Traffic Density Estimation via Unsupervised Domain Adaptation

(Discussion Paper)

Luca Ciampi[1], Carlos Santiago[2], Joao Paulo Costeira[2], Claudio Gennaro[1] and Giuseppe Amato[1]

[1]*Institute of Information Science and Technologies - National Research Council - Pisa, Italy*
[2]*Instituto Superior Técnico (LARSyS/IST) - Lisbon, Portugal*

Abstract

Monitoring traffic flows in cities is crucial to improve urban mobility, and images are the best sensing modality to perceive and assess the flow of vehicles in large areas. However, current machine learning-based technologies using images hinge on large quantities of annotated data, preventing their scalability to city-scale as new cameras are added to the system. We propose a new methodology to design image-based vehicle density estimators with few labeled data via an unsupervised domain adaptation technique.

Keywords

Unsupervised Domain Adaptation, Synthetic Datasets, Deep Learning, Counting Vehicles,

1. Introduction

Traffic problems are constantly increasing, and tomorrow's cities can only be smart if they enable smart mobility. This concept is becoming more critical since traffic congestion caused by the increasing number of people using different road infrastructures to travel anywhere is imposing extra costs that make all activities more expensive and put a damper on the development.

Smart mobility applications such as smart parking and road traffic management are nowadays widely employed worldwide, making our cities more livable and bringing benefits to the cities, a better quality of our life, reducing costs, and improving energy usage.

Images are probably the best sensing modality to perceive and assess the flow of vehicles in large areas. Like no other sensing mechanism, networks of city cameras can observe such large dimensions and simultaneously provide visual data to AI systems to extract relevant information from this deluge of data.

In this work, we propose a CNN-based system that can estimate traffic density and count the vehicles present in urban scenes directly on-board smart city cameras, analyzing the images captured by themselves.

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

luca.ciampi@isti.cnr.it (L. Ciampi)
0000-0002-6985-0439 (L. Ciampi); 0000-0002-4737-0020 (C. Santiago); 0000-0001-6769-2935 (J. P. Costeira); 0000-0002-3715-149X (C. Gennaro); 0000-0003-0171-4315 (G. Amato)

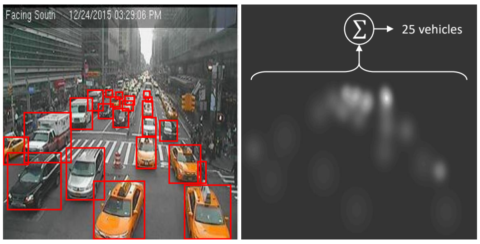

Figure 1: Example of an image with the bounding box annotations (left) and the corresponding density map that sums to the counting value (right).

Current systems address the counting problem as a supervised learning process. They fall into two main classes of methods: a) detection-based approaches [1][2][3] that try to identify and localize single instances of objects in the image and b) density-based techniques that rely on regression techniques to estimate a density map from the image, and where the final count is given by summing all pixel values [4]. Figure 1 illustrates the mapping of such regression. Concerning vehicle counting in urban spaces, where images are of low resolution, and most objects are partially occluded, density-based methods have a clear advantage on detection methods [5][6].

However, since this class of approaches requires pixel-level ground truth for supervised learning, they may not generalize well to unseen images, especially when there is a large *domain gap* between the training (*source*) and the test (*target*) sets, such as different camera perspectives, weather, or illumination. The direct transfer of the learned features between different domains does not work very well because the distributions are different. Thus, a model trained on the source domain usually experiences a drastic drop in performance when applied to the target domain. This problem is commonly referred to as *Domain Shift* [7], and it severely hampers the application of counting methods to very large-scale scenarios since annotating images for all the possible cases is unfeasible.

To mitigate this problem, we introduce a methodology that performs *Unsupervised Domain Adaptation* (UDA) among different scenarios. UDA techniques address the domain shift taking a source labeled dataset and a target *unlabeled* one. The challenge here is to automatically infer some knowledge from the target data to reduce the gap between the two domains. Specifically, in this work, we propose an end-to-end CNN-based UDA algorithm for traffic density estimation and counting, based on adversarial learning performed directly on the generated density maps, i.e., in the *output space*, given that in this specific case, the output space contains valuable information such as scene layout and context. We focus on vehicle counting, but the approach is suitable for counting any other types of objects.

Another contribution of this work is represented by the creation of two new per-pixel annotated datasets made available to the scientific community. One of the two novel datasets is a collection of synthetic images taken from a photo-realistic video game where the labels are automatically assigned while interacting with the API of the graphical engine. We conducted our experiments considering these two datasets and another collection of images already present in the literature, validating our approach over different types of domain shifts: i) the *Camera2Camera* domain shift, where the source images belong to some specific cameras, and the target ones are instead taken from different perspectives and context; ii) the *Day2Night* domain shift, where the source domain is represented by images taken during the day and the target domain by pictures taken at night; iii) the *Synthetic2Real* domain shift, where source images are collected using a video game and automatically annotated, while the target ones are real urban pictures. Experiments show a significant improvement compared to the performance of the model without domain adaptation.

2. The Datasets

This section describes the datasets exploited in this work, focusing mainly on the two novel datasets created on purpose in this work.

2.1. NDISPark Dataset

The *NDISPark - Night and Day Instance Segmented Park* dataset is a small, manually annotated dataset for counting cars in parking lots, consisting of about 250 images. This dataset is challenging and describes the most difficult situations that can be found in a real scenario: seven different cameras capture the images under various weather conditions and angles of view. Furthermore, it is worth noting that pictures are taken during the day and the night, showing utterly different light conditions. The images are precisely annotated with *instance* segmentation labels, and this allowed us to generate accurate ground truth density maps usable for the counting task.

2.2. GTA Dataset

The *GTA - Grand Traffic Auto* dataset is a vast collection of about 15,000 *synthetic* images of urban traffic scenes collected from the highly photo-realistic video game *GTA V - Grand Theft Auto V*. We deploy a framework that can *automatically* and precisely annotate the vehicles present in the scene with per-pixel annotations. To the best of our knowledge, it is the first *instance* segmentation synthetic dataset of city traffic scenarios. Figure 2 shows some examples of images belonging to this dataset together with the annotations.

2.2.1. WebCamT Dataset

The *WebCamT* dataset is a collection of traffic scenes recorded using city-cameras introduced by [6]. It is particularly challenging for analysis due to the low-resolution (352×240), high occlusion, and large perspective. We considered images belonging to different cameras and consequently having different views.

3. Proposed Method

Our method relies on a CNN model trained end-to-end with adversarial learning in the output space (i.e., the density maps), which contains rich information such as scene layout and context.

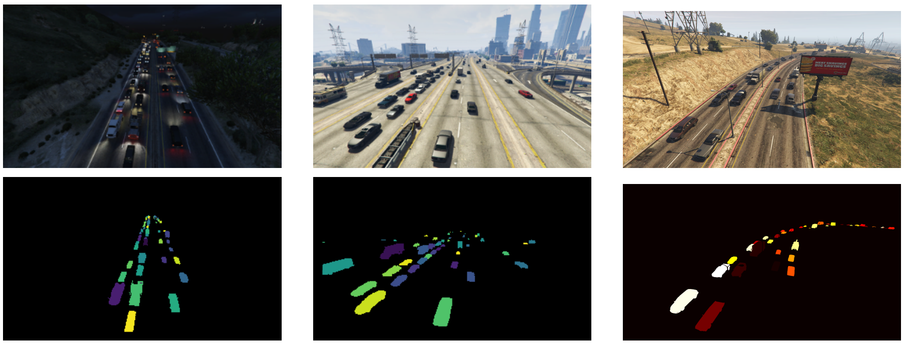

Figure 2: Some examples of images of our *Grand Traffic Auto* dataset, together with the *automatically* generated instance segmentation annotations.

The peculiarity of our adversarial learning scheme is that it forces the predicted density maps in the target domain to have local similarities with the ones in the source domain.

Figure 3 depicts the proposed framework consisting of two modules: 1) a CNN that predicts traffic density maps, from which we estimate the number of vehicles in the scene, and 2) a discriminator that identifies whether a density map (received by the density map estimator) was generated from an image of the source domain or the target domain.

In the training phase, the density map predictor learns to map images to densities based on annotated data from the source domain. At the same time, it learns to predict realistic density maps for the target domain by trying to fool the discriminator with an adversarial loss. The discriminator's output is a pixel-wise classification of a low-resolution map, as illustrated in Figure 3, where each pixel corresponds to a small region in the density map. Consequently, the output space is forced to be locally similar for both the source and target domains. In the inference phase, the discriminator is discarded, and only the density map predictor is used for the target images. We describe each module and how it is trained in the following subsections.

3.1. Density Estimation Network

We formulate the counting task as a density map estimation problem [4]. The density (intensity) of each pixel in the map depends on its proximity to a vehicle centroid and the size of the vehicle in the image so that each vehicle contributes with a total value of 1 to the map. Therefore, it provides statistical information about the vehicles' location and allows the counting to be estimated by summing of all density values.

This task is performed by a CNN-based model [5], whose goal is to automatically determine the vehicle density map associated with a given input image. Formally, the density map estimator, $\Psi : \mathcal{R}^{\mathcal{C}\times\mathcal{H}\times\mathcal{W}} \mapsto \mathcal{R}^{\mathcal{H}\times\mathcal{W}}$, transforms a $\mathcal{W}\times\mathcal{H}$ input image $\mathcal{I}$ with $\mathcal{C}$ channels, into a density map, $D = \Psi(\mathcal{I}) \in \mathcal{R}^{\mathcal{H}\times\mathcal{W}}$.

3.2. Discriminator Network

The discriminator network, denoted by Θ, also consists of a CNN model. It takes as input the density map, D, estimated by the network Ψ. Its output is a lower resolution probability

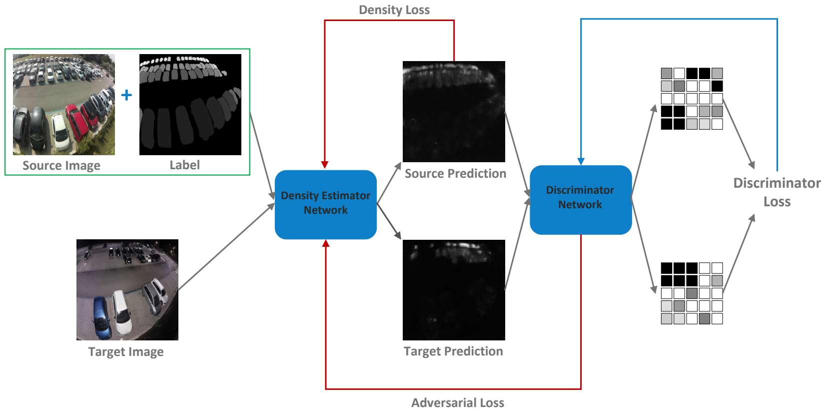

Figure 3: Algorithm overview. Given $C \times H \times W$ images from source and target domains, we pass them through the density map estimation network to obtain output predictions. A density loss is computed for source predictions based on the ground truth. In order to improve target predictions, a discriminator is used to locally classify whether a density map belongs to the source or target domain. Then, an adversarial loss is computed on the target prediction and is back-propagated to the density map estimation and counting network.

map where each pixel represents the probability that the corresponding region (from the input density map) comes either from the source or the target domain. The goal of the discriminator is to learn to distinguish between density maps belonging to source or target domains. Through an adversarial loss, this discriminator will, in turn, force the density estimator to provide density maps with similar distributions in both domains. In other words, the target domain density maps have to look realistic, even though the network Ψ was not trained with an annotated training set from that domain.

3.3. Domain Adaptation Learning

The proposed framework is trained based on an alternate optimization of the density estimation network, Ψ, and the discriminator network, Θ. Regarding the former, the training process relies on two components: 1) density estimation using pairs of images and ground truth density maps, which we assume are only available in the source domain; and 2) adversarial training, which aims to make the discriminator fail to distinguish between the source and target domains. As for the latter, images from both domains are used to train the discriminator on correctly classifying each pixel of the probability map as either source or target.

To implement the above training procedure, we use two loss functions: one is employed in the first step of the algorithm to train network Ψ, and the other is used in the second step to train the discriminator Θ. These loss functions are detailed next.

Network Ψ Training. We formulate the loss function for Ψ as the sum of two main components:

$$\mathcal{L}(\mathcal{I}^{\mathcal{S}}, \mathcal{I}^{\mathcal{T}}) = \mathcal{L}_{density}(\mathcal{I}^{\mathcal{S}}) + \lambda_{adv}\mathcal{L}_{adv}(\mathcal{I}^{\mathcal{T}}), \quad (1)$$

where $\mathcal{L}_{density}$ is the loss computed using ground truth annotations available in the source domain, while $\mathcal{L}_{adv}$ is the adversarial loss that is responsible for making the distribution of the target and the source domain closer to each other. In particular, we define the density loss $\mathcal{L}_{density}$ as the mean square error between the predicted and ground truth density maps, i.e. $\mathcal{L}_{density} = MSE(D^{\mathcal{S}}, D^{\mathcal{S}\_\mathcal{GT}})$.

To compute the adversarial loss $\mathcal{L}_{adv}$, we first forward the images belonging to the target domain through network Ψ, to generate the predicted density maps $D^{\mathcal{T}}$. Then, we forward $D^{\mathcal{T}}$ through network Θ, to generate the probability map $P = \Theta(\Psi(\mathcal{I}^{\mathcal{T}})) \in [0,1]^{H' \times W'}$, where $H' < H$ and $W' < W$. The adversarial loss is given by

$$\mathcal{L}_{adv}(\mathcal{I}^{\mathcal{T}}) = -\sum_{h,w} \log(P_{h,w}), \quad (2)$$

where the subscript h, w denotes a pixel in P. This loss makes the distribution of $D^{\mathcal{T}}$ closer to $D^{\mathcal{S}}$ by forcing Ψ to fool the discriminator, through the maximization of the probability of $D^{\mathcal{T}}$ being locally classified as belonging to the source domain.

Network Θ Training. Given an image $\mathcal{I}$ and the corresponding predicted density map D, we feed D as input to the fully-convolutional discriminator Θ to obtain the probability map P. The discriminator is trained by comparing P with the ground truth label map $Y \in \{0,1\}^{H' \times W'}$ using a pixel-wise binary cross-entropy loss

$$\mathcal{L}_{disc}(\mathcal{I}) = -\sum_{h,w} (1 - Y_{h,w}) \log(1 - P_{h,w}) + Y_{h,w} log(P_{h,w}), \quad (3)$$

where $Y_{h,w} = 0 \ \ \forall \ h, w$ if $\mathcal{I}$ is taken from the target domain and $Y_{h,w} = 1$ otherwise.

4. Experimental Results

We validate the proposed UDA method for density estimation and counting of traffic scenes under different settings. First, we employ the *NDISPark* dataset, and we test the *Day2Night* domain shift considering pictures taken during the day as the source domain, while night images for the target domain. Then, we utilize the *WebCamT* dataset to take into account the *Camera2Camera* performance gap, tackling the domain shift that takes place when we consider a camera different from the ones used during the training phase. Finally, we use the *GTA* dataset to assess the *Synthetic2Real* domain difference, training the algorithm using the synthetic images, and then test it on real data considering the *WebCamT* dataset again.

For all the experiments, we base the evaluation of the models on three metrics widely used for the counting task: (i) Mean Absolute Error (MAE) that measures the absolute count error of each image; (ii) Mean Squared Error (MSE) that instead quantifies the squared count error for each image; (iii) Average Relative Error (ARE), which measures the absolute count error divided by the true count. Note that, as a result of the squaring of each error, the MSE effectively penalizes large errors more heavily than small ones. Instead, the ARE is the only metric that

Table 1
Experimental results obtained for the four considered domain shift in terms of MAE, MSE and ARE. We achieved performance improvements for all the scenarios, considering all the three metrics.

	MAE	MSE	ARE
Day2Night Domain Shift - NDISPark Dataset			
Baseline - CSRNet [5]	3.95	27.45	0.43
Our Approach	**3.49**	**20.90**	**0.39**
Camera2Camera Domain Shift - WebCamT Dataset [6]			
Baseline - CSRNet [5]	3.24	16.83	0.21
Our Approach	**2.86**	**13.03**	**0.19**
Synthetic2Real Domain Shift - GTA Dataset			
Baseline - CSRNet [5]	4.10	25.83	0.28
Our Approach	**3.88**	**23.80**	**0.27**

considers the relation of the error and the total number of vehicles present for each image. Results are summarized in Table 1. We achieved better results compared to the basic model in all the considered scenarios and considering all the three metrics.

5. Conclusions

In this article, we tackled the problem of determining the density and the number of objects present in large sets of images. Building on a CNN-based density estimator, the proposed methodology can generalize to new data sources for which there are no annotations available. We achieved this generalization by exploiting an Unsupervised Domain Adaptation strategy, whereby a discriminator attached to the output forces similar density distribution in the target and source domains. Experiments show a significant improvement relative to the performance of the model without domain adaptation. To the best of our knowledge, we are the first to introduce a UDA scheme for counting to reduce the gap between the source and the target domain without using additional labels. Given the conventional structure of the estimator, the improvement obtained by just monitoring the output entails a great capacity to generalize learned knowledge, thus suggesting the application of similar principles to the inner layers of the network.

Another contribution is represented by the creation of two new per-pixel annotated datasets made available to the scientific community. One of the two novel datasets is a synthetic dataset created from a photo-realistic video game. Here the labels are automatically assigned while interacting with the API of the graphical engine. Using this synthetic dataset, we demonstrated that it is possible to train a model with a precisely annotated and automatically generated synthetic dataset and perform UDA toward a real-world scenario, obtaining very good performance *without* using additional manual annotations.

In our view, this work's outcome opens new perspectives to deal with the scalability of learning methods for large physical systems with scarce supervisory resources.

Acknowledgments

This work was partially supported by H2020 project AI4EU under GA 825619.

References

[1] G. Amato, L. Ciampi, F. Falchi, C. Gennaro, Counting vehicles with deep learning in onboard UAV imagery, in: 2019 IEEE Symposium on Computers and Communications, ISCC 2019, Barcelona, Spain, June 29 - July 3, 2019, IEEE, 2019, pp. 1–6. URL: https://doi.org/10.1109/ISCC47284.2019.8969620. doi:10.1109/ISCC47284.2019.8969620.

[2] L. Ciampi, G. Amato, F. Falchi, C. Gennaro, F. Rabitti, Counting vehicles with cameras, in: S. Bergamaschi, T. D. Noia, A. Maurino (Eds.), Proceedings of the 26th Italian Symposium on Advanced Database Systems, Castellaneta Marina (Taranto), Italy, June 24-27, 2018, volume 2161 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018, pp. 1–8. URL: http://ceur-ws.org/Vol-2161/paper12.pdf.

[3] G. Amato, P. Bolettieri, D. Moroni, F. Carrara, L. Ciampi, G. Pieri, C. Gennaro, G. R. Leone, C. Vairo, A wireless smart camera network for parking monitoring, in: IEEE Globecom Workshops, GC Wkshps 2018, Abu Dhabi, United Arab Emirates, December 9-13, 2018, IEEE, 2018, pp. 1–6. URL: https://doi.org/10.1109/GLOCOMW.2018.8644226. doi:10.1109/GLOCOMW.2018.8644226.

[4] V. S. Lempitsky, A. Zisserman, Learning to count objects in images, in: J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada, Curran Associates, Inc., 2010, pp. 1324–1332. URL: https://proceedings.neurips.cc/paper/2010/hash/fe73f687e5bc5280214e0486b273a5f9-Abstract.html.

[5] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society, 2018, pp. 1091–1100. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Li_CSRNet_Dilated_Convolutional_CVPR_2018_paper.html. doi:10.1109/CVPR.2018.00120.

[6] S. Zhang, G. Wu, J. P. Costeira, J. M. F. Moura, Understanding traffic density from large-scale web camera data, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 4264–4273. URL: http://doi.ieeecomputersociety.org/10.1109/CVPR.2017.454. doi:10.1109/CVPR.2017.454.

[7] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011, IEEE Computer Society, 2011, pp. 1521–1528. URL: https://doi.org/10.1109/CVPR.2011.5995347. doi:10.1109/CVPR.2011.5995347.