the detection of occupied spaces. Vitek and Melnicuk (2017) propose a distributed system of wireless cameras which, through Raspberry Pi modules, HOG filters and SVM classifiers, manage to evaluate 10 slots per second with 90 % accuracy. Paidi et al. (2018) highlight the contributions that the use of computer vision, neural networks, convolutional and multiagent approaches provides in the search for free parking. Masmoudi et al., (2015, 2014) make use of cameras already installed by surveillance systems in cities to locate the open slots and guides the driver. They use a distributed multiagent computer vision system.

Bibi et al. (2018) work on video streams taken from high surveillance cameras, emphasize that the video must be segmented, reducing it to a few static shots that will be analyzed afterwards. Nieto et al. (2018) experiment with a system based on multiple takes, from different cameras, to which they add several types of graphic preprocessing, demonstrating that vision-based systems resist different environmental conditions, while maintaining proper functioning.

De Almeida et al. (2015) experiment with texture descriptors: Local Binary Patterns and Local Phase Quantization, and, through a Support Vector Machine classifier, declare good detection results when the samples come from the same camera-take; in other cases they indicate that the effectiveness decreases remarkably; as an additional contribution they make available a database of parking images. The latter is also done by Zhang et al., (2018).

As can be seen, the proposals vary between systems using IoT and integrated sensors, and systems using computer vision. The systems that make use of images have as a great advantage the minimum need of infrastructure, which significantly reduces the costs, compared with other techniques; additionally, it simplifies the implementation in new environments, which would facilitate the development of a derivative commercial product.

1.2.2. On implementations and prototypes of intelligent parking systems

Pham et al. (2017) suggest the need to implement a standard architecture for intelligent parking systems. In 2016 Khanna and Anand presented a proposal for a smart parking system based on IoT technologies, which is supported on a cloud system in conjunction with a mobile application to guide the search for available places; this path has been followed also by Ordóñez-Camacho, Gómez & Ávalos (2018). Aydin et al. (2017) develop a system for smart cities that involves the use of sensors communicating through IoT. The system searches for the closest free space by means of genetic algorithms.

In 2015, Pham et al. propose a system for the user to automatically find a free parking space, based on an algorithm which, with performance metrics, offers this possibility, with the lowest cost in distance. Mainetti et al. (2015) worked on a system proposal combining RFID, WSN, cloud computing and mobile applications, which could collect real-time information on the use of parking spaces, directing the user to them.

Baroffio et al. (2015) suggest that a better performance in the transmission of data through the network and in general a reduction in energy consumption would be achieved by passing the processing or previous analysis of the images to the visual sensor. Shinde et al. (2017) introduce a system based on IoT and develop an Android application connected to Google Maps that allows locating the free space and being guided to it.

Korman (US20170025010 A1, 2017) patented a community system based on computer vision, where a vehicle captures images while in transit; these images are processed by a centralized system that warns other users in search of parking. In the same way, Salpietro et al. (2015) propose a collaborative system as an alternative for those places where there are no sensors installed and no network infrastructure. In this scenario, users share the information through their mobile phones; this information is then channeled from one user to another by Bluetooth in an *ad-hoc* network. Koster et al. (2014) work in a similar line, but contribute by using the smartphones sensors that automatically detect parking behavior, which allows the detection of free spaces. These works are in line with other research on the use of contextual mobile applications for positioning (E. Cabrera-Goyes & Ordóñez-Camacho, 2017; Edwin Cabrera-Goyes & Ordóñez-Camacho, 2018; Loayza, Proaño, & Ordóñez Camacho, 2013). Roy et al., (2018) also propose an *ad-hoc* network that communicates with vehicles and parking spaces to not only find free slots but also to reserve them.

1.3. Contributions

The main contribution of this work is to propose a distributed architecture and a reference prototype for an intelligent parking management system, dedicated to the detection of free slots in real time, thanks to the use of computerized vision and machine learning. The proposed architecture allows to:

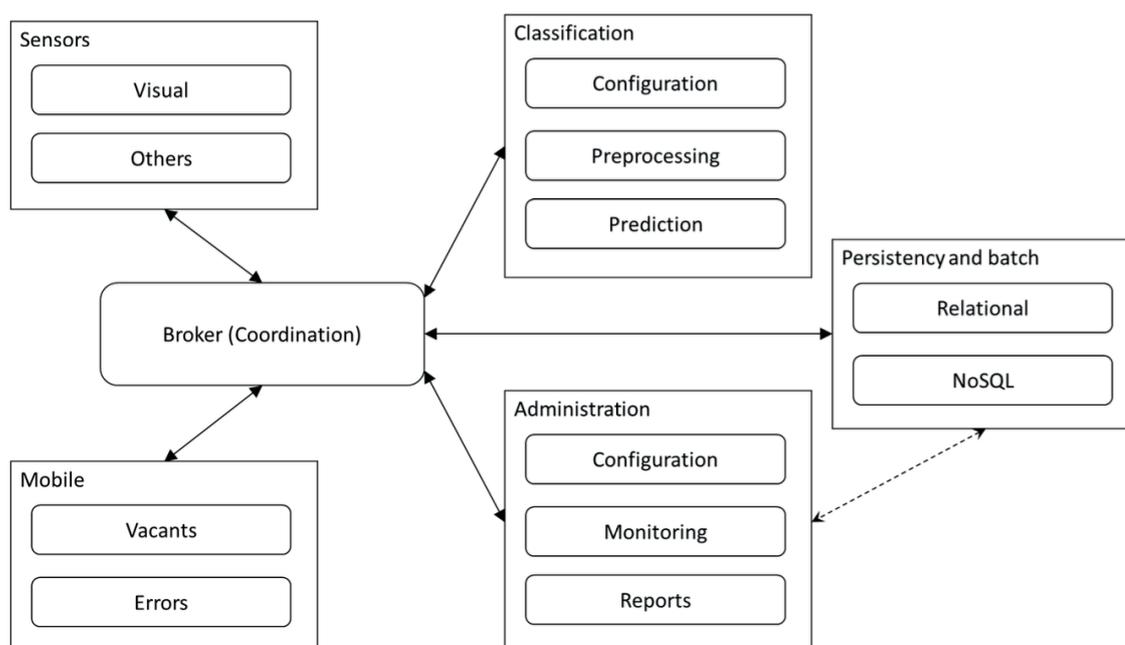
- Take advantage of the existing infrastructure of surveillance cameras to extract the images of the parking lots, which minimizes the costs of implementation.
- Distribute the load of the system on several interconnected equipment to allow timely processing of large amounts of data.
- Provide the users of the system real-time information of the best parking alternatives on their mobile devices.

- Implement a flexible system capable of an easy configuration for the needs of different institutions.
- Reduce the incidence of delays in arriving at the work or study place, reducing the risk for pedestrians circulating in the parking areas.

2. An architectural proposal for an institutional empty parking detection system

Next, we show the detailed architecture proposed in this work, which is outlined in *Figure 1*. The proposal consists of five subsystems: Sensors, to collect the base information of the parking lots; Administration, to configure the system; Classification, to process the basic information and predicts the availability of each place; Persistence and batch, which stores the information produced by the system and allows to execute offline analysis; Mobile, that presents the user with the status of the parking spaces; finally, Broker or Coordination, in charge of transmitting the information among the other subsystems.

Figure 1. Distributed architecture for the empty parking lot recognition system



2.1. Sensors subsystem

This subsystem can make use of several types of sensors for the detection of free parking spaces. Given that the advantages already discussed consider that the use of cameras or visual sensors seems to be the best option in terms of infrastructure, the Visual module is explicitly included. However, the Others module represents, in this case, any family of available sensors.

The Visual module consists of surveillance IP cameras allowing to pull the images at the convenience of the system, as any other type of active video equipment pushing the images to the broker. In several tests carried out with personalized applications for smartphones, the two mechanisms seem appropriate to the objectives.

2.2. Administration subsystem

An intelligent parking system should allow to manage globally the different system options. This subsystem is proposed as a web application exposing a module with configuration options for the administrator user. It includes generic options, such as access control, and specific options, such as connection and calibration of cameras or sensors, and definition and segmentation of parking spaces. Another suggested module allows to define and parameterize various reports and statistics; for this purpose, the possibility of a direct connection with the Persistence and batch module is proposed, thus decongesting the Broker; a third module must enable online monitoring in the same interface.

2.3. Classification subsystem

This subsystem, which can be very simple with other types of sensors, with visual sensors or cameras is more complex due to all the processing necessary to determine where, in an image, the parking spaces are located. This architectural proposal is focused on computerized vision techniques and machine learning, but it is easy to adapt, by simplification, to other techniques.

The Configuration module allows the administrator to assign processing and classification options for each parking lot, parking slot or arbitrary groups of places. The Preprocess module considers that the images may require a transformation process prior to sending them to the classifiers. This module must integrate several image processing algorithms, able to be assigned and organized in a pipeline through which the images must pass as soon as they are received, ending in an appropriate representation for sending them to a classifier.

The Prediction module is dedicated to provide machine learning functionality. It offers the possibility of registering previously trained classification models, which can be assigned slot by slot if desired. Training the models can be done in a batch process into this module.

2.4. Mobile subsystem

It is responsible for communicating relevant system information to the end user, usually thanks to some smartphone application. The Vacant module fulfills these functions, presenting in the first place a schematic map of the disposition of the different parking blocks in an institution; this scheme must already provide a count of free places in each block. This will allow the user to first, mark the block of interest, and then see in detail the scheme of each block where the free spaces can be found. A second module in this subsystem is Feedback, which allows users to report classification errors to the system, indicating false classification decisions; this allows to fine-tune the system.

2.5. Persistence and batch subsystem

All the information generated in the system must be saved in order to obtain statistics, generate reports and, above all, debug and improve the system, especially regarding the classification module. Many predictive techniques require huge amounts of data to train the models.

This subsystem considers that certain processes, such as the reports generation, can behave better by retrieving data from SQL databases, for which reason the Relational module is available. Other types of jobs are more compatible with NoSQL databases or directly with a file system, which is why the Non-Relational module is considered. The entire subsystem additionally considers the batch processing that may be required, through Hadoop, for example.

2.6. Broker subsystem (coordination)

The subsystems presented so far need to share large amounts of information during their operation. This must be done in real time, minimizing the delay in passing the messages. This is the responsibility of the Broker subsystem, which basically provides several message queues to allow for interaction.

Administration sends any configuration update to Broker, which transfers it to all other subsystems. Sensors sends the images to Broker, these are transferred to Classification and Persistence. Classification passes the predictions, parking by parking, to Broker who sends them to Persistence and Mobile. Mobile sends the user's feedback to Broker, who passes it to Persistence. As we already pointed out, only an exception to this centralized step can exist, since to generate reports, Administration could directly access Persistence to decongest Broker.

3. Implementation

The architecture described in the previous section was implemented as a functional prototype allowing to test the required system. From the point of view of a system administrator, this architecture appears as three differentiable modules: parking management, configuration of filters and classifiers, and mobile application for the end user.

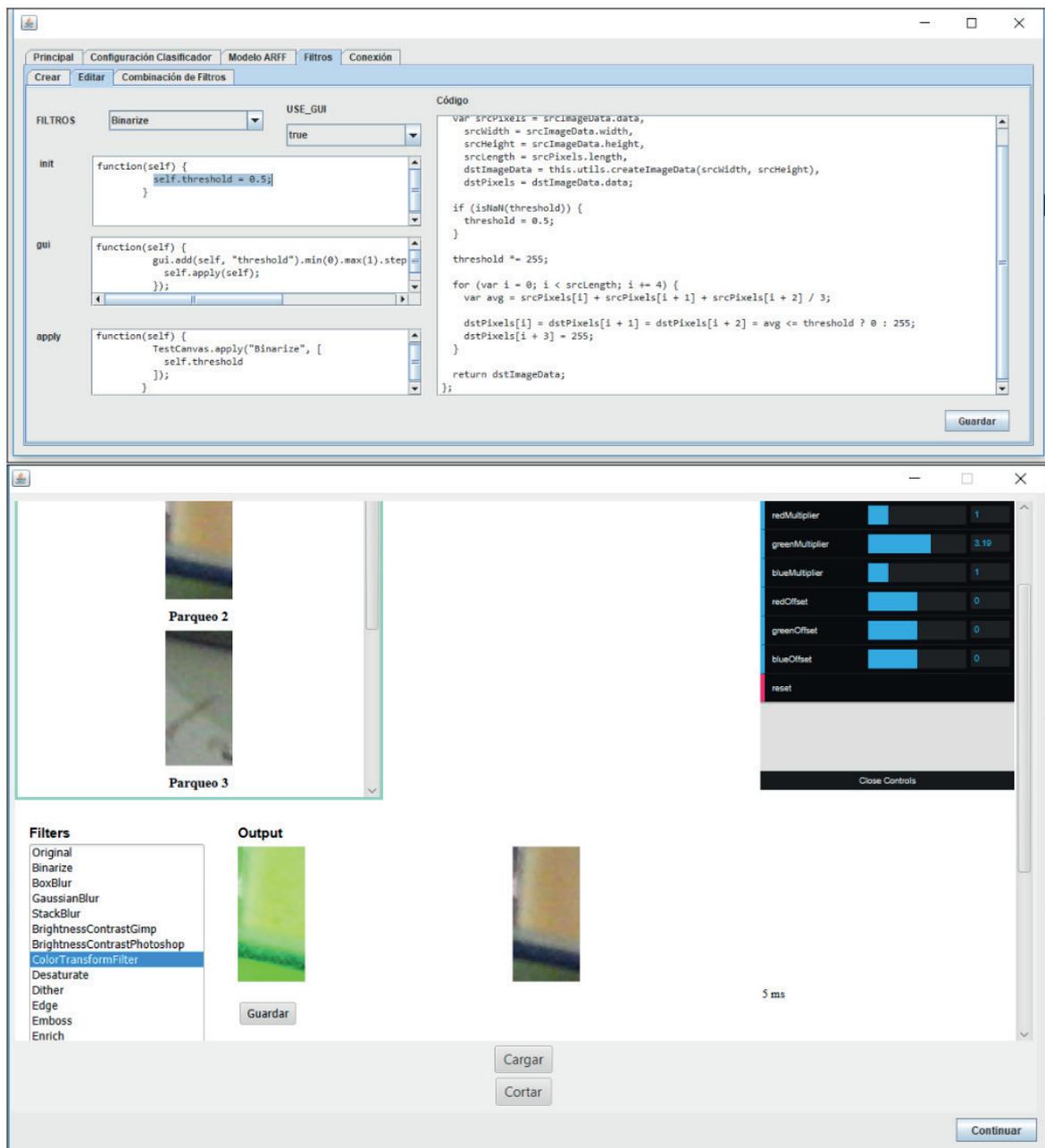
The administration module was implemented, in its front-end, with Angular (Hevery, 2019), Bootstrap (Otto & Thornton, 2019), Fabric (Zaytzev & Cherniak, 2019) and ng2-charts (Valor Software, 2019); Payara (C2B2 Consulting, 2019) and MySQL (MySQL AB, 2019) were used in the back-end, with JWT (Auth0, 2019) and HS256 (HMAC with SHA-256) for security, and Assured (Haleby, 2019) for tests. The Broker uses OpenMQ (Oracle Corporation, 2018) to handle the message queues. In *Figure 2* you can see, as an example, the interface of the parking segmentation module.

Figure 2. Administration module: Parking segmentation



The classification and prediction module was developed and deployed in the central Pa-yara server, and uses image processing libraries based on OpenCV (Intel Corporation, Willow Garage, & Itseez, 2018), ImgLib2 and ImageJ (Pietzsch, Preibisch, Tomančák & Saalfeld, 2012), for the case the static, preconfigured filters, and ImageFilters (Hayato, 2019) to ease the process of creating new filters inside the module. The machine learning core makes intensive use of Weka (Witten, Frank, Hall & Pal, 2016) to build the models and to classify the parking places by occupancy. In *Figure 3* are shown the configurations of the filtering stage, for the images preprocessing.

Figure 3. Classification and prediction module, showing the filtering stage

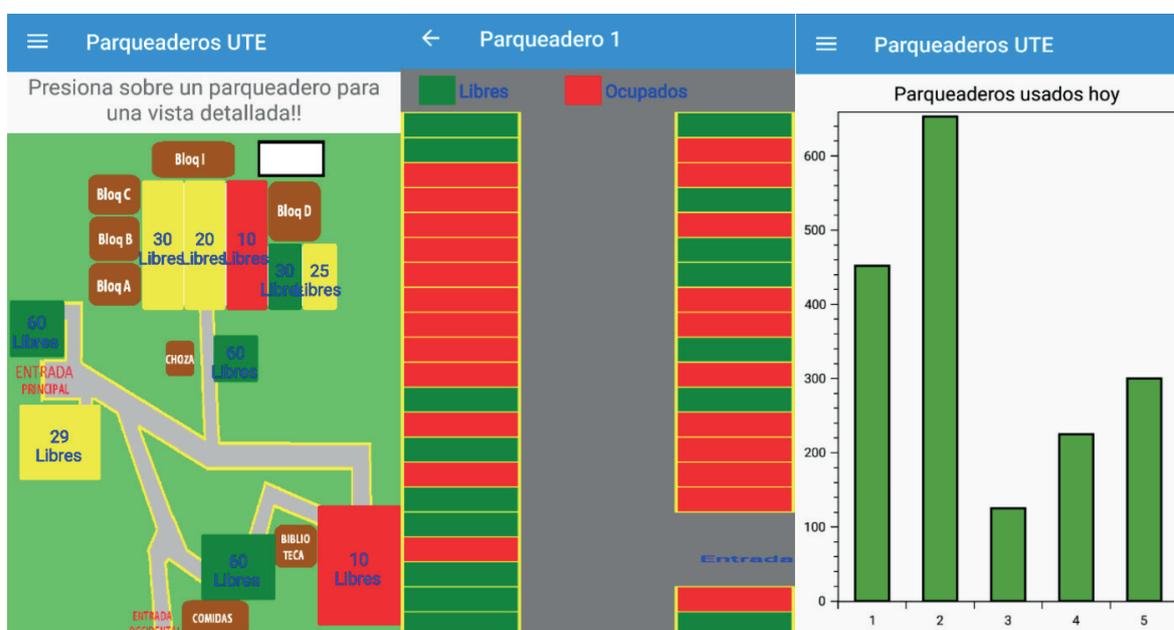


Lastly, the mobile application was developed with Xamarin (Benett, 2018) to facilitate the multiplatform deployment, covering Android and iOS. The libraries OxyPlot (OxyPlot, 2019), Acr User Dialogs (Ritchie, 2019), Json.NET (Newton-King, 2019) and SkiaSharp (Mono Project, 2019) were used to provide interactivity and access to information in the user interface. In *Figure 4* you can see three activities of the mobile application, related to the selection of a parking block, free space and general statistical information on parking occupancy.

4. Discussion

Although the numbers presented in this paper (IBM, 2011; Margreiter et al., 2017), refer to large cities, the problem is currently found at every level, regardless of city size; within private institutions premises, there are different but equally pressing problems. IoT techniques, as mentioned by Salpietro et al. (2015) open up new possibilities to take advantage of.

Figure 4. Mobile application: parking map, parking slots and statistics



The architecture proposed in this work focuses on the use of visual sensors, as presented by other authors (Bibi et al., 2018; de Almeida et al., 2015; Masmoudi et al., 2015, 2014; Nieto et al., 2018; Paidi et al., 2018; Shih & Tsai, 2014; Vitek et al., 2017); compared to those works, this proposal is not limited to the use of a single technique or machine learning algorithm, but allows the inclusion of several alternatives, leaving it up to the user to choose the most appropriate, even for each parking slot.

Regarding the architecture, from a more general point of view, the standardization suggestion of Pham et al., (2017) is adopted, although we follow a less strict line, preferring alternatives more linked to the IoT, like those used by (Aydin et al., 2017; Khanna & Anand, 2016; T. N. Pham et al., 2015).

To help the user find the free space, a map technique is adopted as in (Shinde et al., 2017), but it is improved with the inclusion of personalized maps; the reason is because Google Maps

does not provide information on private parking. Although utilities of collaborative systems, such as those of (de Koster et al., 2014; Roy et al., 2018), have not been considered; the possibility of direct feedback by the user is included, making him a participant in the effectiveness of the system.

5. Conclusions

This paper proposes an architecture for an intelligent parking management system which takes advantage of the existing infrastructure of surveillance cameras, minimizing costs; it distributes the processing of the different subsystems to offer a real-time response, is easily and extensively configurable allowing its adaptation to a variety of environments, and contributes to the reduction of arrival delays in an institution.

The proposed architecture is being implemented as a prototype based on Java, with modules developed through JEE, Xamarin and Weka mainly. At the moment, the different subsystems have been tested individually and have been interconnected to validate the flow of message passing between pairs of subsystems; the next stage is intensive and extensive online testing, which will subsequently allow reporting on issues of processing load and classification accuracy.

References

- Auth0. (2019). *JWT.IO*. Auth0. Recuperado de <http://jwt.io/>
- Aydin, I.; Karakose, M., y Karakose, E. (2017). A navigation and reservation based smart parking platform using genetic optimization for smart cities. En *2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)* (pp. 120-124). <https://doi.org/10.1109/SGCF.2017.7947615>
- Baroffio, L., Bondi, L., Cesana, M., Redondi, A. E., y Tagliasacchi, M. (2015). A visual sensor network for parking lot occupancy detection in Smart Cities. En *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 745-750). <https://doi.org/10.1109/WF-IoT.2015.7389147>
- Bennett, J. (2018). *Xamarin in Action: Creating native cross-platform mobile apps* (1 edition). Shelter Island: Manning Publications.
- Bibi, N.; Majid, M. N.; Dawood, H., y Guo, P. (2018). Automatic Parking Space Detection System. En *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)* (pp. 11-15). <https://doi.org/10.1109/ICMIP.2017.4>
- C2B2 Consulting. (2018). *Payara*. Payara Services Ltd. Recuperado de <https://www.payara.fish/> (Original work published 2014)
- Cabrera-Goyes, E., y Ordóñez-Camacho, D. (2017). Towards a Bluetooth Indoor Positioning System with Android Consumer Devices. En *2017 International Conference on Information Systems and Computer Science (INCISCOS)* (pp. 56-59). <https://doi.org/10.1109/INCISCOS.2017.14>
- Cabrera-Goyes, E., y Ordóñez-Camacho, D. (2018). Posicionamiento en espacios interiores con Android, Bluetooth y RSSI. *Enfoque UTE*, 9 (1), 118-126. <https://doi.org/10.29019/enfoqueute.v9n1.238>
- Das, A.; Dash, P. K., y Mishra, B. K. (2018). An Intelligent Parking System in Smart Cities Using IoT. *Exploring the Convergence of Big Data and the Internet of Things*, 155-180. <https://doi.org/10.4018/978-1-5225-2947-7.ch012>
- De Almeida, P. R. L., Oliveira, L. S.; Britto, A. S., Silva, E. J., y Koerich, A. L. (2015). PKLot_A robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11), 4937-4949. <https://doi.org/10.1016/j.eswa.2015.02.009>
- Haleby, J. (2019). *REST-assured*. Parkster. Recuperado de <https://github.com/rest-assured/rest-assured> (Original work published 2010)

- Hayato, A. (2019). *ImageFilters*. Arahaya. Recuperado de <https://github.com/arahaya/ImageFilters.js> (Original work published 2011)
- Hevery, M. (2019). *Angular*. GitHub. Recuperado de <https://angular.io/>
- IBM. (2011, septiembre 28). IBM Global Parking Survey: Drivers Share Worldwide Parking Woes. *IBM News Room*. Recuperado de <https://www-03.ibm.com/press/us/en/pressrelease/35515.wss>
- Intel Corporation, Willow Garage, & Itseez. (2018). OpenCV library. GitHub. Recuperado de <https://opencv.org/> (Original work published 2000)
- Khanna, A., y Anand, R. (2016). IoT based smart parking system. En *2016 International Conference on Internet of Things and Applications (IOTA)* (pp. 266-270). <https://doi.org/10.1109/IOTA.2016.7562735>
- Korman, R. (2017). *US20170025010 A1*. Recuperado de <http://www.google.com/patents/US20170025010>
- Koster, A; Oliveira, A; Volpato, O.; Delvequio, V., y Koch, F. (2014). Recognition and Recommendation of Parking Places. En *Advances in Artificial Intelligence--IBERAMIA 2014* (pp. 675-685). Springer, Cham. https://doi.org/10.1007/978-3-319-12027-0_54
- Loayza, A; Proaño, R., y Ordóñez Camacho, D. (2013). Aplicaciones sensibles al contexto. Tendencias actuales. *Enfoque UTE*, 4(2), 95-110.
- Mainetti, L.; Patrono, L.; Stefanizzi, M. L., y Vergallo, R. (2015). A Smart Parking System based on IoT protocols and emerging enabling technologies. En *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 764-769). <https://doi.org/10.1109/WF-IoT.2015.7389150>
- Margreiter, M; Orfanou, F., y Mayer, P. (2017). Determination of the parking place availability using manual data collection enriched by crowdsourced in-vehicle data. *Transportation Research Procedia*, 25, 497-510. <https://doi.org/10.1016/j.trpro.2017.05.432>
- Masmoudi, I; Wali, A; Alimi, A. M., y Jamoussi, A. (2015). Architecture of Parking Lots Management System for Drivers' Guidance. En *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 2974-2978). <https://doi.org/10.1109/SMC.2015.517>
- Masmoudi, I.; Wali, A.; Jamoussi, A., y Alimi, A. M. (2014). Vision based system for Vacant Parking Lot Detection: VPLD. En *2014 International Conference on Computer Vision Theory and Applications (VISAPP)* (Vol. 2, pp. 526-533).
- Mono Project. (2019). *SkiaSharp*. GitHub. Recuperado de <https://github.com/mono/SkiaSharp> (Original work published 2016)
- MySQL AB. (2018). *MySQL*. Oracle Corporation. Recuperado de <https://www.mysql.com/> (Original work published 1995)
- Newton-King, J. (2019). *Json.NET*. Newtonsoft. Recuperado de <https://github.com/JamesNK/Newtonsoft.Json> (Original work published 2012)
- Nieto, R. M., García-Martín, Á., Hauptmann, A. G., y Martínez, J. M. (2018). Automatic Vacant Parking Places Management System Using Multicamera Vehicle Detection. *IEEE Transactions on Intelligent Transportation Systems*, 1-12. <https://doi.org/10.1109/TITS.2018.2838128>
- Oracle Corporation. (2018). *OpenMQ*. EE4J. Recuperado de <https://javaee.github.io/openmq/>
- Ordóñez-Camacho, D; Gómez, E., y Ávalos, H. (2018). An Architectural Proposal for an Automatic Vacant Parking Detection System. In *2018 International Conference on Information Systems and Computer Science (INCISCOS)* (pp. 351-355). <https://doi.org/10.1109/INCISCOS.2018.00057>
- Otto, M., y Thornton, J. (2019). *Bootstrap*. GitHub. Recuperado de <https://github.com/twbs/bootstrap> (Original work published 2011)
- OxyPlot. (2019). *OxyPlot*. GitHub. Recuperado de <https://github.com/oxyp/oxyp> (Original work published 2014)
- Paidi, V.; Fleyeh, H.; Håkansson, J., y Nyberg, R. G. (2018). Smart parking sensors, technologies and applications for open parking lots: a review. *IET Intelligent Transport Systems*. <https://doi.org/10.1049/iet-its.2017.0406>

- Pham, N.; Hassan, M.; Nguyen, H. M., y Kim, D. (2017). GS1 Global Smart Parking System: One Architecture to Unify Them All. En *2017 IEEE International Conference on Services Computing (SCC)* (pp. 479-482). <https://doi.org/10.1109/SCC.2017.69>
- Pham, T. N.; Tsai, M.; Nguyen, D. B.; Dow, C., y Deng, D. (2015). A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies. *IEEE Access*, 3, 1581-1591. <https://doi.org/10.1109/ACCESS.2015.2477299>
- Pietzsch, T.; Preibisch, S.; Tomančák, P., y Saalfeld, S. (2012). ImgLib2—generic image processing in Java. *Bioinformatics*, 28 (22), 3009-3011. <https://doi.org/10.1093/bioinformatics/bts543>
- Ritchie, A. (2019). *ACR User Dialogs*. GitHub. Recuperado de <https://github.com/aritchie/userdialogs> (Original work published 2015)
- Roy, A; Paul, J; Baidya, R., y Devi, M. (2018). Parking Places Discovery and Reservation Using Vehicular Ad Hoc Networks. En *Advances in Electronics, Communication and Computing* (pp. 695-703). Springer, Singapore. https://doi.org/10.1007/978-981-10-4765-7_72
- Salpietro, R.; Bedogni, L.; Felice, M. D., y Bononi, L. (2015). Park Here! a smart parking system based on smartphones' embedded sensors and short range Communication Technologies. En *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 18-23). <https://doi.org/10.1109/WF-IoT.2015.7389020>
- Shih, S. E., y Tsai, W. H. (2014). A Convenient Vision-Based System for Automatic Detection of Parking Spaces in Indoor Parking Lots Using Wide-Angle Cameras. *IEEE Transactions on Vehicular Technology*, 63 (6), 2521-2532. <https://doi.org/10.1109/TVT.2013.2297331>
- Shinde, S.; Patil, A.; Chavan, S.; Deshmukh, S., y Ingleshwar, S. (2017). IoT based parking system using Google. En *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (pp. 634-636). <https://doi.org/10.1109/I-SMAC.2017.8058256>
- Valor Software. (2019). *ng2-charts*. GitHub. Recuperado de <https://github.com/valor-software/ng2-charts> (Original work published 2015)
- Vitek, S; Melničuk, P.; Vitek, S., y Melničuk, P. (2017). A Distributed Wireless Camera System for the Management of Parking Spaces. *Sensors*, 18 (1), 69. <https://doi.org/10.3390/s18010069>
- Witten, I. H; Frank, E.; Hall, M. A., y Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4 edition). Amsterdam: Morgan Kaufmann.
- Zaytsev, J., y Chernyak, M. (2019). *Fabric*. GitHub. Recuperado de <https://github.com/fabricjs/fabric.js> (Original work published 2010)
- Zhang, L.; Li, X.; Huang, J.; Shen, Y.; Wang, D.; Zhang, L.,... Wang, D. (2018). Vision-Based Parking-Slot Detection: A Benchmark and A Learning-Based Approach. *Symmetry*, 10 (3), 64. <https://doi.org/10.3390/sym10030064>