

Innovative Solution for Parking-Sharing of Private Institutions Using Various Occupancy Tracking Methods

Adrian Florea*, Valentin Fleaca, Simona Daniela Marcu

Lucian Blaga University of Sibiu, Department of Computer Science and Electrical Engineering, Sibiu, 550025, Romania

ARTICLE INFO

Article history:

Received: 27 August, 2020

Accepted: 28 September, 2020

Online: 12 October, 2020

Keywords:

parking sharing
embedded system
web application
smart parking
image processing

ABSTRACT

This work presents an innovative solution for parking-sharing of private institutions based on daily occupancy patterns and using different real time tracking methods of vacant parking slots. The research objective consists in finding the most accurate cars detection method, for determining of vacant parking slots and updating them on application web page. Beside the technical innovation represented by image processing algorithms used, this paper promotes the concept of sharing economy with many social benefits like car flow optimization, reducing fuel, pollution, loss of time and creating financial advantages for parking owners. The main software component is a web application which is connected with Raspberry Pi microcontroller, 2 Pi cameras and one fix camera for parking management. It facilitates reserving a place, opening the barrier and allows entering, exiting and revising the number of vacant slots and synchronization with the web application and the supporting database. The web application provides the following facilities: real time parking status view, reservation on a specific time by license plate number, administration module that includes payment system and updates about users and prices, implementation of the gamification concept in the management of parking spaces. The solution was piloted at Lucian Blaga University of Sibiu (LBUS) Romania. The developed solution is flexible, extensible and applicable to crowded university cities, but also to other private organizations that have inefficiently operated parking slots.

1. Introduction

This paper extends the previous work that was presented at the 2018 Thirteenth International Conference on Digital Information Management (ICDIM) [1], by presenting an innovative solution for parking-sharing of private institutions based on daily occupancy patterns and using different real time tracking methods of vacant parking slots.

1.1. The need of innovative smart parking solutions

The general-purpose objective of smart cities is finding smart solutions for increasing the citizens' welfare and comfort by providing high quality services like smart mobility, smart health, and smart environment. Unfortunately, the shortage of parking spaces and, the continuous infrastructure development generates city crowding [2] and traffic congestion. The constant stopping and starting in traffic jams burn the fuel at higher rate in comparison to

when someone travels smoothly through the highway. Thus, the congested areas have a higher emission of Carbon Dioxide than other areas [3]. A solution that some countries tried was to put a number of traffic policemen to help manage the traffic in a better way [4]. The problem with this was that a one out of three-traffic policeman suffers from either one or more respiratory morbidities like allergic rhinitis, chest symptoms (cough, wheeze, and breathing difficulty/chest tightness), allergic eye symptoms, sinusitis [5].

Especially in touristic cities, at seaside resorts or others with many visitors yearly, the lack of parking spaces is emphasized by unpopular local authorities' decisions with short or long term effect. For example, in Sibiu, during the theatre festival or international events like auto-moto rallies, the municipality restricts parking places, providing these for guests or organizers. In 2019 there were 700,000 people who visited or participated to shows, more than 4 times the population of Sibiu that produces a further pressure on local drivers. Finding with difficulty a vacant parking place became ordinary activity for most of the drivers.

*Corresponding Author: Adrian Florea, Lucian Blaga University of Sibiu, Department of Computer Science and Electrical Engineering, Sibiu 550025, Romania, Emil Cioran Street, No. 4, Email adrian.florea@ulbsibiu.ro

Studies developed by researchers from Boston University [6] reveals that over 30% of the drivers lost around 8 minutes looking for a free space around the zone where they must stop their car. According to [7], the drivers spend in medium 17 hours yearly looking for vacant places on streets, in lots, or in garages generating tremendous financial losses (tens of billions of dollars) to economy. From our experience in Sibiu, Romania, finding a parking place near the Engineering Faculty takes every morning minimum 10 minutes of searching, beside the fuel costs. Arriving to Faculty earlier with around 20 minutes before starting the classes might solve sometimes this problem. Implementing a smart parking system would shorten the search time for a parking space, reduces CO2 emissions – a key aspect of smart crowded cities, optimizing the car traffic, increasing safety and sometimes even avoid fines caused by illegal parking.

An innovative smart parking system incorporates several important features [8]:

- It helps the business growth and economic development by different solutions like providing incentives for parking or sharing excess spaces of private institutions.
- It helps drivers and municipality to efficiently exploit citywide parking space or providing unexpected (vacant) spaces in real time from private parking which share their spaces.
- It analyses the daily occupancy patterns based on institution / company schedule to understand driver patterns and to predict spaces to be released.

In our smart parking solution, beside innovative technical implementation, the emphasize is on the problem of sharing economy (until the municipalities make new parking lots – a very difficult and time-consuming process, the sharing empty parking places is an acceptable solution with positive economic impact. Especially in big cities finding a parking place represents one of the major challenge of a drivers, and smart parking would represent even urban smart mobility solution for traffic congestion control using parking charges differentiated by time and area. In busy days when most of people are running to work / schools and Saturday afternoon when everybody is engaged in the social and commercial life of the city or during major events when traffic congestion is high, the municipalities’ decision factors could coordinate with traffic planners the city mobility to update parking lot status and maximize use of public transit, and alternative transportation options. However, this would be applied by local municipalities. Our proposal is dedicated for private organizations who may share their excess parking places with ordinary traffic drivers without affecting their own employees.

1.2. Scientific motivation based on bibliometric study

As motivation of our research we also present the following bibliometric study, developed by us at the end of June 2020. 812 articles containing the phrase “smart parking” in the “Title”, “Abstract” or “Keywords” and that appeared at least 25 times in the article’s content were selected. The charts were developed using Microsoft Excel and VOSviewer Software (<https://www.vosviewer.com/>).

Analyzing the period 2011-2020, there is an increasing evolution of the number of articles published in the SCOPUS

database, which refers to the phrase “smart parking”. The statistics (see Figure 1) were compiled at the end of June 2020 and the number of articles in 2017 has already been reached, despite the reduction in the number of conferences (and implicitly of articles) caused by the COVID-19 pandemic.

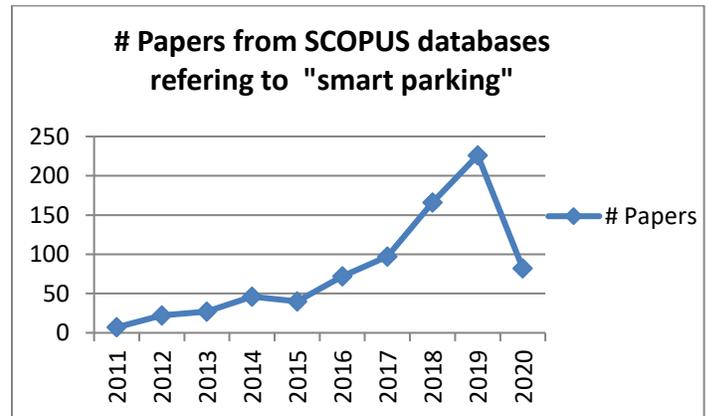


Figure 1. Number of publications indexed in the Scopus databases referring to the concept of smart parking (2011–2020)

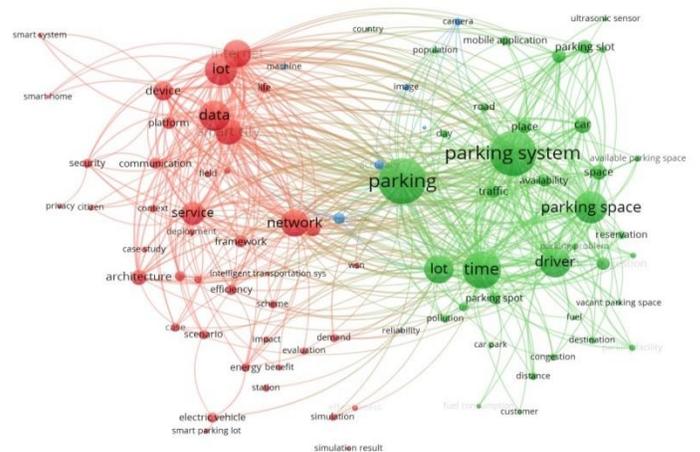


Figure 2. Number of publications indexed in the Scopus databases referring to the concept of smart parking (2011–2020)

By applying clustering algorithms on the selected papers resulted in 87 keywords grouped into three clusters. In Figure 2 is drawn with different colors a map displaying relations between the analyzed keywords. The frequently key terms are presented in the central part of the map. The style information like the size and font of point specific to a certain keyword express its frequency (higher or lower). First, the resulting network is characterized by numerous links and looks rather dense. The first cluster was located around the term “parking” including the most terms defining the problem of smart parking: parking system/lot/space/slot/spot, reservation, road, fuel, congestion, pollution, availability, etc. In the second cluster terms related to solution (most of them technical) can be observed: network, Internet, IoT (of Things), Communication, device, platform, framework, architecture, service, smart system/home/city, etc. The third is the shortest and is strict related to some tools and their performance: image, camera, accuracy, reliability, etc. However, the three clusters do not contain keywords such as “sharing”, “economy”, and very few or not at all contain “image processing”, “machine learning algorithms”, “thermo vision camera”, etc.

The conclusions of this bibliometric study show us that very few researchers consider the economic view of smart parking and especially the sharing concept in order to create benefits.

Other studies show that there are a lot of startups worldwide that target different aspects of a smart parking: payment, access monitoring or even vacancies but do not analyse the problem from perspective of sharing economy and private parking lots and also does not present technical solutions for detection of free parking spaces [9].

1.3. Premises for success. Brief description of solution proposed

In our previous work [10] we predicted “the occupancy rate of a building” based on embedded system which retrieves data from sensors and, using Multi-Layer Perceptrons we predicted the energy consumption in smart buildings knowing alternative data sources, like the number of cars from parking space. It was computed the correlation between the number of vehicles over time and time-related parameters, like day of week, hour of day, and type of day, for validating the machine learning model of parking lot occupancy. This paper describes in detail the smart parking system implemented at Engineering Faculty of Lucian Blaga University of Sibiu (LBUS), Romania. The occupancy pattern during one year includes around 40 weeks, in rest being holidays. Engineering Faculty contains about 120 parking slots. The institution time scheduler shows that usually after 5 p.m. are much less classes than during the morning. In conclusion, there is a quite high degree of vacancy in the parking which could generate additional income by sharing them.

The proposed solution includes the hardware components (Raspberry Pi Model 3B with a Pi camera) for license plate recognition, the opening / closing the barrier, three led that form the traffic light, and software application for real time detection of each empty parking space based on Motion Detection algorithm; the reserving a parking space for a time interval is done through a web application. Raspberry Pi microcontroller and the WEB application communicate through the Node-RED program that use the MQTT services for transmitting real-time information to the application without the need to refresh the webpage ensuring real time synchronization of web application and database with existing vacant parking spaces.

Our solution provides the next benefits:

- Innovate by relying on the principle of sharing economy / parking sharing of private institutions: analyses the problem from the perspective of creating financial advantages for the owner in conjunction with reducing the disadvantages mentioned above, loss of time, pollution, etc.
- It is based on the profile / schedule of the institution (daily and / or on much longer periods – monthly, annually).
- Innovate by applying image processing methods for monitoring vacancies from parking lot based on daily occupancy patterns and real time tracking of vacant parking slots and updating in real time the web application and the supporting database.
- Provides a multidisciplinary character combining different disciplines of Computer Engineering ranging from Image

Processing and classification algorithms specific to Artificial Intelligence, to Embedded Systems and IoT, Web Design, Information Security (payment), targeting social welfare, improving the citizens’ life quality.

The rest of the paper is organized into six sections. Section 2 analyses the related works. Section 3 introduces our smart parking solution, its advantages as well as the analysed use-case scenarios. Section 4 presents the application’s hardware and software architecture, whilst the Section 5 describes the image processing methods used for finding the vacant slots in the parking space. The Section 6 illustrates some results obtained after software implementation of methods from fifth section. Section 7 emphasizes the paper’s conclusions and proposes further work research.

2. Related Work

The reservation-based parking system developed by researchers from Nebraska University, Lincoln, USA [11] focus on parking spaces managing and real-time monitoring. Their solution is based on reservation and the selection of the vacant slot for parking is made based on the walking distance to the destination and the cost. The main drawback of their solution consists in the synchronization problem when more than one motorists apply simultaneously for reservation. A “bottleneck” occur within the system and increases the update time out of the need to serially treat the requests. In our work we solve this problem at the session level of the OSI reference model [12] of communications protocols. Different names for every session are used, and the messages are received and sent only to the specific session.

In [13] the researchers from University of South Florida developed an embedded system for a real-time parking aiming to inform the drivers using a mobile application, about parking spaces availability. Detecting the vacant spaces was done with ultrasonic sensors. When a car is detected in front of the parking lot, the ultrasonic sensor reads and refreshes the number of cars on the entrance display and on the smart phone application. The accuracy suffers in case of golf carts or people because the sensors consider them as cars. Unlike them, our solution of entering and leaving is based on image recognition of license plate number.

The Indian researchers from Pune [14] implemented an intelligent reservation parking system using QR code technology and a camera that reads at parking entrance the QR code and checks its correspondence in database. In our case, the system includes newer and different hardware components than those used in [14] and besides that we implemented image recognition for license plate instead of QR technology, we also use image processing for updating the web application according to cars presence or not from the parking.

Finding vacant parking slot by the drivers becomes an NP-hard problem with various constraints. In [15] a genetic algorithm is used for solving scheduling problems and finding the best parking region into shopping malls and the slot to park the vehicle without affecting medium waiting time. The kernel of their system is a number of autonomous trolleys. These are used to take the vehicles from the entrance to best parking slot identified by the system. The customer can leave their vehicle at the entrance and enjoy the

shopping. The trolley captures the image of the surrounding after the vehicle is successfully parked in the slot. The image and location map is send to the customer mobile application. The main disadvantage is the cost of implementation and the necessity that the customer to have a WiFi connection.

In [16] the author reviews and presents practical examples based on smart technologies for guiding drivers to parking lot in urban areas. There are four categories (from simple to complex) depending on the equipment used and the services provided by parking lot: typical, semi-automatic, automatic and multi-storey. The simplest (typical) involves no equipment and services, the semi-automatic allows payment at the cashier and the operator who operates the barriers, the automatic is fully equipped with an automatic parking equipment and payment on terminals. The disadvantage of first three categories is that these parking systems are not providing information about the occupancy of car parks (free parking spaces) except for the direction and possibly the distance to the destination. The most complex, multi-storey car parks are fully automatic and equipped with smart technologies. Unfortunately, from the administrator's viewpoint their purchasing and correct implementation is costly and from the driver's point of view, the disadvantages are their reliability and detection accuracy (most of the time different of 100%).

Smart parking alternative solutions based on magnetic sensors and IoT systems are revealed by engineering companies. Magnetic sensors, placed on the side of road or parking spot, on the road surface, or in the ground in the form of Inductive-loop detectors, can be used to check the occupancy of the parking lot, determining the number of free spaces left every time a car passes over them and disrupts the magnetic field [17]. The major disadvantages of this solution are the high installation cost and time and require civil engineering.

In [18] are briefly shown examples from Siemens and Huawei IoT-integrated parking management systems. These provide drivers information about the availability of parking lots and connect with municipalities decision factors. The IoT system consists in environmental sensors, a cloud-based web platform enabling also the automated parking payment. Our solution is very close to this system but furthermore promote the idea of parking sharing by private institutions.

In the European project proposal "Unlock the sharing economy for the parking sector" [19] the authors promote the parking sharing concept proposing integration of parking sharing platform into connected and autonomous transportation systems in smart cities. Based on mathematical model and simulation the authors tests how motivated are the drivers to use the platform which includes private parking shared for public. Furthermore they reveal the need for collaboration between academic and industrial partners as co-developers and city councils and local communities as decision makers and private companies as parking sharers.

3. The smart parking solution

This work aims to create an embedded system that solves the parking problem described above. The main software component is a web application which is connected with Raspberry Pi for parking management: reserving a slot and authorizing the entering and exiting and revising the number of vacant slots, real time

detection of each empty parking space based on Motion Detection algorithm and synchronization with the web application and the supporting database. Of course the technical implementation is complementary to the sharing economy concept that in order to be efficiently applied requires appropriate understanding the parking occupancy patterns of each private institution. The project development stages are:

- Implementation of hardware-software application for identifying car license plate number and real time detection of each parking space (to determine if they are vacant or not)
- Development of the web application
- Real time synchronization with the web application and database

The web application provides the following facilities:

- Real time parking status view
- Reservation on a specific time by license plate number
- Contact parking management by email
- Administration module that includes updates about users and prices
- Implementation of the gamification concept in the management of parking spaces.

In order to make drivers estimate an exact time at which their car will be parked on that place, is introduced the concept of "Gamification". In this case, this concept aims at reserving parking spaces for as accurate a period of time as possible, in order to better manage vacant spaces and in addition to this, it also has the purpose of reducing fuel, driving time and streamlining traffic at peak hours. This is done using the web application and encourages drivers to make as accurate estimation of time as possible by rewarding them with points that can be used later for parking payment.

Scalability issues and proposal for solving are treated in [1].

3.1. The use case diagrams

In Figure 3 is presented the possible use cases of the application: the involved actors and the actions performed. There are three types of actors: the administrator, a previously enrolled user and the new user. The administrator manages the parking lots and their status; adds new locations and parks; manages new users; allows sign-in of new users; allows and control making the new reservations and set the price for parking. A registered user may see the parking status, may reserve a parking slot into a location, and obviously the sign-in process. The reservation process supposes: choosing the date and time for access, the parking location and slot and providing the license plate. The new user can see the parking status, the contact information and the geographical area of parking location, and he may register.

3.2. Reservation

The smart parking system is reservations-based. Any registered user may reserve a parking slot in case exists at least one vacant. Reservation includes the parking name, parking space (selected by user), hourly rates, the estimated date and time of arrival and

departure. The parking spaces reservation is made from web application. This reservation can only be made if the user already has an account (as new user performed the registration process and then as registered user make the sign-in process).

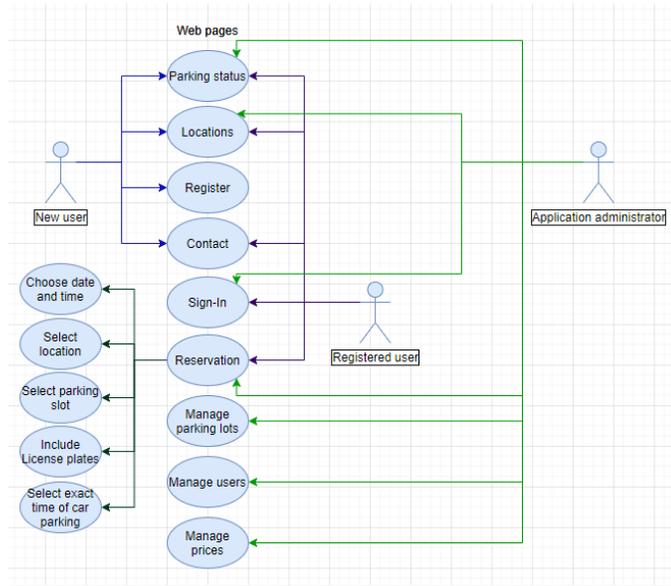


Figure 3. The application use case diagram

Figure 4. Reservation form

Extending the original work [1], in the Reservation page (see Figure 4) we added hourly payment implementation and the type of parking desired based on car's size (less than 5 meters, between 5-10 meters and higher that 10 meters). This constraint has been implemented to ensure an appropriate number of parking spaces according to the size of the car, otherwise there is a risk that the car is too large compared to the parking space and then two places must be invalidated even if only one car is entered. To change the type of parking is required to access the configuration page, and for each parking space the type is changed by the administrator. Payment implementation buttons have also been added using

PayPal technology, which currently perfectly simulates a real payment. This process requires bank protocols which are not finalized yet.

3.3. Real time parking status view

As Figure 5 illustrates, the parking degree of occupancy can view in real time by anyone, and, for the car parked, which are their license plate numbers. The busy spaces are drawn with Red color, and vacant spaces are drawn with Green color. In "Parking status" page can be seen in real time the occupancy of the parking. The parking slots are updated by reloading the page when a message is received from the Node-Red with the parameter "Access". All parking lots are made as dynamic buttons. All parking coordinates together with id and color are taken from database into web application and displayed as buttons of different sizes depending on the parking type. For each set of 4 points that represent a parking space has been made some calculations to determine the parking type. The types of parking can be as follow: normal parking, diagonally parking on the right, diagonally parking on the left and lateral parking (see Figure 5).

In addition to the original work [1], in the Parking Status page all the coordinates corresponding to each parking space are taken and it is determined for each place whether it is straight, lateral or diagonal. These places are displayed on the page according to the coordinates received in the database (x, y) and the corresponding color in real time.

The management of parking in the web application is done in the following way: when a new parking lot is added, the "start" image is taken from the moment the fixed camera is installed in the parking lot, it is sent to Raspberry Pi for processing (is checked if there are parking spaces in the image, how many, and possibly the two-dimensional coordinates x, y for each parking space). This image is added as a background on the real time parking view page and is colored according to coordinates which is occupied in red and the rest in green (similar to Figure 5).

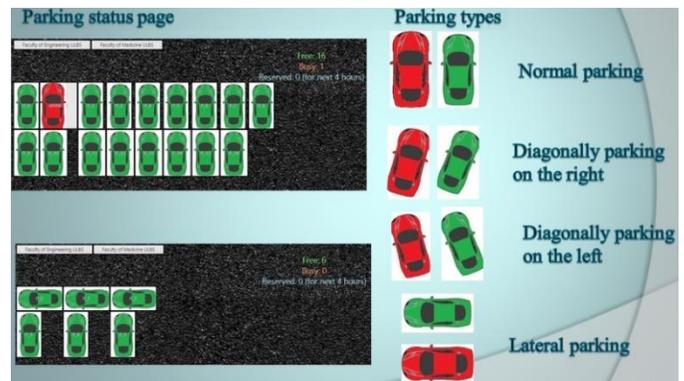


Figure 5. Real time status parking view

3.4. License plate recognition

At the entrance into the parking is checked the car license plate number. When reach in the front of the barrier, the driver push <I arrive> button from "Home" page to send instruction to camera for capturing and checking license plate number. If the drivers have a reservation that is +/- 15 minutes around estimated arrived date and time he has access into the parking. Otherwise, it is

checked if exist vacant slots, that are not reserved in the next 4 hours. In the affirmative case the car is authorize to enter in the parking space, otherwise not.

4. The application architecture, tools and programming environment used

This embedded system consists in the following main components:

- Hardware component which contains a Raspberry Pi Model 3B with a two Pi cameras (one for entering and one for exiting from parking lot), a Hikvision camera from the category “Bullet Series Thermal”, stepper motor and some LEDs. Raspberry Pi is used together with the library that includes License plate recognition algorithm (OpenALPR algorithm from OpenCV library) that checks the number of license plate with the previously made reservation and opening the gate, parking lot and occupancy detection algorithm, Pi camera used for taking picture of each car arrived on parking entrance to apply license plate recognition algorithm, python script to manage barrier using a stepper motor and providing rotation angles, and another scripts that commands LEDs lighting for semaphore simulation. The Hikvision camera is used for parking monitoring and providing inputs for Motion Detection algorithm.
- ASP .NET web application developed using DevExpress controls. Website structure it consists of several modules who communicate with each other for a better functionality of application. MQTT (Message Queuing Telemetry Transport) and Node-Red are used for communication between web application and Raspberry Pi. The main web components are pre-setting the parking configuration based on the architecture of the location, user registration, parking slot reservation, gamification concept, real time parking visualization with counter for each vacant, occupied and reserved (for next two hours) parking slots, administration module to manage users, parking and prices for each zone.
- MySQL database. The database is essential and structured in such a way as to allow a much better flexibility, efficiency and dynamism so that the web application can be applied much more easily from one parking to another. The MySQL database contains few tables with foreign keys and records in database all reservations, user information, prices, history and the number of free parking spaces.

In Figure 6 is presented the application architecture together with all project stages from online reservation to entering car into a parking space. The first step is Motion Detection algorithm (1) in order to set the parking configurations and saving the coordinates in database (2). When the camera is installed and fixed, the parking spaces in an image captured in real time are delimited with the help of a Python script. The delimited places are saved in the database together with the coordinates and the id corresponding to each place, and the initial colour is green (free places). The Motion Detection algorithm is in real time applied and used for updating the colour corresponding to each place (red - occupied, green - free) in the database and web application. Then, it follows the online reservation (3) and inserting a record in database. When the driver arrives in front of the barrier, he has to push the <I

arrive> button from Home page. After all this, the Node-Red is sending using MQTT, messages and instructions from/to application. In step (4), after pushing the <I arrive> button the web application through Nod-Red is sending to Raspberry Pi the instruction to take a picture. Then the checking access in parking is start calling OpenALPR algorithm (5) which provide the license plate number (6) to web application. So, if the car that arrives in the front of barrier has a valid reservation (the license plate number is successfully found in database (7) and the result is sent to web application (8)) then the latest grants the permission to enter in the parking space (9) and send this message through Node-Red to Raspberry Pi, the green led is turning on (10), and the barrier opens (11). Otherwise, the red led is turning on. After verification of license plate number, web application sends instructions to turn on / off the lights and open the barrier or not. While all this checks are made, yellow light is on. Finally, if the access is granted, green light is turning on, and the barrier opens. Otherwise, red light is turning on, and the barrier does not open.

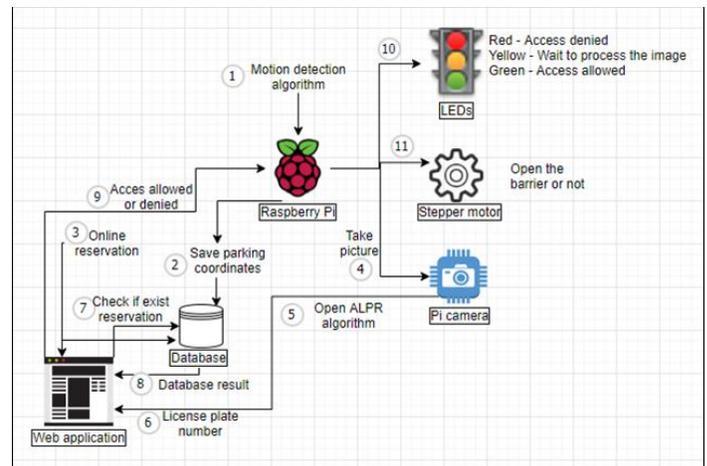


Figure 6. Application hardware-software architecture

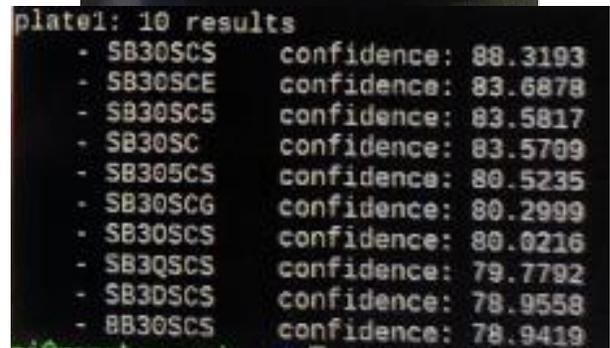


Figure 7. ALPR algorithm (up - image source, down - the result)

For updating the web application and database with the correct number of slots when the car leaves the parking is used the second Pi camera. So, if a car intends to exit from parking the same license plate recognition algorithm is applied. In the web application we check if that license plate number is inside the parking or not. In the affirmative case, is change accordingly the parking slots and opens the barrier and turn on the green led. The contrary case means that the car intends to enter in the parking. Ultrasonic sensors could be alternative solution for detecting the presence or absence of the car. When the car leaves, sensors detect this thing and mark this thing in the database. We opted in this project to use the second Pi camera instead of sensors from simplicity reason.

4.1. Software tools

1) OpenALPR: is an open source library written in C++, Java, Node.js, and Python that analyses video images and streams to identify the license plate numbers [20]. The Computer Vision algorithm is in Python implemented. The output returned is the text represented by the characters of the license plate (see Figure 7).

OpenALPR return license plate number with the higher confidence from ten possible results. The algorithm can be configured to work both with European and American license plate numbers. This library is portable and can be compiled and run on Linux, Mac and Windows. OpenALPR needs the next additional libraries:

- Tesseract OCR: it is one of the most accurate optical character recognition (OCR) engine for various operating systems.
- OpenCV: it is a Computer Vision library enhanced with machine learning algorithms.
- Leptonica: contains software dedicated for image processing and analysis applications. The license plate region detector uses the Local Binary Pattern (LBP) algorithm [21]. LBP is a simple but effective texture operator that labels the pixels of an image by comparing it with neighboring pixels, and considers the result to be a binary number. The advantage of LBP operator consists in its resistance to changes in the level of gray generated mainly by the variation of light.

2) MySQL: is used for database. In Figure 8 can be viewed the database architecture, the structure of the tables, the primary keys of each table and their links to other tables as foreign keys, which allow adjacent selections from several tables (joins), but also allow faster retrieval data in this way.

The Users table contains information about users and their type such as “User”, “Visitor” or “Admin”. The Reservations table contains all reservations made using the online reservation such as parking zone, car size, parking spot where driver should park, time of arrival and departure, license plate number, account that need to be paid before, total price and user id. The History table keeps the temporary information retaining only the vehicles that has entered in the parking and are still there. Here are stored the exactly time of arrival and departure of each car that had access to parking. Price table stores the hourly rate for every parking and ratio for account payment for each parking zone and each parking type. The Parkingzone table stored the name of every parking zone (for the moment “Faculty of Engineering ULBS”, “Faculty of Medicine ULBS”) and in Parking_types table are stored the car size types

such as “Less than 5 meters”, “Between 5 and 10 meters” and “Higher than 10 meters”. In table Parking_coordinates are stored information about each parking slot. Every row from this table contains a set of 4 points which determines a parking slot, id from image file and colour in real time (red or green depending if parking slot is vacant or not).

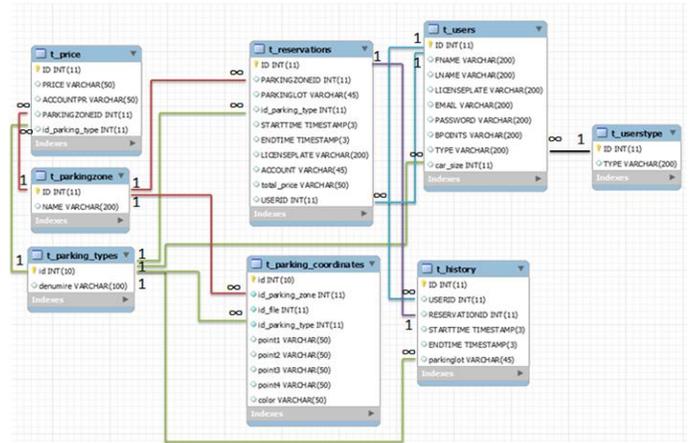


Figure 8. Database architecture

3) Node-Red: represents a programming tool used in general for IoT programming that connects hardware components with API's and online services in new and interesting ways. To connect the Raspberry Pi, the Pi camera, the LEDs and the stepper motor of the web application is used Node-Red programming tool.

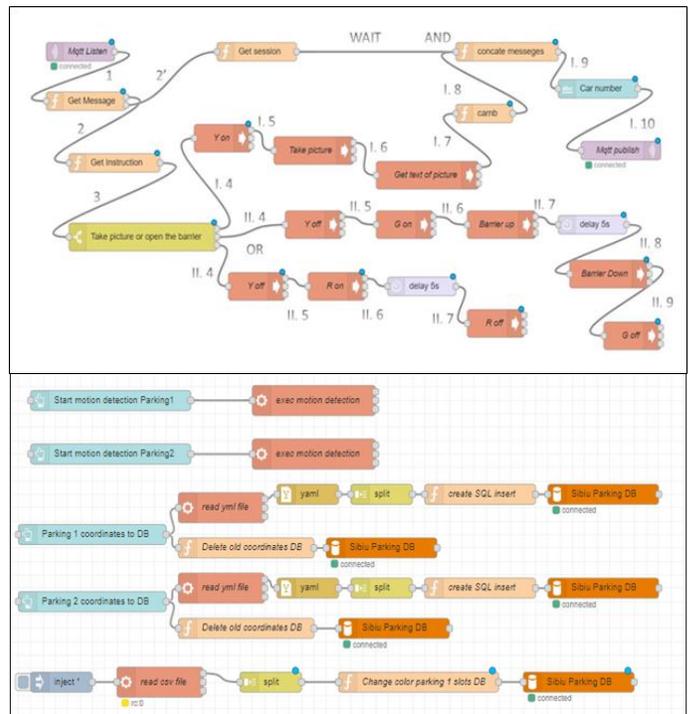


Figure 9. Implementation in Nod-Red of parking access (up) and Motion Detection algorithm (down)

In the upper side of Figure 9 is presented the whole flow of Node-Red. The purple nodes are MQTT nodes. “MQTT in” is waiting for messages from web application while “MQTT out” is sending messages to web application. The nodes with an “f” are functions, the yellow/green node is a switch node, the nodes with

an arrow are execution nodes, delay nodes represent a delay of 5 seconds, and the blue node represents a text output that displays the final message that has to be sent. When a message arrives, using function nodes we get the session and instruction from message. The session is used to know from which web session had been called the instruction, to send back the message only to that session. If instruction is “new Car” we go on the first value of switch. This value generates the execution on commands like turn yellow led on, take a picture, get the license plate number from picture, concatenate with session and send message. The second value of switch determine the fact that the car has access in parking and send the commands to turn off yellow led, turn on the green one, open the barrier, wait 5 seconds, close the barrier and turn off the green led. Last value of switch represents the fact that the car doesn't have access and turns off yellow led, turn on red led, wait 5 second and turn off the red led.

In the down side of Figure 9 are the scripts corresponding to saving coordinates for each parking slot from real time image captured with Hikvision camera when this is installed for the first time; Command to apply Motion Detection algorithm and saving real-time colors to database at each 1 second.

4) DevExpress: provides various controls for ASP .NET web applications. The web application was splited into several modules with different functionalities combined to achieve the goal, and easy understanding from the user perspective. There are implemented the 3 menu types (for Visitor, Users and Administrator) according to the application use case diagram (see Figure 3). The Register form (see Figure 10) is completely made from DevExpress controls. For the web application in order to have similar format for all pages as possible (the only part what differs in pages is the content) a page called “master” is used, in which CSS or JavaScript files can be imported, elements can be added that are desired in the future in all the pages created later, such as menu, footnote, or any other elements whose presence is desired in more than one page. This page is responsible for the layout of the application.

Figure 10. Register form

5) MQTT is a machine-to-machine (M2M) connectivity protocol. The messages are sent and received using an online MQTT broker. The Node-Red receives the message using the MQTT in node. The implementation of MQTT in the web application is done by installing an M2Mqtt service, which uses a PAHO client from Eclipse, the programming being done in JavaScript.

5. Finding the vacant parking slot using Image Processing methods

Finding vacant spaces can be a challenge for large parking lots and especially at rush hours. The technical solution proposed for implementing a smart parking system in this work is based on Image Processing. Different methods was tackled in order to set the best one. The Hikvision camera is positioned on a considerable height (3 meters above the ground), so that it could see the entire parking lot.

In the following are provided brief details about the methods tried and will be discussed in detail the most accurate that is used in the implementation of the smart parking system.

5.1. Using the Hough Transform

First was tried the detection of the parking lot grid using Hough Transform (dedicated to feature extraction and object detection) and Hough Probabilistic for lines [22]. The potential challenge referred to the fact that the parking lot grid is not clear/visible enough from the camera position in case of fog/snow/rain/old paint on pavement/fallen leaves. The approached solutions were:

- i) Manual parking lot grid initialization
 - Reading the image
 - Convert image to grayscale to reduce the info in the photo
 - Apply Gaussian Blur to remove even more unnecessary noise
 - Detect the edges with Canny detector
 - Detect lines with Hough Transform

ii) Machine-Learning algorithm for detecting the grid (fed with multiple photos from different days/weeks/months) which generates as output the parking slots.

On web application, first was tried the rendering of background image every 1 second (and after every 5 seconds) with real time picture from camera. The rendering takes too long. We tried all this on a separated asynchronous thread, but with no success. After that we were thinking about drawing each set of points on a background image from web application. This was a good idea, but wasn't good looking. The main drawback was that other lines appear besides the desired ones and the hard part was that we need to identify the useful lines and remove the rest. We tried to selectively analyze only of the useful part and not of the whole image. Unfortunately the results was not satisfactory. These are presented in the 6.1 section of the paper.

5.2. Car detection via static thermovision camera

The Hikvision camera allows acquiring images both from infrared and visible specter. Thus, the second method aims car detection via static thermovision camera by tracking the engine of the car

(considering that it will dissipate the highest amount of heat). In the database is saved the slot with highest temperature produced by the engine of a new car that is coming from outside the parking. We update as an occupied slot, in the database, and in web application, whenever a new hotspot reaches to a previously vacant slot. A challenge that we faced was finding the correct threshold for the heat level. This is because in the sunny days even the pavement/asphalt can reach the temperature of running engine. To overcome this, we ignore not moving heat spots. This method could be the best for using at night consumes moderate computational power but provides moderate accuracy. The main disadvantage is the cost of camera which is relatively high.

However, were made without success, temperature comparisons relative to the cars that enter the parking lot, against to those that stay and have the engine “cold” or “colder” according to the previous idea that different temperature of the cars represents a concrete indicator regarding a vacant or occupied place (see Figure 13).

5.3. Using Motion Detection algorithm

Detecting a free parking slot based on car motion can be done in multiple ways:

- Car tracking via motion detection – from the moment the barrier is open, is known that only a car would move near the parking entrance and it can be track it up to a point where it parks. The location is saved in the database as occupied. The challenge in this case is that if the driver, on his way to a slot, will stop too much time on the road the tracking will fail (a threshold variable for the motion detection has been reached). Thus, is need variable thresholding time for the tracking (tracking the car until it reaches exactly on a parking spot by calculate the distances towards the nearest available slot’s grid lines). It has low computational impact but may not be accurate enough and it does not work well during night
- The second way suppose no car detection or tracking, actually the motion is indirectly determined. It is just detected if a slot is occupied or not by calculating the average of pixels between 4 grid lines. An empty slot has to have a lot of darker pixels (empty slot case) than any other colored pixels (the case when car is parked). The points method (based of pixels between 4 grid lines) provides the best accuracy and is easy integrated in the web application. Basically, in the pre-processing stage (pre-setup), the image is analyzed and is configured the 2D map of the parking lot that is loaded in the web page. Subsequently, the stored information related to the spatial position of the existing parking spaces is kept in database. This motion detector algorithm is based on some coordinates given by user. Thus, user has to give position for each of the parking spots by clicking on the initial image. This process is made only once, when the camera is fixing (pre-setup stage). Each set of 4 points that made up a parking space for all of the spaces in the lot are stored in an *.yaml* file. Clicking edges of each parking slot is made using a mouse as a “paintbrush”. Then some calculations are made to find out the center of the rectangle (to label each space with a number, used later on web application). After drawing the rectangles, all there was left to do was examine the area of each rectangle to see if there was a car in or not. By taking each rectangle (grayed and

blurred), determining the area and doing an average on the pixels, we can tell that there was not a car in the spot if the average was high (more dark pixels) and the rectangle color is changing. One challenge that could arise is if car has a close color to the pavement. In this case, even if a car has the same color as the slot when it is empty, it will still reflect light from the windshields so will result different average color of pixels than an empty slot.

The most benefits of this method are that provide high accuracy and requires moderate to low computational power. By this indirect Motion Detection algorithm is known exactly that a new car which entered in the parking lot cover a previously empty place (however it is not 100% sure that is the reserved place, it would have been useful to combine the algorithm with the above solution that follows the car from the barrier to the parking slot). The introduction of the type of car is necessary and efficient. It allows the display of parking on the web page in a way as close as possible to reality and the parking spaces are updated in real time depending on what colors the Motion Detection algorithm returns for each parking slot (see Figure 14).

Next, we summarize the strengths and the weakness of car detection methods:

Table 1: Algorithms Summary

| Algorithm | Advantages | Disadvantages |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hough Transform | <ul style="list-style-type: none"> • Handle missing data • Express a high level of detail | <ul style="list-style-type: none"> • Due to the heterogeneity of the objects in the image, the algorithm produces as a result lines that are not part of the parking lot (poles, houses, etc.). • Sensitive to fog / snow / rain / fallen leaves, etc. |
| via static thermovision camera | <ul style="list-style-type: none"> • May be successfully used at night • Consumes moderate computational power | <ul style="list-style-type: none"> • Moderate accuracy • Relative high costs |
| Motion Detection | <ul style="list-style-type: none"> • High accuracy • Low computational power | <ul style="list-style-type: none"> • Limited by the relation between cars color and pavement |

6. Comparative results between algorithms

6.1. Results after using Hough Transform for detection of the parking lot grid

In Figure 11 and Figure 12 are presented the original images (two parking lots) and the results respectively after detecting the lines in parking using Hough Transform. OpenCV encapsulates the math of the Hough Transform into HoughLines(). This approach was tested on different videos and real time images but without success.

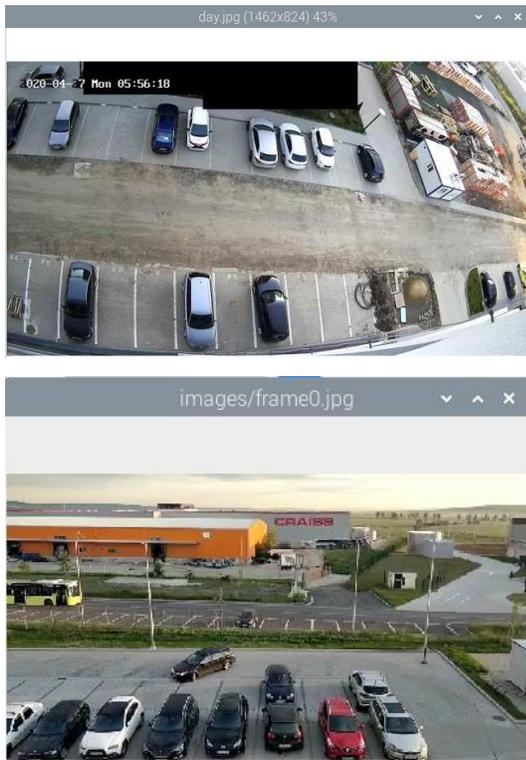


Figure 11. Original images

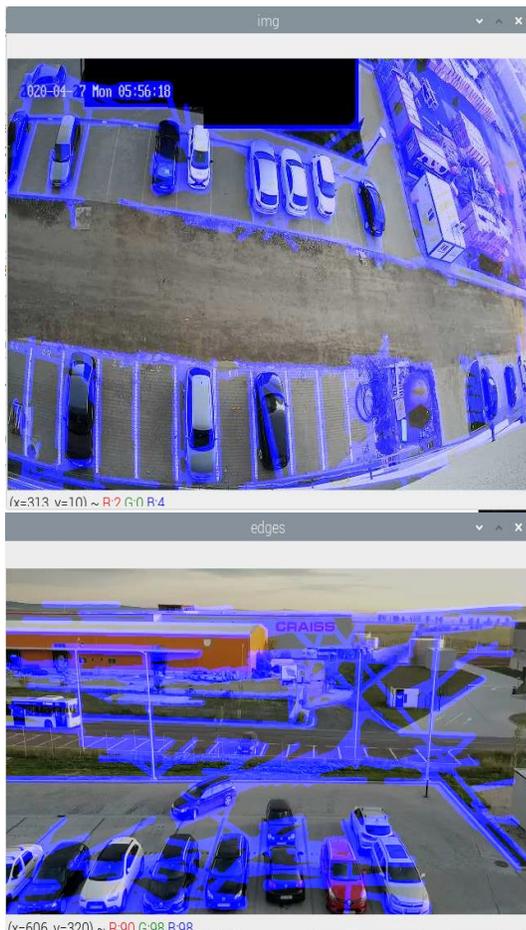


Figure 12. Final result

6.2. Results from thermovision camera

One explanation for the disappointing results may be that the car body insulates the engine from the outside and its high temperature is not captured by the thermovision camera. Also, heated asphalt produces a higher temperature than cars. The Figure 13 illustrates the image of the parking lot captured with the camera both in the infrared and in the visible spectrum. In the Figure 14 it can be seen how the driver getting out of the car has the highest temperature in the image and not the engine. Unfortunately, the method did not help us to accurately identify free parking spaces.

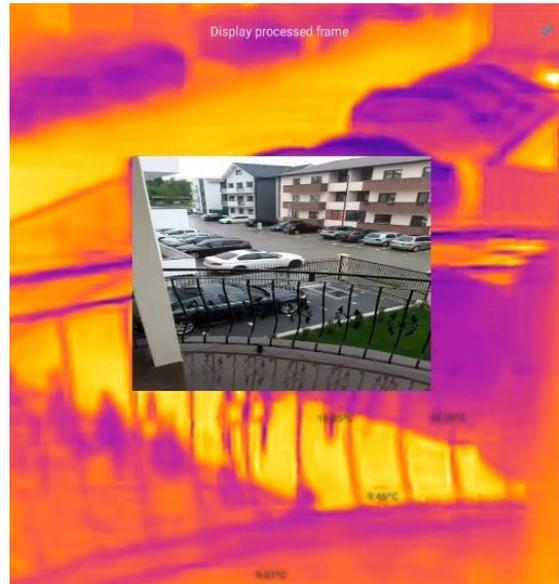


Figure 13. The parking lot seen in both spectrum: infrared and visible



Figure 14. Driver exiting from the car parked seen with thermovision camera

6.3. Results after using Motion Detection algorithm

Looking both to Figure 15 and to Figure 5 it can be seen that the indirect Motion Detection algorithm helps to create a more real 2D structure of the parking lot and using it we can determine the occupancy of the parking space (to know which slots are vacant or not). In the upper side of the Figure 15 is capture the pre-setup stage when are identified the slots (with Blue colour) of the parking, in the middle it can be seen a scenario with two empty spaces and the down side image shows the previous places that are now occupied. It should be mentioned that for a better exemplifying, only the bottom of the test image was treated (since the pre-setup phase) without taking into account the cars and the places from the top of each of the three images. Once the algorithm is implemented, the degree of parking occupancy can be accurately determined at any given time.



Figure 15. Motion Detection algorithm results

7. Conclusions and further work

This paper presents a smart parking embedded system with focus on image processing algorithms as a solution for smart mobility. We implemented and comparatively presented the results of various image processing algorithms like Hough Transform, with thermovision camera and based on motion detection for real time tracking of vacant parking slots or OpenALPR for identifying

car license plate number. The accuracy of the identification system of license plate numbers depends on the quality of the camera used. For example, in the first tests we noticed that for the number SB11MBU the number 11 was not read, the result being SBMBU, probably due to the confusion with the letter 'i' or 'l'. Another problem would be the brightness; the 5MP resolution Pi camera is not efficient enough to take clear pictures when the brightness is low. Anyway, with this cost-effective Pi camera we perfectly detected 188 from 204 cars meaning 92.15% accuracy (a reasonable performance).

The main software component of the system is the web application that implements the payment system with different rates depending on the number of hours or days of parking and integrates the gamification concept rewarding the driver with bonuses if he exactly respects the parking period for which he made the reservation. When booking a parking slot, an account is paid to make sure that the slot is available on arrival thus avoiding the queue for payment and the need to change money.

This work brings social, economic and scientific arguments to promote innovation in smart parking and especially promoting the parking sharing concept. Even if we developed a prototype tested at LBUS Sibiu (a town with around two hundred thousand inhabitants) we appreciate that the smart parking system proposed is functional, scalable [1] at least to moderate sized cities and extendable to many private institutions which might share their excessive parking places according to the daily occupancy patterns of their employees. Excess parking spaces of the parking lots of different faculties / universities, but also of private companies can be shared to users in need! If a power outage occurs or the barrier does not work, to avoid creating a long queue at the entrance that could block traffic then the implemented technical solution involves the automatic lifting of the barrier and allowing free entry and exit (without scanning the license plate number) of cars depending on the remaining available space. Even in these conditions, the Motion Detection algorithm and the takeover of the occupied spaces in the web application works, announcing the exact number of free spaces in the parking lot.

There are at least two work directions we will consider in the near future:

- The web application should provide an indication of what would be the best place to park depending on the size of the car based on genetic algorithms.
- With the help of Google Earth or Google Street applications, we will take over the spatial arrangement of the parking lots and we will generate the parking architecture in the web page (see Figure 5), thus offering a greater flexibility for application.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

This work was partially developed under the ERASMUS+ KA2 project "THE FOF-DESIGNER: DIGITAL DESIGN SKILLS FOR FACTORIES OF THE FUTURE", financing contract no.

2018-2553 / 001-001, project number 601089-EPP-1-2018-1-RO-EPPKA2-KA, web: <https://www.digifof.eu/>

[22] J. Illingworth, J. Kittler, "A survey of the Hough transform", *Computer vision, graphics, and image processing*, **44**(1), 87-116, 1988, [https://doi.org/10.1016/S0734-189X\(88\)80033-1](https://doi.org/10.1016/S0734-189X(88)80033-1)

References

- [1] S. D. Marcu, A. Florea, "Smart parking system-another way of sharing economy provided by private institutions". In 2018 IEEE Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, 2018. <https://doi.org/10.1109/ICDIM.2018.8846986>
- [2] E. J. Tomaszewska, A. Florea, "Urban smart mobility in the scientific literature - bibliometric analysis", *Engineering Management in Production and Services*, **10**(2), 41-55, 2018, <https://doi.org/10.2478/emj-2018-0010>
- [3] A. Florea, L. Berntzen, M. R. Johannessen, D. Stoica, I.S. Naicu, & V. Cazan, "Low Cost Mobile Embedded System for Air Quality Monitoring", In Proceedings of the Sixth International Conference on Smart Cities, Systems, Devices and Technologies (SMART), Venice, Italy (25-29), 2017.
- [4] N. Zhao, & Z. Li, "Optimize traffic police arrangement in easy congested area based on improved particle swarm optimization". *Procedia-Social and Behavioral Sciences*, **138**, 408-417, 2014. <https://doi.org/10.1016/j.sbspro.2014.07.219>
- [5] G. Gowda, & R. Thenambigai, "A Study on Respiratory Morbidities and Pulmonary Functions among Traffic Policemen in Bengaluru City". *Indian journal of community medicine: official publication of Indian Association of Preventive & Social Medicine*, **45**(1), 23-26, 2020. https://doi.org/10.4103/ijcm.IJCM_102_19
- [6] Y. Geng, C. G. Cassandras, "A new smart parking system infrastructure and implementation". *Procedia - Social and Behavioral Sciences*, **54**, 1278-1287, 2012, <https://doi.org/10.1016/j.sbspro.2012.09.842>
- [7] K. McCoy, "Drivers spend an average of 17 hours a year searching for parking spots", *USA TODAY*, December 15, 2019, <https://eu.usatoday.com/story/money/2017/07/12/parking-pain-causes-financial-and-personal-strain/467637001/>, last accessed 25.08.2020.
- [8] B. Chan, "Smart parking is innovative, but what it enables is even more innovative", *Digital Transformation, Innovation Management, Internet of Things, IoT, Smart City*, September 21, 2017, <https://strategyofthings.io/smart-parking-innovation>, last accessed 25.08.2020.
- [9] I.T. Cervera, "Internet of Things & startups: state of the art and emerging trends", Master's thesis, Universitat Politècnica de Catalunya, 2020.
- [10] J. A. Oliveira-Lima, R. Morais, J.F. Martins, A. Florea, C. Lima, "Load forecast on intelligent buildings based on temporary occupancy monitoring", *Energy and Buildings*, **116**, 512-521, 2016, <https://doi.org/10.1016/j.enbuild.2016.01.028>
- [11] H. Wang, "A reservation-based Smart Parking System", Ph.D Thesis, University of Nebraska-Lincoln, 2011.
- [12] International Telecommunication Union ITU-T X.200, "Information technology – Open Systems Interconnection – Basic Reference Model: The basic model", July 1994.
- [13] O. Dokur, "Embedded System Design of a Real-time Parking Guidance System", MSc Thesis, University of South Florida, October 2015.
- [14] F. I. Shaikh, P. N. Jadhav, S. P. Bandarkar, O. P. Kulkarni, N. B. Shardoor, "Smart Parking System Based on Embedded System and Sensor Network", *International Journal of Computer Applications in Technology* **140**(12):45-51, 2016, <https://doi.org/10.5120/ijca2016909532>
- [15] D. Thomas, B. C. Kovoor, "A genetic algorithm approach to autonomous smart vehicle parking system", *Procedia Computer Science*, 125, 68-76. 2018, <https://doi.org/10.1016/j.procs.2017.12.011>
- [16] J. Hanzl, "Parking Information Guidance Systems and Smart Technologies Application Used in Urban Areas and Multi-storey Car Parks", *Transportation Research Procedia*, **44**, 361-368, 2020, <https://doi.org/10.1016/j.trpro.2020.02.030>
- [17] M. Pietruch, "Innovative Technology for Quick and Easy Smart City Parking Solutions", *Comarch Telecommunications Blog*, May 08 2019, <https://www.comarch.com/telecommunications/blog/innovative-technology-for-quick-and-easy-smart-city-parking-solutions/>, last accessed 25.08.2020.
- [18] Digiteum Team, "Parking IoT Technologies: About Smart Parking Systems", *Smart Parking Solutions*, November 27, 2019, <https://www.digiteum.com/iot-smart-parking-solutions>, last accessed 25.08.2020.
- [19] W. Liu; H. Haitao, "Unlock the sharing economy for the parking sector", *ETH Zurich Research Collection, European Transport Innovation Challenge: Creating Mobility Solutions*, 2017, <https://doi.org/10.3929/ethz-b-000171167>
- [20] <https://github.com/openalpr/openalpr>, Retrieved at 25.08.2020.
- [21] D. C. He, L. Wang, "Texture Unit, Texture Spectrum, and Texture Analysis", *IEEE Transactions on Geoscience and Remote Sensing*, **28**, 509-512, 1990, <https://doi.org/10.1109/TGRS.1990.572934>