Proc. Of the 4th International E-Conference on Advances in Engineering, Technology and Management - ICETM 2021
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-191-7 DOI: 10.15224/978-1-63248-191-7-02

Video Based Real Time Application for Parking Spots Availability Management

Bogdan BURTEA, Camelia FLOREA, Ciprian IACOBESCU, Gabriel OLTEAN

***Abstract*— Nowadays, intelligent parking spot availability management is a must for smart cities, because nobody enjoys wasting time in traffic looking for a parking space. In this paper we propose a method to identify the free parking spots in a large parking lot by analyzing the information from the surveillance cameras. The classification result is used in a mobile and/or web application for guide the users to a specific free parking spot. The proposed solution is based on SVM (Support Vector Machines) for classification in two classes free/occupied and quantized DCT (Discrete Cosine Transform) for spots description. The experimental results show very good performance for real time parking management, with a rate over 92% of accuracy. The tests are done on four different large parking areas.**

***Keywords*— Smart Parking, Spots Availability, SVM, DCT, Real Time Management**

I. Introduction

The main purpose of the proposed and developed software described in this paper was the identification of an empty or occupied parking spot in a parking lot or on the side of the road. Nobody enjoys wasting time circling around the city to find a vacant spot. It is estimated that about 30 percent of the cars in traffic are looking for a free parking spot. Aside from wasting time and fuel, the frustration factor and the pollution are very important issues too.

A solution is to manage these parking spots and to share this information using a mobile or web application, so you can check the availability of a spot, in real time, when approaching the desired destination. Using the phone, it is easier because there are lots of navigations applications that can take you to a vacant parking spot in a parking area using GPS or a different guiding system.

In order to automatically detect the free parking spots, it can be done using different solutions [1-10], like: sensors, security cameras or systems with barriers and cameras. According to the specificity of the parking and the budget allocated for this, it can be chosen between either system. Sensors have a high cost of purchase; installation and the sensors must be replaced after a few years. The one with barriers and cameras cannot show the exact position of the available spot, this system being capable only of showing the number of available spots.

Bogdan BURTEA, Camelia FLOREA, Ciprian IACOBESCU, Gabriel OLTEAN

Faculty of Electronics, Telecommunications and Information Technology/ Technical University of Cluj-Napoca
Cluj-Napoca, Romania

The system based on cameras use mainly the existing infrastructure, meaning the surveillance cameras already installed on a parking for security reasons. This will reduce the installation cost, due to the fact that we already have on-site cameras, and a single camera can simultaneously monitor several parking slots.

After mapping the parking image with Google Maps the exact position of the parking spot can be obtained. In this way, the user can be guided to the exact position of the spot in the parking area via a Mobile Application. This scenario is used in our application.

We managed to get availability of a parking spot in real time by analyzing each video stream from cameras in the parking area. First step is to segment the scene so that we know the exact position of all parking spots. Then, each individual parking spot is considered and analyzed to determine if it is free or occupied by a car. For feature extraction we use quantized DCT (Discrete Cosine Transform) coefficients and for classification a machine learning algorithm, SVM (Support Vector Machine), which will decide if that spot is free or occupied. The last step is sending the response to an application which has the role of making the information available and visible to the drivers in real time. The proposed method has been applied on four different parking areas, and the results are good, with a rate of over 92% of accuracy, for real time processing.

The proposed subject is still in today's topics, considering both: 1) real time parking spots detection and management for smart cities [1-10] and 2) the DCT and SVM usage for data processing and classification [11-13].

Most of the papers from the literature discus of the sensor-based system to get the availability of the parking spots like the papers [1, 2, 3]. From our point of view, for an outside parking lot, it is difficult to install sensors on an existing parking because the owner needs to make modification to the parking spot. Another issue it is that every sensor battery has a lifespan around 3 to 5 years and needs to be changed.

A video management system provides security for the cars and people in the parking and a single camera can cover up to 50-60 parking spots for outside parking lots. For indoor parking lots a camera can cover up to 12 spots, depending on the camera type, bullet, or fisheye. In this case the costs for installing the enough cameras and the cost of sensor can be close.

However, there are some articles that use video streaming for parking spot availability management [6-8]. Many existing techniques using video cameras are fine-tuned to specific contexts and scenarios [7-8].

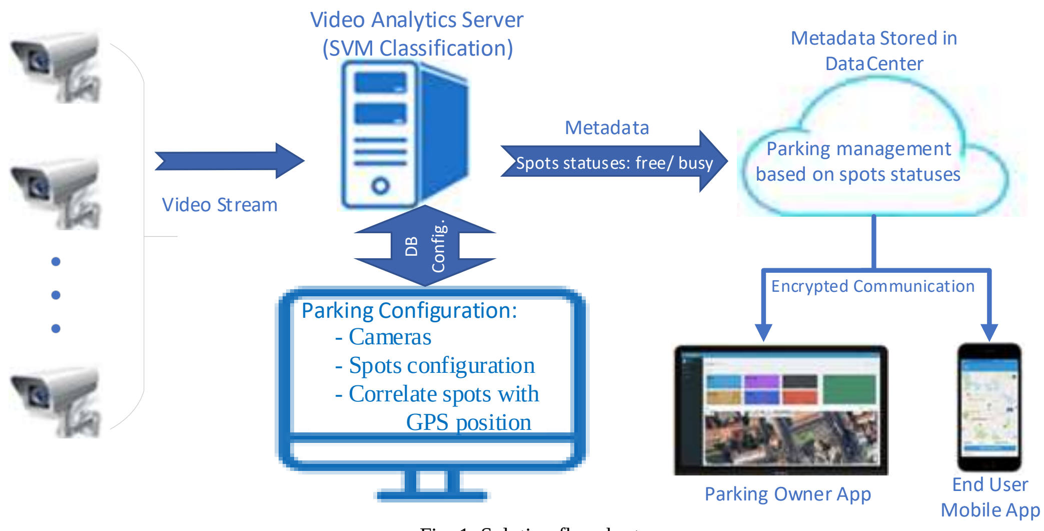


Fig. 1. Solution flowchart

Unfortunately, these techniques cannot be easily generalized, and even the adaptation of one solution to a different parking lot is not easy.

These solutions are mainly based on Convolutional Neural Network (CNN) classifiers. The results presented in [7] and [8] are similar with the ones that we have. We tested our method on CNRPark data set also used in articles [7-8] and we obtained an accuracy of 93%. They report a maximum accuracy of 90.70% on this data set.

An approach like [6] it is difficult because of camera angle and height (the images used here looks like satellite images). In this case the parking spots all have the same number of pixels and the same angle in the image. In real parking lots the spots have different angles, and the marking lines of the parking spots are not always visible.

II. Proposed Method for detection of available parking spots using DCT and SVM

The proposed solution, illustrated in Figure 1, considers the current infrastructure of the parking lots, as are the existing surveillance cameras. We applied the proposed scenario on for different parking lots in real time conditions, with the results illustrated on a web-based application for parking owner and a mobile application for end user, in real-time.

For each individual parking lot, there is needed a one-time configuration of the system. This implies creating an image map with the exact localization of all spots that are visible in the cameras and correlate them with their GPS position. After this configuration we have all individual parking slots (as a list with the slot region coordinates on the video frame from a specific camera). At the processing step, we consider each parking spot information – as an image patch.

Each parking spot, as an image patch, is numerical described as a feature vector in the DCT domain (from which several key features are extracted). These features are used for training and/or prediction of the patch classification (in our case, two classes: free or occupied) using SVM.

The feature extraction is the process of dimensionality reduction to represent an image as a compact feature vector. We realize a first dimensionality reduction by resizing the image patches of each slot at small size, but enough for a good prediction. For further dimensionality redaction we consider using the Discrete Cosine Transform (DCT).

Also, since the luminance component contains more texture/ edges relevant information comparing with chrominance component, we used the YCbCr color space representation. Therefore, we can apply different resize factor, and keep less information on chrominance component and more information from luminance component.

The prediction result is processed in clouds for parking management. The last step is sending the response to an application which has the role of making the information available and visible to the drivers in real time.

A. *DCT feature extraction component*

The DCT concentrates the spatial image content in relatively few coefficients, so we can treat images as sparse signals in the DCT domain. The first coefficient in the DCT coefficients matrix represents the DC coefficient (it represents the average intensity in a block of pixels), and the other DCT coefficients are the AC coefficients (which describe the variance of intensities changes in pixels within the same patch). The DCT is appropriate for sparse representation mainly of natural images, as we have the parking slots from video streams.

Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-191-7 DOI: 10.15224/978-1-63248-191-7-02

Due to the success of texture segmentation/ content description directly in the quantized DCT coefficients domain [11, 12, 13], it is still a state-of-the-art method for feature description.

Our feature extraction approach (detailed in Fig. 2) is using DCT and hyperbolic quantization for an image patch description in YCbCr color space.

Original patch slots have a different resolution, depending of its position and distance from camera. In the first step we resize each parking spot (image patch) at a fixed resolution N×N. We use N=30 for luminance component, and N=20 for chrominance component. On every slot we apply 2D-DCT and re-arrange the coefficients in a 1D vector form using Zig-Zag scan (see Fig.3). We truncate the data by keeping just the left corner of the DCT block (which contains the low frequencies, the most relevant information from the image patch represented in the frequencies domain). For the truncation step we used a shifted complement hyperboloid function, that allows us to keep just the up-left corner in the 2D DCT matrix, containing the relevant variations in the patch.

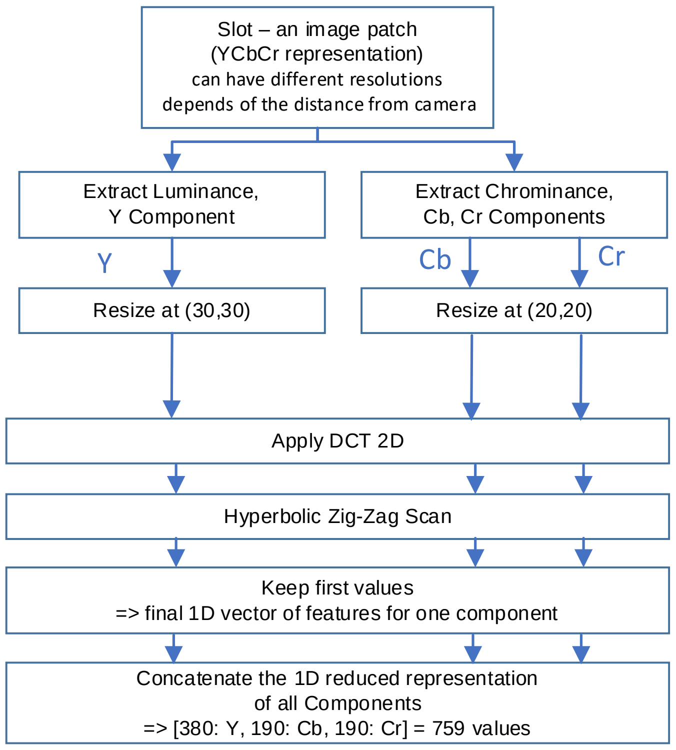
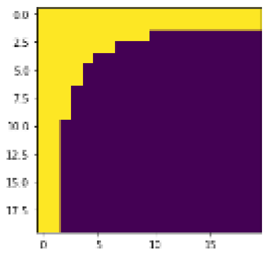


Fig. 2. Feature extraction flowchart



$$Q(i,j)= i\cdot j< K \qquad (1)$$

Fig. 3. Hyperbolic Zig-Zag Scan Matrix (left) obtained with the equation (1), K=20 (right). We keep just the DCT coefficients form yellow region.

If applying DCT transform at large resolutions is time consuming, using it on small patches – as is our case – is very fast and efficient.

We standardize features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as: $z = (x - u) / s$, where u is the mean of the training samples, and s is the standard deviation of the training samples. Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set.

Therefore, the entire slot is represented by a set of data that is much smaller than the original number of pixels. The reduced feature space, as well as the reduced data set, makes computationally easier the implementation of rather complex classification procedures.

B. *Support Vector Machine Classification*

The Support Vector Machines (SVMs) are one of the most popular machine learning methods in classification and regression [14]. In its implicit form, an SVM is a binary classifier which provides the benefit of learning with high precision and recall from a relatively sparse set of training data. The classification problem addressed here is a binary classification of the patches in one of two classes: free or occupied, therefore the use of a binary SVM classifier for this task in the feature space described in the previous sub-section is a suitable and appealing solution.

After some preliminary tests for classification of data in free/busy classes, we chose the RBF kernel (C = 10, gamma= 0.001) as giving the best accuracy and generalization performance on a validation data set.

Still, adding parking spots patches from different spot positions (45/90 degrees parking spot orientation, or different cameras positions) was starting to lower the recognition rate. For this reason, we propose to group the parking spots depending on camera type (fisheye or bullet) and position on the parking spot (straight or oblique). The results improved because the position and the aspect of the objects where more similar.

C. *Motion detection*

Taking in consideration the fact that when a car is parking, it remains unmoved for a while, to reduce the computational power used in classification step we verify parking spots just if there is movement inside them. If there is no motion detected, we keep the last known label (occupied or free). If we detect motion, we reclassify the patch to check if the label is changed or not.

To skip the cases when the motion is determined by people (or shopping carts) passing in front of the car in the parking spot area, was settled a minimum movement percentage into the spot area (is it the parked car that is moving, or something

Proc. Of the 4th International E-Conference on Advances in Engineering, Technology and Management - ICETM 2021
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-191-7 DOI: 10.15224/978-1-63248-191-7-02

in front of it). The movement is also trigger by the sun entering and leaving the clouds. Also, the sun makes shadows

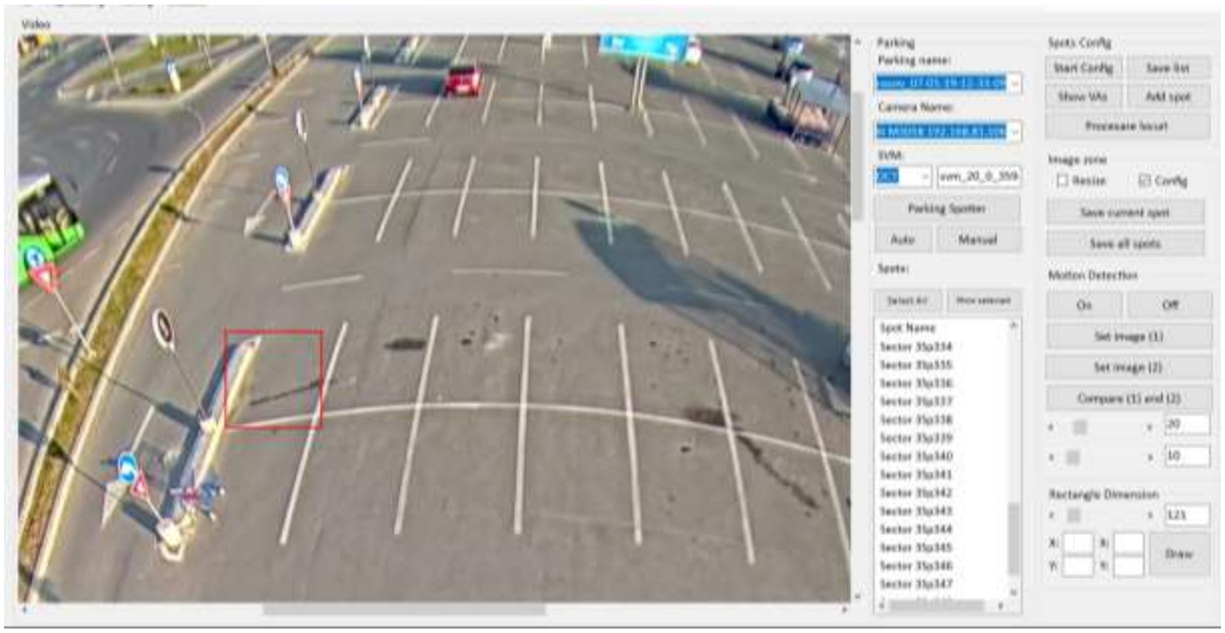


Fig. 4. The GUI and the illustration of drawing parking spots

in the shape of a car. Therefore, to improve the detection of the movement done by cars, we settled the threshold (at the comparison made between the current and last frame) so that small changes, like those made from sun light, doesn't impact the results.

III. Implementation and Results

Our application was made using VC# and EMGU CV for image processing and machine learning. VC# is an object-oriented programming language developed by Microsoft. EMGU CV is a cross platform .Net wrapper of the OpenCV image processing library.

The main application that analyzes the images and predict if each spot is free or occupied runs in a Windows Service application. This way the program starts in the same time as the Windows operating system and it's easy to manage from the Services window. Every error is logged in Windows Event Viewer with different levels of importance.

The first step, Configuration Tool, is a one-time step needed for each new parking area added to the system/ application. This step uses an VC# interactive GUI which helps the user to set all parking spots from each surveillance camera from parking area. For this purpose, it was extracted a single frame from the camera and displayed in GUI (see Fig. 4). Now, the user can draw rectangles on captured image frame to set each parking spot. The rectangles represent the patches where the video analysis is made (see Fig. 6 – at results we can see the blue rectangles for each spot considered).

Each parking spot has GPS coordinates for illustrate them on the map. The configuration is sent to cloud and the information is available for the mobile application and the web application.

A. *Training and classification*

For training and testing the SVM model we collected a number of 60.000 samples of patches (from parking spots) and labeled them as free or busy (the two classes used to train the model). We used 50.000 samples for training and 10.000 for testing and for validation. In Fig. 5 we can see some examples with the image patches extracted for the two classes. We considered frames from different times of day and different weather conditions.

Beside this dataset, we evaluated the application in real time: we extracted frames from times to time and checked the prediction performances.

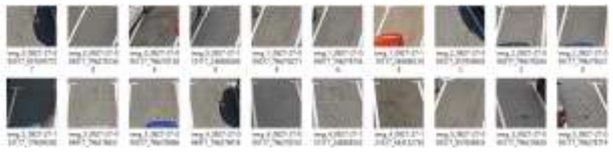


Fig. 5. Examples with the image patches extracted for the two classes: free and occupied

Even if, the accuracy on dataset was 94.5%, and real time 89%, we considered that we can improve the correct predictions by having multiple models. Therefore, we trained 3 models, depending on camera position and parking stile, obtaining the accuracy of 92% for real time scenario.

The system is supervised from time to time to see if there are errors and if there are, we can take the specific spots and retrain the model including that scenario to make the model more stable.

To reduce the computer's power consumption, we try to use classification step as rare as possible. Therefore, we analyze the parking spot only when there is movement in more than 50% of the parking spot rectangle, in this way we manage to exclude people passing by or shopping carts.

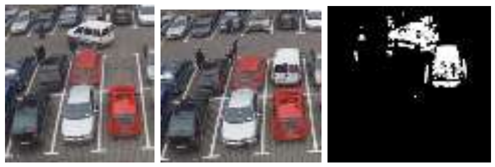


Fig. 6. Previous Frame/ Current Frame/ Motion Estimated

B. *The application structures*

The main application is connected to the cameras and from each stream, having the list of parking spots locations (see on GUI illustrated in Fig. 4), we extract all spots patches, and keep just this information (from inside the spots, represented in blue in Fig. 7). Each parking spot use the corresponding SVM model (from the 3 models trained) depending on camera type and position in the image.

We set the frame rate at 1 or 2 frames per second because the speed of the cars is low into the parking area and the status report from each second it's more than enough to see the live availability of the parking spots.

The status is encrypted and sent to the cloud application that is managing the availability for mobile and web applications. We avoid sending the same status by verifying with the previous status and checking if there was enough moving.

Another important aspect in managing the SVM models is the weather. A good SVM model for summer days can have bad performances for winter with snow days. A solution for this problem can be using local weather forecasts and changing the models if it is needed.

A camera problem (a hardware or sensor issue, communications problem, power outage or anything else) can cause a lot of issues regarding the live information for that parking. In some cases, the camera does not send any stream or can send a bad stream. When the camera doesn't send any stream, the application is sending an inactive status to inform everyone that there is no information for that parking spots. The bad stream can have multiple scenario: black or white screen which can be discovered by having the same color in the entire frame or a frame with an altered image which can be discovered by having very different results comparing with the average of the past days. All of these scenarios trigger alarms for the service, to check the cameras if there are any hardware or software issues.

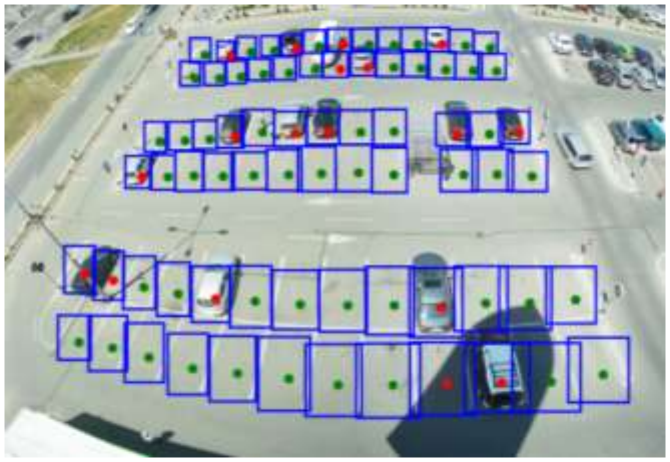


Fig. 7. A result when we do not classify with the optimal SVM model (there is an error on the second row from down/left)

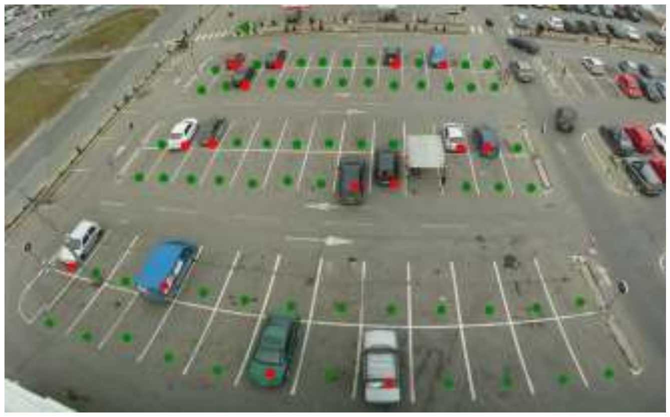


Fig. 8. A result when we use the optimal SVM model for this parking lot

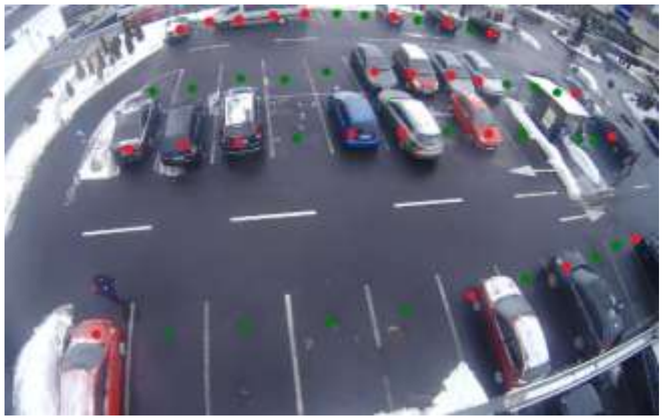


Fig. 9. A result when we use the optimal SVM model for this parking lot, in snow conditions.

C. *The Parking Spotter Application*

The mobile application (Fig. 10.) used by the drivers to navigate exactly to the free spot, named Parking Spotter, is functional as demo, and you can download it for free. Each parking is marked on the map and after selecting it, the map is populated with each free parking spot. Selecting a spot will open your default navigation application with the GPS coordinates and helps you to reach that spot.

The web application is used by the owner of the parking. It contains live information about availability of the parking slopts and different reports that can be used for marketing purposes.

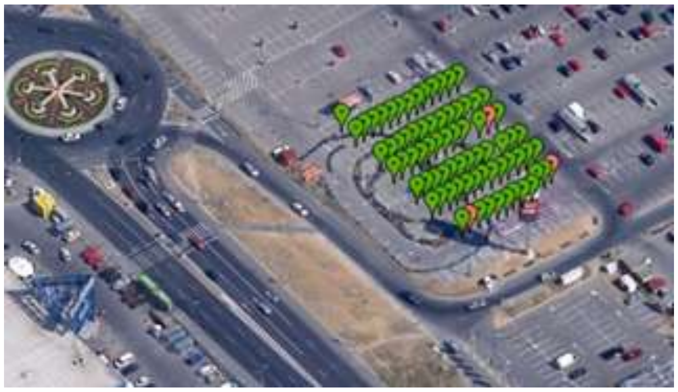


Fig. 10. Mobile/Web Application View

D. *Conclusions*

In order to summarize what has been detailed above we can say that, one of the most common use case for our solution is for the parking lots where, after analyzing the streams from the video cameras, it can recognize the spots that are free and the ones that are occupied and show them in real time in the mobile application. In addition, our AI algorithm is able to identify if a vehicle has been in the parking spot for a specific time span and we can raise an alarm to the parking owner, which will take the appropriate measures according to his parking policy.

Right now, our algorithm can make a binary classification (busy or free) using SVM from the images received from the video cameras. The images are preprocessed, and we extract some key features that we use in order to build three SVM models. The prediction and the ID of the spot are sent to a cloud platform where they are processed. This information is available to the drivers on a mobile app (available on Google Play and Apple Store) and to the parking owners on a dashboard where they can find different reports, heat map and live status of the parking.

During the next phase, we want to develop a Deep Learning based solution to use in our application, to try to improve the results even better.

Acknowledgment

The research leading to these results has received support from the Parking Spotter Solution from Cluj-Napoca, Romania (parking-spotter.com). We thank to Ovidiu CIURTE (product owner) and Mihai GHEORGHE, (product manager) for their support.

References

[1] Khaoula Hassoune ; Wafaa Dachry ; Fouad Moutaouakkil ; Hicham Medromi, “Smart parking systems: A survey”, 11th International Conference on Intelligent Systems: Theories and Applications (SITA), Mohammedia, Morocco, 2016.

[2] M. Alam, D. Moroni, G. Pieri, M. Tampucci, M. Gomes, J. Fonseca, J. Ferreira, G. R. Leone, “Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities”, Hindawi, Journal of Advanced Transportation, Article ID 1485652, 13 pages, 2018.

[3] Fadi Al-Turjmana, Arman Malekloo, “Smart parking in IoT-enabled cities: A survey”, Elsevier, Sustainable Cities and Society, Volume 49, August 2019, https://doi.org/10.1016/j.scs.2019.101608

[4] Harshitha Bura ; Nathan Lin ; Naveen Kumar ; Sangram Malekar ; Sushma Nagaraj ; Kaikai Liu “An Edge Based Smart Parking Solution Using Camera Networks and Deep Learning”

[5] Sepehr Valipour, Mennatullah Siam, Eleni Stroulia, Martin Jagersand, “Parking-Stall Vacancy Indicator System, Based on Deep Convolutional Neural Networks”

[6] Priya Dwivedi, “Find where to park in real time using OpenCV and Tensorflow” https://towardsdatascience.com/find-where-to-park-in-real-time-using-opencv-and-tensorflow-4307a4c3da03

[7] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, Claudio Vairo, “Deep learning for decentralized parking lot occupancy detection”, Expert Systems with Applications 72, 327-334, 2017 (http://cnrpark.it/)

[8] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Claudio Vairo, “Car parking occupancy detection using smart camera networks and deep learning”, IEEE Symposium on Computers and Communication (ISCC) 2016, 1212-1217.

[9] Noah Sieck, Cameron Calpin, Mohammad Almalag (2020), “Machine Vision Smart Parking Using Internet of Things (IoTs) In A Smart University”, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Austin, TX, USA.

[10] Lun-Chi Chen, Ruey-Kai Sheu, Wen-Yi Peng, Jyh-Horng Wu, Chien-Hao Tseng (2020), “Video-Based Parking Occupancy Detection for Smart Control System”, Applied Sciences, 2020, 10(3), 1079.

[11] D. F. Atrevi, D. Vivet, B. Emile, “Rare Events Detection and Localization In Crowded Scenes Based On Flow Signature”, 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, 19 December 2019.

[12] Maxence Bobin, Hamdi Amroun, Mehdi Boukallel, Margarita Anastassova, “Smart Cup to Monitor Stroke Patients Activities During Everyday Life”, iThings, 2018 IEEE 11th International Conference on Internet of Things., At Halifax, Canada, August 2018

[13] Yifan Zhang, Honglei An, Hongxu Ma, Qing Wei , “Human Activity Recognition with Discrete Cosine Transform in Lower Extremity Exoskeleton”, 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR), August 2018

[14] V. N. Vapnik, Statistical Learning Theory. New York: Wiley, 1998.