Unsupervised Domain Adaptation for Traffic Density Estimation and Counting

Luca Ciampi*, Carlos Santiago†, Joao Paulo Costeira†, Claudio Gennaro* and Giuseppe Amato*
*Institute of Information Science and Technologies - National Research Council - Pisa, Italy
name.surname@isti.cnr.it
†Instituto Superior Técnico, (LARSyS/IST) - Lisbon, Portugal

***Abstract*—Convolutional Neural Networks have produced state-of-the-art results for a multitude of computer vision tasks under supervised learning. However, the crux of these methods is the need for a massive amount of labeled data to guarantee that they generalize well to diverse testing scenarios. In many real-world applications, there is indeed a large *domain shift* between the distributions of the train (*source*) and test (*target*) domains, leading to a significant drop in performance at inference time. *Unsupervised Domain Adaptation* (UDA) is a class of techniques that aims to mitigate this drawback without the need for labeled data in the target domain. This makes it particularly useful for the tasks in which acquiring new labeled data is very expensive, such as for semantic and instance segmentation. In this work, we propose an end-to-end CNN-based UDA algorithm for traffic density estimation and counting, based on adversarial learning in the *output space*. The density estimation is one of those tasks requiring per-pixel annotated labels and, therefore, needs a lot of human effort. We conduct experiments considering different types of domain shift, and we make publicly available two new datasets for the vehicle counting task, that were also used for our experiments. One of them, the *GTA - Grand Traffic Auto*, is a *synthetic* collection of images, obtained using the graphical engine of the Grand Traffic Auto video game, *automatically* annotated with precise per-pixel labels. Experiments show a significant improvement using our UDA algorithm compared to the model's performance without domain adaptation. The code, the models and the datasets are freely available at https://ciampluca.github.io/unsupervised_counting/**

I. Introduction

With the advent of Convolutional Neural Networks (CNNs) [1], supervised learning has reached excellent results across many Computer Vision application areas, such as object detection [2] and instance segmentation [3]. However, most CNN-based methods require a large amount of labeled data, and make a common assumption: the training and testing data are drawn from the same distribution. 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 one domain, named *source*, usually experiences a drastic drop in performance when applied on another domain, named *target*. This problem is commonly referred as *Domain Shift* [4].

Domain Adaptation is a common technique to address this problem. It adapts a trained neural network by fine-tuning it with a new set of labeled data, belonging to the new distribution. However, in many real cases, collecting a further collection of labeled data is expensive, especially for tasks that imply per-pixel annotations, like semantic or instance segmentation.

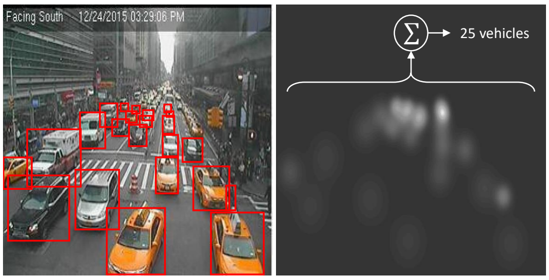
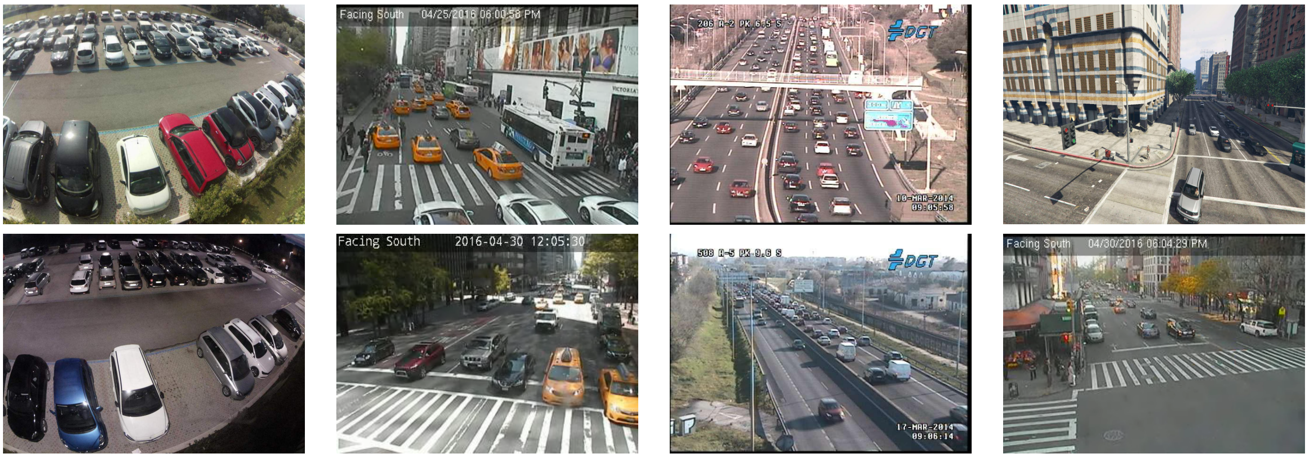


Fig. 1. Example of an image with the bounding box annotations (left) and the corresponding density map that sums up to the counting value (right).

Unsupervised Domain Adaptation (UDA) addresses the domain shift problem differently. It does not use labeled data from the target domain and relies only on supervision in the source domain. Specifically, UDA takes a source labeled dataset and a target *unlabeled* one. The challenge here is to infer some knowledge from the target data automatically to reduce the gap between the two domains.

In this work, we consider the counting task, defined as the estimation of the number of object instances in still images or video frames [5], which has recently attracted significant attention in the Computer Vision community. Specifically, we consider the vehicle counting scenario, where the task is to estimate the number of vehicles occurring in streets, roads, or parking lots. Most current systems address the counting task as a supervised learning process, relying on regression techniques to estimate a pixel-based density map from the image. The final count is obtained by summing all pixel values [5]. Figure 1 illustrates this approach.

We propose an end-to-end CNN-based UDA algorithm for traffic density estimation and counting, based on adversarial learning. Adversarial learning is 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. To the best of our knowledge, we are the first to introduce a UDA scheme for counting that can reduce the gap between the source and the target domain without using additional labels.



(a) (b) (c) (d)

Fig. 2. The Domain Shift scenarios that have been addressed in this work: (a) *Day2Night*; (b) and (c) *Camera2Camera*; (d) *Synthetic2Real*. The first row represents the labeled *source* domain, while the second represents the unlabeled *target* one used for our unsupervised domain adaptation.

We conducted experiments taking into account different types of domain shifts, and validating our approach on various vehicle counting datasets. First, we employed two existing datasets for traffic density estimation, *WebCamT* [6] and *TRANCOS* [7]. To emphasize the domain shift problem, we used as source domain images acquired by a specific subset of cameras. In contrast, the target domain is represented by images captured by a different subset of cameras, seeing different perspectives and visual contexts. We call this type of domain shift as the *Camera2Camera* domain shift. Comparisons with other techniques on these datasets show the superiority of our approach.

In order to test our technique with further types of domain shifts, we created and made publicly available the two additional datasets described in the following.

The *NDISPark - Night and Day Instance Segmented Park* dataset, consisting of images taken from surveillance cameras in a parking lot. Here, on the one hand, source data include annotated images collected by various cameras during the day. On the other hand, the unlabeled target domain contains images collected, in the same scenarios, during the night. We call this domain shift *Day2Night*.

The *GTA - Grand Traffic Auto* dataset, a vast collection of *synthetic* images generated with the highly photo-realistic graphical engine of the *Grand Theft Auto V* video game, developed by *Rockstar North*. This dataset consists of urban traffic scenes, *automatically* and precisely annotated with per-pixel annotations. To the best of our knowledge, it is the first *instance* segmentation synthetic dataset of traffic scenarios. We use this dataset to train the counting algorithm. Then, we performed domain adaptation to be able to count in real images. In this case, the domain shift is represented by the *Synthetic2Real* difference.

Figure 2 shows the described domain shifts that we have addressed.

In all the experiments, we show that our UDA technique always outperforms the non-domain adapted models.

Contributions of this work can be summarized as follows:

- We introduce a UDA algorithm for traffic density estimation and counting, which can reduce the domain gap between a labeled source dataset and an unlabeled target one. To the best of our knowledge, this is the first time that UDA is applied to counting.
- We create and make publicly available two new datasets, both having instance segmentation annotations. One is manually annotated and consists of images of parked cars collected during the day and by night. The second is a synthetic collection of images taken from a photo-realistic graphical engine, where the per-pixel annotations are automatically created.

II. Related Work

In this section, we review some previous work related to the Unsupervised Domain Adaptation and the counting task.

A. Unsupervised Domain Adaptation

Traditional UDA approaches have been developed to address the problem of image classification, and they try to align features across the two domains ([8], [9]). However, as pointed out in [10], they do not perform well in other tasks.

More recent advances also involve the semantic segmentation task. In this case, adversarial training for UDA is the most employed approach. It includes two networks. The first predicts the segmentation maps for the input source image while the second acts as a discriminator, taking the feature maps from the segmentation network and trying to predict the domain of the input. The adversarial loss, computed from the discriminator output, tries to make the distributions of the two domains more similar. The first to apply such a technique is [11]. More recently, the work proposed in [12] employs a residual network and adversarial training to make the source feature maps closer to the target ones. Another interesting work that inspired this paper is [13], where the authors applied adversarial training to the output space taking advantage of the structural consistency across domains.

A very appealing application of domain adaptation concerns synthetic data, encouraging the development of several synthetic datasets, such as ViPeD [14] for pedestrian detection and SYNTHIA [15] for semantic segmentation and autonomous driving applications. In this case, the algorithm is trained using these synthetic images and applied over real images. The domain adaptation algorithm is in charge of filling the gap between the two worlds.

B. The Counting Task

Following the taxonomy adopted in [16], we can broadly classify existing counting approaches into two categories: counting by regression and counting by detection. Counting by *regression* [5] is a supervised learning approach that tries to establish a direct mapping (linear or not) from the image features to the number of objects present in the scene or a corresponding density map (i.e., a continuous-valued function), skipping the challenging task of detecting instances of the objects. Counting by *detection* is, instead, a supervised technique where we localize instances of the objects, and then we count them [17], [18], [19], [20], [21].

Regression techniques have shown superior performance in crowded scenarios where the instances of the objects are sometimes not clearly visible due to occlusions, and they have been applied to a multitude of situations. The first work that employed a pure CNN to estimate the density and count people in crowded contexts is presented by [22]. A more efficient structure is proposed by [23] introducing a Multi-Column CNN-based architecture (MCNN) for crowd counting. A similar idea is developed by [24] with a scale-aware, multi-column counting model named Hydra-CNN able to estimate traffic densities in congested scenes. More recently, authors of [25] introduced CSRNet. This CNN-based algorithm uses dilated kernels to deliver larger reception fields and replace pooling operations. We employ this network as the baseline in our work, and we briefly review its architecture in the next sections.

The main limitations of these approaches are due to the scarcity of data. As a result, existing methods often suffer from overfitting, which leads to performance degradation while transferring them to other scenes. Besides, there is another inherent problem: the labels of these datasets are not very accurate. Most of the existing datasets are dot-annotated. Consequently, the ground truth density maps are just an approximation in which the sizes of the objects are estimated using some heuristics. In this work, we address both problems proposing an unsupervised domain adaptation technique that exploits unlabeled data and introducing two new datasets with per-pixel annotations that allow the creation of precise ground truth density maps.

III. Datasets

As mentioned before, to prove our approach's validity, we performed experiments on various vehicle counting datasets, offering different domain shift characteristics. Specifically, we used two existing datasets for traffic density estimation: *WebCamT* [6] and *TRANCOS* [7]. Then, we used two additional datasets that we created on purpose and made publicly available: the *NDISPark - Night and Day Instance Segmented Park* dataset and the *GTA - Grand Traffic Auto* dataset.

Figure 3 shows some images belonging to these datasets, together with the associate labels and the corresponding generated density maps used for the counting task.

A. 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 consider a total of about 40,000 images belonging to 10 different cameras and consequently having different views. We employ the existing bounding box annotations of the dataset to generate ground truth density maps. In particular, we consider one Gaussian Normal kernel for each vehicle present in the scene, having a value of μ and σ equal to the center and proportional to the size of the bounding box surrounding the vehicle, respectively. We used this dataset to test performance with the *Camera2Camera* domain shift, introduced in Section I.

B. TRANCOS Dataset

The *TRANCOS* dataset is a public traffic dataset containing 1244 dot-annotated images of different congested traffic scenes captured by surveillance cameras, introduced by [7]. The approximated ground truth density maps are generated by putting one Normal Gaussian kernel for each dot present in the scene, having a value of σ empirically decided by the authors. They also provided the regions of interest (ROIs). We used this dataset to test performance with the *Camera2Camera* domain shift, mentioned in Section I.

C. NDISPark - Night and Day Instance Segmented Park

The *NDISPark* dataset was created by us on purpose and made publicly available. It is a small, manually annotated dataset for counting cars in parking lots, consisting of about 250 images. This dataset is challenging and describes most of the problematic situations that we can find in a real scenario: seven different cameras capture the images under various weather conditions and viewing angles. Another challenging aspect is the presence of partial occlusion patterns in many scenes such as obstacles (trees, lampposts, other cars) and shadowed cars. Furthermore, it is worth noting that images are taken during the day and the night, showing utterly different lighting conditions.

Unlike most counting datasets, *NDISPark* is precisely annotated with *instance* segmentation labels, allowing us to generate accurate ground truth density maps for the counting task since the size of the vehicles is well-known.

We employed this dataset to test performance with the *Day2Night* domain shift, explained in Section I.

D. GTA - Grand Traffic Auto

The *GTA - Grand Traffic Auto* was also created by us on purpose and made publicly available. It is a vast collection of about 10,000 *synthetic* images of urban traffic scenes collected using the highly photo-realistic graphical engine of the *GTA V - Grand Theft Auto V* video game. About half of them concern urban city areas, while the remaining involve sub-urban areas and highways. To generate this dataset, we designed a framework that *automatically* and precisely annotates the vehicles present in the scene with per-pixel annotations. To the best of our knowledge, this is the first *instance* segmentation synthetic dataset of city traffic scenarios. As in the NDISPark dataset, the instance segmentation labels allow us to produce accurate ground truth density maps for the counting task since the size of the vehicles is well-known.

We exploited this dataset to test performance with the *Synthetic2Real* domain shift, introduced in Section I.

IV. Proposed Method

Our method relies on CNN models 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. 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.

Fig. 4 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 output of the discriminator is a pixel-wise classification of a low-resolution map, as illustrated in Fig. 4, 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 the train in the following subsections.

A. Density Estimation Network

We formulate the counting task as a density map estimation problem [5]. 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 location of the vehicles and allows the counting to be estimated by summing of all density values.

This task is performed by a CNN-based model, 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, $D = \Psi(\mathcal{I}) \in \mathcal{R}^{\mathcal{H}\times\mathcal{W}}$.

B. 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 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, *i.e.*. 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.

C. 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.

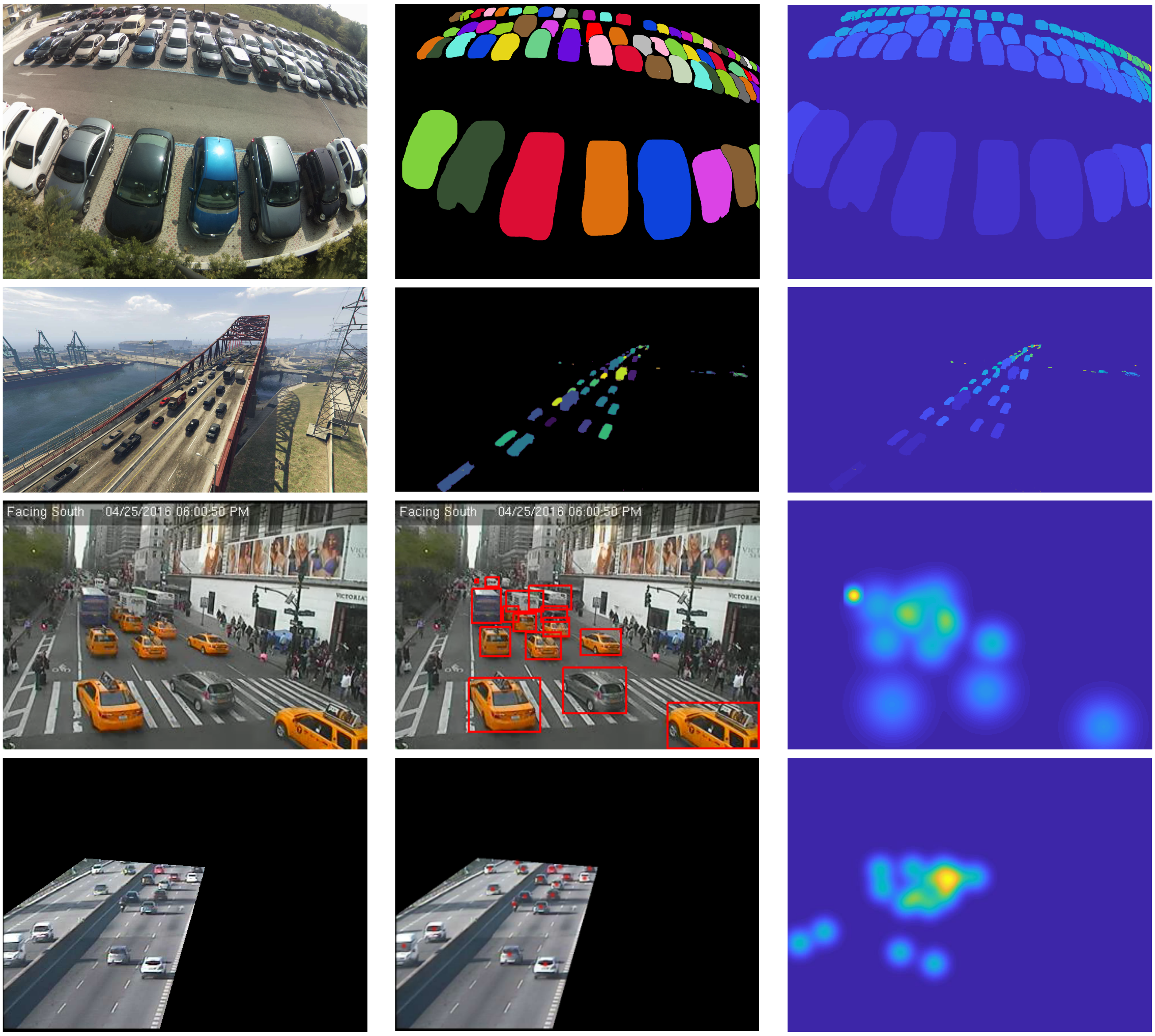
Network Ψ Training. We formulate the loss function for Ψ as the sum of two main components:

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

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

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



(a) (b) (c)

Fig. 3. Some examples taken by the four datasets used in this work: (a) Images; (b) Labels; (c) Density Maps generated from the labels. Each row correspond to a specific dataset: from top to bottom, the *NDISPark* and the *Grand Traffic Auto* datasets introduced in this work, the *WebCamT* dataset [6] and the *TRANCOS* dataset [7]. Note that the densities maps generated in our datasets are accurate since we start from an instance segmentation annotations. Also notice that, in the case of the Grand Traffic Auto dataset, annotations are *automatically* generated without human effort.

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

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

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

V. Experimental Results

A. Implementation Details

Density Map Estimation and Counting Network. We build our density map estimation network based on the Congested Scene Recognition Network (CSRNet) [25]. Here we briefly review some of the features characterizing this algorithm. CSRNet provides a CNN-based method that can understand highly congested scenes and perform accurate density estimation and counting. It is composed of two major components. Authors use the well-known VGG-16 network

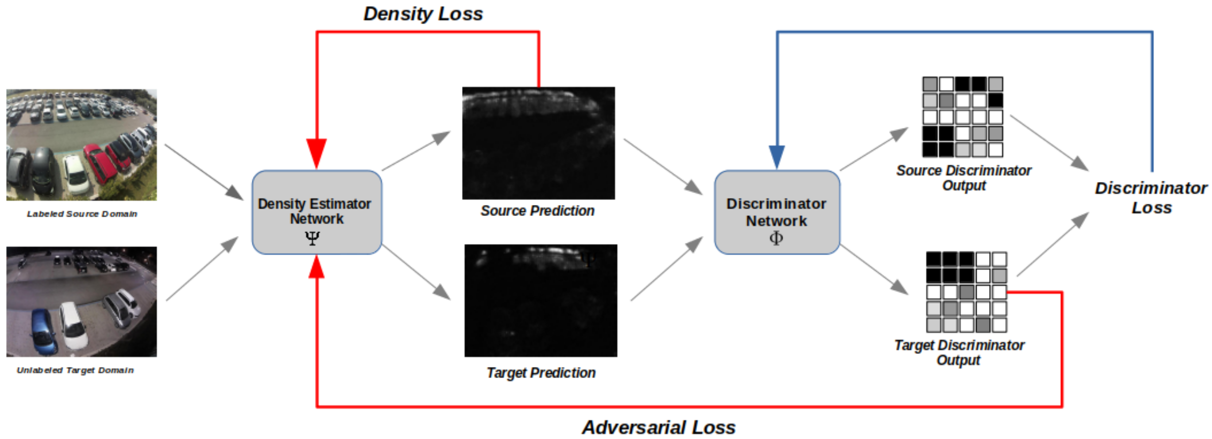


Fig. 4. Algorithm overview. Given $C \times H \times W$ images from source and target domains, we pass them through the density map estimation network to obtain output predictions. A density loss is computed for source predictions based on the ground truth. To improve target predictions, a discriminator 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.

[26] as the front-end for 2D feature extraction because of its strong transfer learning ability. On the other hand, the back-end consists of dilated kernels. The basic concept of using dilated convolutions is to deliver larger reception fields replacing the pooling operations. It is worth noting that the max pool operation is responsible for losing quality in the density generation procedure. Since the output size from VGG is reduced by a factor of 8 of the original input size, we up-sampled the final output to compare it with the ground truth density map.

Discriminator. We use a Fully Convolutional Network similar to [13], [27], composed of 5 convolution layers with kernel 4×4 and stride of 2. The number of channels are $\{64, 128, 256, 512, 1\}$, respectively. Each convolution layer is followed by a leaky ReLU having a parameter equals to 0.2.

We implement the whole system using the PyTorch framework on a single Nvidia RTX 2080 GPU with 12 GB memory. To train the density estimator network and the discriminator, we use Adam optimizer [28] with an initial learning rate set to 10^{-5}. During the training, it is crucial to balance the weight between density and adversarial losses. A small value of λ_{adv} may not help the training process significantly. In contrast, a larger value may propagate incorrect gradients to the density estimator. We empirically choose the value of λ_{adv} depending on the employed dataset.

B. Results and Discussion

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; then, we utilize the WebCamT and the TRANCOS datasets to take into account the *Camera2Camera* performance gap. Finally, we use the Grand Traffic Auto dataset to consider the *Synthetic2Real* domain difference. 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 penalizes large errors more heavily than small ones; (iii) Average Relative Error (ARE), which measures the absolute count error divided by the true count. Results are summarized in Table I.

1) Day2Night Domain Shift: In this scenario, we split the NDISPark dataset into train, validation, and test subsets containing about 100, 50, and 100 images. The former contains only pictures taken during the day (source domain), while the validation and the test subsets contain night images (target domain). To fairly evaluate our method, we first consider the baseline model without the domain adaptation module (i.e., putting the λ_{adv} value to zero). Then, we add the adversarial module comparing the results. In both cases, we train the network for 300 epochs, validating at each iteration. We choose the best validation model in terms of MAE, and we test it against the test set. As showed in Table I, using our solution, we obtained performance improvements considering all the three metrics.

2) Camera2Camera Domain Shift: In this case, we perform two sets of experiments to test the domain shift that takes place when we consider a camera different from the ones used in the training phase.

First, we consider the WebCamT dataset, and we split it into a train and a test set. In the former, we account for about 25,000 images belonging to 7 cameras (source domain). In the latter, we consider the remaining 15,000 pictures of 3 different cameras, having diverse contexts and slightly angle of views (target domain). We compare the baseline and our solution training for 20 epochs, validate it at each iteration and choose the best model in terms of MAE.

Second, we take into account the TRANCOS dataset. We split it into train, validation, and test sets, following [7]. The train set represents the source domain, while the other two belong to the target domain and are collected in different contexts. We train our domain adaptation for 200 epochs, picking the best validation model in terms of MAE, and we evaluate it against the test set. We compare the obtained results with the ones claimed by [25] using only the state-of-the-art CSRNet algorithm (i.e., our baseline) and with other state-of-the-art techniques present in the literature.

As showed in Table I, we obtained performance improvements in both cases, taking into account all the three metrics. Considering the publicly available TRANCOS dataset, we achieved superior results not only with respect to the baseline, but also compared to the other considered approaches.

3) Synthetic2Real Domain Shift: In this scenario, we train the algorithm using synthetic images. Then we test it on real data. In particular, we consider a subset of the Grand Traffic Auto dataset containing about 5,000 images of city traffic scenarios, and we use it as the training set (source domain). On the other hand, we account for the test subset of the WebCamT dataset as the target domain. We compare the results obtained using the baseline model and our solution with the domain adaptation module. In both cases, we train the algorithm for 20 epochs, validating at each iteration. We choose the best model in terms of MAE.

Again, as showed in Table I, we achieved better results compared to the basic model. We believe that this scenario is particularly interesting because we obtained comparable results with the previous one, but this time *without* using manual annotations neither in the source domain.

Finally, we also plot some examples of the outputs obtained using our models showing their visual quality. Figure 5 shows the ground truth and the predicted density maps for some random samples of the considered scenarios.

VI. Conclusions

In this article, we tackle 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 sources of data for which there are no annotations available. We achieve this generalization by adversarial learning, 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. Given the conventional structure of the estimator, the improvement obtained by just monitoring the output entails a great capacity to generalize training, thus suggesting applying 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 new 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, the outcome of this work opens new perspectives to deal with the scalability of learning methods for large physical systems with scarce supervisory resources.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[4] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*. IEEE, 2011, pp. 1521–1528.

[5] V. Lempitsky et al., "Learning to count objects in images," in *Advances in neural information processing systems*, 2010, pp. 1324–1332.

[6] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Understanding traffic density from large-scale web camera data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5898–5907.

[7] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Onoro-Rubio, "Extremely overlapping vehicle counting," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2015, pp. 423–431.

[8] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, 2015, pp. 1180–1189.

[9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[10] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2020–2030.

[11] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[12] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1335–1344.

[13] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7472–7481.

[14] G. Amato, L. Ciampi, F. Falchi, C. Gennaro, and N. Messina, "Learning pedestrian detection from virtual worlds," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 302–312.

[15] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[16] V. A. Sindagi and V. M. Patel, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

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

[18] 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.

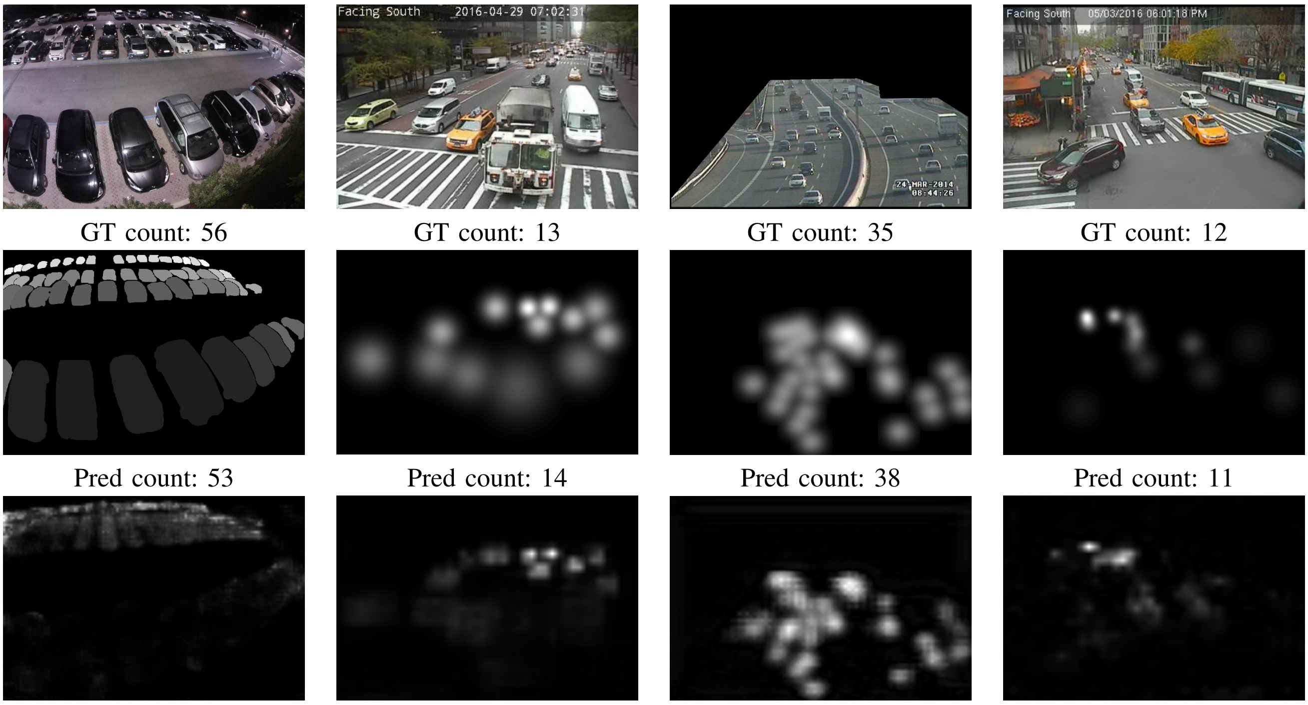
[19] G. Amato, P. Bolettieri, D. Moroni, F. Carrara, L. Ciampi, G. Pieri, C. Gennaro, G. R. Leone, and C. Vairo, "A wireless smart camera network for parking monitoring," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.

[20] S. Aich and I. Stavness, "Improving object counting with heatmap regulation," *arXiv preprint arXiv:1803.05494*, 2018.

	MAE - Mean Absolute Error	MSE - Mean Squared Error	ARE - Average Relative Error
Day2Night Domain Shift - NDISPark Dataset			
Baseline - CSRNet [25]	1.70	5.53	0.26
Our Approach	**1.45**	**4.40**	**0.25**
Camera2Camera Domain Shift - WebCamT Dataset [6]			
Baseline - CSRNet [25]	3.24	16.83	0.21
Our Approach	**2.86**	**13.03**	**0.19**
Camera2Camera Domain Shift - TRANCOS Dataset [7]			
Hydra-CNN [24]	10.99	68.70	0.71
FCN-MT [6]	5.31	-	0.85
LC-ResFCN [21]	3.32	-	-
Baseline - CSRNet [25]	3.56	30.64	0.10
Our Approach	**3.30**	**23.60**	**0.08**
Synthetic2Real Domain Shift - Grand Traffic Auto Dataset			
Baseline - CSRNet [25]	4.10	25.83	0.28
Our Approach	**3.88**	**23.80**	**0.27**

TABLE I

EXPERIMENTAL RESULTS OBTAINED FOR THE FOUR CONSIDERED DOMAIN SHIFT. WE ACHIEVED PERFORMANCE IMPROVEMENTS FOR ALL THE SCENARIOS, CONSIDERING ALL THE THREE EVALUATION METRICS.



(a) (b) (c) (d)

Fig. 5. Examples of the predicted density maps in the considered scenarios: (a) *Day2Nigh* Domain Shift using the NDISPark dataset; (b) and (c) *Camera2Camera* Domain Shift employing the WebCamT and TRANCOS dataset, respectively; (d) *Synthetic2Real* Domain Shift using the Grand Traffic Auto for the training phase and the WebCamT for testing on real images. In the first row we report the input images, in the second row the ground truth, while in the third the predicted density maps obtained with our models.

[21] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 547–562.

[22] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 640–644.

[23] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589–597.

[24] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 615–629.

[25] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] A. Radford et al., "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.