Noname manuscript No.
(will be inserted by the editor)

Pixel-wise Crowd Understanding via Synthetic Data

Qi Wang · Junyu Gao · Wei Lin · Yuan Yuan*

Received: date / Accepted: date

Abstract Crowd analysis via computer vision techniques is an important topic in the field of video surveillance, which has wide-spread applications including crowd monitoring, public safety, space design and so on. Pixel-wise crowd understanding is the most fundamental task in crowd analysis because of its finer results for video sequences or still images than other analysis tasks. Unfortunately, pixel-level understanding needs a large amount of labeled training data. Annotating them is an expensive work, which causes that current crowd datasets are small. As a result, most algorithms suffer from over-fitting to varying degrees. In this paper, take crowd counting and segmentation as examples from the pixel-wise crowd understanding, we attempt to remedy these problems from two aspects, namely data and methodology. Firstly, we develop a free data collector and labeler to generate synthetic and labeled crowd scenes in a computer game, Grand Theft Auto V. Then we use it to construct a large-scale, diverse synthetic crowd dataset, which is named as "GCC Dataset". Secondly, we propose two simple methods to improve the performance of crowd understanding via exploiting the synthetic data. To be specific, 1) supervised crowd understanding: pre-train a crowd analysis model on the synthetic data, then fine-tune it using the real data and labels, which makes the model perform better on the real world; 2) crowd understanding via domain adaptation: translate the synthetic data to photo-realistic images, then train the model on translated data and labels. As a result, the trained model works well in real crowd scenes.

Extensive experiments verify that the supervision algorithm outperforms the state-of-the-art performance on four real datasets: UCF_CC_50, UCF-QNRF, and Shanghai Tech Part A/B Dataset. The above results show the effectiveness, values of synthetic GCC for the pixel-wise crowd understanding. The tools of collecting/labeling data, the proposed synthetic dataset and the source code for counting models are available at https://gjy3035.github.io/GCC-CL/.

Keywords Crowd analysis · pixel-wise understanding · crowd counting · crowd segmentation · synthetic data generation

1 Introduction

Recently, crowd analysis has been a hot topic in the field of computer vision. It has great potential (including visual surveillance, crowd management, public space design and so on) in the real-world crowd scenes: railway station, shopping mall, stadium, airport terminals, theaters, public buildings, etc. (Chan et al., 2009; Li et al., 2014). Some typical crowd analysis tasks includes crowd counting/density estimation (Chan et al., 2008; Chan and Vasconcelos, 2009; Idrees et al., 2013; Wan et al., 2019; Yan et al., 2019), crowd segmentation (Dong et al., 2007; Kang and Wang, 2014), crowd anomaly detection (Mahadevan et al., 2010; Li et al., 2013; Yuan et al., 2014), human behavior recognition (Mehran et al., 2009; Popoola and Wang, 2012), person group detection (Li et al., 2017; Wang et al., 2018a), pedestrian tracking (Zuo et al., 2018; Li et al., 2018a). To be specific, crowd counting and segmentation are two essential tasks in crowd analysis, which are pixel-wise regression and classification task. In this paper, the both are together treated as **Pixel-wise Crowd Understanding**. The former aims to estimate

* Y. Yuan is the corresponding author.
Q. Wang, J. Gao, W. Lin and Y. Yuan are with the School of Computer Science and with the Center for Optical Imagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China. E-mails: crabwq@gmail.com, gjy3035@gmail.com, elonlin24@gmail.com, y.yuan1.ieee@gmail.com.
This work was supported by the National Key R&D Program of China under Grant 2017YFB1002202, National Natural Science Foundation of China under Grant U1864204, 61773316, 61632018, and 61825603.

Fig. 1 Two strategies of using the synthetic GCC dataset for pixel-wise crowd understanding: supervised learning and domain adaptation. The former firstly trains a pre-trained model on GCC and then fine-tune the model on real-world data. This strategy is able to significantly improve performance of the traditional supervise methods. The latter firstly adopts a CycleGAN-based method to translate the GCC data to photo-realistic scenes, then the trained crowd model uses the translated data and labels. The entire domain adaptation process does not need any label of real data.

the crowd density for each region in scenes and produces the number of people. The latter can predict the crowd region at the pixel level. Accurate density estimation and crowd localization provide some important attention information and fine pixel-wise results for other high-level analysis tasks. For example, semantic person group detection needs preliminary crowd segmentation results. Thus, pixel-wise crowd understanding is the most fundamental task in the field of crowd analysis.

With the development of deep learning, many methods (Onororubio and Lopezsastre, 2016; Walach and Wolf, 2016; Xiong et al., 2017; Sam et al., 2018; Shen et al., 2018; Shi et al., 2018) based on CNN attain a remarkable performance in the pixel-wise understanding task on the current research datasets. However, most algorithms only focus on how to learn effective and discriminative features (such as texture patterns, local structural features, global contextual information, multi-scale features and so on) to improve models' performance in the specific dataset but ignore the results in the wild. In fact, there is a large performance degradation during deploying them to the wild or other real-world scenes. The essential reason is that the model over-fits the scarce data.

For the scarce data problem, extending data seems like a very straightforward and effective solution. Unfortunately, annotating pixel-wise data is an expensive task: accurately labeling a scene contains 1,000 people will take more than 40 minutes. Thus, current crowd datasets (Chan et al., 2008; Chen et al., 2012; Zhang et al., 2016a,b; Wang et al., 2018b; Idrees et al., 2013, 2018) are small data volume so that they can not perfectly satisfy the needs of the mainstream CNN-based methods. Take the congested crowd dataset as an example, NWPU-Crowd contains only 5,106 images. In addition, there are disadvantages in the existing crowd dataset. Firstly, for extremely congested scenes, the labels of head locations are not very accurate and some heads are missed, such as some samples in UCF_CC_50 and Shanghai Tech Part A ("SHT A" for short). Secondly, existing datasets lack scenes such as night, variant illumination, and a large-range number of people, which are very common in real life.

Therefore, we first try to reduce the above problems from the data point of view. Our goal is to construct a large-scale, diverse, low-cost crowd dataset. To this end, with the help of a game engine, we develop a data collector and labeler in a popular computer game (Grand Theft Auto V, GTA V for short, an electronic game developed by Rockstar Games), which can generate synthetic crowd scenes and automatically annotate them. By the proposed collector and labeler, we successfully construct a synthetic crowd dataset, which is named as "GTA V Crowd Counting" ("GCC" for short) dataset. Compared with the existing real-world datasets. GCC dataset has four advantages: 1) free collection and annotation; 2) larger data volume and higher image resolution; 3) more diversified scenes, 4) more accurate annotations and label types (head dot and crowd mask). The detailed statistics are reported in Table 1.

After generating the large-scale dataset, then we attempt to exploit it to improve the performance in the wild from the methodological perspective. Here, we have two ways to realize our goal. To be specific, we firstly exploit the synthetic data to pre-train a crowd model, a Fully Convolutional Network, then fine-tune the model using real-world data. This strategy can effectively prompt performance in the real world. Traditional methods (training from scratch (Zhang et al., 2016b; Ranjan et al., 2018; Cao et al., 2018) or image classification models (Babu Sam et al., 2018; Shi et al., 2018; Idrees et al., 2018)) have some layers with random initialization or a regular distribution, which is not a good scheme. Compared with them, our strategy can provide more complete and better initialization parameters. The entire pipeline is shown in the orange box of Figure 1.

The above supervised learning method still needs real-world data and labels. It is an intractable problem that how to get rid of manual labels. Inspired by un-paired image translation (Zhu et al., 2017), we firstly translate the synthetic images to photo-realistic images. Compared with the

original CycleGAN, we propose the Structural Similarity Index (SSIM) loss to maintain the texture features and local patterns in the crowd region during the translation process. Then, we train a crowd model via adversarial learning on the labeled translated domain and the unlabeled real domain, which works well in the wild. The flowchart is demonstrated in the light blue box of Fig. 1.

In summary, this paper's contributions are four-fold:

1) We are the first to develop a data collector and labeler for crowd counting, which can automatically construct synthetic crowd scenes in GTA V game and simultaneously annotate them without any labor costs.
2) We create the first large-scale and synthetic crowd counting dataset by using the data collector and labeler, which contains 15,212 images, a total of 7,625,843 people.
3) We present a pre-trained scheme to facilitate the original method's performance on the real data, which can more effectively reduce the estimation errors compared with random initialization and ImageNet model. Further, through the strategy, our proposed SFCN achieves the state-of-the-art results.
4) We are the first to propose a crowd understanding method via domain adaptation, which does not use any label of the real data. By our designed SE CycleGAN, the domain gap between the synthetic and real data can be significantly reduced. Finally, the proposed method outperforms the two baselines that face the same problem.

This paper is an extension of our previous work on crowd counting (Wang et al., 2019) in the IEEE Conference on Computer Vision and Pattern Recognition. Compared with the conference version, this paper has some extensions as follows:

1) **Data Generation:** The process of scene synthetic is optimized for more efficient data generation, which reduces the simulation time by two-thirds. Furthermore, we give a more detailed description of the entire process for data generation, including scene selection, setting and synthesis.
2) **Dataset:** In addition to the head dot labels, the crowd mask for segmentation is provided. It displays pixel-wise salient regions of crowd, which is also an important fundamental task of crowd analysis.
3) **Methodology:** For supervised crowd understanding, we add a new network to segment the crowd mask based on SFCN. For Crowd understanding via domain adaptation, we add the adversarial learning to jointly train SE CycleGAN and SFCN to further prompt counting performance in the real world.
4) **Experiments:** More further experiments are conducted to verify the effectiveness (improvement of performance, generalization ability, etc.) of the two proposed ways on the real-world datasets.

The rest of this paper is organized as follows. Section 2 reviews the related work briefly in terms of crowd understanding, crowd dataset and sythetic dataset. Section 3 describes the generation process and key features of GCC dataset. Section 4 and 5 respectively focuses on supervised learning and domain adaptation for pixel-wise crowd understanding. Further, the experimental results and discussion are analyzed in Section 6. Finally, this work is summarized in Section 7.

2 Related Works

2.1 Pixel-wise Crowd Understanding

Pixel-wise crowd understanding mainly consists of crowd counting and segmentation task. In the past half-decade, researchers exploit Convolutional Neural Network (CNN) to design the effective crowd model, which attains a significant improvement. Some methods (Wang et al., 2015; Fu et al., 2015) attempt to directly regress the number of people for image patches by modifying the traditional CNN classification models. However, there are more semantic gaps in direct regression than pixel-wise density estimation. Thus, many methods (Marsden et al., 2016; Liu et al., 2018b; Kang and Wang, 2014) adopt Fully Convolutional Networks (FCN) (Long et al., 2015) to produce the pixel-wise density map or predict the crowd region.

Benefiting from FCN's remarkable performance on pixel-wise task (such as semantic segmentation, saliency detection and on), almost all algorithms use FCN to predict crowd density map. Some methods (Zhang et al., 2016b; Idrees et al., 2018; Cao et al., 2018; Ranjan et al., 2018) integrate different feature maps from different layers in FCN to improve the quality of density maps. To be specific, Zhang et al. (2016b); Cao et al. (2018) design a multi-column CNN and fuses the feature map from different columns to predict the final density map. Idrees et al. (2018) compute loss from shallow to deep layers by different loss functions to output fine maps. Jiang et al. (2019) present a combinatorial loss to enforce similarities in different spatial scales between prediction maps and groundtruth. However, it is hard to train a single regression model for density map estimation, which converges slowly and performs not well. Thus, some methods (Sindagi and Patel, 2017a; Gao et al., 2019; Lian et al., 2019; Zhao et al., 2019) exploit multi-task learning to explore the relation of different tasks to improve the counting performance, such as high-level density classification, foreground/crowd segmentation, perspective prediction, crowd depth estimation and so on. In addition to multi-task learning, Sindagi and Patel (2017b); Li et al. (2018b); Liu et al. (2019) attempt to encode the large-range contextual information via patch-level classification, dilatation convolution and multiple receptive field sizes for crowd scenes.

Table 1 Statistics of the eight real-world datasets and the synthetic GCC dataset. Specifically, the real-world datasets include UCSD (Chan et al., 2008), Mall (Chen et al., 2012), UCF_CC_50 (Idrees et al., 2013), WorldExpo'10 (Zhang et al., 2016a), SHT A/B (Zhang et al., 2016b), UCF-QNRF (Idrees et al., 2018), NWPU-Crowd (Wang et al., 2020) and JHU-CROWD++ (Sindagi et al., 2019, 2020).

Dataset	Number of Images	Average Resolution	Count Statistics				Label Forms	
			Total	Min	Ave	Max	Head Dot	Crowd Mask
UCSD	2,000	158×238	49,885	11	25	46	✔	✗
Mall	2,000	480×640	62,325	13	31	53	✔	✗
UCF_CC_50	50	2101×2888	63,974	94	1,279	4,543	✔	✗
WorldExpo'10	3,980	576×720	199,923	1	50	253	✔	✗
SHT A	482	589×868	241,677	33	501	3,139	✔	✗
SHT B	716	768×1024	88,488	9	123	578	✔	✗
UCF-QNRF	1,525	2013×2902	1,251,642	49	815	12,865	✔	✗
NWPU-Crowd	5,109	2311×3383	2,133,238	0	418	20,033	✔	✗
JHU-CROWD++	4,372	910×1430	1,515,005	0	346	25,791	✔	✗
GCC	**15,212**	$\mathbf{1080 \times 1920}$	**7,625,843**	**0**	**501**	**3,995**	✔	✔

For handling scarce training data, Liu et al. (2018c) propose a self-supervised learning method to learn to rank a large amount of unlabeled web data, and Shi et al. (2018) present a deep negative correlation learning to reduce the over-fitting. Sam et al. (2019) present an almost unsupervised dense counting autoencoder, in which 99.9% of parameters are trained without any labeled data.

2.2 Crowd Datasets

In addition to the above algorithms for crowd understanding, the datasets potentially boost the development of crowd counting. The first crowd counting dataset, UCSD (Chan et al., 2008), is released by Chan *et al.* from the University of California San Diego, which records a sparse crowd scene in a pedestrian walkway. In addition to counting labels, UCSD also provides identity information, traveling direction and instantaneous velocity for each person. Mall dataset (Chen et al., 2012) is captured from a surveillance camera by Chen *et al.*, containing over 60,000 pedestrian instances in an indoor shopping mall.

Considering that UCSD and Mall are collected from a single sparse scene, some researchers build extremely congested and diversified-scene crowd counting dataset. Idrees et al. (2013) release a highly congested crowd dataset named UCF_CC_50, containing 50 images. The average number of people per image is more than 1,200. Zhang et al. (2016a) construct a cross-scene crowd dataset, WorldExpo'10. It includes 120 different crowd scenes that are captured from surveillance cameras in Shanghai 2010 WorldExpo. Zhang et al. (2016b) present ShanghaiTech Dataset, including the high-quality real-world images. It consists of 2 parts: Part A is collected from a photo-sharing website [1] and Part B is captured from the walking streets in Shanghai. For covering the more large-range, large-scale crowd scene and more accurate annotation, Idrees et al. (2018) propose a highly congested dataset with higher resolution, UCF-QNRF, of which number range if from 49 to 12,865. UCF-QNRF is by far the largest extremely congested crowd counting dataset, containing 1,525 images, in total of 1,251,642 persons. Recently, JHU and NWPU release two large-scale crowd counting datasets, (Wang et al., 2020; Sindagi et al., 2019, 2020), which will promote the further development of the community of crowd counting.

However, the above datasets suffer some drawbacks mentioned in Section 1 to some extent. More detailed information about them is shown in Table 1.

2.3 Synthetic Dataset

In recent years, the mainstream deep-learning-based methods achieve a remarkable improvement relying on a large amount of training data (Deng et al., 2009; Lin et al., 2014; Abu-El-Haija et al., 2016). But annotating the groundtruth of massive amounts of data is a time-consuming and labor-intensive assignment, especially for pixel-wise tasks (such as semantic segmentation, density map estimation). According to the statistics, annotating a crowd scene with 1,000 people takes more than 40 minutes. To alleviate this problem, many methods about the generation and low-cost annotation of synthetic data are proposed. Richter et al. (2016) collect $\sim 25,000$ synthetic street scenes from GTA V. Meanwhile, they propose a low-cost annotation method that can save 90% manual labeling time. Ros et al. (2016) release a synthetic semantic segmentation dataset by constructing a virtual world. They exploit Unity Engine [2] (an open-source game engine) to design various common object models, such

[1] https://www.flickr.com/

[2] https://unity3d.com/

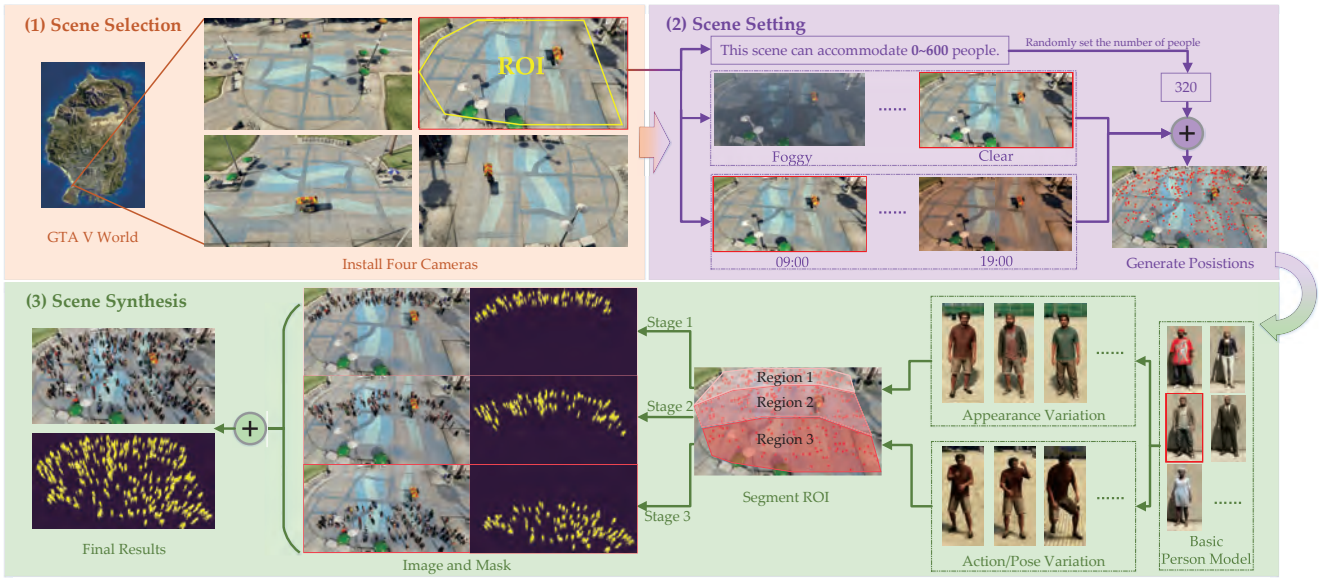


Fig. 2 The illustration of crowd scene generation in GTA 5, including scene selection, setting and synthesis.

as pedestrians, cars, buildings, etc. In the field of autonomous driving, Johnson-Roberson et al. (2017) construct a synthetic auto-driving dataset from GTA V. At the same time, they present an automatic method to analyze the depth information from the game engine to get the accurate object masks. Based on GTA V, Richter et al. (2017) propose a large-scale benchmark for autonomous driving, which provides six types of data, namely video frame, semantic segmentation mask, instance mask, optical flow, 3D layout, and visual odometry. Bak et al. (2018) develop a synthetic person re-identification dataset based on Unreal Engine 4 [3].

In addition to the aforementioned datasets, some famous synthetic data platforms/tools (Kempka et al., 2016; Dosovitskiy et al., 2017; Qiu et al., 2017; Shah et al., 2018) are released. Kempka et al. (2016) propose a test-bed platform for visual reinforcement learning, which adopts a First-Person Perspective (FPP) in the constructed semi-realistic 3D world. CARLA (Dosovitskiy et al., 2017) an open-source simulator for self-driving research, which supports the flexible specification of sensor suites and environmental conditions. UnrealCV (Qiu et al., 2017) is a project to help researchers build virtual worlds using Unreal Engine 4, which provides some key commands to interact with the virtual world and API to connect external programs, such as Caffe (Jia et al., 2014). AirSim (Shah et al., 2018) is a simulator for cars, drones or other unmanned vehicles, which is an open-source and cross platform. It supports hardware-in-loop with flight controllers and provides depth information, RGB images and pixel-level segmentation masks.

[3] https://www.unrealengine.com/

3 GTA5 Crowd Counting (GCC) Dataset

Grand Theft Auto V (GTA5) is an electronic game published by Rockstar Games [4] in 2013. GTA5 utilizes the proprietary Rockstar Advanced Game Engine (RAGE) to improve its draw distance rendering capabilities. Benefiting from the excellent game engine, its scene rendering, texture details, weather effects and so on are very close to the real-world conditions. Rockstar Games constructs a virtual world, including the fictional Blaine County, and the fictional city of Los Santos. GTA5 allows players freely roam the open world and explore more gameplay content. In addition, Rockstar Games allows the players to develop the mod to achieve specific needs in the game. It must be noncommercial or personal use[5,6] and not be used in an online version.

Considering the aforementioned advantages and characteristics, we develop a data collector and labeler to construct crowd scenes in GTA5, which is based on Script Hook V [7]. Script Hook V is a C++ library for developing game plugins, which allows developers to get the game data from rendering stencil. The data collector firstly constructs the congested crowd scenes via controlling the objects (pedestrians, cars, etc.) and setting attributes (weathers, timestamp, etc.) of the virtual world. Then, by analyzing the data from rendering stencil, the labeler automatically annotates the accurate head locations of persons without any manpower.

[4] https://www.rockstargames.com/

[5] https://support.rockstargames.com/articles/115009494848/PC-Single-Player-Mods

[6] https://support.rockstargames.com/articles/200153756/Policy-on-posting-copyrighted-Rockstar-Games-material

[7] http://www.dev-c.com/gtav/scripthookv/

Previous synthetic GTA5 datasets (Richter et al., 2016; Johnson-Roberson et al., 2017; Richter et al., 2017) capture normal scenes directed by the game programming. Unfortunately, there is no congested scene in GTA5. Thus, we need to design a strategy to construct crowd scenes, which is the most obvious difference with them.

3.1 Data Collection

In this section, we briefly describe the key step in each component as shown in Fig. 2. Scene Selection: a) select location in the GTAV world; b) equip four cameras with different appropriate parameters; c) draw a reasonable Region of interest (ROI) for crowd. Scene Setting: a) set the level of scene capacity according to the ROI's size; b) set weather conditions and time randomly; c) set the number and positions of people randomly. Scene Synthesis: a) place pedestrians in order; b) capture crowd information; c) integrate multiple scenes into one scene; d) remove the labels of occluded heads. The video demonstration is available at: https://www.youtube.com/watch?v=Hvl7xWkIueo.

Scene Selection. The virtual world in GTA5 is built on a fictional city, which covers an area of 252 square kilometers. In the city, we select 100 typical locations, such as the beach, stadium, mall, store and so on. To capture the scene from multiple views, the four surveillance cameras are equipped with different parameters (location, height, rotation/pitch angle) for each location. As a result, a total of 400 diverse scenes are built. In order to obtain the realistic constructed crowd scenes, we delimit a polygon area to place the person model, which is named as "Region of Interest (ROI)". The main purpose of this operation is to prevent people from appearing on unreasonable objects.

Scene Setting. After the scene selection, we need to set some basic attributes for each scene. Firstly, according to the area of ROI, we set a range of the number of people. Then, to enhance the diversity of data, the weather condition and time are randomly set to simulate various illumination and brightness during each generation. The distribution of different attributes will be reported in Section 3.2. Finally, we pre-generate the number of people and each people's position.

Scene Synthesis. The last step is the scene synthesis, the most important part of the entire data construction. Due to the limitation of GTA5, the number of people can not exceed a maximum value of 256. To create a congested crowd scene, we segment several non-overlapping regions according to the distance from the Surveillance camera to and place persons in each region, as shown in Fig. 2(3). As for each region, we save the relevant information, such as image, segmentation mask, and head point coordinates. Next, according to the crowd mask information, all images are integrated into one scene. Finally, we remove the labels of heads occluded by other people or objects and update the label information.

The person is a core component in the crowd scene. During the scene synthesis, we employ 265 basic person models from GTAV to simulate the crowd, and every model comes with a different combination of skin color, gender, height, weight, age and so on[8]. Besides, for each person model, it has six variations on external appearance, such as clothing, haircut, etc. Theoretically, we can create far more than person models with different appearance features. For mimicking the various poses in the real world, each person is programmed to do a random action in sparse crowd scenes.

3.2 Properties of GCC

GCC dataset consists of 15,212 images, with a resolution of 1080×1920, containing more than 7,000,000 persons. Compared with the existing datasets, GCC is the largest crowd counting dataset not only in the number of images but also in the number of persons. Table 1 reports the basic information of GCC and the existing datasets, including data volume, image resolution, label forms and so on. In addition to the above advantages, GCC is more diverse than other real-world datasets.

Diverse Scenes. In addition to the advantage in terms of data volume, GCC is more diverse than other real-world datasets. It captures 400 different crowd scenes in the GTA V game, which includes multiple types of locations. For example, indoor scenes: office, convenience store, pub, etc. outdoor scenes: mall, walking street, sidewalk, plaza, stadium and so on. Furthermore, all scenes are assigned with a level label according to their space capacity. The first row in Fig. 3 shows the typical scenes with different levels. In general, for covering the range of people, a larger scene should have more images. Thus, the setting is conducted as follows: the scenes with the first/second/last three levels contain $30/40/50$ images. Besides, these images containing some improper events should be deleted. Finally, the number of images in some scenes may be less than their expected value. Fig. 4 demonstrates the population distribution histogram of our GCC dataset.

Existing datasets only focus on one of the sparse or congested crowd. However, a large scene may also contain very few people in the wild. Considering that, during the generation process of an image, the number of people is set as a random value in the range of its level. Therefore, GCC has more large-range than other real datasets.

Diverse Environments. In order to construct the data that are close to the wild, the images are captured at a random time in a day and under a random weather condition. In

[8] https://wiki.gtanet.work/index.php?title=Peds

Fig. 3 The display of the proposed GCC dataset from three views: scene capacity, timestamp and weather conditions.

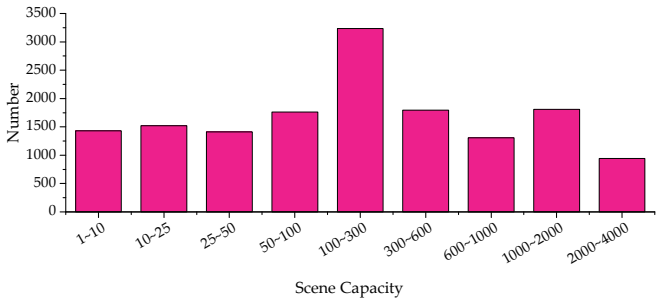
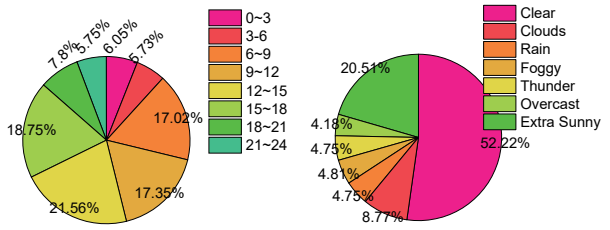


Fig. 4 The statistical histogram of crowd counts on the proposed GCC dataset.



(a) Time stamp distribution. (b) Weather condition distribution.

Fig. 5 The pie charts of time stamp and weather condition distribution on GCC dataset. In the left pie chart, the label "0 $\sim$ 3" denotes the time period during $[0:00,\ 3:00)$ in 24 hours a day.

GTA5, we select common weathers, namely clear, clouds, rain, foggy, thunder, overcast and extra sunny. The last two rows of Fig. 3 illustrate the exemplars at different times and under various weathers. In the process of generation, we tend to produce more images under common conditions. Specifically, we prefer to generate more daytime scenes and fine-weather scenes. The two sector charts in Fig. 5 respectively shows the proportional distribution on the time stamp and weather conditions of the GCC dataset.

Splitting for Evaluation. In order to fully verify the performance of the algorithm, we propose three different schemes to split the dataset into two parts (namely training and testing):

1) **Random splitting**: the entire dataset is randomly divided into two groups as the training set (75%) and testing set (25%), respectively.
2) **Cross-camera splitting**: as for a specific location, one surveillance camera is randomly selected for testing and the others for training.
3) **Cross-location splitting**: we randomly choose 75/25 locations for training/testing, which can be treated as a cross-scene evaluation.

Obviously, the task difficulty of the three strategies is increased sequentially. The last two splitting methods can effectively evaluate the generalization ability of models.

4 Supervised Crowd Understanding

In this section, we propose a Spatial FCN for crowd understanding, focusing on counting and segmentation tasks.

4.1 Spatial FCN for Crowd Understanding

In 2014, Long et al. (2015) and Kang and Wang (2014) propose the Fully Convolutional Network (FCN) almost simultaneously, which focuses on image segmentation. It uses the convolutional layer to replace the fully connected layer so that it can process the image with an arbitrary size and output the map of the corresponding size. To encode the large-range contextual information, Pan et al. (2017) present a spatial encoder via a sequence of convolution on the four directions (down, up, left-to-right and right-to-left).

Inspired by Pan et al. (2017), we design a spatial FCN (SFCN) to estimate the crowd density maps. The spatial encoder is added to the top of the backbones: such as VGG-16 Network (Simonyan and Zisserman, 2014) and ResNet-101 (He et al., 2016). After the spatial encoder, a regression layer is added for crowd counting, which directly outputs the density map with input's $1/8$ size. For predicting a finer crowd mask, we adopt de-convolutional layers to produce the mask with the original input's size. In addition, classification and soft-max layers are also added to the top of the network. The detailed network configurations are shown in supplementary materials.

Loss Function Standard pixel-wise Mean Squared Error is used to optimize the proposed SFCN for crowd Counting.

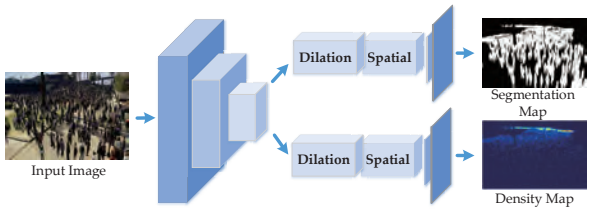


Fig. 6 The architecture of spatial FCN (SFCN).

During the training process of the segmentation model, the objective is standard 2-D Cross Entropy loss.

4.2 Pretraining & Fine-tuning

Many current methods suffer from the over-fitting because of scarce real labeled data. Some methods (Babu Sam et al., 2018; Shi et al., 2018; Idrees et al., 2018) exploit the pre-trained model based on ImageNet Database (Deng et al., 2009) as the backbone. However, the pre-trained classification models(VGG (Simonyan and Zisserman, 2014), ResNet (He et al., 2016) and DenseNet (Huang et al., 2017)) may be not the best selection, because they only provide the initialization for the backbone. Some specific layers (such as regression/classification modules, context encoders) are still initialized at a random or regular distributions. To remedy this problem, we propose a new pre-trained scheme, named as "Pre-GCC" (similarly, the traditional scheme is named as "Pre-ImgNt"). To be specific, the designed model is firstly pre-trained on the large-scale GCC Dataset, then the pre-trained model is fine-tuned using the real data. Compared with the pre-trained model on ImageNet, Pre-GCC scheme provides entire initialized parameters so that it can alleviate over-fitting more effectively.

5 Crowd Understanding via Domain Adaptation

The last section proposes the Pre-GCC scheme that can significantly improve the model's performance on the real data. However, this strategy still relies on the labels of real datasets. In Section 1, we have mentioned that manually annotating extremely congested scenes is a tedious task. Spontaneously, we attempt to find a new way to get rid of the burden of labeling data. Therefore, we propose a crowd counting method via Domain Adaptation (DA) to save human resources. The purpose of CU via DA is to learn the translation and domain-invariant features mapping between the synthetic domain $\mathcal{S}$ and the real-world domain $\mathcal{R}$. The synthetic domain $\mathcal{S}$ provides images $I_{\mathcal{S}}$ and count labels $L_{\mathcal{S}}$. The real-world domain $\mathcal{R}$ only provides images $I_{\mathcal{R}}$. In a word, given $\mathrm{i}_{\mathcal{S}} \in I_{\mathcal{S}}$, $\mathrm{l}_{\mathcal{S}} \in L_{\mathcal{S}}$ and $\mathrm{i}_{\mathcal{R}} \in I_{\mathcal{R}}$ (the lowercase letters represent the samples in the corresponding sets), we want to train a crowd counter to predict density maps of $\mathcal{R}$.

5.1 SSIM Embedding CycleGAN

In this section, we propose a crowd counting method via domain adaptation, which can learn domain-invariant features effectively between synthetic and real data. To be specific, we present an SSIM Embedding (SE) CycleGAN to translate the synthetic image to the photo-realistic image. At the same time, we use the adversarial learning for the SFCN counter to extract domain-invariant features in a hidden space. Finally, we directly apply the model to the real data. Fig. 7 illustrates the flowchart of the proposed method.

5.1.1 CycleGAN

The original CycleGAN (Zhu et al., 2017) focuses on unpaired image-to-image translation. For different two domains, we can exploit CycleGAN to handle DA problem, which can translate the synthetic images to photo-realistic images. As for the domain $\mathcal{S}$ and $\mathcal{R}$, we define two generator $G_{\mathcal{S}\to\mathcal{R}}$ and $G_{\mathcal{R}\to\mathcal{S}}$. The former one attempts to learn a mapping function from domain $\mathcal{S}$ to $\mathcal{R}$, and vice versa, the latter one's goal is to learn the mapping from domain $\mathcal{R}$ to $\mathcal{S}$. To regularize the training process, the cycle-consistent loss $\mathcal{L}_{cycle}$ is introduced. Additionally, two discriminators $D_{\mathcal{R}}$ and $D_{\mathcal{S}}$ are modeled corresponding to the $G_{\mathcal{S}\to\mathcal{R}}$ and $G_{\mathcal{R}\to\mathcal{S}}$. Specifically, $D_{\mathcal{R}}$ attempts to discriminate that where the images are from ($I_{\mathcal{R}}$ or $G_{\mathcal{S}\to\mathcal{R}}(I_{\mathcal{S}})$), and $D_{\mathcal{S}}$ tries to discriminate the images from $I_{\mathcal{S}}$ or $G_{\mathcal{R}\to\mathcal{S}}(I_{\mathcal{R}})$. For training $D_{\mathcal{R}}$ and $D_{\mathcal{S}}$, the standard adversarial loss $\mathcal{L}_{GAN}$ is optimized, which is proposed by Goodfellow et al. (2014). The final loss function is defined as:

$$\begin{aligned}\mathcal{L}_{CycleGAN}(G_{\mathcal{S}\to\mathcal{R}}, G_{\mathcal{R}\to\mathcal{S}}, D_{\mathcal{R}}, D_{\mathcal{S}}, I_{\mathcal{S}}, I_{\mathcal{R}}) \\ = \mathcal{L}_{GAN}(G_{\mathcal{S}\to\mathcal{R}}, D_{\mathcal{R}}, I_{\mathcal{S}}, I_{\mathcal{R}}) \\ + \mathcal{L}_{GAN}(G_{\mathcal{R}\to\mathcal{S}}, D_{\mathcal{S}}, I_{\mathcal{S}}, I_{\mathcal{R}}) \\ + \lambda\mathcal{L}_{cycle}(G_{\mathcal{S}\to\mathcal{R}}, G_{\mathcal{R}\to\mathcal{S}}, I_{\mathcal{S}}, I_{\mathcal{R}}),\end{aligned} \tag{1}$$

where λ is the weight of cycle-consistent loss.

5.1.2 SSIM Embedding CycleGAN

In the crowd scenes, the biggest differences between high-density regions and other regions (including background regions and low-density crowd) are the local patterns and texture features instead of structural head information. Unfortunately, in the translation from synthetic to real images, the original cycle consistency is prone to losing them, which causes that the translated images lose the detailed information and are easily distorted.

To remedy the aforementioned problem, we propose a Structural Similarity Index (SSIM) loss in CycleGAN, which is named as "SE CycleGAN". It can maintain the local structured features in the original crowd scenes. SSIM is proposed by Wang et al. (2004) to asses the reconstruction quality in the field of image denoising, super-resolution and so

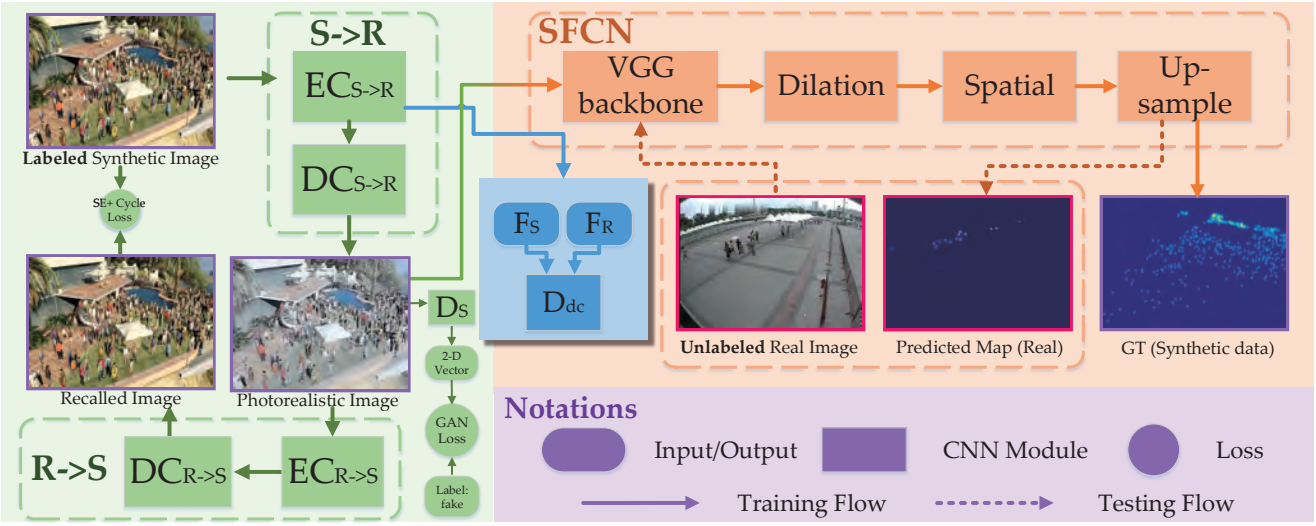


Fig. 7 The flowchart of the proposed crowd counting via domain adaptation. The light green/blue regions are SSIM Embedding (SE) CycleGAN, and light orange region represents Spatial FCN (FCN). Limited by paper length, we do not show the adaptation flowchart of real images to synthetic images (R→S), which is similar to S→R.

on, which computes the similarity between two images in terms of local patterns (mean, variance and covariance). In particular, when the two images are identical, the SSIM value is equal to 1. In the practice, we convert the SSIM value into the trainable form, which is defined as:

$$\begin{aligned}&\mathcal{L}_{SEcycle}(G_{\mathcal{S}\to\mathcal{R}}, G_{\mathcal{R}\to\mathcal{S}}, I_{\mathcal{S}}, I_{\mathcal{R}})\\&= \mathbb{E}_{i_{\mathcal{S}}\sim I_{\mathcal{S}}}[1 - SSIM(i_{\mathcal{S}}, G_{\mathcal{R}\to\mathcal{S}}(G_{\mathcal{S}\to R}(i_{\mathcal{S}})))]\\&+ \mathbb{E}_{i_{\mathcal{R}}\sim I_{\mathcal{R}}}[1 - SSIM(i_{\mathcal{R}}, G_{\mathcal{S}\to\mathcal{R}}(G_{\mathcal{R}\to S}(i_{\mathcal{R}})))],\end{aligned} \tag{2}$$

where $SSIM(\cdot,\cdot)$ is standard computation. The first input is the original image from domain $\mathcal{S}$ or $\mathcal{R}$, and the second input is the reconstructed image produced by the two generators in turns. Finally, the final objective of SE CycleGAN $\mathcal{L}_{trans}$ is the sum of $\mathcal{L}_{CycleGAN}$ and $\mathcal{L}_{SEcycle}$.

5.1.3 Feature-level Adversarial Learning

To further prompt the adaptation performance, we add the feature-level adversarial learning for the outputs of the two generators. According to the size of each layer in $G_{\mathcal{S}\to\mathcal{R}}$ and $G_{\mathcal{R}\to\mathcal{S}}$, the generator can be treated as two components: EnCoder and DeCoder (EC and DC for short, respectively). To be specific, EC contains the down-sampling operation for image and DC has up-sampling operation. For feature-level adversarial learning, a domain classifier is present to discriminate where EC's outputs ($F_{\mathcal{S}}$and $F_{\mathcal{R}}$) are from. By the adversarial learning (Goodfellow et al., 2014), the encoders can extract powerful domain-invariant features to fool the classifier. Specially, the classifier is a fully convolutional network, including four convolution layers with leaky ReLU.

As for SFCN, we select the feature maps of $I_{\mathcal{S}}^{trslt}$ and $I_{\mathcal{R}}$ after Spatial Module as the inputs for a domain classifier D_{dc}. The feature maps are written as $F_{\mathcal{S}}$ and $F_{\mathcal{R}}$, respectively. Through D_{dc}, the $O_{\mathcal{S}}$ and $O_{\mathcal{R}}$ can be obtained. For optimizing D_{dc}, a 2-D pixel-wise binary cross-entropy loss is performed. In order to confuse D_{dc}, the inverse adversarial loss should be added into the training of SFCN, which is defined as:

$$\mathcal{L}_{adv}(F_{\mathcal{R}}) = -\sum_{F_{\mathcal{R}}\in\mathcal{R}}\sum_{h\in H}\sum_{w\in W}\log(p(O_{\mathcal{R}})), \tag{3}$$

where $O_{\mathcal{R}}$ is 2D-channel maps with size of $H\times W$ for real feature input $F_{\mathcal{R}}$, H and W denote the height and width of the inputs, and $p(\cdot)$ is the soft-max operation for each pixel.

5.2 Joint Training

Finally, the joint training of SE CycleGAN and the SFCN counter is implemented by optimizing the following loss:

$$\mathcal{L}(I_{\mathcal{S}}, L_{\mathcal{S}}, I_{\mathcal{R}}) = \alpha\mathcal{L}_{cnt}(I_{\mathcal{S}}, L_{\mathcal{S}}) + \beta\mathcal{L}_{trans} + \lambda\mathcal{L}_{adv}(F_{\mathcal{R}}), \tag{4}$$

where $\mathcal{L}_{cnt}$ is the standard MSE loss on the translated synthetic domain, L_{adv} is the inverse adversarial loss for features $F_{\mathcal{R}}$ from the real domain in Section 5.1.3. α, β and λ are the weights to balance the losses.

5.3 Scene/Density Regularization

For a better domain adaptation from synthetic to real world, we design two strategies to facilitate the DA model to learn domain-invariant feature and produce the valid density map.

Scene Regularization. Since GCC is a large-counter-range and diverse dataset, using all images may cause the

side effect in domain adaptation. For example, ShanghaiTech does not contain the thunder/rain scenes, and WorldExpo'10 does not have a scene that can accommodate more than 500 people. Training all translated synthetic images can decrease the adaptation performance on the specific dataset. Thus, we manually select some specific scenes for different datasets. The concrete strategies are described in Section 6.3.1. In general, it is a coarse data filter, not an elaborate selection.

Density Regularization. Although we translate synthetic images to photo-realistic images, some objects and data distributions in the real world are unseen during training the translated images. As a pixel-wise regression problem, the density may be an arbitrary value in theory. In fact, in some preliminary experiments, we find some backgrounds in real data are estimated as some exceptionally large values. To handle this problem, we set a upper bound $MAX_{\mathcal{S}}$, which is defined as the max density in the synthetic data. If the output value of a pixel is more than $MAX_{\mathcal{S}}$, the output will be set as 0. Note that the network's last layer is ReLU, so the output of each pixel must be greater than or equal to 0.

6 Experiments

6.1 Metrics

In the field of crowd counting, the mainstream evaluation metrics are Mean Absolute Error (MAE) and Mean Squared Error (MSE), which are formulated as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|^2}, \quad (5)$$

where N is the number of samples in testing data, y_i is the count label (real number of people in an image) and $\hat{y}_i$ is the estimated count value for the ith test sample. In addition to the evaluation of final count, we also evaluate the quality of density maps using two mainstream criteria in image assessment: Peak Signal-to-Noise Ratio (PSNR) ans Structural Similarity in Image (SSIM) (Wang et al., 2004).

For crowd segmentation task, we use Intersection-over-Union (IoU) (Everingham et al., 2015) for crowd and background to evaluate crowd models, which is defined as:

$$IoU = \frac{TP}{TP + FP + FN},. \quad (6)$$

where TP, FP and FN are the numbers of true positive, false positive, and false negative samples, respectively.

6.2 Results of Supervised Crowd Understanding

In this section, the two types of experiments are conducted: 1) training and testing within GCC dataset; 2) pre-training on GCC and fine-tuning on the real datasets.

6.2.1 Implementation Details

We use C^3F (Gao et al., 2019) to conduct our designed experiments, which is an open-source PyTorch (Paszke et al., 2017) code framework for crowd counting, and all experiments are performed on NVIDIA GTX 1080Ti GPU. Different from C^3F, we randomly select 10% training data as the validation set to find the best model (which may result in some performance degradation compared with C^3F). As for different networks, the key hyper-parameters are listed in Table 2. In it, "lr" denotes learning rate; "dr" is decay rate of learning rate every epoch; "lnf" is short for "label normalization factor", which means that the density map is multiplied by a factor[9]. Adam (Kingma and Ba, 2014) algorithm is adopted to optimize the network and obtain the best result.

In this section, the experiments involve five networks: MCNN(Zhang et al., 2016b), CSRNet(Li et al., 2018b), FCN, SFCN and SFCN†. The first two are the original version in their published papers. The last three's detailed configurations are shown in Section 2.1 of the supplementary materials.

Other training parameters are logged in C^3F's repository[10]. By flexible design of C^3F, our every result can be effectively reproduced.

Table 2 The key parameters of training different models.

Method	backbone	lr	dr	lnf
MCNN	None	10^{-4}	1	100
CSRNet	VGG-16	10^{-5}	0.995	100
FCN	VGG-16	10^{-5}	0.995	100
SFCN	VGG-16	10^{-5}	0.995	100
SFCN†	ResNet-101	10^{-5}	0.995	100

6.2.2 Experiments on GCC Dataset

Performance of Overall Evaluation

We report the results of the extensive experiments within the GCC dataset, which verifies SFCN from three different training strategies: random, cross-camera and cross-location splitting. Table 3 reports the performance of our SFCN and two popular methods on the proposed GCC dataset. In the table, "fg" and "bg" respectively denotes the foreground and background in the scenes, and "mIoU" is the mean value of two classes of IoU.

[9] This trick effectively improves the counting performance (Gao et al., 2019)

[10] https://github.com/gjy3035/C-3-Framework/tree/python3.x/results_reports

Table 3 The results (MAE↓/MSE↓/PSNR↑/SSIM↑/IoU↑) of our proposed SFCN and the two classic methods (MCNN (Zhang et al., 2016b) and CSRNet (Li et al., 2018b)) on GCC dataset.

Performance of random splitting

Method	Counting				Segmentation(%)		
	MAE	MSE	PSNR	SSIM	fg	bg	mIoU
MCNN	100.9	217.6	24.00	0.838	77.8	40.2	59.0
CSRNet	38.2	87.6	29.52	0.829	94.0	73.9	83.9
FCN	42.3	98.7	30.10	0.889	93.7	71.2	82.5
SFCN	36.2	81.1	30.21	0.904	94.3	74.7	84.5
SFCN†	**28.1**	**70.2**	**31.03**	**0.927**	**94.7**	**76.1**	**85.5**

Performance of cross-camera splitting

Method	Counting				Segmentation(%)		
	MAE	MSE	PSNR	SSIM	fg	bg	mIoU
MCNN	110.0	221.5	23.81	0.842	75.5	40.1	57.8
CSRNet	61.1	134.9	29.03	0.826	94.5	73.8	84.1
FCN	61.5	156.6	28.92	0.874	93.5	70.7	82.1
SFCN	**56.0**	129.7	29.17	0.889	93.9	74.3	84.1
SFCN†	57.3	**127.3**	**30.01**	**0.895**	**94.9**	**76.6**	**85.7**

Performance of cross-location splitting

Method	Counting				Segmentation(%)		
	MAE	MSE	PSNR	SSIM	fg	bg	mIoU
MCNN	154.8	340.7	24.05	0.857	76.3	37.4	56.9
CSRNet	92.2	220.1	28.75	0.842	94.4	73.3	83.9
FCN	97.5	226.8	29.33	0.866	93.4	69.9	81.7
SFCN	89.3	216.8	29.50	0.906	94.9	73.8	84.3
SFCN†	**83.9**	**209.7**	**29.76**	**0.914**	**95.1**	**76.0**	**85.5**

From the table, we find SFCN† attains the best performance in both crowd counting and segmentation tasks, which is due to the more powerful learning ability of ResNet-101 than VGG-16 Net. For a fair comparison, we select CSRNet, FCN and SFCN that use the same backbone (VGG-16) to show the effectiveness of the proposed SFCN. We find SFCN is better than CSRNet and FCN in terms of seven metrics on counting and segmentation.

In addition, from the counting performance of the three aspects (random, cross-camera and cross-location splitting), the performances are decreased significantly, which means the difficulty of three tasks is rising in turn. The main reason is that there is a big difference in the distribution of people in different crowd scenes. In contrast, the segmentation results of different models in the three evaluations are very similar, which implies that crowd region segmentation is not sensitive to different evaluation strategies. The essential reasons are: 1) GCC's person model is fixed though the crowd scenes are different; 2) the segmentation focuses on appearance feature.

Multi-task Learning for Counting and Segmentation

Counting and segmentation are two complementary tasks: the former focuses on the local density, and the latter aims at the difference between foreground and background. On the one hand, introducing segmentation can effectively reduce the error density estimation in background regions. On the other hand, density maps provide rich information to represent different-density crowd regions, which aids the segmentation branch in tackling them via different priors. Here, we also conduct the experiments of multi-task learning using SFCN on GCC. During the training stage, the loss weights for counting and segmentation are 1, 0.01, respectively. Table 4 shows the counting (MAE/MSE) and segmentation (mIoU) performance of single-task learning (STL) and multi-task learning (MTL). From the table, we find that MTL outperforms the STL in terms of counting and segmentation performance.

Table 4 The results of STL and MTL on GCC dataset.

Data	Single task	Multi task
	MAE/MSE/mIoU	MAE/MSE/mIoU
rd	36.2/81.1/84.5	33.9/80.6/85.7
cc	56.0/129.7/84.1	52.6/125.4/84.7
cl	89.3/216.8/84.3	85.7/209.9/85.5

6.2.3 Comparison of Different Pre-trained Models

In Section 4.2, we propose a pre-training scheme to provide a model with better-initialized parameters, which can significantly improve the performance on small-scale counting datasets. To verify our strategy, we conduct the MCNN, CSRNet, SFCN and SFCN† on the two datasets (UCF-QNRF and SHT B) and compare different pre-training data. Notably, there are five strategies:

FS: train the model From Scratch (light-wight models use it, such as MCNN);

Pre-ImgNt: Pre-train the model on ImageNet and fine-tune on a specific dataset (mainstream VGG-backbone or ResNet-backbone models use it, such as CSRNet);

Pre-GCC: Pre-train the model on GCC dataset and fine-tune on a specific dataset;

Pre-UR: Pre-train the model on the Union of seven Real-world datasets (UCSD, Mall, UCF_CC_50, WorldExpo'10, SHT A, SHT B and UCF-QNRF), and fine-tune on a specific dataset;

Pre-GU: Pre-train the model on GCC and UR (Union of Real datasets, same as the seven aforementioned datasets), and fine-tune on a specific dataset.

Note that all pre-trained data are from the training set. Considering the disparity in the data volume of each dataset, each subset is sampled with the same probability during the training stage of Pre-UR and Pre-GU. In addition, small images (such as UCSD) will be resized to at least 480px. Other settings are the same as Section 6.2.1. Table 5 shows the fine-tuning SFCN's results on the two real-world datasets by using three different pre-trained models. The bold blue fonts

Table 5 The fine-tuning SFCN's results (MAE/MSE) on the two real-world datasets by using three different pre-trained models: Pre-GCC, Pre-UR and Pre-GU. The **blue bold** texts denote the baseline results. The relative reduction is computed based on the corresponding baseline.

Method	UCF-QNRF				SHT B				Avg. Reduction
	MCNN	CSRNet	SFCN	SFCN†	MCNN	CSRNet	SFCN	SFCN†	
FS	**281.2/445.0**	-	-	-	**26.3/39.5**	-	-	-	-
Pre-ImgNt	-	**120.3/208.5**	**134.3/240.3**	**114.8/192.0**	-	**10.6/16.0**	**11.0/17.1**	**8.9/14.3**	-
Pre-GCC	199.8/311.2 (↓ 29/30%)	112.4/185.6 (↓ 7/11%)	124.7/203.5 (↓ 7/15%)	102.0/171.4 (↓ 11/11%)	18.8/28.2 (↓ 29/29%)	10.1/15.7 (↓ 5/2%)	9.4/14.4 (↓ 15/16%)	7.6/13.0 (↓ 15/9%)	↓ 15%
Pre-UR	194.5/304.9 (↓ 31/31%)	115.7/180.4 (↓ 4/13%)	110.4/187.1 (↓ 18/22%)	107.8/186.2 (↓ 6/3%)	19.5/30.2 (↓ 26/24%)	10.2/16.0 (↓ 4/0%)	9.3/14.6 (↓ 15/15%)	8.3/13.6 (↓ 7/5%)	↓ 14%
Pre-GU	177.5/271.5 (↓ 37/39%)	104.2/171.9 (↓ 13/18%)	106.6/171.9 (↓ 21/28%)	**98.6/170.2** (↓ 14/11%)	17.4/27.5 (↓ 34/30%)	9.9/15.3 (↓ 7/4%)	9.1/14.5 (↓ 15/17%)	**7.2/12.4** (↓ 19/13%)	↓ 20%

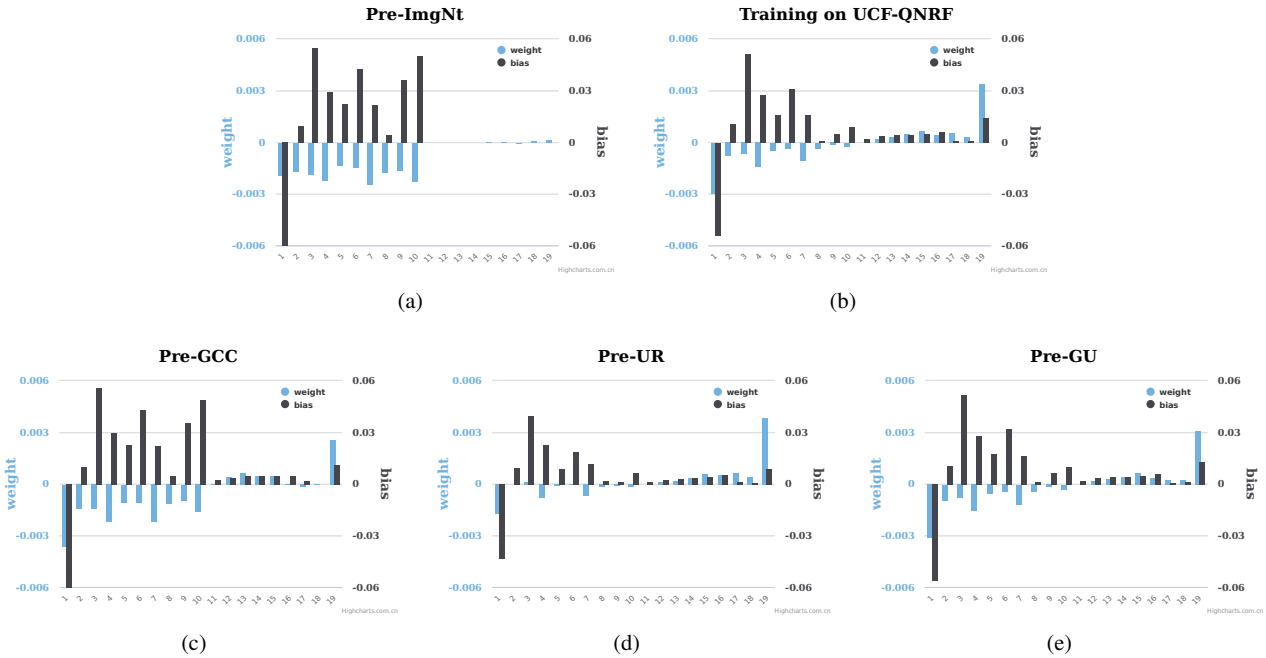


Fig. 8 The mean value of weight and bias in each layer of SFCN models with different training strategies. (a) Pre-ImgNt: pre-trained model on ImgNt, (b) traditional supervised training on UCF-QNRF, (c,d,e) pre-trained model on GCC, UR and GU.

represent the baseline results, and the red percentages indicate the relative reduction compared with the corresponding baseline. From the table, there are two interesting findings:

1) Using the extra pre-trained counting data can effectively prompt the performance. The proposed Pre-GCC, Pre-UR and Pre-GU reduce the average error by 15%, 14% and 20%, respectively. We also find that Pre-GU's results are better than Pre-GCC and Pre-UR.
2) The average performance improvement is more significant based on the model trained from Scratch (MCNN) than the model pre-trained on ImageNet (CSRNet, SFCN and SFCN†): $\sim 31\%$ *v.s.* $\sim 12\%$.

Comparison on the Parameter Level

In addition to the comparison of the final estimation results, we further explore the differences in the pre-training as mentioned above models at the parameter level. Take SFCN as an example, we compute the average distribution of weights and bias for each layer for four pre-trained models (Pre-ImgNt, Pre-GCC, Pre-UR, Pre-GU) and a fine-tuning model based on Pre-GU. Fig. 8 illustrates the mean value of weight and bias in each layer of SFCN models with different training strategies. For Fig. 8(a), the first ten layers are VGG-16 backbone, and the others are randomly initialized. By comparing Fig. 8(a) and (b), the distribution difference between the two is very obvious. However, other pre-trained models' distributions on counting datasets (namely Pre-GCC, UR and GU) are very close to Fig. 8(b). The similarity of the last four models shows that crowd counting models have a certain distribution. Introducing pre-training scheme on counting data provides better installation parameters than training from scratch or pre-training from other tasks. Besides, we also find Pre-UR and Pre-GU are more similar to

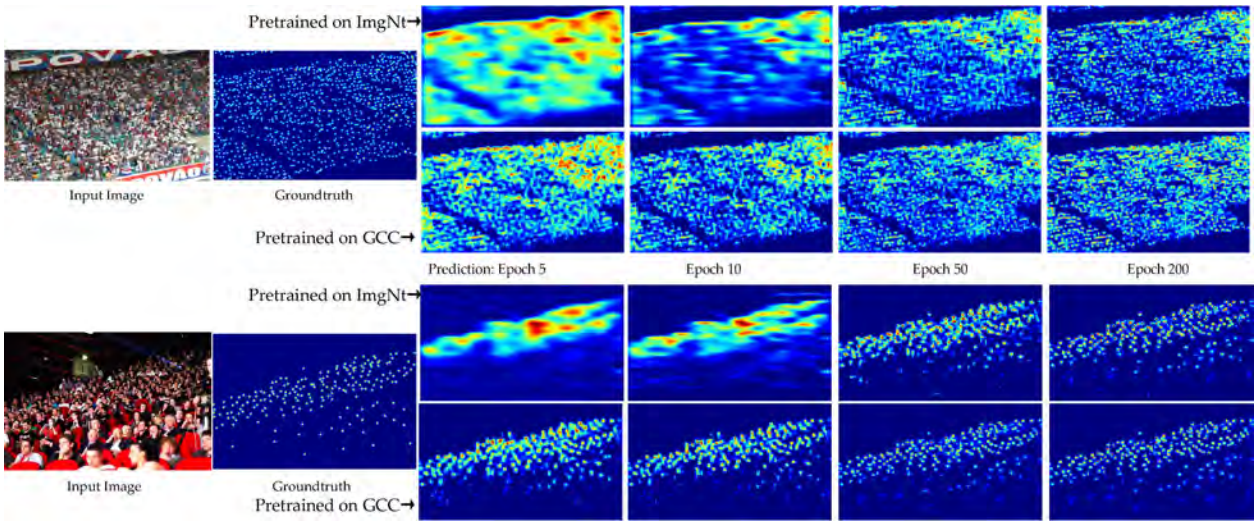


Fig. 9 Visual comparison of different pre-trained models on UCF-QNRF.

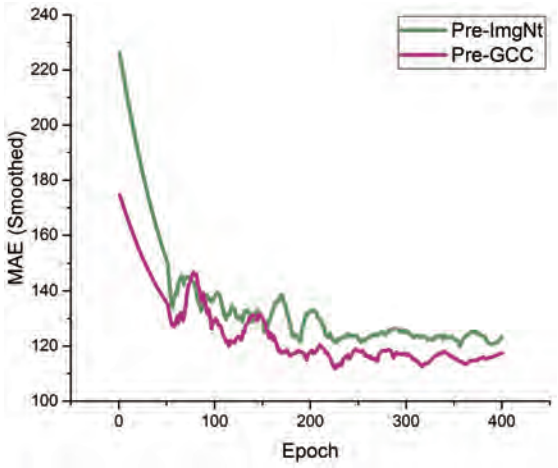


Fig. 10 The MAE curve of two different training schemes on UCF-QNRF test set.

Fig. 8(b) than Pre-GCC. The main reason is that the first two pre-training methods use the labeled UCF-QNRF data.

Visual Comparison of Different Pre-trained Models

The last section shows the improvement of performance after using Pre-GCC. In order to show this effectiveness, we report the MAE curve and different visual density maps of two pre-trained strategies (namely pre-trained on ImgNt and GCC) during the entire training process. Fig. 10 depicts the variation curve of the loss during the training process. At the beginning of training, the model using Pre-GCC can converge rapidly than the model using Pre-ImgNt, which means that the initialized parameter provided by the former is better the latter. Besides, we find the purple line is lower than the green curve. In other words, exploiting the Pre-GCC strategy can achieve better training results. For an intuitive comparison, we record the visualization results of density map estimation during different training phases, which are shown in Fig. 9. To be specific, we select two typical test images and show their density map at Epoch 5, 10, 50 and 200 in the entire training stage. In the early stages of training (before Epoch 10), Pre-GCC can easily get an acceptable result, while the Pre-ImgNt can only output a coarse density distribution. This phenomenon confirms the convergence curve in Fig.10. As the training continues, Pre-ImgNt can also output a fine density map after Epoch 200, though the result is worse than Pre-GCC.

Generalization Ability of Different Pre-trained Models

Here, we further compare the cross-dataset generalization ability of counters using different pre-trained models. To be specific, we apply the two SFCN† models trained on UCF-QNRF dataset using different training schemes to the two other real-world counting dataset, namely SHT A and B. Table 6 shows the experimental results. From it, we find Pre-GCC can significantly prompt the model's generalization ability. Specifically, compared with Pre-ImgNt, Pre-GCC can reduce the MAE by 6.1% (108.0 $\rightarrow$ 101.4) and 17.4% (17.2 $\rightarrow$ 14.2) on SHT A and B, respectively. The better generalization ability means that Pre-GCC makes the model perform better in unseen real data than the traditional Pre-ImgNt.

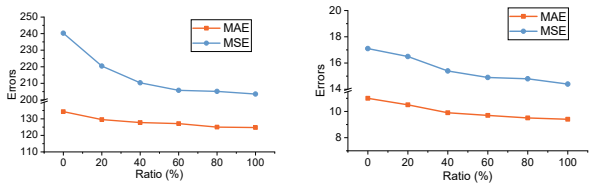
Table 6 The performance (MAE/MSE) of different UCF-QNRF counting models (SFCN†) under the different pre-trained models on SHT A and B dataset.

Strategy	SHT A	SHT B
Pre-ImgNt	108.0/184.1	17.2/24.9
Pre-GCC	**101.4/179.2** ($\downarrow$ 6.1/2.7%)	**14.2/21.4** ($\downarrow$ 17.4/14.1%)

6.2.4 The Effect of Pre-GCC on the Fine-tuning Results

In this section, we analyze how different GCC data affect the final fine-tuning result on the real-world datasets (UCF-QNRF and SHT B), such as different data volumes, different combinations of scene levels, and different-luminance crowd scenes.

For the first factor, more images mean that the pre-trained data is more diverse. Thus, we implement Pre-GCC SFCN models using 20%, 40%, 60%, 80% and 100% GCC data, respectively. Fig. 11 demonstrates the fine-tuning SFCN's estimation errors on UCF-QNRF and SHT B datasets by using different pre-trained GCC data volumes. From the figure, as the pre-training data gradually increases, the estimation errors (MAE and MSE) become smaller. It means that more diverse data can better help fine-tune the model on real-world data.



(a) UCF-QNRF. (b) SHT B.

Fig. 11 The fine-tuning SFCN's results on UCF-QNRF and SHT B datasets by pre-training different pre-trained GCC data volumes. "0%" means the model do not use any GCC data to pre-train, namely Pre-ImgNt.

For the second factor, different combinations of scene levels indicate different density distributions in the pre-trained data. In GCC, all scenes are divided into nine categories, denoting $L0$, $L1$, ..., and $L8$. Here, we merge them into four categories: $\{L0, L1, L2\}$, $\{L3, L4\}$, $\{L5, L6\}$ and $\{L7, L8\}$. To eliminate the impact of data volume, we sample $1,800$ images from the 4 classes as the pre-trained data. Table 7 reports the performance (MAE/MSE) of the fine-tuning SFCN's results on UCF-QNRF and SHT B datasets. Since UCF-QNRF is an extremely congested dataset, the results of pre-training on $\{L5, L6\}$ and $\{L7, L8\}$ are better than that of pre-training on sparse crowd scenes (namely $\{L0, L1, L2\}$ and $\{L3, L4\}$. Similarly, SHT B's density range is in $[9, 578]$ (reported in Table 1), so when pre-training on $\{L5, L6\}$ (the density range is in $[0, 600]$ and $[0, 1000]$), the errors are the lowest (MAE/MSE of 9.1/14.8). According to the above results, we find that when the density distribution of pre-trained data is closer to that of real data, the fine-tuning performance will be better.

Finally, we explore the effect of GCC data's luminance changes on the fine-tuning results. To be specific, GCC is roughly divided into high-luminance and low-luminance data according to the time of shooting (high-luminance range is 6:00 $\sim$ 17:59 and the others are low-luminance data). To eliminate the impact of data volume, we sample $2,500$ images from the two types of data classes as the pre-trained data. Table 8 shows the fine-tuning SFCN's results (MAE/MSE) on UCF-QNRF and SHT B datasets. The final estimation errors using low-luminance data is larger than that of using high-luminance data. The main reason is that the real-world datasets (UCF-QNRF and SHT B) rarely contain low-luminance data. In other words, the high-luminance GCC data is closer to the real data.

Table 7 The fine-tuning SFCN's results (MAE/MSE) on UCF-QNRF and SHT B datasets by pre-training GCC data with different scene levels.

Pre-trained data	QNRF	SHT B
$\{L0, L1, L2\}$	131.4/223.5	10.8/17.4
$\{L3, L4\}$	121.6/201.9	9.3/15.0
$\{L5, L6\}$	117.0/**196.1**	**9.1/14.8**
$\{L7, L8\}$	**115.0**/207.6	9.8/17.1

Table 8 The fine-tuning SFCN's results (MAE/MSE) on UCF-QNRF and SHT B datasets by pre-training high-/low- luminance GCC data.

Pre-trained data	QNRF	SHT B
Low luminance	126.7/231.9	10.6/16.1
High luminance	**117.2/199.4**	**9.2/14.3**

In summary, to further prompt the performance of Pre-GCC, we may need as much pre-training data as possible that is more similar to the real data. It will be an interesting question about how to select the proper pre-training data.

6.2.5 Comparison with the SOTA Methods

For comparison with other State-of-the-art methods, we conduct the experiments of SFCN† with the Pre-GCC strategy on five mainstream crowd counting datasets, namely UCF-QNRF, SHT A, SHT B, UCF_CC_50 and WorldExpo'10. Table 9 reports the results of them. Our proposed method refreshes the six records in all nine metrics of the five datasets. To be specific, we achieve the best MAE performance on UCF-QNRF (**102.0**), SHT A (**64.8**), SHT B (**7.6**), UCF_CC_50 (**214.2**).

6.3 Results of Domain-adaptation Crowd Understanding

In this section, we conduct the adaptation experiments and further analyze the effectiveness of the proposed CycleGAN-based methods.

Table 9 The comparison with the state-of-the-art performance on the five real datasets.

Method	UCF-QNRF		SHT A		SHT B		UCF_CC_50		WorldExpo'10 (MAE)					
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	S1	S2	S3	S4	S5	Avg.
MCNN (Zhang et al., 2016b)	277	426	110.2	173.2	26.4	41.3	377.6	509.1	3.4	20.6	12.9	13.0	8.1	11.6
Switching-CNN (Sam et al., 2017)	-	-	90.4	135.0	21.6	33.6	318.1	439.2	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN (Sindagi and Patel, 2017b)	-	-	73.6	106.4	20.1	30.1	298.8	320.9	2.9	14.7	10.5	10.4	5.8	8.86
ACSCP (Shen et al., 2018)	-	-	75.7	102.7	17.2	27.4	291.0	404.6	2.8	14.05	9.6	**8.1**	**2.9**	**7.5**
CSRNet (Li et al., 2018b)	-	-	68.2	115.0	10.6	16.0	266.1	397.5	2.9	**11.5**	**8.6**	16.6	3.4	8.6
DRSAN (Liu et al., 2018b)	-	-	69.3	**96.4**	11.1	18.2	219.2	**250.2**	2.6	11.8	10.3	10.4	3.7	7.76
SANet (Cao et al., 2018)	-	-	67.0	104.5	17.0	8.4	258.4	334.9	2.6	13.2	9.0	13.3	3.0	8.2
CL (Idrees et al., 2018)	132	191	-	-	-	-	-	-	-	-	-	-	-	-
ic-CNN (Ranjan et al., 2018)	-	-	68.5	116.2	10.7	16.0	260.9	365.5	17.0	12.3	9.2	**8.1**	4.7	10.3
SFCN† with Pre-GCC	**102.0**	**171.4**	**64.8**	107.5	**7.6**	**13.0**	**214.2**	318.2	**1.8**	17.5	11.1	13.5	3.0	9.4

6.3.1 Implementation Details

Like Section 6.2.1, we randomly select 10% training data of the real domain as the validation set to find the best model. During the training phase, α, β and λ in Eq. 4 are set as 1, 0.1 and 0.01, respectively. SE CycleGAN's and SFCN's training parameter is same as the original CycleGAN and Section 6.2.1, respectively. D_{dc}'s learning rate is set as 10^{-4}.

In Section 5.2, we introduce Scene Regularization (SR) to select the proper images to avoid negative adaptation. Here, Table 10 shows the concrete filter condition for adaptation to the five real datasets. Specifically, ratio range means that the numbers of people in selected images should be in a specific range. For example, during adaptation to SHT A, there is a candidate image with level 0∼4000, containing 800 people. According to the ratio range of 0.5∼1, since 800 is not in 2000∼4000 (namely 0.5*4000 ∼1*4000), the image can not be selected. In other words, the ratio range is a restriction in terms of congestion.

Table 10 Filter condition on eight real datasets.

Target Dataset	level	time	weather	count range	ratio range
SHT A	4,5,6,7,8	6:00∼19:59	0,1,3,5,6	25∼4000	0.5∼1
SHT B	1,2,3,4,5	6:00∼19:59	0,1,5,6	10∼600	0.3∼1
UCF_CC_50	5,6,7,8	8:00∼17:59	0,1,5,6	400∼4000	0.6∼1
UCF-QNRF	4,5,6,7,8	5:00∼20:59	0,1,5,6	400∼4000	0.6∼1
WorldExpo'10	2,3,4,5,6	6:00∼18:59	0,1,5,6	0∼1000	0∼1

Other explanations of Arabic numerals in the table is listed as follows:

[Level Categories] 0: 0∼10, 1: 0∼25, 2: 0∼50, 3: 0∼100, 4: 0∼300, 5: 0∼600, 6: 0∼1k, 7: 0∼2k and 8: 0∼4k.

[Weather Categories] 0: clear, 1: clouds, 2: rain, 3: foggy, 4: thunder, 5: overcast and 6: extra sunny.

6.3.2 Adaptation Performance on Real-world Datasets

In this section, we conduct the adaptation experiments from GCC dataset to five mainstream real-world counting datasets: ShanghaiTech A/B (Zhang et al., 2016b), UCF_CC_50 (Idrees et al., 2013), UCF-QNRF (Idrees et al., 2018), WorldExpo'10 (Zhang et al., 2016a) and a real-word crowd segmentation dataset, CityScapes (Cordts et al., 2016). For the best performance, all models adopt the Scene/Density Regularization mentioned in Section 5.3. Notably, each model is explained as follows:

NoAdpt: Train SFCN on the original GCC and evaluate on the real dataset.

CycleGAN: Translate GCC images to photo-realistic data using CycleGAN, and then train SFCN on them.

SE CycleGAN: Translate GCC images to photo-realistic data using SE CycleGAN, and then train SFCN on them. It is the method of the conference version (Wang et al., 2019).

SE CycleGAN (Joint Training, JT): Jointly train SE CycleGAN (introducing feature-level adversarial learning) model and SFCN.

Crowd Counting via Domain Adaptation

Table 11 shows the four metrics (only MAE on WorldExpo'10) of the No Adaptation (NoAdpt), CycleGAN, the proposed SE CycleGAN and SE CycleGAN (Joint Training, JT). From it, we find the results after adaptation are far better than that of no adaptation, which indicates the adaptation can effectively reduce the domain gaps between synthetic and real-world data. After embedding SSIM loss in CycleGAN, almost all performances are improved on five datasets. There are only two reductions of PSNR on Shanghai Tech A and UCF_CC_50. In general, the proposed SE CycleGAN outperforms the original CycleGAN. When utilizing jointing training, the performance is increased in most metrics, which means that adversarial learning and joint training can further reduce the domain gap between translated synthetic images and real-world data.

In addition, we find the counting results on SHT B achieves a good level (MAE/MSE of 16.4/25.8), even outperform some early supervised methods (Zhang et al., 2016b; Sindagi and Patel, 2017a; Sam et al., 2017; Sindagi and Patel, 2017b; Liu et al., 2018a). The main reasons are: 1) the real data is strongly consistent, which is captured by the same sensors; 2) the data has high image clarity. The two characteristics

Table 11 The counting performance of no adaptation (No Adpt), CycleGAN (Zhu et al., 2017), SE CycleGAN and SE CycleGAN (Joint Training, JT) on the five real-world datasets.

Method	DA	SHT A				SHT B				UCF_CC_50			
		MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM	MAE	MSE	PSNR	SSIM
NoAdpt	✗	160.0	216.5	19.01	0.359	22.8	30.6	24.66	0.715	487.2	689.0	17.27	0.386
CycleGAN	✔	143.3	204.3	**19.27**	0.379	25.4	39.7	24.60	0.763	404.6	548.2	**17.34**	0.468
SE CycleGAN	✔	123.4	193.4	18.61	0.407	19.9	28.3	24.78	0.765	373.4	528.8	17.01	**0.743**
SE CycleGAN (JT)	✔	**119.6**	**189.1**	18.69	**0.429**	**16.4**	**25.8**	**26.17**	**0.786**	**370.2**	**512.0**	17.11	0.689

Method	DA	UCF-QNRF				WorldExpo'10 (MAE)					
		MAE	MSE	PSNR	SSIM	S1	S2	S3	S4	S5	Avg.
NoAdpt	✗	275.5	458.5	20.12	0.554	4.4	87.2	59.1	51.8	11.7	42.8
CycleGAN	✔	257.3	400.6	20.80	0.480	4.4	69.6	49.9	29.2	9.0	32.4
SE CycleGAN	✔	230.4	**384.5**	21.03	**0.660**	4.3	59.1	43.7	**17.0**	7.6	26.3
SE CycleGAN (JT)	✔	**225.9**	385.7	**21.10**	0.642	**4.2**	**49.6**	**41.3**	19.8	**7.2**	**24.4**

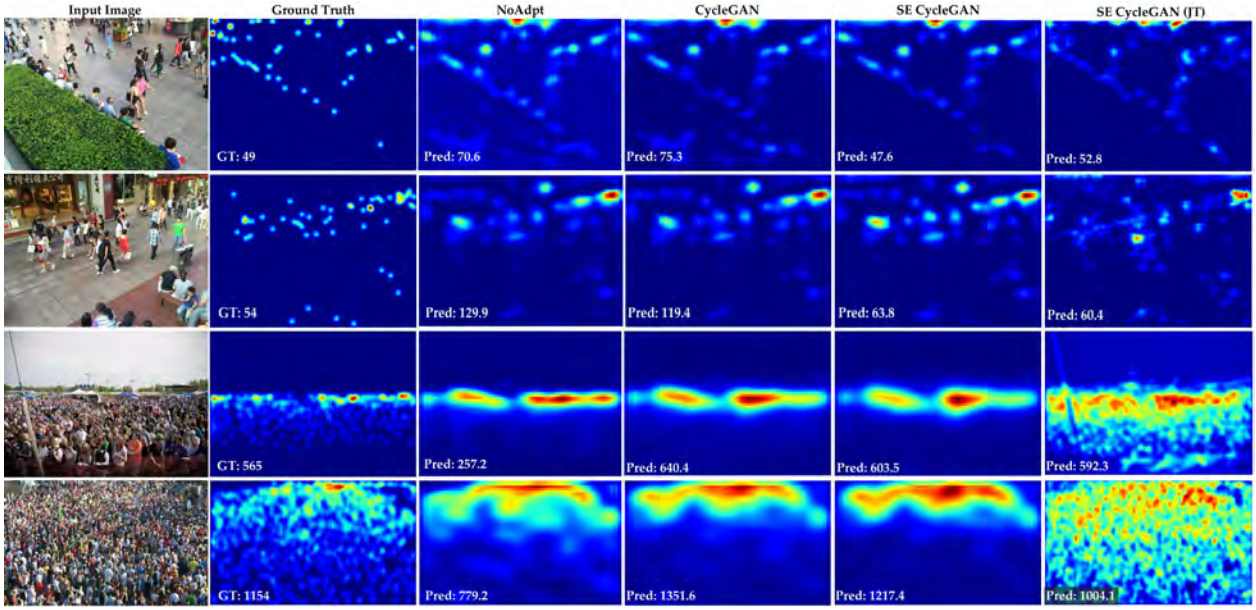


Fig. 12 The demonstration of different methods on SHT dataset. "GT" and "Pred" represent the labeled and predicted count, respectively.

guarantee that the SE CycleGAN's adaptation on SHT B is more effective than others.

Fig. 12 demonstrates four groups of visualized results on SHT A and B dataset. Compared with NoAdpt, the map quality via CycleGAN has a significant improvement. Row 1 and 2 demonstrate the Part B visualization results. We find the predicted maps are very close to the groundtruth. When jointly training SE CycleGAN and SFCN, we find the misestimation in the background region can be effectively alleviated compared with the original SE CycleGAN. However, for the extremely congested scenes (in Row 3 and 4), the predicted maps are very far from the ground truth. We think the main reason is that the translated images lose the details (such as texture, sharpness and edge) in high-density regions.

Crowd Segmentation via Domain Adaptation

Considering that there is no real-world dataset for crowd segmentation, we transform a semantic segmentation dataset (CityScapes, Cordts et al. (2016)) to a crowd segmentation dataset. To be specific, the "pedestrian" class to generate the crowd mask and the other objects are treated as "background". Same as the counting task, we conduct two groups of experiments: NoAdpt and SE CycleGAN (Joint training, JT). The results are reported in Table 12. From the table, after introducing domain adaptation, mIoU is increased by 10.1% compared with NoAdapt, which evidences that the proposed framework is also suitable for the domain-adaptation-style crowd segmentation task.

Table 12 The segmentation performance (%) of no adaptation (NoAdpt) and SE CycleGAN (Joint Training, JT) on CityScapes dataset.

Method	fg	bg	mIoU
NoAdpt	98.9	20.7	59.8
SE CycleGAN (JT)	96.3	35.4	65.9

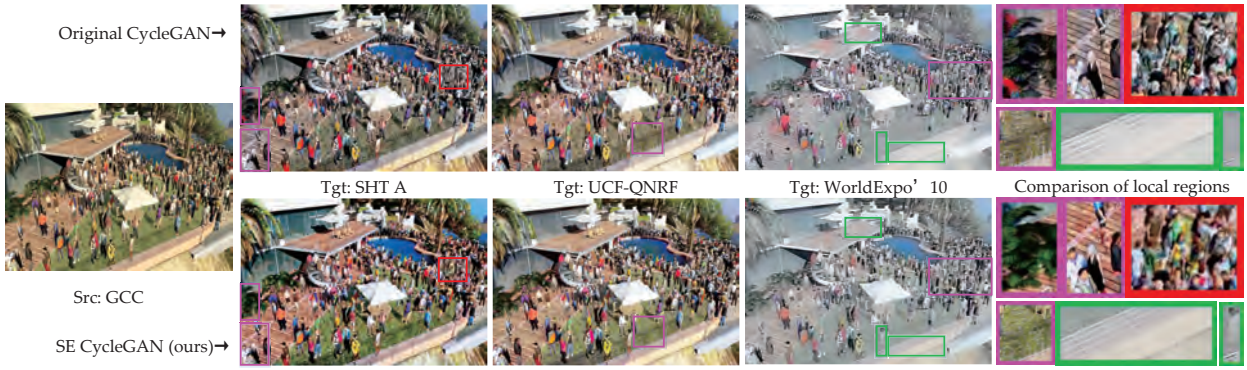


Fig. 13 The visualization comparisons of CycleGAN and SE CycleGAN.

6.3.3 The Effectiveness of SDR

Here, we compare the performance of three models (No Adpt, CycleGAN and SE CycleGAN) without Scene/Density Regularization (SDR) and with SDR. Table 13 reports the performance of with or without SDR on SHT A dataset. From the results in the first column, we find these two adaptation methods cause some side effects. In fact, they do not produce ideal translated images. When introducing SDR, the nonexistent synthetic scenes in the real datasets are filtered out, which improves the domain adaptation performance.

Table 13 The results of NoAdpt, CycleGAN (Zhu et al., 2017) and SE CycleGAN on SHT A.

Method	w/o SDR	with SDR
NoAdpt	**163.6**/244.5	160.0/216.5
CycleGAN	180.1/290.3	143.3/204.3
SE CycleGAN	169.8/**230.2**	**123.4**/**193.4**

6.3.4 Analysis of SSIM Embedding

SSIM Embedding can guarantee the original synthetic and reconstructed images have high structural similarity (SS), which prompts two generators' translation for images to maintain a certain degree of SS during the training process. Fig. 13, demonstrate the translated images from GCC to the three real-world datasets. "Src" and "Tgt" represent the source domain (synthetic data) and the target domain (real-world data). The top row shows the results of the original CycleGAN and the bottom is the results of the proposed SE CycleGAN.

We compare some obvious differences between CycleGAN and SE CycleGAN (ours) and mark them up with rectangular boxes. To be specific, ours can produce a more consistent image than the original CycleGAN in the green boxes. As for the red boxes, CycleGAN loses more texture features than ours. For the purple boxes, we find that CycleGAN produces some abnormal color values, but SE CycleGAN performs better than it. For the regions covered by blue boxes, SE CycleGAN maintains the contrast of the original image than CycleGAN in a even better fashion.

In general, from the visualization results, the proposed SE CycleGAN generates more high-quality crowd scenes than the original CycleGAN. The complete translation results are available at this website [11].

7 Conclusion and Outlook

In this paper, we focus on promoting the performance of crowd understanding in the wild via utilizing the synthetic data. Exploiting the generated data, we then propose two effective ways (pre-training scheme and domain adaptation) to improve the counting performance in the real world significantly. The proposed pre-training scheme provides a better installation parameter than the traditional strategy, namely pre-training on ImageNet. Experiments show that the counting performance is improved by an average of 12%. The presented domain adaptation provides a new direction for crowd understanding, which liberates humans from the tedious labeling work. By the joint training for adaptation and source-domain crowd understanding, the trained crowd model works better than traditional no adaptation method in the real-world data. To be specific, in some typical subservience scenes (ShanghaiTech Part B, WorldExpo'10, UCSD and Mall), the estimations of domain adaptation are very close to that of the traditional supervised learning.

According to the results of experiments from this paper, we think there are some interesting directions in the crowd understanding:

1) **Data generation** Based on the open-sourced tools, the researchers are allowed to re-develop customized software for generating synthetic image or video data including but not limited to the following tasks: object

[11] http://share.crowdbenchmark.com:2443/home/Translation_Results

counting/localization/tracking (crowd, vehicle, *etc.*), crowd instance segmentation, crowd flow analysis, group detection, person re-identification, anomaly event detection, and human trajectory prediction.

2) **Domain-adaptive crowd understanding** The traditional supervised learning requires a large amount of labeled data, which hinders the landing of the crowd model in the real world. Considering this problem, we think domain-adaptive crowd counting is a more practical research area than supervised learning: unsupervised/few-shot domain adaptation will reduce the cost of collecting and annotating real scene data.

In future work, we will focus on the two types as mentioned above of tasks and attempt to promote the practical application of crowd understanding in the real world.

References

Abu-El-Haija S, Kothari N, Lee J, Natsev P, Toderici G, Varadarajan B, Vijayanarasimhan S (2016) Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:160908675

Babu Sam D, Sajjan NN, Venkatesh Babu R, Srinivasan M (2018) Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3618–3626

Bak S, Carr P, Lalonde JF (2018) Domain adaptation through synthesis for unsupervised person re-identification. arXiv preprint arXiv:180410094

Cao X, Wang Z, Zhao Y, Su F (2018) Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European Conference on Computer Vision, pp 734–750

Chan AB, Vasconcelos N (2009) Bayesian poisson regression for crowd counting. In: 2009 IEEE 12th international conference on computer vision, IEEE, pp 545–551

Chan AB, Liang ZSJ, Vasconcelos N (2008) Privacy preserving crowd monitoring: Counting people without people models or tracking. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 1–7

Chan AB, Morrow M, Vasconcelos N, et al. (2009) Analysis of crowded scenes using holistic properties. In: Performance Evaluation of Tracking and Surveillance workshop at CVPR, pp 101–108

Chen K, Loy CC, Gong S, Xiang T (2012) Feature mining for localised crowd counting. In: Proceedings of the British Machine Vision Conference, vol 1, p 3

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3213–3223

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 248–255

Dong L, Parameswaran V, Ramesh V, Zoghlami I (2007) Fast crowd segmentation using shape indexing. In: 2007 IEEE 11th International Conference on Computer Vision, IEEE, pp 1–8

Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V (2017) CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning, pp 1–16

Everingham M, Eslami SA, Van Gool L, Williams CK, Winn J, Zisserman A (2015) The pascal visual object classes challenge: A retrospective. International journal of computer vision 111(1):98–136

Fu M, Xu P, Li X, Liu Q, Ye M, Zhu C (2015) Fast crowd density estimation with convolutional neural networks. Engineering Applications of Artificial Intelligence 43:81–88

Gao J, Lin W, Zhao B, Wang D, Gao C, Wen J (2019) C^3 framework: An open-source pytorch code for crowd counting. arXiv preprint arXiv:190702724

Gao J, Wang Q, Li X (2019) Pcc net: Perspective crowd counting via spatial convolutional network. IEEE Transactions on Circuits and Systems for Video Technology pp 1–1, DOI 10.1109/TCSVT.2019.2919139

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Proceedings of the Advances in Neural Information Processing Systems, pp 2672–2680

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 770–778

Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 4700–4708

Idrees H, Saleemi I, Seibert C, Shah M (2013) Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 2547–2554

Idrees H, Tayyab M, Athrey K, Zhang D, Al-Maadeed S, Rajpoot N, Shah M (2018) Composition loss for counting, density map estimation and localization in dense crowds. arXiv preprint arXiv:180801050

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multime-

dia, ACM, pp 675–678

Jiang X, Xiao Z, Zhang B, Zhen X, Cao X, Doermann D, Shao L (2019) Crowd counting and density estimation by trellis encoder-decoder networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 6133–6142

Johnson-Roberson M, Barto C, Mehta R, Sridhar SN, Rosaen K, Vasudevan R (2017) Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In: Proceedings of the IEEE International Conference on Robotics and Automation, pp 1–8

Kang K, Wang X (2014) Fully convolutional neural networks for crowd segmentation. arXiv preprint arXiv:14114464

Kempka M, Wydmuch M, Runc G, Toczek J, Jaśkowski W (2016) Vizdoom: A doom-based ai research platform for visual reinforcement learning. In: 2016 IEEE Conference on Computational Intelligence and Games (CIG), pp 1–8

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980

Li C, Lin L, Zuo W, Tang J, Yang MH (2018a) Visual tracking via dynamic graph learning. IEEE transactions on pattern analysis and machine intelligence 41(11):2770–2782

Li T, Chang H, Wang M, Ni B, Hong R, Yan S (2014) Crowded scene analysis: A survey. IEEE transactions on circuits and systems for video technology 25(3):367–386

Li W, Mahadevan V, Vasconcelos N (2013) Anomaly detection and localization in crowded scenes. IEEE transactions on pattern analysis and machine intelligence 36(1):18–32

Li X, Chen M, Nie F, Wang Q (2017) A multiview-based parameter free framework for group detection. In: Thirty-First AAAI Conference on Artificial Intelligence

Li Y, Zhang X, Chen D (2018b) Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1091–1100

Lian D, Li J, Zheng J, Luo W, Gao S (2019) Density map regression guided detection network for rgb-d crowd counting and localization. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1821–1830

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: European conference on computer vision, Springer, pp 740–755

Liu J, Gao C, Meng D, Hauptmann AG (2018a) Decidenet: Counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5197–5206

Liu L, Wang H, Li G, Ouyang W, Lin L (2018b) Crowd counting using deep recurrent spatial-aware network. arXiv preprint arXiv:180700601

Liu W, Salzmann M, Fua P (2019) Context-aware crowd counting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 5099–5108

Liu X, van de Weijer J, Bagdanov AD (2018c) Leveraging unlabeled data for crowd counting by learning to rank. arXiv preprint arXiv:180303095

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3431–3440

Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, pp 1975–1981

Marsden M, McGuinness K, Little S, O'Connor NE (2016) Fully convolutional crowd counting on highly congested scenes. arXiv preprint arXiv:161200220

Mehran R, Oyama A, Shah M (2009) Abnormal crowd behavior detection using social force model. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp 935–942

Onororubio D, Lopezsastre RJ (2016) Towards perspective-free object counting with deep learning pp 615–629

Pan X, Shi J, Luo P, Wang X, Tang X (2017) Spatial as deep: Spatial cnn for traffic scene understanding. arXiv preprint arXiv:171206080

Paszke A, Gross S, Chintala S, Chanan G (2017) Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration

Popoola OP, Wang K (2012) Video-based abnormal human behavior recognition—a review. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42(6):865–878

Qiu W, Zhong F, Zhang Y, Qiao S, Xiao Z, Kim TS, Wang Y, Yuille A (2017) Unrealcv: Virtual worlds for computer vision. ACM Multimedia Open Source Software Competition

Ranjan V, Le H, Hoai M (2018) Iterative crowd counting. arXiv preprint arXiv:180709959

Richter SR, Vineet V, Roth S, Koltun V (2016) Playing for data: Ground truth from computer games. In: Proceedings of the European Conference on Computer Vision, pp 102–118

Richter SR, Hayder Z, Koltun V (2017) Playing for benchmarks. In: Proceedings of the International conference on computer vision, vol 2

Ros G, Sellart L, Materzynska J, Vazquez D, Lopez AM (2016) 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, pp 3234–3243

Sam DB, Surya S, Babu RV (2017) Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol 1, p 6

Sam DB, Sajjan NN, Babu RV (2018) Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. arXiv: Computer Vision and Pattern Recognition

Sam DB, Sajjan NN, Maurya H, Babu RV (2019) Almost unsupervised learning for dense crowd counting. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, vol 27

Shah S, Dey D, Lovett C, Kapoor A (2018) Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics, Springer, pp 621–635

Shen Z, Xu Y, Ni B, Wang M, Hu J, Yang X (2018) Crowd counting via adversarial cross-scale consistency pursuit. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5245–5254

Shi Z, Zhang L, Liu Y, Cao X, Ye Y, Cheng MM, Zheng G (2018) Crowd counting with deep negative correlation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5382–5390

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Sindagi VA, Patel VM (2017a) Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, pp 1–6

Sindagi VA, Patel VM (2017b) Generating high-quality crowd density maps using contextual pyramid cnns. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1879–1888

Sindagi VA, Yasarla R, Patel VM (2019) Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1221–1231

Sindagi VA, Yasarla R, Patel VM (2020) Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. Technical Report

Walach E, Wolf L (2016) Learning to count with cnn boosting pp 660–676

Wan J, Luo W, Wu B, Chan AB, Liu W (2019) Residual regression with semantic prior for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4036–4045

Wang C, Zhang H, Yang L, Liu S, Cao X (2015) Deep people counting in extremely dense crowds. In: Proceedings of the 23rd ACM international conference on Multimedia, ACM, pp 1299–1302

Wang Q, Chen M, Nie F, Li X (2018a) Detecting coherent groups in crowd scenes by multiview clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence DOI 10.1109/TPAMI.2018.2875002

Wang Q, Wan J, Yuan Y (2018b) Deep metric learning for crowdedness regression. IEEE Transactions on Circuits and Systems for Video Technology 28(10):2633–2643

Wang Q, Gao J, Lin W, Yuan Y (2019) Learning from synthetic data for crowd counting in the wild. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 8198–8207

Wang Q, Gao J, Lin W, Li X (2020) Nwpu-crowd: A large-scale benchmark for crowd counting. arXiv preprint arXiv:200103360

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4):600–612

Xiong F, Shi X, Yeung DY (2017) Spatiotemporal modeling for crowd counting in videos. arXiv: Computer Vision and Pattern Recognition

Yan Z, Yuan Y, Zuo W, Tan X, Wang Y, Wen S, Ding E (2019) Perspective-guided convolution networks for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision, pp 952–961

Yuan Y, Fang J, Wang Q (2014) Online anomaly detection in crowd scenes via structure analysis. IEEE transactions on cybernetics 45(3):548–561

Zhang C, Kang K, Li H, Wang X, Xie R, Yang X (2016a) Data-driven crowd understanding: a baseline for a large-scale crowd dataset. IEEE Transactions on Multimedia 18(6):1048–1061

Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016b) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp 589–597

Zhao M, Zhang J, Zhang C, Zhang W (2019) Leveraging heterogeneous auxiliary tasks to assist crowd counting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 12736–12745

Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint

Zuo W, Wu X, Lin L, Zhang L, Yang MH (2018) Learning support correlation filters for visual tracking. IEEE transactions on pattern analysis and machine intelligence 41(5):1158–1172