Counting Vehicles with Cameras

Luca Ciampi[1], Giuseppe Amato[1], Fabrizio Falchi[1], Claudio Gennaro[1], and Fausto Rabitti[1]

Institute of Information, Science and Technologies of the National Research Council of Italy (ISTI-CNR), via G. Moruzzi 1, 56124 Pisa, Italy

Abstract. This paper aims to develop a method that can accurately count vehicles from images of parking areas captured by smart cameras. To this end, we have proposed a deep learning-based approach for car detection that permits the input images to be of arbitrary perspectives, illumination, and occlusions. No other information about the scenes is needed, such as the position of the parking lots or the perspective maps. This solution is tested using Counting CNRPark-EXT, a new dataset created for this specific task and that is another contribution to our research. Our experiments show that our solution outperforms the state-of-the-art approaches.

Keywords: Counting · Convolutional Neural Networks · Machine Learning · Deep Learning

1 Introduction

This paper is motivated by the need to address the challenging real-world counting problem related to the estimation of the number of vehicles present in a park area, using images captured by smart cameras. The visual understanding of the collected images must face many challenges, perhaps common to all the counting tasks, like variations in scales and perspectives, the inter-object occlusions, the non-uniform illumination of the scene, and many others.

In order to address these challenges, we propose a deep learning-based approach that is able to accurately count cars in images *without* any extra information of the scenes, like the position of the parking lots or the perspective map. The latter aspect is a key feature since in this way our solution is directly applicable in unconstrained contexts.

To validate our approach, we also built a dataset, called *Counting CNRPark-EXT* dataset, collecting images from the parking lots in the campus of the National Research Council (CNR) in Pisa. The images are taken by nine different cameras in challenging conditions since they are captured from different perspectives, they present different illuminations and many occlusions. The result of the proposed methodology significantly outperforms the ones obtained using the state-of-the-art baseline methods.

SEBD 2018, June 24-27, 2018, Castellaneta Marina, Italy. Copyright held by the author(s).

2 Related Work

Objects counting has been tackled in computer vision by various techniques, especially for the estimation of the number of people in crowded scenes. Following the taxonomies adopted in [11] and [14] we can divide counting approaches into four main categories: counting by detection, counting by clustering, counting by regression and counting by density estimation.

Counting by *detection* is a supervised approach where a sliding window detector (i.e. a classifier that is slid over the entire image) previously trained is used to detect objects in the scene. This information is then used to count the number of objects. In the *monolithic* detection the classifier is trained in order to recognize the whole object we want to detect [7], while in the *part-based* detection we looking for specific parts of the object (such as head and shoulders for people detection) [9]. Finally, in the *shape-matching* detection, the classifier is about object shapes, for example composed of ellipses [15]. Even if these methods are quite simple to understand, they suffer in scenes with occlusions.

Counting by *clustering* tackles the counting problem in an unsupervised way. A clear advantage is that such an approach does not need to be trained and it is out of the box. However, the counting accuracy of such fully unsupervised methods is in general limited. The clustering by *self-similarities* technique relies on tracking simple image features and probabilistically group them into clusters, like in [1]. Then we can count clusters belonging to a certain category. The clustering by *motion similarities* approach relies instead on the assumption that a pair of points that appears to move together is likely to be part of the same individual, hence coherent feature trajectories can be grouped together to represent independently moving-entities, like in [13]. The main drawback in such a method is that it only works with continuous image frames and not with static images.

Counting by *regression* is a supervised method that tries to establish a direct mapping (linear or not) from the image features to the number of objects present in an image without explicit object detection or tracking. Since it does not rely on a specific classifier or model previously trained, it is more robust to occlusions and perspective distortions.

Finally, counting by *density estimation* is a supervised technique that extends in some way counting by regression approach, introduced in [8]. In this case, the (linear or not) mapping is between image features and a corresponding density map (i.e. a continuous-valued function), and not between features and the number of objects. Then we can calculate the integral over any region in the density map obtaining the count of objects within that region. This approach is robust to occlusions and perspective distortions because it does not rely on a specific classifier previously trained, just like counting by regression approach as well, but the key difference is that now we are exploiting a *pixel level mapping*: each pixel of the image is represented by a feature vector and mapped to a pixel of the corresponding density map. In some way, unlike in counting by regression, now we are incorporating spatial information in the learning process.

The extraction of suitable features is a crucial operation for all the four approaches that have been described. Since, in general, handcrafted features suffer a drop in accuracy when subjected to challenging situations (variances in illumination, perspective distortion, severe occlusion, etc.), most state-of-the-art counting methods employ deep-learning approaches, in particular exploiting Convolutional Neural Networks (CNNs), in order to extract suitable task-specific features automatically. Some works that use CNNs are [4] and [3] for crowd counting, and [12] for the vehicles counting task.

3 Dataset

A contribution of this paper is the creation of *"Counting CNRPark-EXT"*, a dataset of roughly 4,000 images containing more than 79,000 labeled cars. This dataset is based on the *CNRPark-EXT* dataset, presented in [2], a collection of roughly 150,000 bounding-box annotated images of vacant and occupied parking slots in the campus of the National Research Council (CNR). This dataset is challenging and describes 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.

The *CNRPark-EXT* dataset is specifically designed for parking lot occupancy detection and it is not directly usable for the counting task, since each image, called *patch*, contains one single park space labeled according to the occupancy status of it, 0 for vacant and 1 for occupied. Since the purpose of this work is to count cars present in an image, the information needed to build the ground truth are full images (and not patches), the total number of vehicles considered in an entire scene, and, at least for some techniques, the locations of these vehicles. Therefore, starting from the *CNRPark-EXT* dataset, we summed up the occupancy status of the patches belonging to a same full image (note that a car space is occupied if and only if a car is present in the car space), obtaining the total number of cars considered in the whole scenes (i.e. obtaining a per-image ground truth). Only bounding boxes corresponding to occupied spaces are considered, identifying cars to be counted (i.e. obtaining a per-object ground truth). Table 1 reports the composition of *Counting CNRPark-EXT* dataset.

Table 1. Details of the dataset used in the experiments, with the various proposed subsets.

Subset	Number Images	Number Slots	Number Cars
All	4.081	144.965	79.307
Train	2.661	94.493	47.616
Validation	524	18.647	13.415
Test	896	31.825	18.276

4 A new vehicles counting solution using Mask R-CNN

Our solution is based on *Mask R-CNN* [6], 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 [5], taking an 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 [10], 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 and bounding boxes. As mentioned before, since the labels of our dataset are bounding boxes (perhaps not very accurate, since they localize parking lots and not the cars), we needed to add the masks on the vehicles to be detected in order to make *Counting CNRPark-EXT* dataset useful for our purposes. Mask creation is a very time-expensive operation - for each car in each image, we should associate the pixels that model the vehicles with a label. Since in the dataset we have hundreds of cars, this problem has been solved by taking advantage of the output of the pre-trained model of *Mask R-CNN*. Since this network produces in output accurate masks localizing objects, the idea is to save these masks automatically generated, parse them, and then manually add some of the remaining masks for the cars that were not detected.

First, we randomly selected a subset of our training set (about the 10%), fed the previous pre-trained Mask R-CNN model with these images, and saved the vertex coordinates of the polygons surrounding the masks localizing objects that the network produces in output along with the category associated with them. Next, a further analysis of these saved masks has been performed, changing the wrong associations between some objects labeled with categories that are not possible in our context (like *airplane* and *boat*), and that were instead vehicles we want to detect. In this way, we forced the network to learn to recognize these objects as cars and not as other wrong objects. Finally, we manually add some of the remaining masks localizing vehicles that were not automatically detected. At this point, we retrained the network using this new mask-labeled and task-specific dataset, frozen the weights of the backbone, and saving the new weights of the head after a few epochs. Figure 1 reports the pipeline of the described approach.

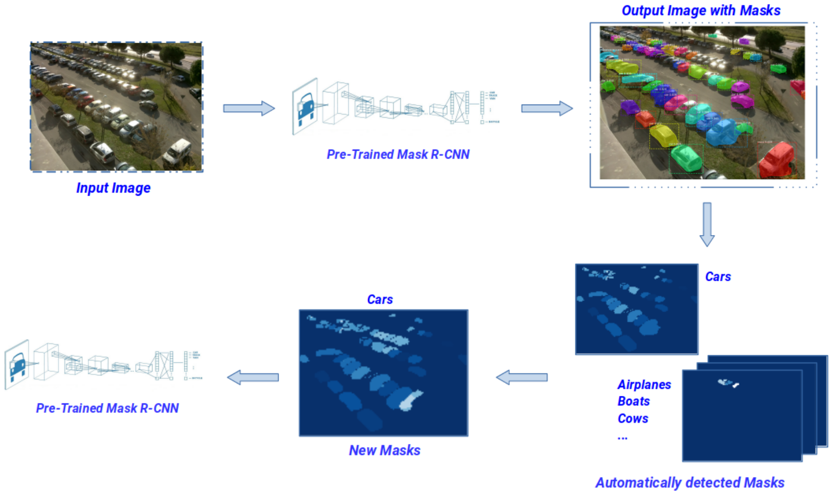

Fig. 1. Pipeline of our approach: we generate labels for the training set saving masks automatically generated by a pre-trained version of Mask R-CNN, parsing them and manually adding the missing ones. Then we use this annotated-dataset in order to re-train Mask R-CNN, specializing it in the detection of the vehicles we want to count.

5 Evaluation

In this section, we present the methodology and the results of the experimental evaluation. For testing purposes, we have used the test subset of the *Counting CNRPark-EXT* dataset.

Evaluation Metrics Following other counting benchmarks, we use Mean Absolute Error (MAE) and Root Mean Square Error ($RMSE$) as the metrics for comparing the performance of our solution against other counting approaches present in literature. MAE is defined as follows:

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

while RMSE is defined as:

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

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 [11], 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:

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

where num_slots_n is the total number of parking lots in the current scene. In the next, this evaluation metric is expressed as a percentage.

Comparisons with the state of the art We have compared our vehicles counting solution against the method proposed in [2], a state-of-the-art approach for car parking occupancy detection, based on *mAlexNet*, a deep CNN specifically designed for smart cameras. This work represents an indirect method for counting cars in a car park, where the counting problem is cast as a classification problem: if a parking lot is occupied we increment the total number of cars, otherwise not.

The main drawback of such an approach is that the locations of the monitored objects of a scene must be known in advance. This technique fails if it is applied on a new camera added in the car park, just because it does not have the knowledge about where car slots are located: a preliminary annotation of the new camera and a new train of the network are then mandatory operations. So, this issue makes this car counting method not directly applicable in unconstrained contexts.

On the other hand, as already mentioned before, our counting approach is able to count vehicles without any extra information about the parking lots locations. Nevertheless, results using this classification approach applied to the *Counting CNRPark-EXT* dataset can be used as a baseline and as a basis of comparison.

Results The results of the experimental evaluation are reported in Table 2. Note that our solution not only outperforms the original pre-trained Mask-RCNN, but also it works better than the state-of-the-art approach using *mAlexNet*, considering all the three performance metrics.
Since counting errors could be due to different weather conditions (since different weather conditions might produce significant different illuminations of the scenes) and/or to different camera views (different camera views correspond to different perspectives), we perform a further analysis of the results by dividing them according to the number of cameras and to the weather condition they belong to. Figure 2 reports the errors scatter-plots of the errors. Note that, in general, our solution tends to underestimate the number of vehicles present in

Table 2. Results in terms of MAE, RMSE and MOE.

Method	MAE	RMSE	MOE
mAlexNet	1.34	2.83	4.17%
Mask R-CNN	1.56	2.89	5.23%
Re-trained Mask R-CNN	1.05	2.1	3.64%

the scene, but it responds well to perspective and illumination changes, having only a small performance decrease considering frames belonging to camera nine.

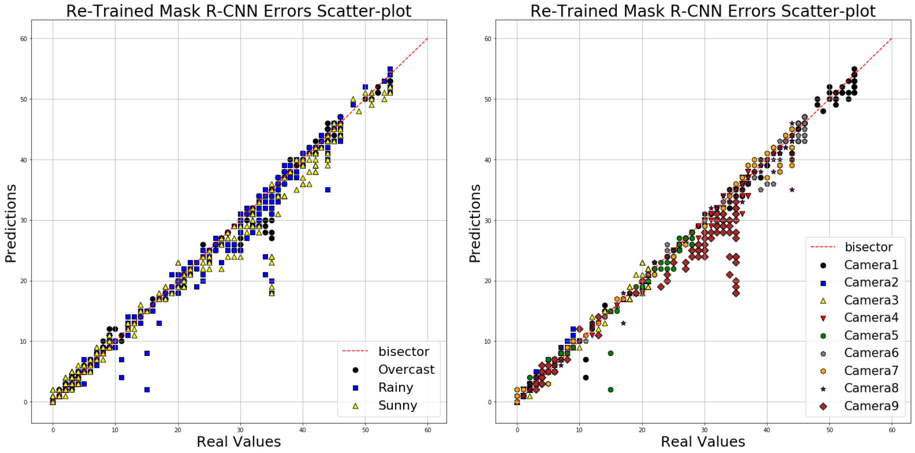

Fig. 2. Scatter-plots of the errors.

6 Conclusions

In this paper, we presented an efficient solution for counting vehicles in parking areas that exploit deep Convolutional Neural Networks (CNNs) to detect vehicles present in challenging scenes. Our proposed methodology does not require any manually entered information about the parking lots locations, allowing a simple 'plug-and-play' installation. The results outperform the ones obtained using the state-of-the-art baseline methods.

As a further contribution, we collected *Counting CNRPark-EXT*, a dataset containing images of real parking areas captured by nine cameras, with different weather conditions and perspective views.

References

1. Ahuja, N., Todorovic, S.: Extracting texels in 2.1 d natural textures. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. pp. 1–8. IEEE (2007)
2. Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., Vairo, C.: Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications **72**, 327–334 (2017)
3. Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 2016 ACM on Multimedia Conference. pp. 640–644. ACM (2016)
4. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Computer vision and pattern recognition (CVPR), 2012 IEEE conference on. pp. 3642–3649. IEEE (2012)
5. Gu, C., Lim, J.J., Arbeláez, P., Malik, J.: Recognition using regions. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 1030–1037. IEEE (2009)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
7. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 878–885. IEEE (2005)
8. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Advances in neural information processing systems. pp. 1324–1332 (2010)
9. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. pp. 1–4. IEEE (2008)
10. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
11. Loy, C.C., Chen, K., Gong, S., Xiang, T.: Crowd counting and profiling: Methodology and evaluation. In: Modeling, Simulation and Visual Analysis of Crowds, pp. 347–382. Springer (2013)
12. Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision. pp. 615–629. Springer (2016)
13. Rabaud, V., Belongie, S.: Counting crowded moving objects. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 1, pp. 705–711. IEEE (2006)
14. Sindagi, V.A., Patel, V.M.: A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognition Letters (2017)
15. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. IEEE transactions on pattern analysis and machine intelligence **30**(7), 1198–1211 (2008)