A Comparison of Techniques based on Image Classification and Object Detection to Count Cars and Non-empty Stalls in Parking Spaces

D. Di Mauro[1], A. Furnari[1], G. Patanè[2], S. Battiato[1] and G. M. Farinella[1]
[1]*Department of Mathematics and Computer Science, University of Catania, Catania, Italy*
[2]*Park Smart s.r.l., Catania, Italy*

Keywords: Counting, Deep Learning, Classification, Object Detection, Smart Cities.

Abstract: The world-wide growth of population in urban areas demands for the development of sustainable technologies to manage city services, such as transportation, in an efficient way. Motivated by the cost-effectiveness of image-based solutions, in this paper we investigate the exploitation of techniques based on image classification and object detection to count cars and non-empty stalls in parking areas. The analysis is performed on a dataset of images collected in a real parking area. Results show that techniques based on image classification are very effective when parking stalls are delimited by marking lines and the geometry of the scene is known in advance.

1 INTRODUCTION AND MOTIVATIONS

In 2007, as stated in (United Nations, 2008), due to an unprecedented urban growth, the world's population was evenly split between urban and rural areas. As a result, urban life problems such as air pollution, traffic congestion, and lack of parking spaces, have worsened sharply. Among other possible solutions, the new concept of *Smart Cities* has been developed. This concept proposes the application of recent technological advancements in the areas of *Internet of Things*, *Computer Vision*, *Machine Learning*, *4G Networks*, etc., to improve the liveability of cities. Among the different applications concerning the concept of smart cities e.g., traffic analysis (Raymond, 2001; Battiato et al., 2018), vehicle tracking (Battiato et al., 2015), etc. in this paper we focus on optimizing the use of parking spaces in urban areas.

The use of private cars has massively increased over the years, without a proportional expansion in the number of parking spaces. Drivers waste a great deal of time looking for a free parking space and favor the worsening of traffic and air pollution. Various systems aimed at optimizing the use of parking spaces require methods to count the number of vehicles and free spaces in parking areas. There are three major types of solutions in this field. *Counter-based technologies* allow to detect whenever a car enters or leaves in a closed parking area, equipped with a barrier, in order to update the number of vehicles and the available parking spaces. *Sensor-based technologies* make use of sensors plunged into the asphalt to detect the presence or absence of cars upon it. *Image-based technologies* use cameras monitoring the parking area and rely on Computer Vision to count vehicles.

Due to the Computer Vision and Machine Learning advances in the last decade, we believe that image-based solutions are economically advantageous over other methods since they do not require specific sensors and can be easily implemented in the context of free-access parking areas. Nevertheless, such approaches have to face several variabilities depending on the positioning of the cameras, the shape of the parking spaces, different lighting conditions, presence of shadows, occlusions, etc.

In this paper, we investigate the application of two methods based on Image Classification and Object Detection to tackle two different but related problems: 1) counting the number of empty and non-empty stalls in a parking area, 2) counting how many cars are present in a parking area. The first problem is common in managed parking areas, where parking spots are delimited by lines. The second one arises when there are not lines delimiting the parking spots and hence the configuration of the parking lot depends on how drivers park. To perform the analysis, we collected and labeled images of a parking area in a real scenario.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 explains the investigated methods. Section 4 introduces

Mauro, D., Furnari, A., Patanè, G., Battiato, S. and Farinella, G.
A Comparison of Techniques based on Image Classification and Object Detection to Count Cars and Non-empty Stalls in Parking Spaces.
DOI: 10.5220/0006851403280336
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS*, pages 328-336
ISBN: 978-989-758-319-3

the proposed dataset and the experimental settings. Section 5 reports the results. Section 6 concludes the paper.

2 RELATED WORKS

We use the verb "counting" to indicate three different processes which we exploit depending on the number of instances we need to count (Davis and Pérusse, 1988):

Estimation: the process to approximate the correct number of elements in a scene. We use this process when the number of elements is too big to extensively count each object instance, e.g. the number of leaves in a tree;

Subitizing: the process employed when the number of objects ranges between 1 and 4 instances. The term has been proposed (Kaufman et al., 1949) to denote the capability to determine the number of objects instantly (i.e. at glance); When the number of objects is higher than 4, the ability to count at glance decreases in terms of accuracy and speed;

Counting: the process of finding the exact number of elements given a finite set of objects of a specific type.

Past neuroscience research has investigated the counting processes at neural level. Eiselt and Nieder (Eiselt and Nieder, 2013) discovered in the primate brain the presence of rule-selective neurons specialized in guiding decisions related to a specific magnitude type only, as well as generalizing neurons that respond abstractly to the concept of magnitude rules. These specialized neurons "number neurons" (Nieder, 2016) encode the number of elements in a set, as well the numerosity or cardinality, from both spatial and temporal presentation arrays.

For the Computer Vision community, Object counting is still a challenging problem which needs a fine-grained understanding of the scene. The task has been typically studied considering specific contexts, e.g., counting people in crowded scenes (Chan et al., 2008; Chen et al., 2015; Li et al., 2008; Lempitsky and Zisserman, 2010; Zhang et al., 2015), cells in biological images (Lempitsky and Zisserman, 2010), bacterial colonies (Ferrari et al., 2017), penguins (Arteta et al., 2016), etc.

In particular, counting approaches can be divided into three groups (Loy et al., 2013):

Counting by Detection: these methods use object detection and count extensively (Chen et al., 2015).

Counting by Clustering: these methods assume the presence of individual entities presenting unique yet coherent patterns which can be clustered to approximate the final number of instances (Rabaud and Belongie, 2006).

Counting by Regression: these methods count entities by learning a direct mapping from low-level imagery to numbers (Chan et al., 2008; Lempitsky and Zisserman, 2010; Arteta et al., 2014; Fiaschi et al., 2012).

The increasing use of surveillance systems is pushing the use of image-based solutions to understand the semantics of parking areas in order to infer the presence of free spaces and to count cars. Some efforts have been done to create datasets useful to tackle this problem. De Almeida et al. (De Almeida et al., 2015) built a dataset containing pictures acquired in different climatic conditions (cloudy, sunny, rainy) and considering three different parking areas. They have benchmarked the problem of discriminating empty vs non-empty spaces comparing two different hand-crafted features: Local Binary Patterns and Local Phase Quantization. The extracted features are exploited with a SVM classifier.

An approach to analyze events and object trajectories in order to discriminate empty stalls from non-empty ones is proposed in (Ng and Chua, 2012). The authors employed motion trajectories as features and applied the adaptive Gaussian Mixture Model (GMM) followed by connected component analysis for background modeling and objects tracking.

Wu et al. (Wu et al., 2007) described a pipeline in which an image patch is sampled every three stalls present in a frame. Hand-crafted features are hence computed to infer the likelihood that a pixel belongs to ground regions. The features extracted from the patch representing the three contiguous parking spaces are given to a eight-way Support Vector Machine (SVM) to classify the 2^3 possible configurations of free or occupied stalls in which the three spaces can be. Conflicts between two neighboring patches are refined by employing a Markov Random Field (MRF).

Deep learning techniques have been recently adopted in this domain obtaining promising performances. In (Amato et al., 2017), a new dataset (*CNRPark-EXT*) to train deep learning based models has been introduced. A modified *AlexNet* CNN is employed to obtain a reduced-size model in order to make inference possible in real-time on low-cost embedded devices. Di Mauro et al. (Di Mauro et al., 2016) proposed an evaluation of supervised and semi-supervised approaches based on a Convolutional Neural Network (CNN) fine-tuned using respectively labeled and pseudo-labeled data. The main aim of the

Figure 1: An example image acquired in a real parking lot using a fixed camera with parking stalls highlighted in green (the image is best seen in digital format). To count non-empty parking spaces, we assume that the positions of the stalls are known in advance.

work was to assess which approach is the most convenient to balance labeling effort over classifications performance.

3 METHODS

As mentioned before, we are interested in counting the number of empty and non-empty stalls in a parking area, as well as in counting how many cars are present. These two tasks are related but not identical. The number of cars present in the parking area cannot be directly inferred from the number of free parking spaces and the total number of stalls. Indeed, some vehicles may, for instance, be traversing the parking spaces, while others might be parked outside the stalls or they might occupy more than one stall. We assume a single camera pointing to the parking area, which acquires images similar to the one shown in Figure 1. The involved setup is convenient because it reduces the costs of deployment, trying to monitor all the parking lot with the minimum number of cameras.

3.1 Counting Non-empty Spaces

This task consists in counting the number of occupied parking spaces in the monitored area. We assume that the position of each stall is known in advance (Figure 1). It should be noted that, since the camera is fixed, labeling the position of the stalls is part of a calibration process which needs to be performed only once (i.e., when the camera is installed). We consider two approaches to count non-empty parking spaces: an approach based on image classification, and an approach based on object detection.

Image Classification. The first approach considers the problem as an image-based binary classification task. For each stall, we first extract the smallest square image patch containing it. Each image patch is labeled as "empty" or "full" depending on the occupancy status of the related stall. A classifier is hence trained to discriminate between "empty" and "full" stalls. At inference time, the trained classifier is used to determine the status of each stall in order to obtain the number of non-empty parking spaces.

Object Detection. The second approach employs a car detector to localize all the cars present in the image. All bounding boxes detected with a score lower than a given threshold d_1 are discarded. The Intersection Over Union (IoU) measure between each stall and each retained bounding box is hence computed. A stall is deemed to be occupied if the IoU with at least one detected car is higher than a given threshold d_2. The method allows to count the number of non-empty parking spaces by determining the status of each stall. This approach allows to obtain also information about cars which are parked on non-marked spaces. Such information can be useful to allow for better management of parking areas, e.g. detecting misparked cars.

3.2 Counting Cars

The question we want to address is the following: "How many cars are present in a given *Region of Interest* (RoI)?" This problem focuses on a more general scenario which does not assume the geometry of the scene to be known in advance or delimited parking stalls to be present. Also in this case, we consider two approaches, one based on image classification, and the other one based on object detection.

Image Classification. The first approach uses the output of the binary Image Classification described in the previous section to approximate the number of cars present in the scene as the number of occupied stalls. It should be noted that this method requires the geometry of the scene to be known. Moreover, this approach cannot deal with cars which are not placed in any of the marked stalls.

Object Detection. This method uses a vehicle detector to find cars present in the scene. Bounding boxes with detection score lower than the threshold r_1 or with IoU score with respect to the given Region of Interest lower than a threshold r_2 are discarded. We obtain the total number of cars present in the RoI by considering all retained bounding boxes.

Table 1: Videos contained in the dataset, along with the corresponding number of labeled frames.

Camera	Video	Frame Number
Camera 1	Video 1.1	1,801
	Video 1.2	1,801
Camera 2	Video 2.1	1,801
	Video 2.2	2,241
Camera 3	Video 3.1	1,321
	Video 3.2	2,101
	Total	11,066

4 DATASET AND EXPERIMENTAL SETTINGS

For the purpose of this study, we collected a dataset comprising a total of 11,066 images captured during 2 days in our living lab which is located at the campus of the University of Catania (Figure 2). Each frame has been labeled with annotations in the form of bounding boxes and parking space configurations.

The dataset has been acquired using three Full-HD cameras looking at different parking spaces. The three cameras are referred to as "Camera 1", "Camera 2" and "Camera 3". "Camera 1" observes 12 parking spaces (Figure 3), "Camera 2" monitors 14 parking spaces (Figure 4), and "Camera 3" acquires images of 12 parking spaces (Figure 5). Given the different viewpoints of the cameras, the acquired scenes are characterized by different scene geometries. We recorded two long videos per camera at $1fps$. The two videos have been acquired in different days and care has been taken to make sure covering many as possible configurations of the parking spaces, including the cases in which the parking area was empty and fully occupied. Table 1 summarizes the videos contained in the dataset and reports the number of frames of the considered videos.

For each image frame contained in the dataset, we labeled a Region of Interest specifying the area within which the parking spaces are comprised. Each image has been manually labeled to report:

- the total number of cars present in the monitored parking space;
- a bounding box around each car inside the monitored parking space;
- a binary vector, each component of which represents the status of the i-th parking space as empty (0) or non-empty (1);
- the coordinates of the four corners for each stall present in the frame (see Figure 1).

We propose two different ways of splitting the data into training and testing sets. The first split assumes that training and testing data have been acquired using a single camera. This gives rise to 6 different data subsets (one for each camera), where one of the two videos is used for training, and the other one is used for testing (subset "Nx" in Table 2). The 6 subsets are intended to assess the performance of methods when exposed to data acquired from a single camera. The second data split assumes that both training and test data have been acquired using the three cameras. In this case, we obtain two subsets (subset "X" in Table 2) where data acquired using the three cameras, but belonging to one of the two videos is used for training, while the remaining is used for test. These two subsets are intended to assess the ability of methods to generalize to different scenes.

All experiments have been performed using the Caffe library (Jia et al., 2014) on a machine equipped with four *NVIDIA GeForce TITAN X with 12Gb of DDR5 RAM*.

4.1 Evaluation Measures

To assess the discrepancy between predicted counts and ground truth counts, we evaluate the investigated approaches by computing the Absolute Errors (AE). Given a test frame I_i, the predicted count $\hat{y}_i$, and the ground truth count y_i, we compute the absolute error corresponding to I_i as follows:

$$AE_i = |y_i - \hat{y}_i| \tag{1}$$

To evaluate the performance on a set of test frames $I = \{I_i\}_{i=1}^{N}$, we also compute statistics of the AE values computed for each frame, including minimum, maximum, median and mean. In particular, we consider the standard Mean Absolute Error (MAE), which is given by:

$$MAE(I) = \frac{1}{N}\sum_{i}^{N} AE_i = \frac{1}{N}\sum_{i}^{N} |\hat{y}_i - y_i| \tag{2}$$

It should be noted that the absolute errors and the derived statistics are easy to interpret, as they are expressed with the same unit measure of the original data. E.g., a method reporting a MAE equal to 1 is, in average, overestimating or underestimating the count by 1 unit.

To evaluate the performance of the different components employed in the investigated methods (i.e., image classification and object detection), we use the most appropriate measures. Specifically, we evaluate binary image classification using accuracy (fraction of correctly classified images), whereas object detection using mean Average Precision (mAP) as proposed in (Everingham et al., 2010).

Figure 2: A satellite image (Google, 2018) of the monitored parking area located at the campus of the University of Catania.

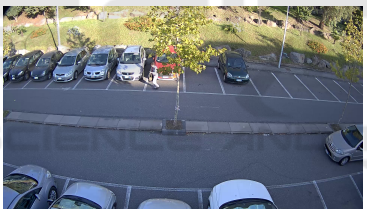

Figure 3: Camera 1 observes 12 parking spaces.

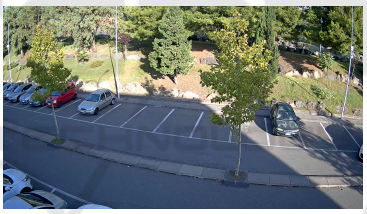

Figure 4: Camera 2 observes 14 parking spaces.

Figure 5: Camera 3 observes 12 parking spaces.

Table 2: Data subsets arising from the two considered data splits and performance of the binary stall classifier, using accuracy (fraction of correctly classified stalls); performance of the car detector is measured using standard mean Average Precision (mAP).

Subset	Training Data	Testing Data	VGG16 Accuracy	FasterRCNN mAP
Subset 1a	Video 1.1	Video 1.2	0.991	0.224
Subset 1b	Video 1.2	Video 1.1	0.987	0.381
Subset 2a	Video 2.1	Video 2.2	0.986	0.358
Subset 2b	Video 2.2	Video 2.1	0.988	0.185
Subset 3a	Video 3.1	Video 3.2	0.949	0.228
Subset 3b	Video 3.2	Video 3.1	0.989	0.340
Subset A	Videos 1.1, 2.1, 3.1	Videos 1.2, 2.2, 3.2	0.911	0.551
Subset B	Videos 1.2, 2.2, 3.2	Videos 1.1, 2.1, 3.1	0.952	0.698

4.2 Image Classification

Training and testing data for the binary classification component are obtained by extracting an image patch around each labeled stall. This is repeated for each frame of the dataset. The extracted image patches are hence assigned a binary label depending on the occupancy status of the stall: 0 for "empty" and 1 for "full". Using this procedure, we obtain a total of 17,688 samples for *Video 1.1* (10325 occupied, 7363 empty), 17,712 samples for *Video 1.2* (8463 occupied, 9249 empty), 20,636 samples for *Video 2.1*

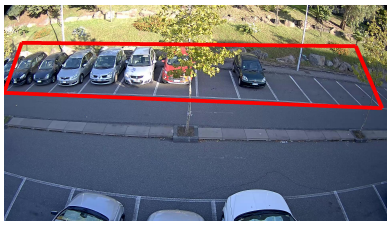

Figure 6: Region of Interest (RoI) for Camera 1.

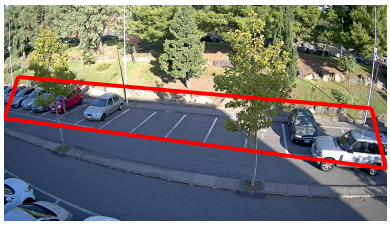

Figure 7: Region of Interest (RoI) for Camera 2.

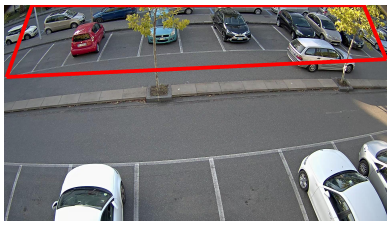

Figure 8: Region of Interest (RoI) for Camera 3.

(10530 occupied, 10106 empty), 25,676 samples for *Video 2.2* (12310 occupied, 13366 empty), 13,032 samples for *Video 3.1* (5911 occupied, 7121 empty), 20,556 samples for *Video 3.2* (11304 occupied, 9252 empty).

The binary image classification component to discriminate between "empty" and "full" stalls has been implemented by fine-tuning a VGG16 Convolutional Nerual Network (CNN) (Simonyan and Zisserman, 2014) pre-trained on ImageNet (Russakovsky et al., 2015). The fine-tuning process is carried out for 10 epochs. A different model is trained on each data subset. Table 2 reports accuracy values on the test sets for each of the considered data subsets.

4.3 Object Detection

To implement a car detector, we fine-tuned the FasterRCNN (Ren et al., 2015) object detector based on VGG16 starting from weights pre-trained on ImageNet. The training process is carried out for 70,000 iterations using a batch size of 1. As for the detection, we train a separate model for each data subset. Table 2 reports mAP values on the test sets for each of the considered data subsets.

Threshold Selection. Counting using Object Detecion as discussed in Section 3 make use of two different thresholds to set parking stalls as "empty" or "occupied" and to count cars. We set such thresholds to the values which optimize the performance of the considered method on a validation set which is formed randomly selecting 15% of the training samples. The search for optimal values is performed independently on each data subset.

When counting non-empty parking spaces we choose the values of d_1 and d_2 which minimize the MAE on the validation set. When counting cars we choose the values of r_1 and r_2 which minimize the MAE on the validation set.

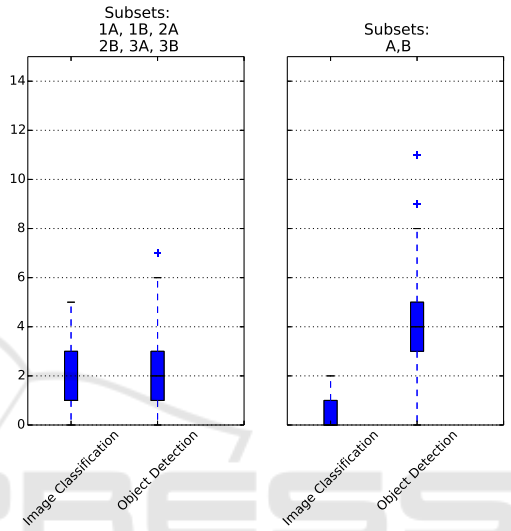

Figure 9: Box plots for counting non-empty spaces. We plot the mean absolute error for counting non-empty spaces in the single camera experiment and in the multiple camera. Higher is the worst.

4.4 RoI Selection

As stated in Section 3.2, we are interested in counting the number of cars present in a Region of Interest (RoI) of the frame. For each camera, we annotated a RoI corresponding with the area including the stalls. All methods have been tested considering only the selected RoI, hence discarding any detection result not included in the RoI. Figures 6, 7 and 8 show the RoIs considered in our experiments.

5 RESULTS

Counting Parking Spaces Results. Table 3 reports some statistics of the AE values computed for the different experiments performed on the considered data subsets. Specifically, the table reports the minimum, maximum, mean and median Absolute Error over the considered subset. We also report results for the aggregation of different data subsets. In particular, we

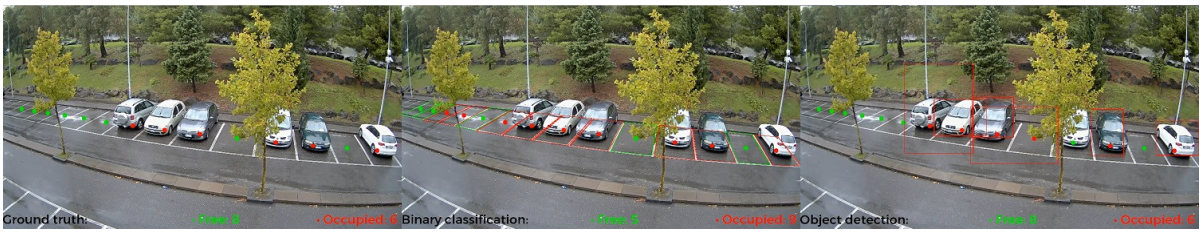

Figure 11: An example of discriminating empty and non-empty stalls with considered methods.

consider the aggregation of subsets 1a to 3b, each of which contains images acquired by a single camera, as well as the aggregation of Subsets A and B, which contain images acquired by different cameras. Figure 9 also reports box plots for the AE values contained in the two aggregated sets discussed above.

The best performance is achieved by the method based on Image Classification. This method always obtains a minimum AE equal to 0 and maximum AE values not exceeding 5 units. Median errors are often close to zero. The Mean Absolute Errors of Image Classification methods are overall significantly lower than methods based on Object Detection both in the case of single camera tests (Subsets 1a to 3b) and multiple camera tests (Subsets A and B). This observation is made particularly clear by Figure 9, in which the box plots related to the Image Classification method exhibit median values and quartile positions lower than Object Detection. Interestingly, the method based on Image Classification benefits from the presence of different geometries in the training set, allowing to further lower the MAE of 1.63 to 0.68 (compare "Subsets: 1a, 1b, 2a, 2b, 3a, 3b" to "Subsets: A, B" in Table 3 and compare left and right plots in Figure 9). On the contrary, the method based on Object Detection achieve much worse results. This observation suggests that the method based on Image Classification is more capable to generalize despite can be used only when stalls are marked. Figure 11 shows a visual example of the counting stalls results.

Counting Cars Results. Table 4 and Figure 10 report the statistics of AE values and box plots related to experiments on counting cars. It should be noted that, in this case, the method based on Image Classification should be considered as a baseline, since it is affected by a systematic error due to the fact that the number of non-empty spaces is not always equal to the number of cars (i.e., the method cannot count cars parked outside the stalls). Nevertheless, similarly to the previous experiment, the method based on Image Classification achieves the best results (MAE equal to 1.57 in the case of a single camera geometry and 1.16 for multiple camera geometries). This suggest that,

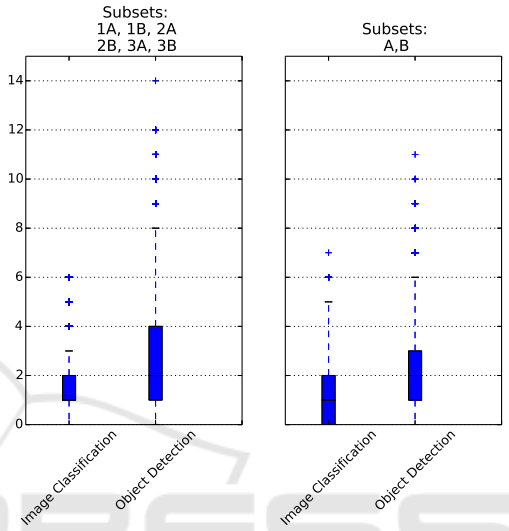

Figure 10: Box plots for counting cars. We plot the mean absolute error for counting cars in the single camera experiment and in the multiple camera. Higher is the worst

when stalls are present and their location is known, this information can be exploited to count cars. Interestingly, all the methods benefit from the presence of different camera geometries in this experiment (compare the two plots in Figure 10).

6 CONCLUSIONS

This paper investigated and compared two different approaches to count non-empty spaces and cars in parking areas. To perform the analysis, a dataset of videos has been collected in a real scenario and each frame has been labeled according to the position of parking stalls, the number of occupied stalls, and the number of cars in the frame. Results show that, when the geometry of the scene is known (i.e., stalls are marked), the system can take advantage of binary classification methods to obtain competitive results.

Table 3: Absolute Errors for the task of counting non-empty stalls. Best results for each data subset are reported in bold numbers.

		Absolute Errors			
Subset	**Method**	**Min**	**Max**	**Mean (MAE)**	**Median**
Subsets: 1a, 1b, 2a 2b, 3a, 3b	Image Classification	**0.00**	**5.00**	**1.63**	**1.00**
	Object Detection	**0.00**	7.00	2.15	2.00
Subset 1a	Image Classification	**0.00**	**3.00**	**1.48**	**1.00**
	Object Detection	**0.00**	7.00	4.20	4.00
Subset 1b	Image Classification	**0.00**	**2.00**	**0.54**	**0.00**
	Object Detection	**0.00**	5.00	2.26	3.00
Subset 2a	Image Classification	**0.00**	**3.00**	2.49	3.00
	Object Detection	**0.00**	5.00	**1.54**	**1.00**
Subset 2b	Image Classification	**0.00**	5.00	3.22	4.00
	Object Detection	**0.00**	**4.00**	**2.04**	**2.00**
Subset 3a	Image Classification	**0.00**	**2.00**	**0.88**	**1.00**
	Object Detection	**0.00**	4.00	1.28	**1.00**
Subset 3b	Image Classification	**0.00**	**3.00**	**0.86**	**1.00**
	Object Detection	**0.00**	**3.00**	1.82	2.00
Subsets: A, B	Image Classification	**0.00**	**2.00**	**0.56**	**0.00**
	Object Detection	**0.00**	11.00	4.23	4.00
Subset A	Image Classification	**0.00**	**1.00**	**0.40**	**0.00**
	Object Detection	**0.00**	8.00	4.09	4.00
Subset B	Image Classification	**0.00**	**2.00**	**0.68**	**0.00**
	Object Detection	**0.00**	11.00	4.34	4.00

Table 4: Absolute Errors for the task of counting cars. Best results for each data subset are reported in bold numbers.

		Absolute Errors			
Subset	**Method**	**Min**	**Max**	**Mean (MAE)**	**Median**
Subsets: 1a, 1b, 2a 2b, 3a, 3b	Image Classification	**0.00**	**6.00**	**1.57**	**1.00**
	Object Detection	**0.00**	14.00	2.83	2.00
Subset 1a	Image Classification	**0.00**	**6.00**	**1.44**	**1.00**
	Object Detection	**0.00**	11.00	2.38	2.00
Subset 1b	Image Classification	**0.00**	**4.00**	**0.56**	**0.00**
	Object Detection	**0.00**	11.00	3.43	3.00
Subset 2a	Image Classification	**0.00**	**6.00**	2.37	2.00
	Object Detection	**0.00**	10.00	2.30	2.00
Subset 2b	Image Classification	**0.00**	**6.00**	**3.15**	**4.00**
	Object Detection	**0.00**	14.00	5.03	5.00
Subset 3a	Image Classification	**0.00**	**5.00**	**0.85**	**1.00**
	Object Detection	**0.00**	7.00	1.92	2.00
Subset 3b	Image Classification	**0.00**	**2.00**	**0.74**	**1.00**
	Object Detection	**0.00**	9.00	1.98	**1.00**
Subsets: A, B	Image Classification	**0.00**	**7.00**	**1.16**	**1.00**
	Object Detection	**0.00**	11.00	1.95	2.00
Subset A	Image Classification	**0.00**	**5.00**	**0.87**	**1.00**
	Object Detection	**0.00**	9.00	1.77	**1.00**
Subset B	Image Classification	**0.00**	**7.00**	**1.39**	**1.00**
	Object Detection	**0.00**	11.00	2.10	2.00

REFERENCES

Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., and Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, 72(15):327–334.

Arteta, C., Lempitsky, V., Noble, J. A., and Zisserman, A. (2014). Interactive Object Counting. In *European Conference on Computer Vision*, pages 1–15.

Arteta, C., Lempitsky, V., and Zisserman, A. (2016). Counting in the wild. In *European Conference on Computer Vision*, pages 483–498.

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

Battiato, S., Farinella, G. M., Gallo, G., and Giudice, O. (2018). On-board monitoring system for road traffic safety analysis. *Computers in Industry*, 98:208 – 217.

Chan, A. B., Liang, Z. S.-J., and Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE.

Chen, S., Fern, A., and Todorovic, S. (2015). Person count localization in videos from noisy foreground and detections. In *Conference on Computer Vision and Pattern Recognition*, pages 1364–1372. IEEE.

Davis, H. and Pérusse, R. (1988). Numerical competence in animals: Definitional issues, current evidence, and a new research agenda. *Behavioral and Brain Sciences*, 11(4):561–579.

De Almeida, P. R., Oliveira, L. S., Britto, A. S., Silva, E. J., and Koerich, A. L. (2015). PKLot-A robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949.

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

Eiselt, A.-K. and Nieder, A. (2013). Representation of abstract quantitative rules applied to spatial and numerical magnitudes in primate prefrontal cortex. *The Journal of neuroscience: the official journal of the Society for Neuroscience*, 33(17):7526–34.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

Ferrari, A., Lombardi, S., and Signoroni, A. (2017). Bacterial colony counting with convolutional neural networks in digital microbiology imaging. *Pattern Recognition*, 61:629–640.

Fiaschi, L., Koethe, U., Nair, R., and Hamprecht, F. A. (2012). Learning to count with regression forest and structured labels. In *International Conference on Pattern Recognition*, pages 2685–2688.

Google (2018). Google Maps. https://www.google.it/maps/@37.5264537,15.0741852,230m/data=!3m1!1e3?hl=en. [Online; accessed 12-March-2017].

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

Kaufman, E. L., Lord, M. W., Reese, T. W., and Volkmann, J. (1949). The discrimination of visual number. *The American journal of psychology*, 62(4):498–525.

Lempitsky, V. and Zisserman, A. (2010). Learning to Count Objects in Images. In *Advances in Neural Information Processing Systems*, pages 1324–1332.

Li, M., Zhang, Z., Huang, K., and Tan, T. (2008). Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection. In *International Conference on Pattern Recognition*, pages 1–4.

Loy, C. C., Chen, K., Gong, S., and Xiang, T. (2013). *Crowd Counting and Profiling: Methodology and Evaluation*, pages 347–382. Springer, New York, NY.

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

Nieder, A. (2016). The neuronal code for number. *Nature Reviews Neuroscience*, 17(6):366–382.

Rabaud, V. and Belongie, S. (2006). Counting Crowded Moving Objects. In *Conference on Computer Vision and Pattern Recognition*, pages 705–711. IEEE.

Raymond, J.-F. (2001). Traffic analysis: Protocols, attacks, design issues, and open problems. In *Designing Privacy Enhancing Technologies*, pages 10–29.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

United Nations (2008). World urbanization prospects: The 2007 revision. *United Nations Publications*.

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

Zhang, C., Li, H., Wang, X., and Yang, X. (2015). Cross-scene crowd counting via deep convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 833–841. IEEE.