DOI 10.1016/j.patcog.2021.108484

Scene-specific Crowd Counting Using Synthetic Training Images

Rita Delussu, Lorenzo Putzu, Giorgio Fumera

Department of Electrical and Electronic Engineering, University of Cagliari Piazza d'Armi, 09123 Cagliari, Italy

Abstract

Crowd counting is a computer vision task on which considerable progress has recently been made thanks to convolutional neural networks. However, it remains a challenging task even in *scene-specific* settings, in real-world application scenarios where no representative images of the target scene are available, not even unlabelled, for training or fine-tuning a crowd counting model. Inspired by previous work in other computer vision tasks, we propose a simple but effective solution for the above application scenario, which consists of automatically building a scene-specific training set of *synthetic* images. Our solution does not require from end-users any manual annotation effort nor the collection of representative images of the target scene. Extensive experiments on several benchmark data sets show that the proposed solution can improve the effectiveness of existing crowd counting methods.

Keywords: Crowd counting, Scene-specific settings, Synthetic training images

1. Introduction

Crowd counting is a potentially very useful computer vision functionality in applications involving monitoring and analysis of crowds [1, 2], in particular, security-related applications based on video surveillance systems. Despite the considerable effort spent so far by the research community and the performance improvements achieved by recent methods based on Convolutional Neural Networks (CNNs) on benchmark data sets [3, 4, 5], it remains a challenging task in unconstrained settings characterised by illumination changes, perspective and scale variations or distortions due to camera views, static and dynamic occlusions, complex backgrounds, and dense crowds. Early methods followed two different approaches: pedestrian or body part detection, which were effective only on sparse crowds with very limited or no overlapping, and regression of the people count from local or global low-level image features [1]. State-of-the-art methods are based on CNNs [2]. Most of them are regression-based, but CNNs are enabling effective detection-based methods also for dense crowds [6]. All regression-based methods, as well as recent detection-based ones, require a training set of manually annotated crowd images, with annotations consisting either in the number of people, for early methods, or in the position of each pedestrian, for CNN-based ones.

Existing work aim at developing crowd counting models capable of generalising to unseen scenes, e.g., to different perspectives and background. This is a very challenging task since it requires training data representative of a large variety of possible crowd scenes. In this work we focus instead on a *scene-specific* setting where accurate estimation of crowd size on a *given* target scene is required, but collecting, and even more manually annotating a suitable amount of representative crowd images for training or fine-tuning a regression model, is too demanding, or even infeasible, for end-users. This is a real-world, challenging application scenario which was inspired by our work in a recent project,[1] involving the development of real-time video analytics tools to support Law Enforcement Agencies (LEAs) in guaranteeing the security of mass gatherings. For instance, the above scenario can occur when a new, temporary installation of surveillance cameras is required in a public area, and should be operational in a short time.

In the above scenario, a regression model can only be trained on already available annotated images from other scenes, e.g., using benchmark data sets,

[1]LETSCROWD, Law Enforcement agencies human factor methods and Toolkit for the Security and protection of CROWDs in mass gatherings, EU H2020, `https://letscrowd.eu/`

which can differ from the target scene in one or more of the above-mentioned factors, e.g., perspective, scale and background. However, in such a *cross-scene* setting, the performance of data-driven regression-based methods can be severely affected [7, 8]. A fine-tuning to the target scene is, therefore, required [7]. However, existing solutions to address cross-scene issues require a collection of representative images of the target scene, which in some cases should also be manually annotated [9, 10, 11, 7]: this does not fit the considered application scenario.

To address the above issue, we propose an approach based on the use of *synthetic* training images. Our approach is inspired by the use of synthetic images to overcome the scarcity of manually annotated training data in other computer vision tasks related to crowd analysis and pedestrian detection [12]. Our approach aims to build a scene-specific training set for a given target camera view, made up *only* of synthetic images, which can be *automatically* annotated. It only requires the user (e.g., a LEA operator) a background image of the target scene, the binary map (BMAP) of the corresponding region of interest (ROI) and its perspective map (PMAP). Synthetic training images are then automatically generated by superimposing images of pedestrians to the background image of the target scene, on locations allowed by the ROI, re-scaled according to the PMAP. Such images are then automatically annotated, and finally, they are used to train or fine-tune a given regression-based crowd counting model.

In this paper, which extends our preliminary work [13], we evaluate the effectiveness of a simple implementation of the above solution through extensive experiments on several benchmark data sets and state-of-the-art regression-based methods, as well as early ones not based on CNNs. We compare our solution against the usual cross-scene one, i.e., using *real* training data from *other* scenes. Our results show that even in the simple implementation considered in this paper, using synthetic images of the target scene can improve the performance of existing crowd counting methods and is therefore useful toward satisfying challenging real-world application requirements.

2. Related work

Crowd counting approaches can be categorised into counting by detection, by clustering, and by regression [1, 2]. The first two approaches rely on detecting pedestrians or body parts (e.g., head and shoulders) [1] from still images, or on clustering pedestrian trajectories from videos [1]. Although these approaches can provide the exact number of people in a scene, they are severely affected by the presence of occlusions and are therefore effective only for sparse crowds with little or no overlapping among people [1].

Regression-based methods *estimate* people count from low-level image features, instead, and can be more effective for dense crowd scenes. Early approaches used classical regression models [1] to map from holistic scene descriptors (e.g., segment, edge and texture descriptors) to crowd size. This requires a training set of crowd images manually annotated with the number of people. More recent CNN-based methods estimate the density map of the input image, instead, from which the number of people can be easily derived [2]. In this case, the training set is made up of the ground truth crowd density map, which is obtained from the manually annotated head positions of all pedestrians: this requires a higher effort than just counting them. The density map is then computed by superimposing 2D Gaussian kernels centred on pedestrians head positions, each one normalised to sum to one. Therefore, the pixel-wise sum of the density map equals the number of people in the corresponding image [14]; this simple computation is also carried out during inference to obtain the crowd size from the estimated density map. More refined definitions of the density map based on the use of adaptive kernels have also been proposed to improve robustness to scale and perspective variations [15, 5].

Existing CNN architectures are either modifications of "generic" ones, such as VGG [16, 11, 17, 5, 18, 15, 19, 20], or are specifically devised for crowd density estimation [21, 4, 3, 14]. Many architectures share the same backbone and differ in details, such as the number of branches or columns. The simplest ones use a single-column architecture [5], whereas others use multiple columns to address specific issues such as scale variations [17, 5, 15, 14, 19]. Some approaches fuse low- and high-level features [21], local and global information [4], and information from the ROI [16].

Some solutions have been proposed so far to address cross-scene issues specifically. A simple one is to use multi-scene training sets [17, 19, 15, 5]. Transfer learning and domain adaptation approaches have been proposed both for early regression-based [9] and for CNN-based methods [11]; however, they require manually annotated images of the target scene. A weakly supervised learning method has been proposed in [7], which also requires manually annotated images of the target scene, although only in terms of a categorical annotation into six classes (from "zero" to "very high" density) to reduce user's effort. An unsupervised solution has been proposed in [10], which, however, requires representative, although unlabelled, images of the target scene; furthermore, it carries out fine-tuning by retrieving *similar* images from the available training set; therefore, its effectiveness relies on the availability of training images representative of the target scene.

Our solution is inspired by the use of synthetic images in several computer vision tasks related to crowd analysis, such as anomalous crowd behaviour detection, pedestrian detection or tracking and crowd analysis based on optical flow [12], as well as in person re-identification [22], to mitigate the lack of representative, manually annotated training data. To our knowledge, using synthetic images has already been proposed for regression-based crowd counting by only one work [11], where a large data set of synthetic images was built using the Grand Theft Auto V (GTA5) video game to pre-train a CNN model. However, to create more realistic synthetic images this method also trains or fine-tunes a generative adversarial network (GAN) using real images of the target scene, which is not feasible in the application scenario considered in this work.

3. A method for constructing scene-specific synthetic training data for crowd counting

In this section we describe the proposed method for building scene-specific regression-based crowd counting models. Its goal is to reduce the gap between the cross-scene performance of existing methods and challenging requirements of real-world applications, such as real-time crowd monitoring tasks carried out by LEAs during mass gatherings. For instance, this is the case of ad hoc installations of video surveillance systems for short-lived mass gathering events. In such a scenario, a crowd counting model previously trained on annotated images from *different* scenes, e.g., benchmark data sets, has to be provided to end-users.

To mitigate the resulting cross-scene issues, we propose to train or fine-tune a crowd counting model using *only* synthetic images of the target scene. This can be made during system operation with minimal support from LEA operators, particularly without requiring them to collect, and even more to manually annotate, a suitable amount of representative crowd images of the target scene. One of the advantages of synthetic images is indeed the automatic definition of the ground truth [12], which in crowd counting tasks amounts to automatically annotate the position of each pedestrian and their exact number. Moreover, in such tasks, synthetic images allow to reproduce the same perspective, background and lighting conditions of the target scene, and to choose the spatial configuration of people.

This work extends two previous conference papers where we evaluated the cross-scene performance of several regression-based methods [8], and preliminarily investigated the effectiveness of synthetic images for early regression-based methods [13]. In this work, we better formalise the generation procedure of synthetic images and evaluate them also for CNN-based methods, including three additional ones with respect to [8], using two additional data sets. Finally, we evaluate how several factors (including the synthetic training set size, the number of pedestrians in synthetic images and their scale) affect crowd counting accuracy. In the following, we describe the requirements of the proposed method and its steps.

3.1. Requirements

To create accurate, scene-specific crowd counting models, it is crucial to reproduce the perspective and the background of the target scene, and to define the ROI, i.e., the region of the image where people can appear [1, 11, 13], as a binary map. Accordingly, our method requires a background image of the target scene and the corresponding BMAP and PMAP. Since we focus on real application scenarios where a crowd counting functionality can be deployed as a component of dedicated software suites for video surveillance system management, the above data can be easily provided by end-users during camera set-up through a suitable graphical user interface (GUI). Another useful information that end users can easily provide is the expected value of the largest crowd size: this allows to generate synthetic images with a different number of people in the corresponding range, which may help to better fit the underlying crowd counting model to the target scene. In case of uncertainty, an overestimate of the largest crowd size should be provided to guarantee examples of the actual largest crowd size in the training data. The above elements are described in the following and are exemplified in Fig. 1.

Among the existing techniques for background extraction and perspective map definition, in this work we consider two techniques that require very limited operator supervision. First, the **background** (BG) image can be automatically extracted during camera set-up. A still image is sufficient if no pedestrians or other non-static objects (e.g., cars) are present. Otherwise, a background extraction algorithm (e.g., by image subtraction) can be applied to a short video that can be easily acquired.

The **binary map** of the **ROI** is then necessary to define the region of the target scene where synthetic pedestrian images can be placed. It can be easily defined (e.g., as a polygon) on the background image acquired in the previous step through a suitable GUI. If possible, static objects (if any) should be excluded from the ROI to avoid inconsistencies with synthetic pedestrians.

Finally, the **perspective map** should be computed to re-scale synthetic pedestrians at each location of the BMAP. It consists of an image of the same size as the target ones, where the value of each pixel is the height, in pixels, of a standard adult individual at the corresponding location [10]. The PMAP can be obtained during camera set-up as well, for instance, by manually computing it on-site or by approximating it through linear interpolation of the height of a few pedestrians in one or more images of the target scene, assuming they have a standard height [10]. In practice, this requires end-users only to manually select the corresponding bounding boxes (BB).

3.2. Synthetic image generation

Complex approaches have been proposed so far to create data sets of synthetic images for various computer vision tasks, based on graphics engines [11] or GANs [22]. We propose a more straightforward method that can be easily implemented in video surveillance software suites. Based on the above requirements, our method consists of superimposing pedestrians' images to the BG image, randomly positioned on the ROI and re-scaled using the PMAP. To this aim, a set of suitable pedestrian images, that we call *gallery*, should previously be collected by the system *designer*, e.g., real images from the Web or synthetic ones generated by computer graphics tools. To guarantee a sufficient appearance variability, the gallery should include a sufficiently large number of pedestrians in different poses. Furthermore, gallery images should contain no background (e.g., they should contain a transparency layer or a foreground binary mask) and should be tightly cropped to the height of pedestrians to allow exact re-scaling through the PMAP. The above requirements are easy to satisfy during design, especially if computer graphics tools are used to generate pedestrian images.

Synthetic crowd images of the target scene can then be generated by superimposing to the BG image the desired number of pedestrians randomly selected from the gallery, located in randomly chosen and mutually exclusive positions inside the ROI, and re-scaled according to the PMAP. It is also easy to reproduce realistic overlapping between people by adding pedestrians one at a time from the farthest to the closest location to the camera. A smoothing operation can also be performed to blend pedestrian outlines with the BG image (different techniques can be used to this aim). The number N of synthetic images to be generated depends on the underlying crowd counting model. The number n of pedestrians in such images can be determined based on the maximum number of pedestrians $n_{\max}$ specified by the user. This allows to select a set of (approximately) evenly spaced values of n in the range $[1, n_{\max}]$, and to generate a fixed number of synthetic images for each value in this set. More precisely, if $n_{\max} = qN$, for some $q \in \mathbb{R}^+$, then one image containing n pedestrians can be generated, for each $n = 1, \lceil 1 + q, \lceil 1 + 2q, \ldots, n_{\max}$.

Finally, each synthetic image can be automatically annotated with the ground truth, i.e., the number of pedestrians and (if required by a CNN-based model) their location. Basic notions of human anatomy allow this task to be automated as well: assuming that gallery images are tightly cropped and contain adult individuals with standard height and body part proportions, the head height is 1/8 of the total body height [23], and the head points are directly located at 1/16 height and 1/2 width of the image.

In Fig. 1 we show an example of the above procedure for generating a synthetic image. Although such images may look unrealistic, e.g., due to unnatural pedestrians' pose and to the absence of perspective distortions typical of surveillance cameras, they reproduce the perspective and the background of the target view, which are the most relevant features to obtain accurate crowd counting models. Moreover, the proposed image generation procedure is very simple to implement and has a low processing cost.

4. Experimental setting

The goal of our experiments is to evaluate the effectiveness of the proposed method for training or fine-tuning existing crowd counting models using only scene-specific synthetic images of the target camera view, and to compare it

Figure 1: Example of the proposed procedure for generating synthetic images of a target scene (best viewed in colour). Top row, left to right: BG image (taken from the UCSD data set, see Sect. 4.3), pedestrian BBs selected by the user on a real image to compute the PMAP, and the resulting PMAP. Bottom row, left to right: ROI provided by the user, some pedestrian images from the gallery used in our experiments (see Sect. 4.4), and a synthetic image with 80 pedestrians and their annotated head positions shown as white dots.

with the alternative cross-scene solution based on using real images from different scenes. To this aim, we carried out extensive experiments on a representative selection of four early regression-based crowd counting methods (Sect. 4.1) and nine state-of-the-art CNN-based ones (Sect. 4.2), using five single-scene and one multi-scene benchmark data sets of real crowd images (Sect. 4.3). Each single-scene data set is used in turn as the target scene (testing set), and a synthetic, scene-specific training set is built using the proposed method (Sect. 4.4). Its performance is then compared with the one achieved by using each one of the other single-scene data sets for training, to simulate a *cross-scene* setting through cross-data set experiments. A comparison is also made with the performance attained using the multi-scene data set for training, since this is one of the existing solutions for improving cross-scene accuracy. For completeness, a comparison is also made against the *same-scene* performance of each target data set, which is evaluated using training images of the same data set, to assess cross-scene performance degradation.

4.1. Early regression-based methods

Despite the substantial progress achieved through CNN-based methods, early regression-based ones are still used [2, 24], since they exhibit a lower complexity, require a lower manual annotation effort, and can nevertheless provide accurate and fast results, especially in the presence of severe occlusions. Various approaches have been proposed to extend these methods through new feature representations or more sophisticated regression models [1, 24], but they still share a similar processing pipeline. In the following, we describe their main components, namely feature representations and regression models, focusing on the ones chosen for our experiments.

4.1.1. Feature extraction

Several kinds of features have been proposed so far, and often different complementary features are combined. For our experiments, we considered segment and edge features, which are among the most common foreground ones, as well as the Grey-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP) textural features. Foreground features can be obtained through background subtraction: segment features aims at capturing *global* properties of image regions, such as area and perimeter, whereas edge features focus on complementary information about *local* image characteristics, such as the number of edge pixels and edge orientation. Textural features encode spatial relationships among image pixels [1], instead. GLCM is defined as the number of occurrences of pairs of pixels with certain values in a given spatial relationship; several global statistical features can then be extracted from it [1]. The well-known LBP descriptor characterises local image textures [1]; it is rotation invariant and robust to grey-scale variation. A drawback of most of the above features is that they are strongly affected by image background [25]. In our experiments, we concatenated all the above features.

4.1.2. Regression models

Early regression-based methods can be subdivided into global and local [1]. They estimate the people count on the whole image, or as the sum of estimates on different image patches, respectively. Although local methods can handle scenes characterised by non-uniform crowd density more effectively, their processing cost is too high for real-time applications. We focused therefore on global methods and selected four representative regression models [1]: two linear models, namely simple Linear Regression (LR) and Partial Least Squares (PLS) regression; and two non-linear models, Random Forests (RF) and Support Vector Regression (SVR) with a radial basis function (RBF) kernel. Gaussian Process Regression has also been proposed as a global crowd counting method [25]; however it exhibits several drawbacks in crowd counting tasks with respect to other non-linear models such as RF: it is not scalable, its processing cost at the prediction phase is too high for real-time applications, and it is more sensitive to parameter selection.

4.2. CNN-based methods

Among the large number of CNN-based crowd counting methods recently proposed, we selected nine representative methods whose source code was available. They are described below and summarised in Table 1, and can be categorised according to the following criteria: network architecture (backbone, number of parallel columns and loss function), type of input used for training, including the augmentation process ("images") and the type of kernel ("head points", either fixed or adaptive), and inference time ("speed") evaluated in ms on a reference input size of 640×480.

The Multi-Column CNN (MCNN) architecture [14] aims at achieving robustness to scale variations. It is made up of three parallel and identical columns (except for filter dimensions), whose feature maps are merged by a final block. The Cascaded Multi-task Learning (CMTL) architecture [3] uses two columns that share the first layers to address two related sub-tasks: crowd count categorisation into ten qualitative levels and density map estimation. The Deformation

Aggregation Network (DAN) [20] consists of two parts: a VGG backbone, made up of eight blocks, and a multi-layer aggregation that learns adjustable weights to estimate the density map by an adaptive fusion of feature maps of different layers. The Spatial Fully Connected Network (SFCN) [11] uses a ResNet-101 backbone to improve density map estimation on congested crowd scenes. The Congested Scene Recognition Network (CSRN) [17] consists of a dilation module on top of a VGG-16 backbone that aggregates multi-scale information without increasing the number of parameters to keep processing time low. The Context-Aware Network (CAN) [19] encodes multi-scale contextual information exploiting a VGG-16 backbone, concatenates the output with weighted feature maps and obtains the density map using dilated convolutions. The Spatial-/Channel-wise Attention Regression (SCAR) network [18] uses spatial-wise and channel-wise attention modules to encode large-range contextual information, to improve the accuracy of head location and alleviate estimation errors. The Deep Structure Scale Integration (DSSI) network [15] aims at handling large scale variations through three parallel sub-networks that process the same input image with different scales; their outputs are merged to increase the resolution of the density map. Finally, the Bayesian Loss for crowd counting estimation architecture (BL+) [5] exploits a loss function designed to directly use the head point supervision to handle large scale variations.

4.3. Real data sets

As explained in previous sections, we focus on crowd counting systems that have to be deployed on a *specific* target scene (camera view). To reproduce this setting in our experiments, data sets containing a sufficient number of manually annotated training and testing images from a *single* camera view should be used. Unfortunately, existing benchmark data sets do not fulfil all the above requirements together. To our knowledge, only three of them contain dense crowd scenes, namely ShanghaiTech, UCF-QNRF and World Expo Shanghai 2010 [14, 21, 2]. However, the first two are made up of single images taken from different scenes. The latter contains five one-hour test videos, each one from a

Table 1: Main features of the CNN-based methods used in our experiments. Network architecture: pre-trained backbone network (– denotes training from scratch), number of columns, loss function (MSE: Mean Squared Error; BCE: Binary Cross Entropy; Bayesian loss). Input: type of input images (whole or cropped image, and augmentation technique: flip, noisy, scale), and kernel used for computing the density map. Speed: inference time (in ms) on a reference input image of size 640×480.

Method	Network architecture			Input		Speed
	backbone	columns	loss	images	kernel	
MCNN [14]	–	3	MSE	Crop	Fixed	130
CMTL [3]	–	2	MSE&BCE	Crop&Flip&Noisy	Fixed	350
DAN [20]	VGG16	5	MSE	Crop	Fixed	210
SFCN [11]	ResNet	–	MSE	Whole	Fixed	900
CSRN [17]	VGG16	–	MSE	Crop&Flip	Fixed	480
CAN [19]	VGG16	4	MSE	Crop&Flip	Fixed	450
SCAR [18]	VGG16	2	MSE	Whole	Fixed	412
DSSI [15]	VGG16	3	MSE	3 scales	Adaptive	510
BL+ [5]	VGG19	–	Bayesian	Crop&Flip	Adaptive	260

single camera, but only one frame every 30 seconds is manually annotated, that is only 120 frames in total, which is not suitable to our experiments.

The only data sets containing a sufficient number of frames from a *single* camera view (from 1,299 to 2,000 frames, see below) manually annotated with the head position, are Mall [26], UCSD [27] and PETS [28]. Although they do not contain dense crowd scenes (at most 53 people per image are present), they are challenging data sets as they exhibit lighting variations, perspective distortions and severe occlusions. We therefore used them as target data sets, as well as training data sets for cross-data set experiments.

Mall is made up of 2,000 frames with a size of 640×480 pixels, collected from a single scene by a surveillance camera in a shopping mall. It contains a total of 62,325 pedestrians, with 13 to 53 people per frame (on average, 31). Mall is a challenging data set with severe perspective distortions and frequent occlusions caused by static objects or by other people. According to recent work [1, 2] we used the first 600 frames for training, the next 200 ones for validation, and the remaining 1,200 frames for testing. **UCSD** contains 70 videos acquired from a low-resolution camera (238×158 pixels) installed in a pedestrian walkway at a university campus. It contains a total of 49,885 pedestrians, with an average of

25 people per frame. We used a subset of 2,000 frames: frames from 600 to 1,399 for training (600 frames) and validation (200 frames), and the remaining 1,200 frames for testing [1, 2]. **PETS2009** was released at the 11th IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance [28], for different visual surveillance tasks. Part “S1” is devoted to crowd counting and is subdivided into three difficulty levels (different crowd density and people behaviour), and each level contains two sequences (frame size of 576×768) acquired with different cameras, at different times under different illumination and shading. We grouped the images from the first three cameras (for different difficulty levels and acquisition time) to create three single-scene data sets named PETSview1, PETSview2 and PETSview3. These new data sets contain in total 1,229 frames that we split into training, validation and testing sets of size 361, 128 and 740, respectively. Since the original PETS2009 does not include the head position for each frame, we used the ground truth provided in [29].

We also used the above mentioned **ShanghaiTech** data set to evaluate the cross-scene performance achieved using *multi-scene* training data. ShanghaiTech is widely used in the literature, especially for training CNN models, since it contains images acquired from different cameras, with different illumination, perspective and crowd density. It contains 1,198 images, for a total of 330,165 pedestrians, and is usually divided into parts two parts, Part_A and Part_B, containing 482 and 716 images, respectively. Each part is further subdivided into 300 images for training and the remaining ones for testing [14, 2]. Fig. 2 shows some examples of frames from each of the above data sets.

4.4. Synthetic data sets

We first collected a gallery of pedestrian images from the Web, according to the requirements described in Sect. 3.2. Taking into account the crowd size in the considered target data sets, for our experiments, we set the gallery size to 100 and chose images of pedestrians of standard height and in an upright pose; we also avoided to purposely select pedestrian images whose appearance was similar to the ones of target data sets. In principle, in applications where much

Figure 2: Example of images from the data sets used in our experiments. Top row, left to right: Mall, UCSD, PETSview1. Bottom row, left to right: PETSview2, PETSview3, ShanghaiTech.

larger crowd sizes can occur in (unknown) target scenes, a larger gallery may be necessary. In sect. 5.4 we shall evaluate the influence of the gallery size on crowd counting accuracy.

For each of the five target scenes (Mall, UCSD, PETSview1, PETSview2 and PETSview3) we extracted one BG image through a simple image subtraction algorithm applied to all training images. More effective techniques may be necessary for more complex scenes to avoid a noisy background image, which may affect the accuracy of crowd counting models.

We then manually defined the ROI as a polygon, without removing static objects (if any) inside it as mentioned in Sect. 3.1. Although this may result in inconsistencies between foreground and background objects when synthetic pedestrians are added to the background image, such inconsistencies are not likely to significantly affect the accuracy of crowd counting models, since early regression-based ones mainly focus on fine textures and foreground objects (pedestrians), and CNN-based ones mainly localise pedestrians heads.

We then computed the PMAP from a single training image by manually selecting the BBs of three pedestrians at different locations. This simple procedure was sufficient to provide an accurate PMAP for the considered target data sets. Other more accurate techniques can be used to take into account more

Figure 3: Examples of synthetic images from each of the considered target data sets. Top row, left to right: Mall, UCSD, PETSview1. Bottom row, left to right: PETSview2, PETSview3.

complex scenes (see Sect. 3.1).

We finally set the number of synthetic training images to $N = 1,000$, and the maximum number of pedestrians in each target scene to $n_{\max} = 100$, taking into account the characteristics of the target scenes and the size of the respective ROIs (see Fig. 2). Note that the chosen value of $n_{\max}$ overestimates the actual maximum crowd size of the real data sets by about twice. According to Sect. 3.2, for each target scene we generated $n_{\max}/N = 10$ synthetic images containing n pedestrians, for each $n = 1, 2, \ldots, n_{\max}$, for a total of 50,500 pedestrians. We finally subdivided this data set into a training and a validation set of 800 and 200 images, respectively. In Section 5 we shall evaluate how the values of N and $n_{\max}$ affect the performance of the considered crowd counting models.

Fig. 3 shows some examples of synthetic images for each target scene.[2] Table 2 reports the main characteristics of real and synthetic data sets.

4.5. Performance measures

We evaluated crowd counting accuracy using two common metrics that are defined over a single image: the absolute error (AE) and the root squared error (RSE). We report their average values across all testing images of a

[2]All our synthetic data sets are available at here.

Table 2: Statistics of real and synthetic data sets used in our experiments.

Type	Data set	Image size	Number of images				Pedestrian count			
			total	training	validation	test	total	min	avg	max
Real	Mall	480×640	2,000	600	200	1,200	62,235	13	31	53
	UCSD	158×238	2,000	600	200	1,200	49,885	11	25	46
	PETSview1	576×768	1,229	361	128	740	32,719	1	27	40
	PETSview2	576×768	1,229	361	128	740	36,458	2	30	40
	PETSview3	576×768	1,229	361	128	740	41,873	11	34	40
Synthetic	Mall	480×640	1,000	800	200	–	50,500	1	50	100
	UCSD	158×238	1,000	800	200	–	50,500	1	50	100
	PETSview1	576×768	1,000	800	200	–	50,500	1	50	100
	PETSview2	576×768	1,000	800	200	–	50,500	1	50	100
	PETSview3	576×768	1,000	800	200	–	50,500	1	50	100

given target scene, i.e., the mean absolute error (MAE) and the root mean squared error (RMSE), which are defined as $\text{MAE} = \frac{1}{N_\text{t}} \sum_{i=1}^{N_\text{t}} |\eta_i - \hat{\eta}_i|$ and $\text{RMSE} = \left(\frac{1}{N_\text{t}} \sum_{i=1}^{N_\text{t}} (\eta_i - \hat{\eta}_i)^2 \right)^{\frac{1}{2}}$, where N_t is the number of testing images, η_i is the ground truth (pedestrian count) and $\hat{\eta}_i$ is the estimated pedestrian count for the i-th image. As a result of the squaring operation, the RMSE penalises larger errors more heavily than MAE.

5. Experimental results

We first present the cross-scene results attained using single-scene (Sect. 5.1) and multi-scene (Sect. 5.2) real training images, then the ones attained using scene-specific, synthetic training data, and finally we compare them (Sect. 5.3).

5.1. Cross-scene results for real single-scene training data

Tables 3 and 4 report the results of cross- and same-data set (scene) experiments for early regression-based and CNN-based methods, respectively. For ease of comparison, same-scene results are highlighted in grey.

Early regression-based methods (Table 3) achieved a high same-scene performance, especially on Mall and UCSD. The best models turned out to be LR and PLS. However, the performance of LR and PLS considerably worsened in cross-scene settings, whereas the one of RF and SVR degraded only slightly; in particular, for training and target scenes characterised by similar perspective and scale, which is the case of Mall and the three views of PETS (see Fig. 2), in

Table 3: Cross-scene MAE and RMSE of early regression-based methods (LR, RF, SVR and PLS) using single-scene training sets. Same-scene results (training and testing on the same data set) are also reported for comparison, highlighted in grey. The best cross-scene result for each target data set is reported in bold.

	Training set	Testing set (target scene)									
		Mall		UCSD		PETSview1		PETSview2		PETSview3	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LR	Mall	2.74	3.49	9.59	11.63	289.2	294.4	348.7	349.0	268.1	270.9
	UCSD	67.3	78.75	2.9	3.54	334.6	347.9	369.2	374.0	128.2	146.6
	PETSview1	276.9	277.0	577.1	577.2	6.25	7.91	33.43	38.04	9.35	11.17
	PETSview2	210.2	210.3	308.4	308.4	97.86	127.0	4.85	5.98	159.4	160.2
	PETSview3	12.15	14.01	29.09	29.93	110.3	110.7	125.1	126.6	6.84	8.42
RF	Mall	3.82	4.85	5.12	7.42	9.27	12.43	12.15	13.96	**4.44**	**6.59**
	UCSD	5.83	6.98	3.82	4.66	9.12	11.45	**8.06**	**10.46**	5.22	5.94
	PETSview1	3.89	5.07	6.92	8.12	9.47	11.03	13.59	14.98	8.36	9.31
	PETSview2	6.88	8.57	5.38	7.31	**8.01**	**8.94**	9.56	11.05	6.27	8.14
	PETSview3	5.52	7.07	6.34	7.73	10.11	11.54	11.59	12.54	11.41	12.49
SVR	Mall	4.8	6.29	8.15	9.18	9.56	10.45	9.8	10.68	8.74	9.55
	UCSD	7.68	9.32	5.38	7.31	10.74	12.08	12.09	13.15	12.86	13.88
	PETSview1	12.26	13.57	6.21	8.52	12.82	15.25	14.85	16.79	17.67	18.56
	PETSview2	8.54	10.12	5.13	7.3	11.06	12.62	12.6	13.81	13.78	14.8
	PETSview3	**5.11**	**6.71**	7.52	8.61	9.76	10.61	10.2	11.04	9.5	10.37
PLS	Mall	3.16	4.1	110.7	110.9	51.97	65.77	16.97	20.94	53.4	61.05
	UCSD	266.3	268.0	2.6	3.23	99.38	109.1	428.7	429.9	460.9	467.7
	PETSview1	49.0	49.37	13.0	14.21	8.46	10.13	20.39	24.53	21.07	26.56
	PETSview2	23.01	23.42	103.9	104.1	57.72	68.15	7.65	9.06	103.1	103.8
	PETSview3	18.05	18.67	**5.1**	**7.27**	14.55	16.86	25.12	26.75	9.03	10.06

some cases the cross-scene performance by RF and SVR was even better than the corresponding same-scene one. **CNN-based methods** (Table 4) exhibited a similar behaviour: they achieved a high same-scene performance (with the exceptions of DAN on PETSview2 and of DSSI on UCSD and PETS) and a lower cross-scene performance, with some exceptions as well. Also, for CNN-based methods, the cross-scene performance was in some cases close or even better than the same-scene one on Mall and PETS, whose perspective and scale is similar. Instead, the most noticeable gap between same- and cross-scene performance can be observed when UCSD is used as either the training or the target scene since its scale and perspective are very different from those of the other data sets (see Fig. 2). A comparison between **early regression-based** and **CNN-based** methods shows that the latter generally achieved a better or slightly better same-scene performance, as one may expect, with the largest improvement occurring mainly on the three views of PETS. On the

Table 4: Cross-scene MAE and RMSE of CNN-based methods using single-scene training sets. Same-scene results are also reported for comparison, highlighted in grey. The best cross-scene result for each target data set is reported in bold.

		Testing set (target scene)									
	Training set	Mall		UCSD		PETSview1		PETSview2		PETSview3	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN	Mall	5.33	6.17	24.64	25.75	5.94	7.83	9.67	10.95	9.9	11.22
	UCSD	86.39	88.04	2.3	2.84	144.9	149.6	49.4	56.85	180.6	181.2
	PETSview1	19.54	20.16	24.18	25.28	6.2	7.86	22.05	23.59	9.77	11.75
	PETSview2	3.39	4.27	19.62	20.92	20.93	22.19	4.23	5.08	24.29	27.72
	PETSview3	4.31	5.35	21.28	22.47	19.54	21.63	10.37	11.66	4.18	5.13
CMTL	Mall	5.53	6.39	23.42	24.58	**5.77**	**7.42**	17.65	19.28	11.41	12.79
	UCSD	189.1	191.1	2.04	2.50	213.7	217.9	111.9	113.7	298.5	300.8
	PETSview1	9.93	10.73	24.18	25.13	5.11	6.29	15.56	17.20	4.46	5.95
	PETSview2	4.68	5.95	24.63	25.76	36.85	38.49	4.80	6.06	47.34	50.96
	PETSview3	4.61	5.79	21.94	23.12	21.90	24.54	11.50	13.97	4.23	5.06
DAN	Mall	5.43	6.42	25.42	26.54	7.51	9.43	11.7	13.14	8.84	10.27
	UCSD	164.1	166.1	5.18	6.39	185.9	192.1	61.76	66.53	227.3	228.5
	PETSview1	7.97	9.06	26.1	27.09	4.92	6.15	16.41	19.12	6.34	7.74
	PETSview2	28.95	29.54	27.86	29.0	26.43	28.38	28.68	30.37	32.89	33.38
	PETSview3	7.9	9.48	**18.8**	**20.12**	18.02	20.45	13.2	15.15	4.63	5.92
SFCN	Mall	4.05	5.02	28.15	29.27	19.37	20.85	27.66	28.72	71.38	71.87
	UCSD	880.2	882.1	2.91	3.64	853.5	859.6	634.3	635.5	988.4	990.6
	PETSview1	8.33	9.64	27.13	28.1	6.32	7.57	12.83	14.5	10.74	12.05
	PETSview2	36.55	38.35	25.93	26.85	85.29	87.81	8.1	9.81	106.9	108.6
	PETSview3	14.78	15.98	28.23	29.36	11.49	13.64	10.03	12.74	4.35	5.68
CSRN	Mall	6.57	7.73	24.51	25.8	21.55	23.89	19.08	21.61	15.37	16.38
	UCSD	70.78	71.46	6.2	7.01	57.52	61.86	28.29	31.21	69.06	69.36
	PETSview1	14.51	14.96	27.33	28.43	5.54	6.83	15.62	17.46	20.57	21.11
	PETSview2	12.15	12.66	27.06	28.16	10.14	11.82	7.09	7.9	8.42	9.53
	PETSview3	9.21	9.89	27.49	28.62	5.84	6.8	9.66	10.56	2.9	3.76
CAN	Mall	2.59	3.21	28.09	29.23	8.28	10.36	17.49	20.02	29.54	30.11
	UCSD	281.6	283.1	4.73	6.16	173.5	176.9	133.4	135.2	252.0	252.4
	PETSview1	10.5	11.17	27.5	28.56	6.33	7.5	8.43	9.25	**3.94**	**4.84**
	PETSview2	27.59	28.51	27.1	28.15	24.62	26.03	6.07	7.67	5.09	6.77
	PETSview3	6.73	7.7	27.55	28.7	7.5	9.07	11.54	12.78	6.82	7.84
SCAR	Mall	3.99	4.75	372.28	372.8	42.3	45.41	55.78	56.46	93.3	93.51
	UCSD	19.43	20.98	4.19	5.24	19.45	21.11	**6.67**	**8.19**	15.3	17.83
	PETSview1	265.37	265.53	503.0	504.1	3.38	4.07	122.04	128.9	134.72	135.23
	PETSview2	314.63	315.81	574.18	577.12	13.47	17.53	5.09	6.32	123.88	124.16
	PETSview3	36.1	37.14	575.91	578.83	11.88	13.53	38.03	44.03	8.39	10.32
DSSI	Mall	5.44	7.09	37.35	37.81	22.81	23.56	18.1	19.03	13.78	14.98
	UCSD	25.6	26.84	21.75	23.2	27.36	28.53	26.92	28.11	26.52	27.72
	PETSview1	9.87	14.1	69.02	69.8	18.0	20.44	12.63	15.0	10.31	11.36
	PETSview2	8.02	12.5	66.81	67.57	20.25	22.26	14.64	16.51	11.31	12.21
	PETSview3	**4.14**	**6.47**	62.54	62.8	24.09	24.75	17.32	18.22	11.46	12.46
BL+	Mall	2.18	2.74	152.76	153.63	6.9	7.86	15.12	16.08	8.22	9.98
	UCSD	23.96	25.05	2.5	3.57	22.65	23.8	21.17	22.0	23.66	24.77
	PETSview1	10.09	11.81	127.26	129.71	3.75	5.12	12.41	14.34	10.49	12.86
	PETSview2	15.73	17.91	77.63	80.9	15.35	17.78	5.8	6.57	10.22	11.68
	PETSview3	26.01	26.69	132.99	133.57	18.69	19.53	7.44	9.0	4.72	5.61

other hand, the best early regression-based methods (RF and SVR) turned out to be generally more robust than CNN-based ones in cross-scene settings. For instance, the cross-scene MAE and RMSE values of RF and SVR (Table 3) never exceed 20, whereas for *all* CNN-based methods *many* cross-scene MAE and RMSE values are above 20, and, except for DSSI and CSRN, several such values are even one order of magnitude higher.

5.2. Cross-scene results for real multi-scene training data

As mentioned in Sect. 2, multi-scene training sets are commonly used to improve the cross-scene performance of CNN-based models [17, 19, 15, 5]. Accordingly, for all the considered CNN-based models, we also carried out experiments using the multi-scene data set ShanghaiTech, either part_A or part_B, for training, with a similar setting as in Sect. 5.1. The results are reported in Table 5. To speed up these experiments, whenever possible, we used CNN models already trained on ShanghaiTech and made available by the respective authors. To ease the comparison with cross-scene results achieved using single-scene training data, we also report for each model the best and worst cross-scene results from Table 4. We did not carry out this experiment on early regression-based methods since holistic features require a BG image of each training image, which is not available for ShanghaiTech, and cannot be computed since each image of this data set is taken from a different scene.

As one may expect, the performance achieved using multi-scene training data is almost always better than the worst performance achieved over all the considered single-scene training sets. More significantly, in several cases (see the entries in boldface), it is even better than the *best* single-scene performance, up to be comparable to the “ideal” same-scene one (see Table 4). However, these latter results were achieved mainly by BL+, DSSI and CAN, and only in a minority of cases by other models; moreover, even for BL+, DSSI and CAN, there are several exceptions, especially on PETSview3.[3] Moreover, it turns out that the performance on a *given* target scene strongly depends on the multi-scene training set used. Indeed, some models achieved a higher performance using

[3]The behaviour of SCAR emerges as a clear outlier, as its performance with multi-scene training data was very poor for all target scenes. We could not find the cause of this behaviour.

Table 5: Cross-scene MAE and RMSE of CNN-based methods attained using for training either part_A (ShTechA) or part_B (ShTechB) of the multi-scene ShanghaiTech data set. For comparison, best and worst cross-scene results achieved on single-scene training data (S-best and S-worst) are reported from Table 4. For each method and target data set, multi-scene results that are better than the *best* single-scene ones are highlighted in boldface.

	Training set	Testing set (target scene)									
		Mall		UCSD		PETSview1		PETSview2		PETSview3	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN	ShTechA	16.16	16.77	**18.88**	**19.64**	9.3	10.04	10.26	11.98	33.9	38.67
	ShTechB	21.03	21.58	22.01	22.86	7.51	8.58	23.2	24.86	**6.55**	**8.12**
	S-best	3.39	4.27	19.62	20.92	5.94	7.83	9.67	10.95	9.77	11.75
	S-worst	86.39	88.04	24.64	25.75	144.9	149.6	49.4	56.85	180.6	181.2
CMTL	ShTechA	17.71	18.33	**21.0**	**21.84**	8.51	9.39	**10.36**	**11.92**	33.46	40.68
	ShTechB	13.92	14.6	22.26	23.02	10.32	11.38	17.95	19.89	9.61	12.39
	S-best	4.61	5.79	21.94	23.12	5.77	7.42	11.5	13.97	4.46	5.95
	S-worst	189.1	191.1	24.63	25.76	213.7	217.9	111.9	113.7	298.5	300.8
DAN	ShTechA	16.76	17.32	23.96	24.67	8.88	10.21	14.49	16.56	15.68	16.68
	ShTechB	18.02	18.64	22.82	24.01	8.93	10.71	19.19	22.03	20.13	21.11
	S-best	7.9	9.48	18.8	20.12	7.52	9.43	11.7	13.14	6.34	7.74
	S-worst	163.1	166.1	27.86	29.0	185.9	192.1	61.76	66.53	227.3	228.5
SFCN	ShTechA	773.2	777.4	**5.42**	**7.55**	30.59	31.5	802.1	802.3	683.6	687.4
	ShTechB	31.21	32.4	322.7	323.7	**10.88**	**12.46**	238.5	238.5	33.8	34.3
	S-best	8.33	9.64	25.93	26.85	11.49	13.64	10.03	12.74	10.74	12.05
	S-worst	880.2	882.1	28.23	29.36	853.5	859.6	634.3	635.5	988.4	990.6
CSRN	ShTechA	14.64	15.1	26.58	27.63	8.58	10.08	**8.92**	**10.17**	15.45	16.55
	ShTechB	10.61	11.1	28.06	29.2	10.97	12.11	12.28	13.83	15.44	16.62
	S-best	9.21	9.89	24.51	25.8	5.84	6.8	9.66	10.56	8.42	9.53
	S-worst	70.78	71.46	27.49	28.62	57.52	61.86	28.29	31.21	69.06	69.36
CAN	ShTechA	9.72	10.28	**27.04**	**28.16**	**5.04**	**5.87**	**6.2**	**7.46**	10.3	11.67
	ShTechB	**3.6**	**4.56**	28.05	29.18	**6.53**	**8.25**	10.31	11.49	15.57	16.55
	S-best	6.73	7.7	28.09	29.23	7.5	9.07	8.43	9.25	3.94	4.84
	S-worst	281.6	283.1	28.09	29.23	173.5	176.9	133.4	135.2	252.0	252.4
SCAR	ShTechA	738.4	739.2	520.4	521.1	997.9	999.5	918.9	919.9	911.7	913.5
	ShTechB	512.9	513.5	**326.2**	**327.2**	813.5	815.5	829.9	811.7	825.6	826.1
	S-best	19.43	20.98	372.3	372.8	11.88	13.53	6.67	8.19	15.3	17.83
	S-worst	314.6	315.8	575.9	578.8	42.3	45.4	122.5	128.9	134.7	135.2
DSSI	ShTechA	8.44	9.16	**20.41**	**21.06**	**7.91**	**9.46**	**8.91**	**9.9**	11.73	13.55
	ShTechB	12.93	13.47	**26.24**	**27.2**	**13.47**	**15.52**	**9.88**	**11.68**	25.65	26.1
	S-best	4.14	6.47	37.35	37.81	20.25	22.46	12.63	15.0	10.31	11.36
	S-worst	25.6	26.84	69.02	69.8	27.83	28.53	26.92	28.11	26.52	27.72
BL+	ShTechA	**6.07**	**7.05**	**16.63**	**17.08**	**5.28**	**6.28**	7.77	9.48	16.51	17.36
	ShTechB	**6.78**	**7.57**	**18.52**	**19.2**	**4.21**	**5.34**	**7.05**	**8.9**	10.07	11.85
	S-best	10.09	11.81	77.63	80.9	6.9	7.86	7.44	9.0	8.22	9.98
	S-worst	26.01	26.69	152	153.63	22.65	23.8	21.17	22.0	23.66	24.77

part_A of ShanghaiTech rather than part_B, whereas the opposite happened for other models; moreover, the performance gap between different multi-scene training sets can be large (see, e.g., MCNN and CMTL on PETSview2 and PETSview3). Similar behaviour can be observed for each model with respect to the different target scenes. To sum up, the results in Table 5 do not show a clear pattern of improvement due to the use of multi-scene over single-scene training data, but a mixed behaviour depending on the specific crowd counting method, target scene and training data set. This means that, in the considered application scenario where a crowd counting model has to be trained before deployment without any information on target scenes, using multi-scene training data is not guaranteed to be an effective solution.

5.3. Results for scene-specific synthetic data sets

In this section, we present the main results of this work. Table 6 shows the results attained on each target data set using scene-specific synthetic training images, together with a comparison with the *best* cross-scene results attained using real training data. In particular, the best cross-scene results over all single-scene training sets is reported for early regression-based methods, from Table 3, and over multi-scene training sets for CNN-based methods, from Table 5. The "ideal" same-scene results are also reported from Tables 3 and 4.

For early regression-based methods, in many cases, synthetic images provided a better (see the entries in boldface) or close performance to the *best* cross-scene one. In particular, the performance of RF and SVR was even better than the "ideal" same-scene one. Only in a few cases, mainly on PETS target scenes, synthetic images achieved a significantly lower performance than the corresponding *best* cross-scene one.

For CNN-based models, synthetic images attained a better or similar performance to the *best* cross-scene one on almost half of the cases. This is especially evident for SCAR, which performed poorly for multi-scene training data. On the other hand, the largest gap between the performance of synthetic data and the *best* cross-scene one (in favour of the latter) was observed for MCNN, CSRN, CAN, DSSI and BL+, although not for all target data sets; for CAN, DSSI and BL+ this result is coherent with the one of section 5.2, where these methods turned out to be the ones that most benefited from multi-scene training data. Nevertheless, a significant result that emerges from Table 6 is that using synthetic images allowed *all* the considered models (including early regression-based

Table 6: MAE and RMSE attained by all the considered crowd counting models, using as a training set: target scene-specific synthetic images ("Synthetic"), real images from the same scene ("Real-same"), and real images from different scenes ("Real-cross": best results over all single-scene training sets for early regression-based methods, and over the two ShanghaiTech training sets for CNN-based methods). For each data set and model the cases in which using synthetic training sets outperformed the best cross-data set results are highlighted in bold.

Method	Training set	Testing set (target scene)									
		Mall		UCSD		PETSview1		PETSview2		PETSview3	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LR	Real-same	2.74	3.49	2.9	3.54	6.25	7.91	4.85	5.98	6.84	8.42
	Real-cross	12.15	14.01	9.59	11.63	97.86	127.0	33.43	38.0	9.35	11.17
	Synthetic	14.94	16.34	**4.74**	**7.09**	**23.25**	**27.08**	**19.14**	**30.16**	26.6	33.19
RF	Real-same	3.82	4.85	3.82	4.66	9.47	11.03	9.56	11.05	11.41	12.49
	Real-cross	3.89	5.07	5.12	7.42	8.01	8.94	8.06	10.46	4.44	6.59
	Synthetic	6.76	8.1	**3.12**	**3.59**	**7.51**	9.13	18.35	23.41	7.82	9.61
SVR	Real-same	4.8	6.29	5.38	7.31	12.82	15.25	12.6	13.81	9.5	10.37
	Real-cross	5.11	6.71	5.13	7.3	9.56	10.45	9.8	10.68	8.74	9.55
	Synthetic	7.98	9.57	**2.85**	**4.13**	**6.96**	**8.66**	**8.83**	**10.63**	**4.6**	**6.54**
PLS	Real-same	3.16	4.1	2.6	3.23	8.46	10.13	7.65	9.06	9.03	10.06
	Real-cross	18.05	18.67	5.1	7.27	14.55	16.86	16.97	20.94	21.07	26.56
	Synthetic	**13.39**	**16.29**	5.16	**6.46**	17.06	21.32	29.1	30.59	**11.22**	**14.05**
MCNN	Real-same	5.33	6.17	2.3	2.84	6.2	7.86	4.23	5.08	4.18	5.13
	Real-cross	16.16	16.77	18.88	19.64	7.51	8.58	10.26	11.98	6.55	8.12
	Synthetic	20.73	21.68	**2.94**	**3.65**	12.22	13.43	17.86	18.67	11.39	13.69
CMTL	Real-same	5.53	6.39	2.04	2.50	5.11	6.29	4.80	6.06	4.23	5.06
	Real-cross	13.92	14.6	21.0	21.84	8.51	9.39	10.36	11.92	9.61	12.39
	Synthetic	22.96	23.47	**8.4**	**9.65**	9.43	11.09	**9.39**	**10.57**	**8.74**	**11.19**
DAN	Real-same	5.43	6.42	5.18	6.39	4.92	6.15	28.68	30.37	4.63	5.92
	Real-cross	16.76	17.32	22.82	24.01	8.88	10.21	14.49	16.56	15.68	16.68
	Synthetic	17.51	18.49	**10.31**	**12.21**	**4.05**	**5.37**	19.37	22.32	**10.55**	**12.56**
SFCN	Real-same	4.05	5.02	2.91	3.64	6.32	7.57	8.1	9.81	4.35	5.68
	Real-cross	31.21	32.4	5.42	7.55	10.88	12.4	238.5	238.5	33.8	34.3
	Synthetic	**17.76**	**18.57**	6.34	**7.34**	15.56	16.85	**23.22**	**24.82**	**10.19**	**12.46**
CSRN	Real-same	6.57	7.73	6.2	7.01	5.54	6.83	7.09	7.9	2.9	3.76
	Real-cross	10.61	11.1	26.58	27.63	8.58	10.08	8.92	10.17	15.45	16.55
	Synthetic	19.9	20.18	**3.45**	**4.8**	13.35	15.42	21.33	23.78	20.01	20.55
CAN	Real-same	2.59	3.21	4.73	6.16	6.33	7.5	6.07	7.67	6.82	7.84
	Real-cross	3.6	4.56	27.04	28.16	5.04	5.87	6.2	7.46	10.3	11.67
	Synthetic	16.77	17.26	**7.35**	**8.0**	12.78	14.4	16.99	19.19	30.95	31.36
SCAR	Real-same	3.99	4.75	4.19	5.24	3.38	4.07	5.09	6.32	8.39	10.32
	Real-cross	512.93	513.47	326.24	327.2	813.47	815.52	829.88	811.68	825.65	826.1
	Synthetic	**23.54**	**24.0**	**7.83**	**8.88**	**8.35**	**9.59**	**7.77**	**10.53**	**15.18**	**16.61**
DSSI	Real-same	5.44	7.09	21.75	23.2	18.0	20.44	14.64	16.51	11.46	12.46
	Real-cross	8.44	9.16	20.41	21.06	7.91	9.46	8.91	9.9	11.73	13.55
	Synthetic	28.91	29.5	**14.86**	**16.91**	19.18	21.81	21.29	23.58	29.48	30.02
BL+	Real-same	2.18	2.74	2.5	3.57	3.75	5.12	5.8	6.57	4.72	5.61
	Real-cross	6.07	7.05	16.63	17.08	4.21	5.34	7.05	8.9	10.07	11.85
	Synthetic	15.5	15.87	**7.85**	**8.59**	8.01	10.1	12.23	13.71	18.74	19.42

ones) to exceed the *best* cross-scene performance on the UCSD target scene, which differs in scale and perspective from the other single-scene data sets, as well as from many images of the multi-scene ShanghaiTech; the only exceptions are the cross-scene MAE values of PLS and SFCN, which are nevertheless very close to the corresponding values achieved using synthetic images. Therefore, despite some models may benefit from multi-scene training data, most of the considered ones exhibited a performance degradation if few or no training images exhibited a similar perspective to the one of the target scene. This result confirms the conclusion drawn at the end of Sect. 5.2 about the limited benefit of multi-scene training data in the considered application scenario.

Since the considered CNN-based models compute the crowd count from the estimated density map, we also examined and compared the quality of the density maps obtained using scene-specific synthetic training images with the ones attained using real training images from other scenes. We considered, in particular, the accuracy of the density map in locating the regions of the target (testing) images containing pedestrians: the rationale is that high accuracy in crowd count may be achieved even if localisation accuracy is low. To this aim, we focused on MCNN, which is one of the models that achieved the *lowest* benefit in crowd counting accuracy from synthetic training data (see Table 6). A first *qualitative* evaluation on some testing images, carried out through a visual comparison, showed an interesting result, i.e., density maps produced by synthetic training data turned out to locate pedestrian regions more accurately. Fig. 4 shows an example on two testing images from PETSview1 and PETSview2 data sets: despite using synthetic images provided (on average) *worse* crowd count results on these data sets (Table 6, row 'MCNN'), it can be seen that the corresponding density maps are *more* accurate with respect to the ones obtained using real training images from PETSview3 (the most similar scene to PETSview1 and PETSview2) and from the multi-scene ShanghaiTech partB.

To *quantitatively* analyse MCNN localisation accuracy on each target data set, we used the Grid Average Mean absolute Error (GAME) metric [6]. GAME subdivides the density map into a grid of 4^L cells, computes the MAE values within each cell and averages them over the whole grid. The higher the value of L, the more precise the corresponding evaluation of localisation accuracy (note

Figure 4: Density maps produced on two frames of PETSview1 (top) and PETSview2 (bottom) by MCNN trained on synthetic images (left), single-scene PETSview3 (middle), multi-scene ShanghaiTech PartB (right). Ground truth (red) and estimated (green) density maps are superimposed to the original frames. Yellow regions are the ones where the two maps coincide, corresponding to perfect localisation of pedestrians. The highest localisation accuracy is achieved when synthetic training images are used (left). Best viewed in colour.

that, for $L = 0$, GAME = MAE). Table 7 shows the GAME values for $L = 3, 5$ attained on each target data set, using as training data scene-specific synthetic images and real multi-scene images (from ShanghaiTech). It can be seen that using synthetic training images produced more accurate density maps for some target data sets, for $L = 3$, and for *all* of them for $L = 5$. Moreover, the increase in GAME from $L = 3$ to $L = 5$ is *lower* for synthetic images. To sum up, the above results provide evidence that scene-specific synthetic images can be an effective solution also for obtaining more accurate crowd density maps.

5.4. Ablation study

As explained in Sect. 4.4, synthetic data sets built for our experiments for each target scene were made up of $N = 1,000$ images (800 for training and 200 for validation) containing from 1 to $n_{\max} = 100$ pedestrians re-scaled according to the PMAP. In this section, we evaluate how the accuracy of the resulting models

Table 7: Cross-scene GAME values of MCNN for $L = 3, 5$, using as training data scene-specific synthetic images, and real multi-scene images from ShanghaiTech part_A (ShTechA) or part_B (ShTechB).

Test set	Syntetic		ShTechA		ShTechB	
	$L=3$	$L=5$	$L=3$	$L=5$	$L=3$	$L=5$
Mall	27.56	33.8	26.13	35.12	27.04	35.13
UCSD	17.41	24.04	23.59	26.62	24.65	26.97
PETSview1	16.03	23.0	15.99	26.54	14.57	26.81
PETSview2	21.87	25.25	22.18	31.13	25.92	29.79
PETSview3	24.35	32.94	58.55	74.12	22.13	37.32

is affected by the parameters N and $n_{\max}$, and by pedestrian scale variations in training images. To avoid re-training all the considered models, we selected a subset of models with the aim of including at least one early regression-based model, one CNN-based model trained from scratch, one trained using image patches, one trained using whole images, one using fixed kernels and one using an adaptive kernel. Accordingly, we selected four methods that fulfil all the above requirements: RF, MCNN, DAN and BL+. **Effect of training set size.** To analyse the effect of N we carried out experiments using randomly selected subsets of the original 800 synthetic training images for each target data set. Fig. 5 shows the MAE values of RF, MCNN, DAN and BL+ for N ranging from 200 to 800 with a step of 200. The behaviour of the RMSE metric was similar and is not reported due to lack of space. Apart from small fluctuations, which are likely caused by the randomness of image selection from the original training sets, the MAE values do not show a decreasing trend as N increases. We point out that the same behaviour was observed both for models obtained by transfer learning (DAN and BL+) and for MCNN, which is trained from scratch. This means that even a relatively small synthetic data set can be adequate to train a scene-specific regression model, which in turn can speed up the training procedure.

Effect of the maximum number of pedestrians. To analyse this aspect, we carried out experiments for $n_{\max}$ ranging from 20 to 100 with a step of 20, both in training and in validation images. Considering the size of the original

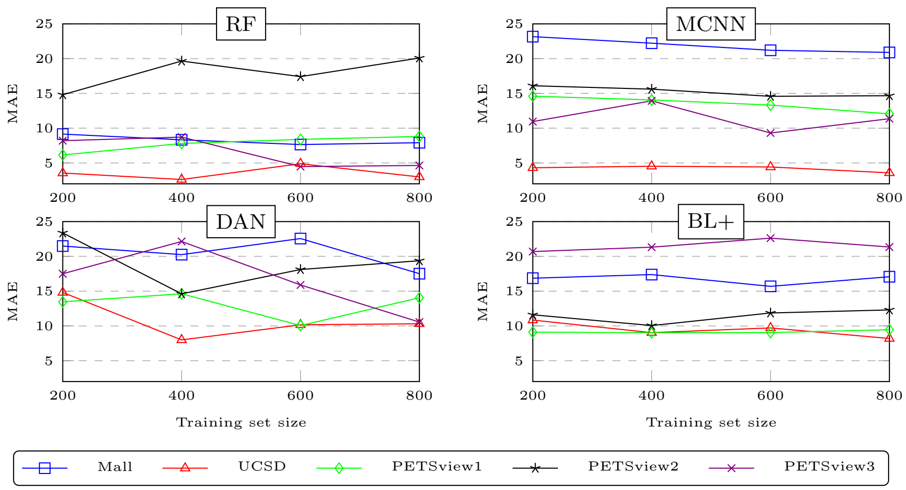

Figure 5: MAE values achieved by RF, MCNN, DAN and BL+ on the five target scenes using synthetic training data, as a function of training set size. Best viewed in colour.

data sets ($N = 1000$ images), to guarantee an equal number of images for each $n_{\max}$ value, these experiments were carried out using 200 training and 200 validation images. The results, reported in Fig. 6, show that in this case the behaviour of the early regression-based model RF turned out to be different from the one of CNN-based models. The MAE values of MCNN and BL+ showed a slightly decreasing trend as $n_{\max}$ increased, whereas no definite trend emerged for DAN. Instead, the MAE value of RF attained a minimum when $n_{\max}$ was closest to the maximum number of pedestrians actually present in the corresponding target scene. This suggests that early regression-based models are more sensitive than CNN-based ones to $n_{\max}$. Accordingly, the guideline we provided in Sect. 3.2 on how to set $n_{\max}$, i.e., overestimating it in case of uncertainty, seems more suited to CNN-based models.

Effect of pedestrian scale variations. If the PMAP is not accurate or the height of the pedestrians in the gallery is not precisely estimated, the scale of pedestrians in synthetic training images can be different than in real images. To analyse the effect of scale variations, we created four alternative synthetic data sets for each target scene, where pedestrian images are re-scaled

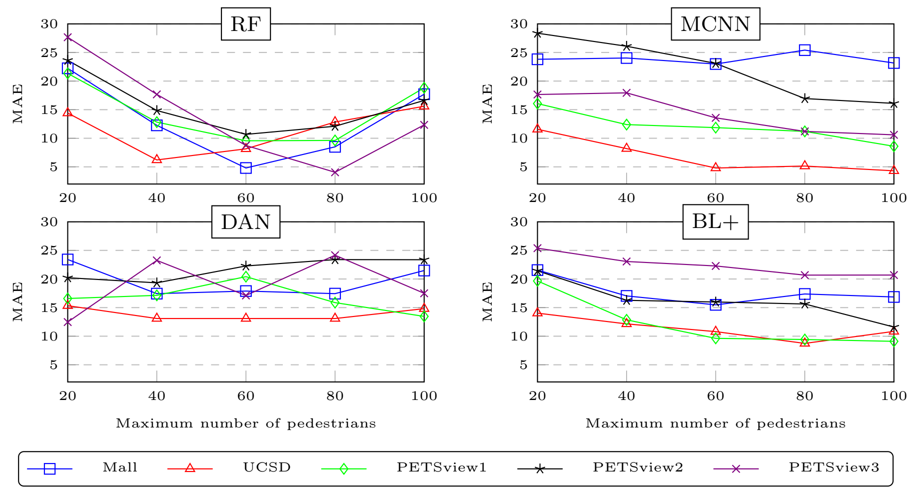

Figure 6: MAE values achieved by RF, MCNN, DAN and BL+ on the five target scenes using synthetic training data, as a function of the maximum number of pedestrians in training images. Best viewed in colour.

by a factor of 0.5 to 2 with respect to the corresponding original PMAP (note that a re-scaling factor of 1 corresponds to the original PMAP). The results are reported in Fig. 7. Generally, scale variations resulted in a sensible increase of MAE. Exceptions can be observed for RF, BL+ and DAN: RF attained a lower MAE on PETSview2 when pedestrians were undersized by a factor of 0.75; similarly, BL+ attained a lower MAE on Mall and PETSview3 for undersized pedestrian images; the performance of DAN on the PETSview1 target scene was only slightly affected even by large scale variations. The behaviour of BL+ may be due to the fact that the corresponding ground truth density map of training images is computed using adaptive kernels whose size is related to the distances between pedestrians.

Effect of gallery size. To analyse the effect of gallery size, we created four alternative synthetic data sets for each target scene, where the gallery size was set to 1, 5, 20 and 50 (note that the gallery size of 100 corresponds to the original synthetic data set). The results, reported in Fig. 8, show that apart from few exceptions, the MAE values show a decreasing trend as the gallery size

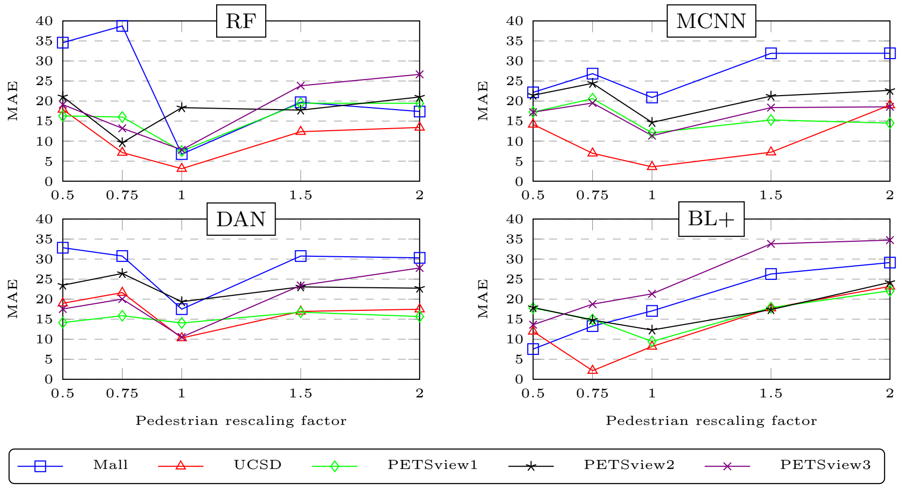

Figure 7: MAE values achieved by RF, MCNN, DAN and BL+ on the five target scenes using synthetic training data for different rescaling factors of pedestrians with respect to the original PMAP (from 0.5 to 2). Best viewed in colour.

increases. However, in most cases, in particular involving BL+ for all the target scenes, the MAE values decrease only slightly for gallery sizes larger than 20. This means that even a relatively small gallery can be adequate. This is likely to hold also for larger and dense crowds, characterised by severe overlapping among pedestrians, whose heads are often almost the only visible part, and whose size (in pixel) is relatively small. Moreover, since the ground truth for CNN-based models consists in pedestrians' head positions, they tend to locate heads in testing images (see Fig. 4 as an example) which makes them less sensitive to pedestrian appearance, including pose and height.

6. Conclusions

We proposed a simple method for building *scene-specific* crowd counting models, focusing on challenging application scenarios where a suitable set of representative crowd images from the target camera is not available, not even unlabelled, for model training or fine-tuning. In such scenarios, the usual cross-scene solution based on training images from other scenes (i.e., benchmark data sets)

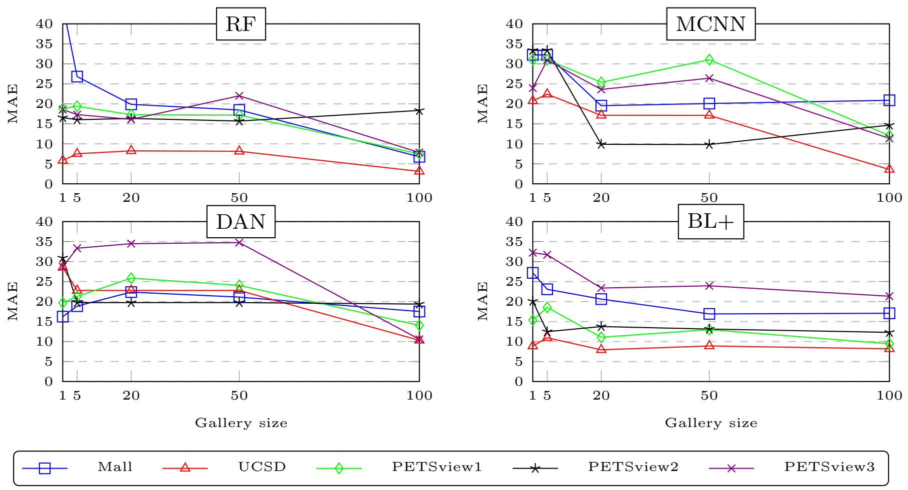

Figure 8: MAE values achieved by RF, MCNN, DAN and BL+ on the five target scenes using synthetic training data, as a function of the number of pedestrian images in the gallery. Best viewed in colour.

can significantly reduce the performance of existing models, including state-of-the-art CNN-based ones, up to the one of early regression-based methods. Our method generates synthetic training images of the target scene characterised by the same background, scale and perspective. To this aim, a background image of the target scene is required, together with its perspective map and region of interest; these three components can be obtained in practice during camera set-up, using different techniques, at the cost of a minimal effort from end-users (e.g., LEA operators). In particular, no collection nor manual annotation of images of the target scene is required. Additionally, the proposed method can be applied to *any* regression-based crowd counting model.

Experiments carried out on several benchmark data sets provided evidence that our solution can improve the effectiveness of existing crowd counting methods, especially on target scenes whose background, scale and perspective significantly differ from the ones of training images. This is a relevant result for real-world applications such as the one mentioned above, where an "out of the box" crowd counting functionality embedded into a video surveillance software suite has to be deployed at several, different target cameras. We showed that synthetic training images can also improve the quality of crowd density maps, which are estimated by most CNN-based models as an intermediate step, in terms of pedestrian localisation; in particular, this can occur even if the corresponding crowd count accuracy does not improve.

Possible limitations to the effectiveness of the proposed method can arise from an inaccurate estimation of the perspective map, as pointed out in our experiments. Robust techniques are therefore recommended to estimate it. A further and well-known issue could arise from variations in weather conditions and daytime lighting, affecting image illumination and colours. Nevertheless, synthetic images can be an effective solution to mitigate this issue: for instance, synthetic images simulating lighting and colour variations and specific weather conditions can be generated, and different models can be trained for specific conditions, which can then be easily selected by end-users depending on the particular environmental conditions [30]. Another interesting issue for future investigations is to improve the realism of synthetic images using computer graphics tools or GANs [11], to transfer the style of the target cameras to pedestrian images in the gallery.

ACKNOWLEDGEMENT

This work was supported by the projects "Law Enforcement agencies human factor methods and Toolkit for the Security and protection of CROWDs in mass gatherings" (LETSCROWD), EU Horizon 2020 programme, grant agreement No. 740466, and "IMaging MAnagement Guidelines and Informatics Network for law enforcement Agencies" (IMMAGINA), European Space Agency, ARTES Integrated Applications Promotion Programme, contract No. 4000133110/20/NL/AF.

References

[1] C. C. Loy, K. Chen, S. Gong, et al., Crowd counting and profiling: Methodology and evaluation, in: Model., sim. and vis. anal. of crowds, 2013, pp.

347–382.

[2] V. Sindagi, V. M. Patel, A survey of recent advances in cnn-based single image crowd counting and density estimation, Patt. Rec. Lett. 107 (2018) 3–16.

[3] V. A. Sindagi, V. M. Patel, Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting, in: AVSS, 2017, pp. 1–6.

[4] A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao, L. Shao, Relational attention network for crowd counting, in: ICCV, 2019, pp. 6787–6796.

[5] Z. Ma, X. Wei, X. Hong, Y. Gong, Bayesian loss for crowd count estimation with point supervision, in: ICCV, 2019, pp. 6141–6150.

[6] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, R. V. Babu, Locate, size, and count: Accurately resolving people in dense crowds via detection, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 2739–2751.

[7] V. A. Sindagi, V. M. Patel, HA-CCN: hierarchical attention-based crowd counting network, IEEE Trans. on Image Processing 29 (2020) 323–335.

[8] R. Delussu, L. Putzu, G. Fumera, An empirical evaluation of cross-scene crowd counting performance, in: VISIGRAPP, 2020, pp. 373–380.

[9] C. Change Loy, S. Gong, T. Xiang, From semi-supervised to transfer counting of crowds, in: ICCV, 2013, pp. 2256–2263.

[10] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: CVPR, 2015, pp. 833–841.

[11] Q. Wang, J. Gao, W. Lin, Y. Yuan, Learning from synthetic data for crowd counting in the wild, in: CVPR, 2019, pp. 8198–8207.

[12] J. C. Silveira Jacques Jr., S. R. Musse, C. R. Jung, Crowd analysis using computer vision techniques, IEEE Sign. Proc. Mag. 27 (5) (2010) 66–77.

[13] R. Delussu, L. Putzu, G. Fumera, Investigating synthetic data sets for crowd counting in cross-scene scenarios, in: VISIGRAPP, 2020, pp. 365–372.

[14] Y. Zhang, D. Zhou, S. Chen, et al., Single-image crowd counting via multi-column convolutional neural network, in: CVPR, 2016, pp. 589–597.

[15] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, L. Lin, Crowd counting with deep structured scale integration network, in: ICCV, 2019, pp. 1774–1783.

[16] N. Liu, Y. Long, C. Zou, et al., Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding, in: CVPR, 2019, pp. 3225–3234.

[17] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, in: CVPR, 2018, pp. 1091–1100.

[18] J. Gao, Q. Wang, Y. Yuan, SCAR: spatial-/channel-wise attention regression networks for crowd counting, Neurocomputing 363 (2019) 1–8.

[19] W. Liu, M. Salzmann, P. Fua, Context-aware crowd counting, in: CVPR, 2019, pp. 5099–5108.

[20] Z. Zou, X. Su, X. Qu, P. Zhou, Da-net: Learning the fine-grained density distribution with deformation aggregation network, IEEE Access 6 (2018) 60745–60756.

[21] Y. Zhang, C. Zhou, F. Chang, A. C. Kot, A scale adaptive network for crowd counting, Neurocomputing 362 (2019) 139–146.

[22] Y. Chen, X. Zhu, S. Gong, Instance-guided context rendering for cross-domain person re-identification, in: ICCV, 2019, pp. 232–242.

[23] D. Mansur, M. Haque, K. Sharma, et al., Use of head circumference as a predictor of height of individual, Kathmandu University Medical Journal (KUMJ) 12 (2) (2014) 89–92.

[24] M. Shen, T. Sun, X. Jiang, K. Xu, Crowd counting estimation in video surveillance based on linear regression function, CISP-BMEI (2017) 60–65.

[25] D. Ryan, S. Denman, S. Sridharan, C. Fookes, An evaluation of crowd counting methods, features and regression models, Comput. Vis. Image Underst. 130 (2015) 1–17.

[26] K. Chen, C. C. Loy, S. Gong, T. Xiang, Feature mining for localised crowd counting., in: BMVC, 2012, pp. 1–11.

[27] A. B. Chan, Z.-S. J. Liang, N. Vasconcelos, Privacy preserving crowd monitoring: Counting people without people models or tracking, in: CVPR, 2008, pp. 1–7.

[28] J. Ferryman, A. Shahrokni, Pets2009: Dataset and challenge, in: Int. work on PETS, 2009, pp. 1–6.

[29] Q. Zhang, A. B. Chan, Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns, in: CVPR, 2019, p. 8297–8306.

[30] A. Kerim, U. Celikcan, E. Erdem, A. Erdem, Using synthetic data for person tracking under adverse weather conditions, Image Vis. Comput. 111 (2021) 104187.