A wireless smart camera network for parking monitoring

Giuseppe Amato* Paolo Bolettieri* Fabio Carrara* Luca Ciampi* Claudio Gennaro* Giuseppe Riccardo Leone* Davide Moroni* Gabriele Pieri* Claudio Vairo*

***Abstract*—In this paper we present a Wireless Sensor Network (WSN), which is intended to provide a scalable solution for active cooperative monitoring of wide geographical areas.**

The system is designed to use different smart-camera prototypes: where the connection to the power grid is available a powerful embedded hardware implements a Deep Neural Network, otherwise a fully autonomous energy-harvesting node based on a low-energy custom board employs lightweight image analysis algorithms. Parking lots occupancy monitoring in the historical city of Lucca (Italy) is the application where the implemented smart cameras have been deployed. Traffic monitoring and surveillance are possible new scenarios for the system.

***Index Terms*—smart camera, parking monitoring, wireless sensor network, deep learning, embedded vision sensors**

I. Introduction

About 30% of rush hour traffic is generated by cars searching for a parking lot [1], [2]. This is a common issue in every big or medium size city. These cars increase the congestion and the emission of noxious gases in the air; furthermore, the long amount of this wasted time is a cause of stress and late arrivals. An automatic system which monitors the urban streets could be very effective in addressing this problem by pointing on a map the areas in which a parking space is easier to be found. Very often inductive field sensors installed under the street or one to one infrared sensors in indoor parking space are used to keep track of the status of the lots. These technologies become highly ineffective due to the high costs whenever large outdoor parking areas should be monitored. In addition, they require major and invasive work to be installed in existing parking areas and curb spaces, which are typical of historical cities. To address these issues, the research project SmartPark@Lucca aims at the development of a system based on a network of "intelligent" cameras able to monitor one or more parking areas of variable dimensions and at the validation of the proposed technology in the medieval city of Lucca, Italy. Processing of the pictures captured by the cameras is performed on board of the smart cameras themselves, so that no transmission of the pictures is required. Each camera is able to evaluate with extreme precision the occupancy level of the monitored parking area and to identify free or occupied spaces. Given that a single camera is able to monitor dozens of spaces, this infrastructure has a significantly lower cost, compared to other solutions. Inside the medieval city of Lucca there are places in which the wired electrical network is sometimes not present and it is not possible to put new artifact due to cultural heritage constraints. In order to manage this scenarios, our research project envisages the development of two solutions: a *Custom vision board* and a *Powered vision board*. The former solution is based on an embedded architecture that has a low consumption and that allows us to use a battery pack and a module for harvesting energy through a photovoltaic panel. The latter one is based on more commercial but more powerful solutions available off-the-shelf. In Section II the detail description of the smart camera prototype is given. Related works and the computer vision logic implemented in the different case scenarios are described in section III Section IV illustrates the experimental results of the first year of the project. They are mainly based on the single camera performance, while for the cooperative sensing we present the first available and encouraging results, whose extension will be the main focus of the next, second, year of the research project. Conclusion and future works conclude the paper.

*All the authors are with the Institute of Information Science and Technologies (ISTI) of the National Research Council of Italy (CNR), Pisa, Italy, `Name.Surname at isti.cnr.it`

II. Materials

The *ad hoc* realization of the smart camera prototype started from a deep study on the design of the architectural side. The guiding principle has been to be able to use state of the art computer vision technology using, at the same time, low cost sensors and electronic components, so that, once engineered, the device can be manufactured at low-cost in large quantities.

Another design requirement is represented by the ease of installation of the device, thus, the protective shield that has been considered for the sensor node is compact but able to accommodate all the components of the device.

The last and optional requirement is represented by the possibility to have a completely untethered device which communicates via wireless technologies and which has no need of electric power from the main supply. In this case the system has to have very low power consumption.

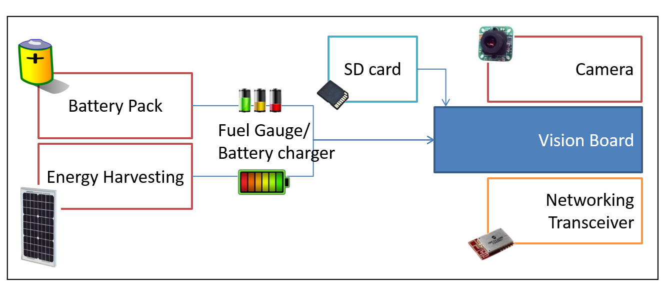

Fig. 1. Design of the architecture of the sensor node

The design of the sensor node architecture is depicted in Fig. 1. Each single sensor node is composed by two main units that will be described in more details in the next sections:

- the Sensor Unit: the vision board equipped with the camera sensor, the wireless communication module and the logic for image analysis. This is the part of the sensor devoted to the acquisition and processing of the images, and to the communication of the results to the system users.
- the (optional) Energy Harvesting Unit: a photo-voltaic panel, the batteries and the charge controller, which serves as a regulator for battery charging and allows an optimal choice in terms of energy savings policies.

As mentioned above, in order to achieve all the goals we used two different types of processing boards: the powerful Nvidia Jetson when the node is connected to the grid and a very low-powered custom board for the energy harvesting scenario.

A. Powered vision board

The embedded processor is the NVIDIA Jetson TX2, an ARM processor couples with parallel GPU which has outstanding processing power for such size and consumption. It is already the de-facto standard among similar custom drone projects. It is mounted on a credit card sized carrier board from Auvidea, because the original development kit is too large and too heavy for this purpose. The connection with the ground station uses standard Wi-fi channels.

B. Custom vision board

For the realization of the custom vision board, an embedded Linux architecture has been selected in the design stage for providing enough computational power and ease of programming. A selection of ready-made Linux based prototyping boards had been evaluated with respect to computing power, flexibility, price and support. All the candidates have as common disadvantages high power consumption. It has been therefore decided to realize a custom vision component by designing, printing and producing a new printed circuit board (PCB). The new PCB has been designed to have the maximum flexibility of use while maximizing the performance/consumption ratio. A good compromise has been achieved by using a *Freescale CPU* based on the ARM architecture, with support for MMU-like operating systems GNU/Linux. The chosen architecture has been proved to have an average consumption measured at the highest speed (454MHz) less than $500mW$. A microSD slot is present, which is essential for booting the system, booting the kernel and file-system associated (EXT4); the board can be upgraded simply by changing the contents of the microSD. In Fig. 2 the realized operational vision board is shown. After a period of testing, the ArchLinux distro has been choosen as embedded Linux operating system. ArchLinux aims at being a lightweight Linux implementation also targeted at embedded devices. Besides the i686 and x86-64 architectures, which are officially supported, there are a number of user-contributed architectures whose packages are collected in ArchLinux User Repository (AUR). Development and deployments are currently based on Linux Kernel 3.14.39.

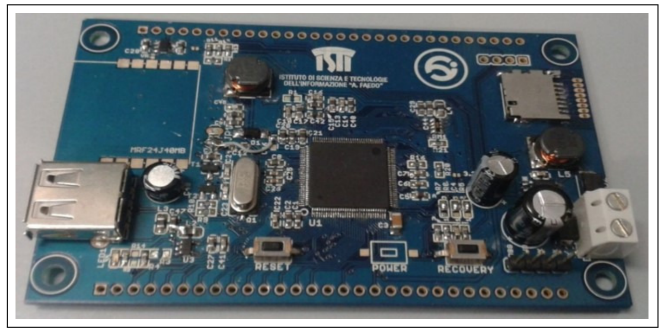

Fig. 2. The realized vision board for the energy harvesting sensor node

C. Energy harvesting and housing

The developed and previously described board and camera are housed into a polycarbonate IP67 shield, which is subject to less electromagnetic attenuation with respect to metallic cases. The other important component of the smart camera is the power supply and energy harvesting system that controls charging and permits to choose optimal energy savings policies. The power supply system includes the lead (Pb) acid battery pack and a module for harvesting energy through a photo-voltaic panel. For the experimentation, a 12V panel with a nominal power of 15W was used and managed by a controller outputting up to 4A DC current. A 12Ah battery has been normally used for tests. The vision board has also been used to measure the charging status of the batteries. To this end, an ADC Conditioning module has been used to adapt the voltage level of the power supply system to the voltage range of the vision board ADC input. In Fig. 3 an installation of two sensors on the same existing pole is presented; in this configuration, the sensors share the same battery and energy harvesting module.

Fig. 3. An installed prototype with two complete sensor nodes

III. METHODS

The traditional image classification approach is based on ad-hoc functions to search for special feature in the image, like angles or straight lines, which is representative of the certain objects [3]–[6]. Color features and a Support Vector Machine (SVM) is used in [7] to distinguish the available spots in a parking area. A classification based on color histogram of three adjacent lots instead of a single one is presented in [8]. To deal with light changes Tsai et al. [9] trained a Bayesian classifier searching for angles and edge features. A 3D model of the parking area and a hierarchical Bayesian framework [10] achieves good results during day and night. Jermsurawong et al. [11] use an ad-hoc trained neural network to solve the task. Using a video sequence instead of a single frame the work proposed in [12] constantly update the transience map of the parking area, while in [13] the authors show good results dealing with partial occlusions. Recently Deep Convolutional Neural Networks (DNN) achieved very good performance in image classification [14]. Vehicles detection in high resolution satellite images is presented in [15].

A. Deep learning approach

In this section, we present two solutions based on deep learning and deployed on the powered vision board presented in Section II-A. The first solution is used to monitor the occupancy status of individual parking spaces in a parking lot. The second one is used to globally count cars parked in the parking lot.

1) Parking space classification: We propose a solution for monitoring the occupancy status of individual parking spaces, that uses a Convolutional Neural Network (CNN) to determine if a parking space is free or occupied. This approach exploits a network of smart cameras (computing boards equipped with a camera module for acquiring images) where each smart camera monitors the set of parking spaces included in the portion of the parking lot seen by the camera. The system also comprises a server that receives the occupancy information of the parking spaces and visualizes them.

We used a modified version of the AlexNet CNN [24], that is called mAlexNet [25], to perform the parking space classification. This network is composed of three convolutional layers and two fully-connected layers, and has been training on the *CNRPark-EXT* dataset [21], a collection of roughly 150,000 bounding-box annotated images of vacant and occupied parking slots (called *patches*) in the campus of the National Research Council (CNR) describing most of the difficult situations that can be found in a real scenario: the images are captured by nine different cameras under various weather conditions, angles of view and light conditions. Furthermore, another challenging aspect is due to the presence of partial occlusion patterns in many scenes such as obstacles (trees, lampposts, other cars) and shadowed cars.

For each camera, we defined the Region Of Interest (ROI) for all the parking spaces included in the field of view of that camera. This operation is performed manually once before deploying the system. At run-time, we use these ROIs to segment the image captured by the smart camera to obtain a set of patches comprising the individual parking spaces. Each patch is then given in input to our CNN which processes it and provides the occupancy status for the parking space coupled with its confidence value.

All the processing is executed on board the smart camera. No image is sent outside. The only information transmitted to the server is the occupancy status for each parking space. This preserves the privacy of the license plates and reduces considerably the bandwidth needed for the system to run.

2) Cars counting: We also propose a deep learning-based approach that is able to count the cars in captured images *without* any extra information of the scenes, like the regions of interest (i.e. the position of the parking lots) or the perspective map. This is a key feature since in this way our solution is directly applicable in unconstrained contexts. The proposed approach is based on *Mask R-CNN* [18], a very popular deep convolutional neural network, employed in many detection systems. Unlike previous methods that tackle the localization problem by building a sliding-window detector, *Mask R-CNN* solves the problem by operating within the recognition using regions paradigm [19], taking a full image as input and producing as output labels for each detected object together with bounding boxes and masks localizing them. The authors differentiate between the convolutional backbone architecture, used for features extraction over an entire image, and the network head for bounding-box recognition (classification and regression) and mask prediction that is applied to each proposed region.

As a starting point, we considered a model of Mask R-CNN pre-trained on the COCO dataset [20], a large dataset composed of images describing complex everyday scenes of common objects in their natural context, categorized in 80 different categories. In order to count vehicles, we considered the detected objects belonging to the *car* and *truck* categories. Since this network is a generic objects detector, we specialize it to recognize the vehicles we want to count.

The first step has been the creation of a suitable labeled training set. In the case of Mask R-CNN, these labels correspond to masks. As a baseline, we used again the *CNRPark-EXT* dataset [21]. In order to make this dataset useful for our purposes, we aggregate all the patches belonging to the same scene in a single full image, and then we add the masks on the vehicles to be detected. Since mask creation is a very time-expensive operation, we take advantage of the output of the pre-trained model of Mask R-CNN. As mentioned before, this network produces in output accurate masks localizing objects, so the idea is to save these masks automatically generated, parse them, changing the associations between wrong categories and the underlying objects that were instead vehicles we want to count, and, finally, manually add some of the remaining masks for the cars that were not detected. At this point, we retrained the network using this new mask-labeled and task-specific dataset, freezing the weights of the backbone, and saving the new weights of the head after a few epochs.

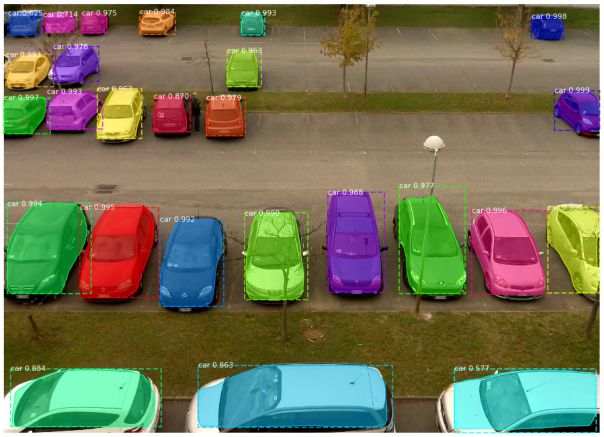

Fig. 4. An example of an image captured by a camera, where we have applied the masks localizing the vehicles present in the scene. Counting the instances, we have an estimation of the total number of cars present in the monitored car park.

Again, all the processing is executed on board the smart camera, and this time the only information transmitted to the server is the total number of vehicles present in the scene. Optionally, it is possible to transmit also the coordinates of the bounding boxes or the masks localizing the detected cars. An example is shown in Fig. 4.

B. Lightweight image processing

The low powered custom board is not suitable for the deep learning approach. Specific artificial vision algorithms have been studied, designed and deployed for this board. The application defines some Regions of Interest (ROI) in the source image and uses a background model computed through lightweight methods [16], [17]. An adaptive background is computed because it proved to be the most robust for use in uncontrolled outdoor scenes. The background is continuously updated using both the previous background model and the latest acquired actual image. In order to reach a strong belief of the status of each the parking slot, we perform two different image analyses, considering that an empty slot should appear as plain asphalt without nothing inside. The first is the so-called asphalt detection: periodically checking small rectangular asphalt samples on the driveway (using the current background image so that no moving vehicle is on the region of interest) we identify similar hue and saturation values in the rest of the image. For each ROI R_k the index $a_k(t)$ is computed. This index is proportional to the ratio of asphalt pixels with respect to the total number of pixels in the R_k.

Then a very neat image of the contours of the vehicles is obtained with a Canny edge detection of the current foreground image. Similar to the first one the computed index $e_k(t)$ is proportional to the ratio of edge pixels in ROI R_k with respect to the total number of pixels in R_k. The combination of the two indexes creates the final belief of the sensor, which indicates the probability of the occupation of the parking lot.

$$P_k(t) = e_k(t) \cdot (1 - a_k(t)) \tag{1}$$

The occupancy status becomes effective only after being observed consecutively for a specific number of acquired frames.

IV. EXPERIMENTAL RESULTS

A. Parking space monitoring evaluation

The CNRPark-EXT dataset presented in [21] is used to perform our experiments.

The Overall Error Rate (OER) is the metric proposed in [23] and is defined as the ratio of total errors (i.e., False Positive plus False Negative) and the total responses (i.e., False Positive plus False Negative plus True Positive plus True Negative):

$$\mathrm{OER} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{FP} + \mathrm{FN} + \mathrm{TP} + \mathrm{TN}} \tag{1}$$

The CNN operates at single-frame level (the output is the state of each slot based on the analysis of a single image); the lightweight image analysis, instead, is a time-related system. In order to compare the two methods, it seems reasonable to consider every single frame output as the complete set of the slots' statuses; thus the denominator of 1 for the second method corresponds to the multiplication of the slots monitored and the total frames considered:

$$\mathrm{ErrorRate} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{TotalFrames}} (\mathrm{FP_i} + \mathrm{FN_i})}{\mathrm{TotalFrames} * \mathrm{TotalSlots}} \tag{2}$$

where $\mathrm{FP}_i + \mathrm{FN}_i$ is the total number of errors made when analyzing frame i.

The average error rate for the CNN-based method results to be 0.4%, while the lightweight image analysis achieves 0.65%. Although a number of papers in literature have been presented on the topic, a fair comparison is difficult to be conducted, since most of the approaches are far from being based on

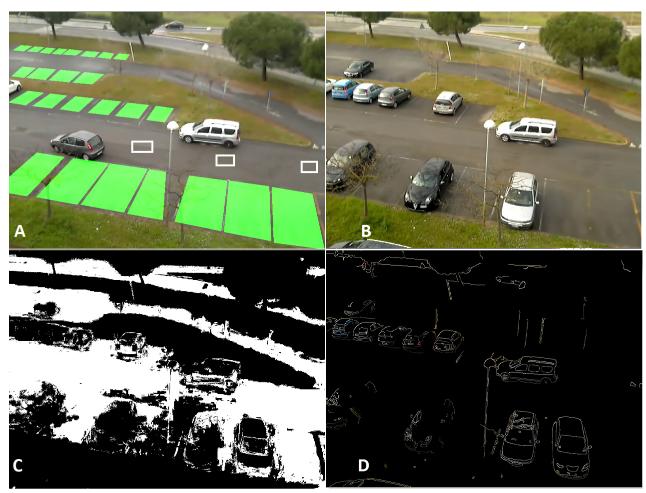

Fig. 5. A) RoI for a set of parking lots are set up manually with the help of a graphics tool. Small rectangles on the driveway define the samples for asphalt detection B) Background image C) White regions represent areas where asphalt is detected D) The output of the Canny edge detector

TABLE I
COMPARISON OF RELATED WORK

Reference	Error rate (%)	Features
Wu et al. 2007 [8]	6.5	*Color*
Sastre et al. 2007 [26]	2.2	*Gabor filters*
Bong et al. 2008 [27]	7.0	*Color*
Hichihashi et al. 2009 [28]	2.0	*PCA*
Huang and Wang 2010 [10]	1.2	*Color*
DeAlmeida et al. 2015 [23]	0.4	*Texture*
Proposed method 1	0.4	*CNN*
Proposed method 2	0.65	*Edge and color*

a "low-cost embedded platform". Yet, their performance is not radically different from the one achieved by our methods. To the best of our knowledge, the approach presented in [23], which is based on hand-crafted features (LPB and LPQ) and targeted for high-end computers, has similar performance with respect to our deep-learning approach but has a lower generalization capability [21]. A comparison with some related work can be found in Table I.

The CNN-based approach takes about 15 seconds to analyze an image and compute its output, while the lightweight image analysis takes about 2 seconds.

Cooperative monitoring: In order to test and validate the cooperative sensing functionalities of the system, we focused on the slots that were in the field of view of more than one sensor. In particular, for each of these slots, *event composition* was performed, i.e. aggregating the measures produced by each sensor in charge of its monitoring. A weighted average was used in aggregation; each sensor contributes to the average with a weight proportional to the area in pixel of the region in the image corresponding to the monitored slot. By evaluation of the log of the network, it resulted that event composition leads to a reduced number of fluctuations, allowing to filter out events due to temporary occlusions in the fields of view. In addition, event composition allowed to cope with potential failures of one of the nodes, that have been tested switching off artificially a sensor in the network. In Table II a resume of the cooperative monitoring performance is reported. The monitoring regards the slots which are watched by two cameras, comparing each individual camera result only for the slots monitored by both, versus the cooperative weighted results (i.e. shown in the column *COOP.*). As it can be seen, the results from single cameras are heavily improved, as the error rates drop from 6% and almost 5% down to only 1% of errors, counting on a total of 78,840 events on which the occupancy detection algorithm is called.

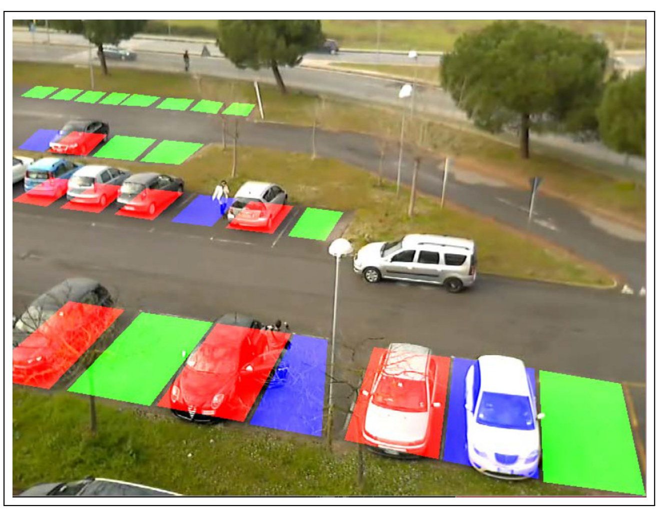

Fig. 6. Real-time output: red is busy, green is available, blue is uncertain due to partial occupation or vehicle not integrated in the background yet

TABLE II
COOPERATIVE MONITORING RESULTS.

	CAM A	CAM B	COOP.
ERRORS	4870	3674	804
%	**6.2%**	**4.7%**	**1.0%**

B. Cars counting evaluation

We performed our experiments using the test subset of the *CNRPark-EXT* dataset [21]. Following other counting benchmarks, we use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as the metrics for evaluating the performance of our solution. MAE is defined as follows:

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|c_n^{gt} - c_n^{pred}| \tag{3}$$

while RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(c_n^{gt} - c_n^{pred})^2} \tag{4}$$

where N is the total number of test images, c_{gt} is the actual count, and c_{pred} is the predicted count of the n-th image. Note that as a result of the squaring of each difference, RMSE effectively penalizes large errors more heavily than small ones. Then RMSE should be more useful when large errors are particularly undesirable.

The above two metrics are indicative of quantifying the error of estimation of the objects count. However, as pointed out by [22], these metrics contain no information about the relation of the error and the total number of objects present in the image. To this end, another performance metric is taken into account, which is essentially a normalized MAE, that we call Mean Occupancy Error (*MOE*) because in this work quantifies the error in the evaluation of the occupancy of a car park, defined as:

$$MOE = \frac{1}{N}\sum_{n=1}^{N}\frac{|c_n^{gt} - c_n^{pred}|}{num\_slots_n} \tag{5}$$

where num_slots_n is the total number of parking lots in the current scene. We express this evaluation metric as a percentage.

The results of the experimental evaluation are reported in Table III.

TABLE III
RESULTS IN TERMS OF MAE, RMSE AND MOE.

MAE	RMSE	MOE
1.05	2.1	3.64%

Finally, we evaluate the execution time. Using the NVIDIA Jetson TX2 described in Section II-A, we process a full image in a time between 10 and 15 seconds, depending on the total number of cars present in the scene.

V. CONCLUSION AND FUTURE WORK

This research has carried out in the framework of the SmartPark@Lucca project, which aims at the development of a system based on a network of "intelligent" cameras able to monitor one or more parking areas of variable dimensions in the medieval city of Lucca, Italy. Given that a single camera is able to monitor dozens of spaces, this infrastructure has a significantly lower cost, compared to other solutions. We presented the work done in the first year of the project: the design and realization of a customizable smart camera prototype and the computer vision logic to evaluate the occupancy level of the monitored parking area and to identify free or occupied spaces. The prototype can host a powerful vision board to run state of the art Convolutional Neural Network which achieve an error rate of 0.4%. Due to cultural heritage constraints of the medieval historical center of Lucca sometimes the wired electrical network is not present: in this case an energy harvesting unit in combination with a custom design very low consumption embedded vision board is able to run lightweight image processing with a slightly higher error rate of 0.65%. In both cases the processing of the pictures captured by the cameras is performed on board of the smart cameras themselves, so that no transmission of the pictures is required. We presented also the very first result of collaborative sensing that show how the overlapping of two cameras improve the monitoring in case of fixed or temporal occlusion. Cooperative sensing with three or more cameras will be the main focus of the next, second, year of the research project.

REFERENCES

[1] C. Dowling, T. Fiez, L. Ratliff, and B. Zhang,"Optimizing curbside parking resources subject to congestion constraints," In: 56th IEEE Annual Conf. on Decision and Control (CDC), pp.5080-5085, 2017.

[2] D. Shoup, The high cost of free parking, volume 7, Planners Press, American Planning Association, 2005.

[3] T. Ojala, M. Pietikainen, and T. Maenpaa,"Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," In: IEEE Trans. on Pattern Analysis and Machine Intelligence, pp.971987, 2002.

[4] V. Ojansivu and J. Heikkila,"Blur insensitive texture classification using local phase quantization," In: Image and signal processing, pp.236243, Springer, 2008.

[5] J. Platt,"Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," In: Advances in large margin classifiers, pp.6174, MIT press, 1999.

[6] E. Rahtu, J. Heikkila, V. Ojansivu, and T. Ahonen,"Local phase quantization for blur-insensitive image analysis," In: Image and Vision Computing, 30(8):501512, 2012.

[7] N. Dan, Parking management system and method, US Patent App. 10/066,215, 2002.

[8] Q. Wu, C. Huang, S. y. Wang, W. c. Chiu, and T. Chen,"Robust parking space detection considering inter-space correlation," In: IEEE Int. Conf. on Multimedia and Expo, pp. 659662, 2007.

[9] L. W. Tsai, J. W. Hsieh, and K. C. Fan,"Vehicle detection using normalized color and edge map," In: IEEE Trans. on Image Processing, 16(3):850864, 2007.

[10] C. C. Huang, Y. S. Tai, and S. J. Wang,"Vacant parking space detection based on plane-based bayesian hierarchical framework," In: IEEE Trans. on Circuits and Systems for Video Technology, 23(9):15981610, 2013.

[11] J. Jermsurawong, U. Ahsan, A. Haidar, D. Haiwei, and N. Mavridis,"One-day long statistical analysis of parking demand by using single-camera vacancy detection," In: J. of Transportation Systems Engineering and Information Technology, 14(2):3344, 2014.

[12] C. G. del Postigo, J. Torres, and J. M. Mendez,"Vacant parking area estimation through background subtraction and transience map analysis," In: IET Intelligent Transport Systems, 2015.

[13] I. Masmoudi, A. Wali, A. Jamoussi, and A. M. Alimi,"Parking spaces modeling for inter spaces occlusion handling," In: 22nd Int. Conf. on Computer Graphics, Visualization and Computer Vision, pp.119124, 2014.

[14] D. Ciresan, U. Meier, and J. Schmidhuber,"Multi-column deep neural networks for image classification," In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp.36423649, 2012.

[15] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan,"Vehicle detection in satellite images by hybrid deep convolutional neural network," In: IEEE Geoscience and Remote Sensing Letters, Vol. 11, Issue 10, Oct. 2014.

[16] Kim, K., Chalidabhongse, T., Harwood, D., and Davis, L.,"Background modeling and subtraction by codebook construction," In: IEEE Int. Conf. on Image Processing, 2004.

[17] Stauffer, C., and Grimson, W. E.,"Adaptive background mixture models for real-time tracking," In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, vol. 2, pp.246-252, 1999.

[18] K. He, G. Gkioxari, P. Dollr, and R. Girshick,"Mask r-cnn," In: IEEE Int. Conf. on Computer Vision, pp.29802988, IEEE, 2017.

[19] C. Gu, J. J. Lim, P. Arbelez, and J. Malik,"Recognition using regions," In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp.10301037, 2009.

[20] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick,"Microsoft coco: Common objects in context," In: European conf. on computer vision, pp.740755, Springer, 2014.

[21] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo,"Deep learning for decentralized parking lot occupancy detection," In: Expert Systems with Applications, vol. 72, pp.327334, 2017.

[22] C. C. Loy, K. Chen, S. Gong, and T. Xiang,"Crowd counting and profiling: Methodology and evaluation," In: Modeling, Simulation and Visual Analysis of Crowds, pp.347382. Springer, 2013.

[23] P. R. L. De Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich,"PKLot-A robust dataset for parking lot classification," In: Expert Systems with Applications, vol. 42, pp.4937-4949, 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton,"Imagenet classification with deep convolutional neural networks," In: Advances in neural information processing systems, pp.10971105, 2012.

[25] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo,"Car parking occupancy detection using smart camera networks and deep learning," In: 21th IEEE symposium on computers and communications, pp.12121217, 2016.

[26] R. J. Lopez Sastre, P. Gil Jimenez, F. Acevedo, and S. Maldonado Bascon,"Computer algebra algorithms applied to computer vision in a parking management system," In: IEEE int. symposium on industrial electronics, pp.1675-1680, 2007.

[27] D. B. L. Bong, K. C. Ting, and K. C. Lai,"Integrated approach in the design of car park occupancy information system," In: IAENG Int. J. of Computer Science, vol. 35, pp.1-8, 2008.

[28] H. Ichihashi, A. Notsu, K. Honda, T. Katada, and M. Fujiyoshi,"Vacant Parking Space Detector for Outdoor Parking Lot by Using Surveillance Camera and FCM Classifier," In: 18th IEEE Int. Conf. on Fuzzy Systems, pp.127-134, 2009.