# A multi-camera solution for counting vehicles on the edge

Luca Ciampi*, Claudio Gennaro*, Fabio Carrara*, Fabrizio Falchi*, Claudio Vairo* and Giuseppe Amato*
*Institute of Information Science and Technologies - National Research Council - Pisa, Italy

***Abstract*—Smart mobility applications, such as intelligent parking and road traffic management, are nowadays widely employed worldwide, making our cities more livable, bringing benefits to our lives, reducing costs, and improving energy usage. We propose a multi-camera system to automatically count vehicles in a parking lot using images captured by smart cameras. Unlike most of the literature on this task, which focuses on the analysis of *single* images, this paper proposes the use of multiple visual sources to monitor a wider parking area from different perspectives. Experiments show that our solution is robust, flexible, and can benefit from redundant information from different cameras while improving overall performance.**

## I. Introduction

In this work, we propose a novel solution to automatically estimate the number of vehicles present in a parking lot using images captured by smart cameras. This counting task is challenging as the process of understanding the captured images faces many problems, such as shadows, light variation, weather conditions, and inter-object occlusions [1]. Most of the existing works concerning the vehicles counting task focus on the analysis of *single* images [2]–[4]. However, in many real-world scenarios, one can benefit from using multiple cameras to monitor the same parking lot from different perspectives and viewpoints. Furthermore, multiple neighboring cameras can also be helpful to cover a wider area. At the same time, such an approach introduces issues related to merging the knowledge extracted from the single cameras with partially overlapping fields of views (FOVs).

In this paper, we discuss a multi-camera system that combines a CNN-based technique, in charge of locate and count vehicles present in images belonging to individual cameras, along with a decentralized geometry-based approach, responsible for merge the data from nearby devices with an overlapping field of view and, finally, estimate the number of cars present in the *entire* parking lot. Our solution performs the task directly on the edge devices (i.e., the smart cameras) without using a central server or cloud, consequently reducing the communication overhead. Hence, our system scales better when the number of monitored parking spaces increases. Moreover, our solution does not require any extra information about the monitored parking area, such as the location of the parking spaces, nor any geometric information about the camera positions in the parking lot. In short, it is a flexible and ready-to-use solution that allows a simple "plug-and-play" insertion of new cameras into the system, and that takes advantage of the redundant information deriving from the different visual sources.

This work was partially supported by H2020 projects AI4EU under GA 825619 and AI4media under GA 951911.

## II. Proposed Method

We model our system as a graph $G$, comprised of $n$ nodes $\nu_i$ and one Sink node $S$, $V = \{\nu_1, \nu_2, \ldots, \nu_n, S\}$. An example is shown in Figure 1. Each node $\nu_i$ represents an independent edge device, i.e., a smart camera in our case. Two nodes $\nu_i$ and $\nu_j$ are considered neighbors if their FOVs overlap, and in this case, a directed edge of the graph connects them. Each edge device $\nu_i$ can capture images, localize and count the vehicles present in its FOV exploiting a deep learning-based detector, and communicate with its neighboring nodes through messages $m_i$ containing the computed detections. Furthermore, each node $\nu_i$ can also run a local counting algorithm in charge of computing partial counting results concerning the estimation of the number of vehicles present in overlapped areas between its FOV and the ones belonging to its neighbors. The fusion of the partial results is performed by the Sink node $S$, which provides the final result and synchronizes all the algorithm steps through synchronization signals headed towards the other nodes $\nu_i$. On the other hand, the nodes $\nu_i$ can also communicate through messages with the Sink node. They can be of two types: i) messages $\eta_i$ containing the number of cars captured by the node $\nu_i$ in its FOV, and ii) messages $\mu_{j,i}$ representing the partial counting estimation related to the overlapping area between two neighboring nodes $\nu_i$ and $\nu_j$.

In the next, we describe the steps of our solution. Firstly, the node $S$ starts the initialization phase of the system, sending a synchronization signal to all the other nodes. Once received, each smart camera captures an image of the scene it monitors and sends it to all its neighbors. Once a smart camera $i$ receives an image from a neighboring camera $j$, it computes a homographic transformation $H_{j,i}$ between the image $j$ and the image $i$ describing its monitored scene. This allows us to establish a correspondence between the points belonging to the pair of images taken by the two cameras, which will be used subsequently in the algorithm. Then, each node $\nu_i$ exploits a CNN-based counting technique that analyzes the monitored view and outputs a set of masks $\text{masks}_i$ localizing the vehicles present in the scene. The cardinality of this set of masks corresponds to the number of cars present in the monitored area, i.e., the quantity $\eta_i$, that is sent through a message to the Sink node $S$. Next, the node $\nu_i$ packs this set of masks $\text{masks}_i$ in a message $m_i$, sending it to all its neighboring nodes

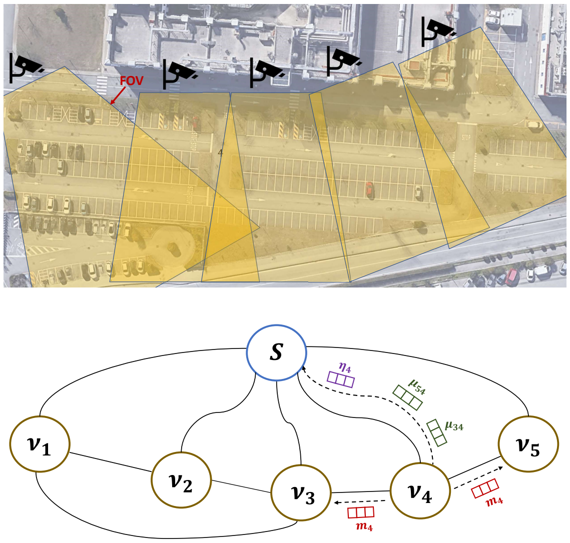


Fig. 1: An example of our multi-camera counting system with $n = 5$ smart cameras, together with its graph modeling.

$\nu_j$, and receiving from them their corresponding set of masks masks$_j$ packed in a message $m_j$. Once received a message $m_j$, the node $\nu_i$ is responsible for analyzing the potential vehicles present in the overlapped area between its FOV and one of the nodes $\nu_j$. To this end, it employs the homographic transformation $H_{j,i}$ computed during the system initialization, projecting the masks belonging to the set masks$_j$ into its image plane, filtering them and discarding the ones that overlap with the masks belonging to the set masks$_i$ having a value of Intersection over Union (IoU) greater than a fixed threshold. These masks indeed localize vehicles already detected, and that should not be considered a second time. On the other hand, the cars left after this filtering are vehicles that were not detected in the FOV underlying the node $\nu_i$, but instead found by the node $\nu_j$, probably because of having a better view of this object. Referring to our graph modeling the system and reported in Figure 1, the number of the discarded cars after this filtering operation corresponds to the message $\mu_{j,i}$, that is sent to the Sink node $S$. Finally, the Sink node $S$ starts the final phase. In particular, for each overlapped area shared between a pair of nodes $\nu_i, \nu_j$, the node $S$ receives two messages $\mu_{j,i}$ and $\mu_{i,j}$, the contents of which are computed by the two nodes employing two homographic transformations $H_{j,i}$ and $H_{i,j}$, respectively. These two quantities can be potentially different. We choose the best value aggregating them, choosing between three different functions - max, min and mean, finding that the latter is the best one. Lastly, the node $S$ builds the final result, i.e., the estimation of the number of vehicles present in the *entire* parking lot, by summing up all the $\eta_i$, and subtracting the aggregated values.

## III. Experimental Evaluation

To validate our solution, we employ the *CNRPark-EXT* dataset [5], a collection of images taken from the parking lot on the campus of the National Research Council (CNR) in Pisa, Italy. The pictures are acquired by multiple cameras having partially overlapping fields of views and describing challenging scenarios, with different perspectives, illuminations, weather conditions, and many occlusions. Since the annotations of this dataset concern single images, we extended it by relabeling a part of it to be consistent with our algorithm that instead considers the entire parking area.

We compare our solution against a system that is not aware of the other cameras' overlapped areas, and so it just sums all the vehicles detected by all the cameras belonging to a sequence (Baseline). Then, we consider a more conservative approach, where the nodes employ the homographic transformations only to black-masking the overlapped areas (Simplified algorithm). This latter baseline then loses the ability to take advantage of monitoring the same lots from different views. However, it is still aware of the locations of the overlapping areas, and it considers the vehicles inside them only once. Table III shows the results terms of Mean Absolute Error, Mean Squared Error and Mean Relative Error (the lower is better).

| | *MAE* | *MSE* | *MRE (%)* |
|---|---|---|---|
| *Baseline* | 111.6 | 12,736.6 | 63.9 |
| *Simplified algorithm* | 1,351.6 | 20.7 | 2.86 |
| *Our Method* | **2.8** | **10.5** | **1.6** |

## IV. Conclusions

This paper presented a distributed artificial intelligence-based system that automatically counts the vehicles present in an *entire* parking area using images taken by multiple smart cameras. The main peculiarities of this approach are that all the computation is performed in a distributed manner at the edge of the network and that there is no need for any extra information of the monitored parking area, such as the location of the parking spaces, nor any geometric information about the position of the cameras in the parking lot. Our solution is simple but effective, combining a deep-learning technique with a distributed geometry-based approach. We evaluated our algorithm on the CNRPark-EXT dataset, which we specifically extended to show how we benefit from redundant information from different cameras while improving overall performance.

## References

[1] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Advances in neural information processing systems*, vol. 23, pp. 1324–1332, 2010.

[2] G. Amato, L. Ciampi, F. Falchi, and C. Gennaro, "Counting vehicles with deep learning in onboard uav imagery," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1–6.

[3] L. Ciampi, C. Santiago, J. P. Costeira, C. Gennaro, and G. Amato, "Domain adaptation for traffic density estimation." in *VISIGRAPP (5: VISAPP)*, 2021, pp. 185–195.

[4] L. Ciampi, G. Amato, F. Falchi, C. Gennaro, and F. Rabitti, "Counting vehicles with cameras." in *SEBD*, 2018.

[5] 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*, vol. 72, pp. 327–334, 2017.