Park Smart

D. Di Mauro [1], M. Moltisanti [2], G. Patanè [2], S. Battiato [1], G. M. Farinella [1]

[1] Department of Mathematics and Computer Science – University of Catania
{dimauro,battiato,gfarinella}@dmi.unict.it

[2] Park Smart s.r.l. – Corso Italia, 298 – Catania
{marco.moltisanti,giuseppe.patane}@parksmart.it

Abstract

*The paper presents Park Smart, a solution which aim is to solve the pain of finding a free parking space in public and private areas (*e.g. *cities, malls, etc.), and hence to optimize parking stalls allocation as well as to increase revenues for the companies which manage them. The proposed solution exploits cutting edge technologies such as IoT, Cloud Computing and Deep Learning.*

1. Introduction

In the last 20 years, the number of people that live in urban areas has been constantly increasing, especially in less developed regions [17]. Together with the growth in the urban population (Fig. 1), the number of vehicles in use has been increasing [1], as shown in Fig. 2.

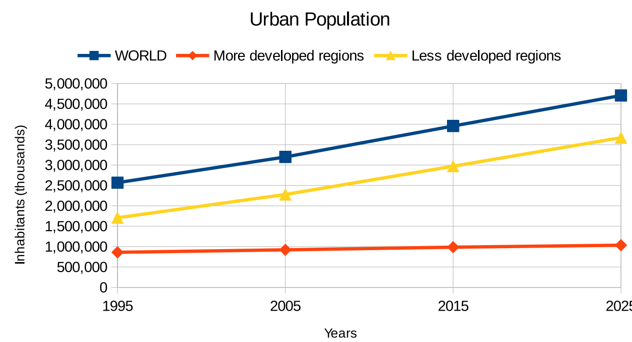

Figure 1. Urban Population 1995 – 2015 and trend to 2025

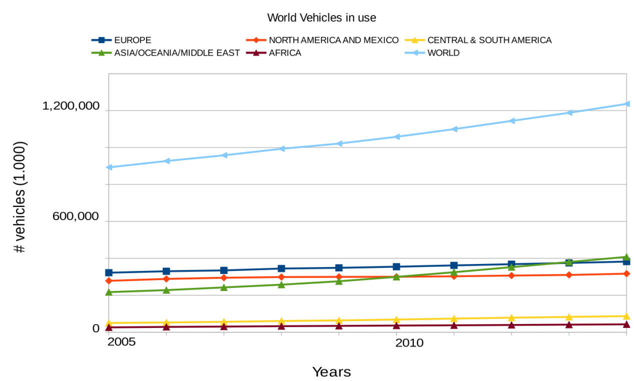

Figure 2. World vehicles in use 2005 – 2014.

These trends lead to a significant reduction in the availability of parking lots. Consequently the amount of time spent driving increases, together with stressful conditions and air pollution. Therefore, smart monitoring of parking stalls aiming to optimize the path from the current driver position to a free parking lot is not only a matter of maximizing profits for the owner or the manager of the parking, but also a matter of public health.
To face this kind of problems, leveraging new technologies to ease their solutions, the scientific community and the governments elaborated the concept of *Smart Cities*. A Smart City is a city where, with a massive use of ICT solutions, classic problems such as traffic, health monitoring, mobility, efficient governance, etc. are tackled in an innovative fashion. Moreover, in the last few years there has been a significant interest about the Internet of Things paradigm [4]. The IoT has been defined as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [9]. Thus Smart Cities are intrinsically correlated to the implementation of an IoT framework. In such a scenario, sensors are distributed in the area of the cities and generally connected via cloud, to exploit the computational power of remote machines. This model though is subject to the limitation brought by the availability of a high speed connection. Recently, major

companies have started to focus on a new paradigm, known as *edge computing* [2]. In this model, the computation is "moved" from inside the cloud to its borders, making use of distributed processing units. Thus the IoT device is transformed from a mere sensor to an intelligent unit.

Here we present Park Smart as a case study. We will discuss the end-to-end system that allows to design a low-cost solution to detect whether a parking stall is free or not.

The remainder of this paper is structured as follows. In Section 2 we briefly review relevant works. In Section 3 we present the Park Smart full pipeline. In Section 4 we discuss the classification system. Whereas Section 5 conclusions and final considerations are drawn.

2. Related works

The problem of detecting empty vs non-empty parking lots is not new in Computer Vision. Wu *et al*. [18] proposed a simple pipeline, where patches were extracted and normalized into rectangular patches by using perspective transformation. The color distribution on these patches is computed by the authors and used to feed a Multi-class Support Vector Machine (SVM) for classification purposes. As a final stage, the results of the classification are processed using a Markov Random Field (MRF) to refine potential conflicts between two neighboring patches. Among other works based on colors histogram we can cite [16, 15]

A method based exclusively on image processing techniques was proposed by Yusnita *et al*. in [19]. The authors mark the real scene painting each stall with a brown circle in the center. In order to decide if a place is available or not the images are thresholded and enhanced using morphological operators. Then the system looks for the circles that are still visible, using an eccentricity based measure to check if the detected blobs are roughly circular. As a last step, the system applies a threshold and counts the remaining spots, giving in output the number of free stalls.

Another approach in literature makes use of trajectories or events to separate empty stalls from non-empty ones. Specifically, Lin *et al*. [12] employ motion trajectories as feature vectors and then apply the adaptive Gaussian Mixture Model (GMM) and connected component analysis for background modeling and objects tracking.

Park lot classification has been addressed by de Almeida *et al*. [7], Di Mauro *et al*. [8], Amato *et al*. [3]. In [7], the main objective of the authors was to build a dataset in order to test and assess both old and new algorithms to solve the free parking slots classification problem. The pictures were taken in three different climatic conditions (i.e.: cloudy, sunny, rainy) to provide a large variability. In order to validate the "goodness" of the dataset, the authors performed three kind of tests. The first was related to the evaluation of using two different hand-crafted features (Local Binary Patterns and Local Phase Quantization). The second aimed to evaluate the generalization properties of the approach, using images from a single stall as training and testing the algorithms on the images representing other stalls. The purpose of the third test was to measure the learning ability of the system.

In [8] supervised and semi-supervised approaches have been compared to solve the problem of classify a parking space as empty or non-empty. In particular, a fine-tuned convolutional neural network (*AlexNet*), and a semi-supervised method, using a *CNN* fine-tuned with pseudo-labeled data have been tested. They compared results have been obtained using different loss functions and different dataset from video recordings. In [3] a smaller version of *AlexNet* (named *mAlexNet*) was adopted to make the detection task executable in real-time on an low-energy embedded device. The authors tested the network developed on the PKLot dataset and on a new dataset, *CNRPark-EXT*, which is now freely available for the community.

3. Park Smart: The Overall System at Glance

Despite the problem to solve is "simple" it can be easily intractable. Lets think about a single camera which streams every frame to a centralized server. Then multiply the band needed by one stream for the several cameras, hundreds, or thousands, or even millions, needed to monitor several smart cities areas. Note that a mid-sized city could need a number between 800 and 1000 cameras to monitor all the parking spaces.

It is clearly infeasible, or at least really expensive and not economically profitable to manage such kind of streaming traffic in real-time. We thus decided for a more scalable approach by bringing the computation close to the camera which acquire the stream using dedicated embedded systems that will send the results to the main server system.

Our architecture is described by Fig. 3. It has four main components:

Cameras We use wide angle cameras to optimize the number of parking spaces monitored. Our approach is not vendor locked. To have best results the resolution needed is at least 50px per side for each parking space.

AISEE IoT We analyze the video stream as closest as possible to the camera. It is an embedded system capable of elevated computing power, enough to do inference using deep learning models. Once inference is done the results are sent to the cloud platform. The embedded operating system has been developed with security, privacy and resilience in mind. We can deploy several AISEE IoT boxes depending on the number of cameras and the dimension of the installation.

Cloud We collect all the information from several installed embedded systems through a cloud platform which is

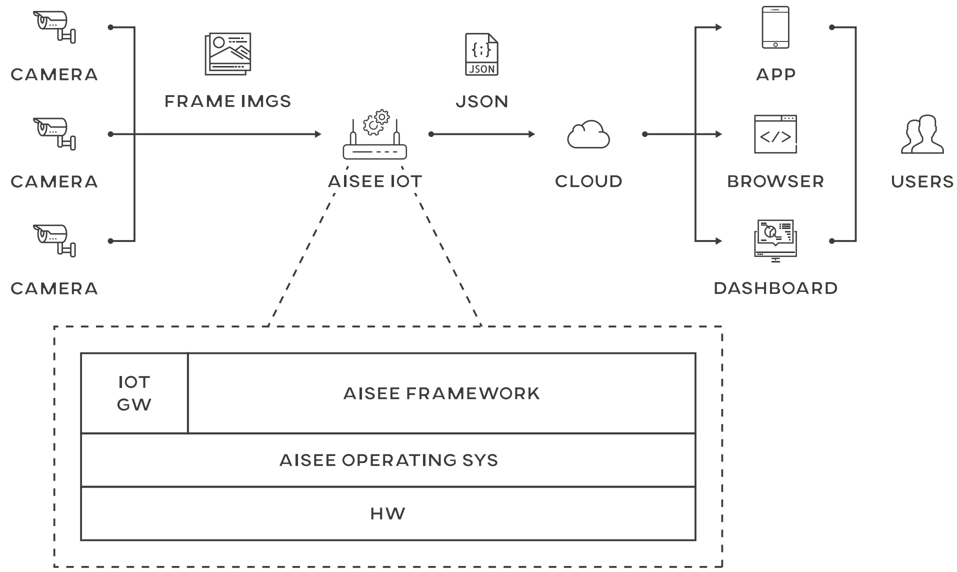

Figure 3. This diagram show the current Park Smart system: images and videos are captured by cameras which send them to the AISEE embedded where the computation is done. From there the information about the parking status is send to the cloud in order to be viewed by users.

scalable by design.

Presentation layer The system is accessible through different kind of appliances:

- The *dashboard* is the business and administration front-end which allows all the operations and to manage the installations (*e.g.* to add new cameras, configure cameras, add embedded, remove embedded and upgrade them, etc.).
- The *mobile app* or *browser* are the ending point for the people who are looking for a free spot where to park.

4. Recognizing free parking stalls

Classification is the task where the computer vision community has obtained great results since the introduction of deep CNNs. Thus we decided to tackle the problem to decide if a parking space is empty or not as a classification task over patches corresponding to parking lots. This approach is well suited for the most cases. The main idea is to divide each frame captured by the camera in several crops, where every crop is a square image corresponding to a parking space.

To investigate the approach, before producing our dataset, we used PKLot dataset [7], it has 12417 images with resolution of 1280×720 pixels. This dataset is really interesting for us for the following key features

- images were taken from three different parking areas;
- cameras were positioned at different heights;
- images have strong variability: such as the presence of shadows, over-exposition, low light, difference in perspective.

We sampled three datasets, one for each parking area, and fine-tuned AlexNet. The results are reported in Table 1.

Sample	Train	Val	Test	Accuracy
UFPR05	19281	4820	24101	99.93%
UFPR04	20000	5000	25000	99,96%
PUC	20000	5000	25000	99,92%

Table 1. Results using a fine-tuned AlexNet on PKLot

We tested the system on other three dataset considering images of a parking area composed by 46 parking spaces. The images were acquired in Catania, Sicily, during summer, autumn and winter 2015 in order to have as much variabilities as possible (light, weather, different cars, etc) and at different time of the day. To cover the parking space area the images have been acquired from eight cameras with Full-HD resolution extracted from motion jpeg streams. The sampled images have been cropped to extract stalls and manually labeled. Specifically each crop has been assigned a free or occupied label.

CNN Models	DS1	DS2	DS3	Avg. Accuracy	Footprint
AlexNet [11]	98,80%	99.20%	93.82%	97,27%	217M
GoogLeNet [14]	99.72%	99.58%	99.26%	99.52%	40M
VGG16 [13]	99.13%	98.70%	94.91%	97.58%	528M

Table 2. Results obtained considering different cnn models and three dataset. In particular, DS1 has 17688 train images, 3924 in val and 21612 in test; DS2 has 20636 train images, 4578 in val and 31374 in test; DS3 has 13032 train images, 2820 in val and 25212 in test

We analyzed different methods of cropping images, but, in most cases, the methods did not have an impact on the final classification results.

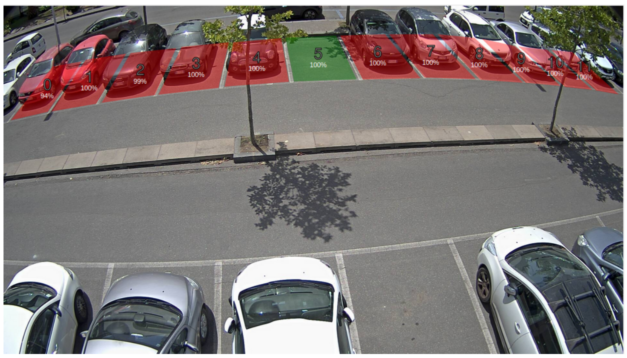

Figure 4. An example of classification of one camera. Best viewed in colors.

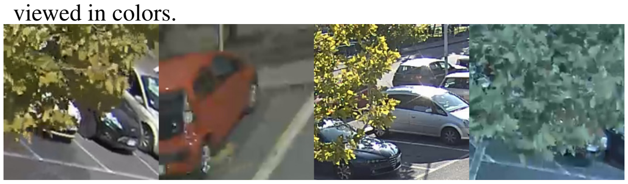

Figure 5. Here there are some misclassified images of parking spaces. Best viewed in colors.

To perform our experiments we used the Caffe library [10] taking advantage of GPU optimized code. To fine-tune the networks we used a machine equipped with four *NVIDIA GeForce TITAN X with 12Gb of DDR5 RAM*.

In order to find the best solution, balancing accuracy, classification speed and model footprint, we have investigated different models known in literature. The results are reported in Table 2. As we can see all the different models work quite well, with accuracy of 97% or more. GoogLenet (without the fully-connected layer) is the slowest to train but is the faster at inference time and the one with the smallest footprint, while VGG16 and AlexNet weigh far more.

In Figure 4 we show an example of classification. The camera is monitoring 12 parking spaces. On every parking space, an overlay displaying the confidence of belonging to empty or non empty class is shown. In Figure 5 we show some misclassification examples. Most of the errors depends from high occlusions and unconventional geometries. To better asses the results, a video demonstrating the proposed solution is available at the following url: `http://iplab.dmi.unict.it/ParkSmart/`

5. Conclusions

We have presented the Park Smart system. Park Smart developed an end-to-end pipeline for smart parking assistance and management. The infrastructure makes use of an IoT device, designed and developed by the company itself, which allows to perform the computation on the "borders" of cloud, implementing the *Edge Computing* paradigm. The system relies on a computer vision algorithm able to classify parking spaces, given their spatial configuration. Current and future developments include car detection, to solve the parking stall detection problem even on streets' side parking spaces (also in the locations where lines are not painted), camera calibration together with depth estimation [5] in order to be able to measure the size of vacant spaces, as well as traffic flow monitoring [6] and license plate recognition.

References

[1] OICA report on vehicles in use. http://www.oica.net/category/vehicles-in-use/, 2014.

[2] A. Ahmed and E. Ahmed. A survey on mobile edge computing. In *Intelligent Systems and Control (ISCO), 2016 10th International Conference on*, pages 1–8. IEEE, 2016.

[3] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, 72:327–334, 2017.

[4] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[5] S. Battiato, S. Curti, M. La Cascia, M. Tortora, and E. Scordato. Depth map generation by image classification. In *Proceedings of SPIE*, volume 5302, pages 95–104, 2004.

[6] S. Battiato, G. M. Farinella, A. Furnari, G. Puglisi, A. Snijders, and J. Spiekstra. An integrated system for vehicle tracking and classification. *Expert Systems with Applications*, 42(21):7263–7275, 2015.

[7] P. R. de Almeida, L. S. Oliveira, A. S. B. Jr., E. J. S. Jr., and A. L. Koerich. Pklot a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937 – 4949, 2015.

[8] D. Di Mauro, S. Battiato, G. Patanè, M. Leotta, D. Maio, and G. M. Farinella. Learning approaches for parking lots classification. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 410–418. Springer, 2016.

[9] International Telecommunication Union. ITU-T Y.4000/Y.2060 – Overview of the Internet of Things, 2012.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] L. L. Ng and H. S. Chua. Vision-based activities recognition by trajectory analysis for parking lot surveillance. In *Circuits and Systems (ICCAS), 2012 IEEE International Conference on*, pages 137–142. IEEE, 2012.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[15] N. True. Vacant parking space detection in static images. *University of California, San Diego*, 17, 2007.

[16] M. Tschentscher and M. Neuhausen. Video-based parking space detection. In *Proceedings of the Forum Bauinformatik*, pages 159–166, 2012.

[17] UN-Habitat. *World Cities Report 2016: Urbanization and Development Emerging Futures*. UN-Habitat, 2016.

[18] Q. Wu, C. c. Huang, S. y. Wang, W. Chiu, and T. Chen. Robust parking space detection considering inter-space correlation. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 659–662. IEEE, 2007.

[19] R. Yusnita, F. Norbaya, and N. Basharuddin. Intelligent parking space detection system based on image processing. *International Journal of Innovation, Management and Technology*, 3(3):232, 2012.