Drone-Based Car Counting via Density Map Learning

Jingxian Huang[1,†], Guanchen Ding[1,†], Yujia Guo[1], Daiqin Yang[1,*],Sihan Wang[2], Tao Wang[2], Yunfei Zhang[2]
[1]*School of Remote Sensing and Information Engineering, Wuhan University, China*
[2]*Tencent Inc., Shenzhen 518052 China*
dqyang@whu.edu.cn

***Abstract*—Car counting on drone-based images is a challenging task in computer vision. Most advanced methods for counting are based on density maps. Usually, density maps are first generated by convolving ground truth point maps with a Gaussian kernel for later model learning (generation). Then, the counting network learns to predict density maps from input images (estimation). Most studies focus on the estimation problem while overlooking the generation problem. In this paper, a training framework is proposed to generate density maps by learning and train generation and estimation subnetworks jointly. Experiments demonstrate that our method outperforms other density map-based methods and shows the best performance on drone-based car counting.**

***Index Terms*—Car counting, density map learning, drone-based image**

I. Introduction

Counting is an important task in computer vision and has attracted a lot of attention in recent years. Given an image as input, the mission of counting is to predict the number of specific objects in the image. Counting has a wide range of applications, such as counting the total number of people in surveillance video [1], estimating the number of vehicles in aerial image [2], and counting the number of cells in microscope image [3]. Severe occlusion and overlap between objects, scale changing caused by perspective, and other factors make counting full of challenges. Most advanced methods [4], [5] are based on density maps, in which the sum of pixels indicates the count of its corresponding image. These methods consist of two steps: generation and estimation. First, density maps are generated from the ground truth point maps, usually by convolving with Gaussian kernels. Second, the mapping from input images to density maps are learned by a network, with the density maps generated in the previous step. Recently, studies have made efforts to improve the performance of density maps estimation. While the generation step, which also plays an important role in counting [6], is often overlooked.

Images captured by drones contain large-scale ground information. Although counting has made significant progress in the past period, few studies have been applied to drone-based images. Almost all density map-based methods rely on fixed Gaussian kernels to estimate density maps. Due to messy background and different sizes of individual cars, this method of generating density maps may lead to inaccurate counting in drone-based images. To solve this problem, in this paper, we propose a new method to generate density maps for drone-based counting. In this method, the density maps used for model training are generated by learning so that their shapes and sizes can be talored to individual cars. The training and estimating network we proposed consists of a Density Map Generator (DMG) subnetwork and a Density Map Estimator (DME) subnetwork. Firstly, the DMG subnetwork is trained with ground truth point maps and extracted features of drone-based images. Through the DMG, density maps are generated. Then, the generated density maps are used by the DME subnetwork to train its model for estimating density map. Both DMG and DME are trained jointly within the network. Experiments demonstrate the superiority of the proposed method over state-of-the-art methods.

The contributions of this paper are summarized as follows:

- A new method is proposed for counting, which learns to generate density maps for model training. With this network, car counting for drone-based images can be more accurate.
- An innovative loss function is proposed. With the function, the distribution of the generated density map is closer to reality.

Despite there is no comprehensive information and complex architecture in our proposed network, it achieves state-of-the-art performance on CARPK dataset.

II. Related Work

The key to density map-based methods is accurate density maps and reasonable estimation models. Recent studies have made efforts to improve the performance of density maps estimation. Ranjan et al. [4] present a Convolutional Neural Network to estimate high-resolution density maps from low-resolution density maps. Sam et al. [5] refine the prediction by the proposed feedback mechanism. Although the generation of density maps also plays an important role in object counting, few methods focus on it. Wan et al. [6] proposed a refiner to refine the density map which is generated by Gaussian kernel.

†These authors with contribute equally to the work.
*Corresponding Author: Daiqin Yang. This work was supported in part by grants from the National Natural Science Foundation of China under Grant 61771348, and Tencent.

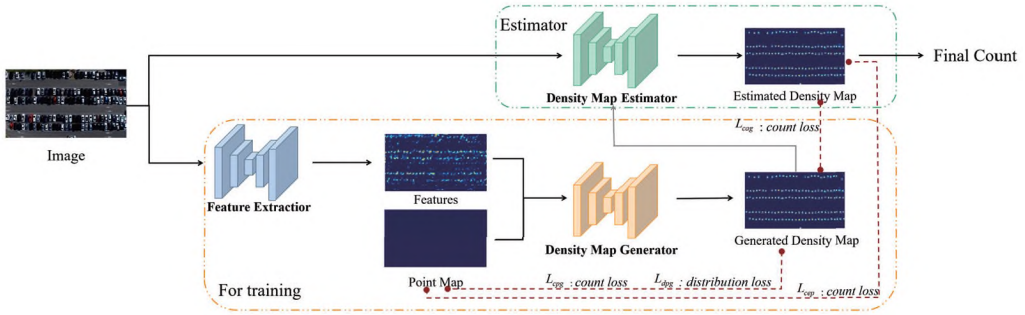

Fig. 1. Architecture of the Network. For training, point map and features extracted from image are input into the Density Map Generator (DMG) subnetwork. Through DMG, density map is generated to supervise the training of Density Map Estimator (DME) subnetwork. Both DMG and DME are trained jointly. For testing, the result is obtained from image by the DME only. Dotted lines connecting maps represent the loss functions that measure the difference between the maps.

Recently, car counting for drone-based images developed rapidly. Mundhenk et al. [7] proposed a neural network, named ResCeption, combining residual learning with inception-style layers to count cars. Cai et al. [8] proposed Guided Attention Network (GANet), which consists of weakly-supervised Background Attention and Foreground Attention Module, to detect and count cars on drone-based scenes. Gao et al. [9] designed a neural network, which consists of a convolution block attention module, scale pyramid module, and deformable convolution module, to predict density map. Most of counting methods depend on the detection of objects, which is not suitable for low-resolution images.

III. Proposed Method

In this paper, we propose a method to generate proper density maps for learning for drone-based car counting. In this section, the architecture of the proposed network is described including the details of subnetworks. After that, the loss functions used in the network are introduced in detail. At the end of this section, the model implementation details of the network are depicted.

A. Network Architecture

The architecture of the proposed network is shown in Fig. 1. The proposed network is mainly composed of two subnetworks: DMG and DME.

To learn more information, the features of images are extracted by Feature Extractor first. Given point maps and the features as input, DMG subnetwork is trained to generate proper density maps. The generated density maps are then used by DME to train its model.

With the supervision of density maps generated by DMG subnetwork, DME subnetwork learns to estimate density maps from input images. These subnetworks are trained jointly, and the training and estimating network are fine-tuned.

After training, DME can estimate density maps independently. During testing, the learned DME estimates density maps from images directly. Finally, the counts are obtained from estimated density maps.

B. Details of the network

1) Density Map Generator Subnetwork: DMG learns to generate density maps adaptively. To improve the efficiency of the model, DMG uses a simple encoder-decoder architecture.

The DMG uses the first ten convolutional layers of the VGG16 [10] model as the encoder. The point maps provided by the datasets are sparse matrix mathematically. Considering that the sparse matrix contains little information that can reflect the characteristics of the objects, we use features extracted by VGG16 together with point maps as the input of the subnetwork. To ensure the simplicity of the network, we use simple nearest upsampling as the decoder to directly obtain the density map.

2) Density Map Estimator Subnetwork: In DME, we use the Resnet101 [11] as the skeleton. To ensure that the generated density map is suitable in size, we modify the size of stride in res.layer 3 to 1 (the original is 2) as the encoder. Following the simple principle, we use a two-layer convolution to form a decoder structure.

C. Loss Function

1) Density Map Generator loss: We combine two kinds of loss functions to penalize the DMG. The first loss measures the count error between the generated density map and the ground-truth point map with the function:

$$L_{cpg} = |C^{ag} - C^{gt}|, \tag{1}$$

where C^{gt} denotes the count of point map, which can be seen as ground-truth, C^{ag} denotes the count of density map generated in DMG.

Besides, the distribution of the density map should be adapted to the point map. To measure the difference between the generated density map and point map, we used the loss function of:

$$L_{dpg} = \frac{1}{N} \sum_{(x,y)\in I} \left(C^{ag}_{(x,y)} * C^{gt}_{(x,y)} \right), \tag{2}$$

where N denotes the number of pixels in point map. (x, y) denotes the location of pixel, and $C^{\Phi}_{(x,y)}$ $(\Phi \in \{ag, gt\})$ denotes the the value of (x, y) in density maps.

To generate optimal density map, the difference between it and the point map should be as small as possible, while the similarity between corresponding points should be as large as possible, when we train the DMG, the loss is summarized as:

$$L^{DMG} = \lambda_1 L_{cpg} + \lambda_2 L_{dpg}, \tag{3}$$

where λ_1 is set to be 0.001, and λ_2 is set to be -0.01.

2) Density Map Estimator loss: In DME, the first loss measures the count error between the predict density map and generated density map, with this loss, the generated density map supervises the counter to estimate density map:

$$L_{ceg} = |C^{est} - C^{ag}|, \tag{4}$$

where C^{est} denotes the count of predicted density map, similar to $eq.1$.

Besides, DME should meet the conditions that the prediction result is close to reality, therefore we utilize the loss:

$$L_{cep} = \left(C^{est} - C^{gt}\right)^2. \tag{5}$$

Then the loss function of DME subnetwork can be expressed as:

$$L^{DME} = L_{ceg} + L_{cep}. \tag{6}$$

The final loss of the network is summarized as the following:

$$L = L^{DMG} + L^{DME}. \tag{7}$$

D. Training Strategy

We use the Adam [12] to optimize the proposed network and set the β_1 and β_2 to 0.9 and 0.999. The training patch size was randomly cropped into 256×256 pixels. We first trained our DMG subnetwork with a learning rate schedule of $1e-5$, dropping by 0.2 when the loss does not fall within the range of 5 epochs. After the network stabilizes, we use the generated density map and train the car counting subnetwork with the same strategy. Finally, we fine-tuned the entire network with a learning rate schedule of $1e-6$.

IV. Experiments

A. Dataset and Evaluation Metrics

We test the proposed network on CARPK [13] dataset. CARPK dataset contains 89777 cars from 4 different parking lots. The 1448 images of CARPK are captured by drones at approximately 40 meters height.

For quantitative evaluation, we compare the performance of the proposed method with state-of-the-art methods using two widely used evaluation metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE):

$$MAE = \frac{1}{K}\sum_{k=1}^{K}|N_k - C_k|, \tag{8}$$

TABLE I
The Comperison of Different Density Maps

Counter	Density Map	MAE	MSE
MCNN	Gaussian kernel (size=5)	39.10	43.30
	Gaussian kernel (size=15)	10.91	14.18
	Gaussian kernel (size=25)	39.77	43.95
	Density Map Generator (ours)	**10.57**	**13.91**
CSRNet	Gaussian kernel (size=5)	11.48	13.32
	Gaussian kernel (size=15)	7.80	9.76
	Gaussian kernel (size=25)	15.90	21.07
	Density Map Generator (ours)	**5.97**	**7.73**
Rsenet101	Gaussian kernel (size=5)	17.56	20.95
	Gaussian kernel (size=15)	6.05	8.16
	Gaussian kernel (size=25)	14.29	17.25
	Density Map Generator (ours)	**5.13**	**6.75**

Bold mumbers depict the best preformance.

$$MSE = \sqrt{\frac{1}{K}\sum_{k=1}^{K}|N_k - C_k|^2}, \tag{9}$$

where K denotes the number of test images,N_k and C_k denote the ground truth count and the estimated count of $k-th$ image.

B. Ablation Study

We attempt to generate proper density maps for car counting. To evaluate the effectiveness of the proposed DMG subnetwork, we compare it with two variations: 1) counting networks trained with traditional Gaussian kernel-based density maps, the Gaussian kernel sizes are set to 5, 15, and 25; 2) counting networks jointly trained with the proposed DMG. The baseline networks used in the experiment include MCNN [14], CSRNet [15], and Resnet101 [11].

The experimental results are shown in Table I. Counters jointly trained with DMG outperform that trained with tradition density maps generated by Gaussian kernels. Among Gaussian kernel-based density maps, the kernel of size 15 is superior than the other two.

In Fig. 2, we visualize the estimated density maps of baseline networks trained with density maps generated by Gaussian kernels and the proposed DMG. The distribution of highlight areas on the Gaussian kernel-based density maps is scattered. While in density maps estimated from DMG, the highlight areas are concentrated on the actual location of the vehicle. It proves the superiority of the density maps generated by DGM over fixed Gaussian kernels. Fig. 2(a) and Fig. 2(b) give a close look to the generated density map of DGM. As demonstrated in these two figures, the generated density map of each cars can have varying shapes according to the sizes of individual cars and their similarity to the background.

C. Comparison with state-of-the-art

We conduct the experiment to compare the performance of our proposed method with state-of-the-art methods, including GAP [16], GSP [17], NFCNN [18], and the baseline methods MCNN, CSRNet, and Resnet101 which are mentioned above. GAP, GSP, and NFCNN are methods proposed for objecting counting in recent years. Since these studies have not yet open

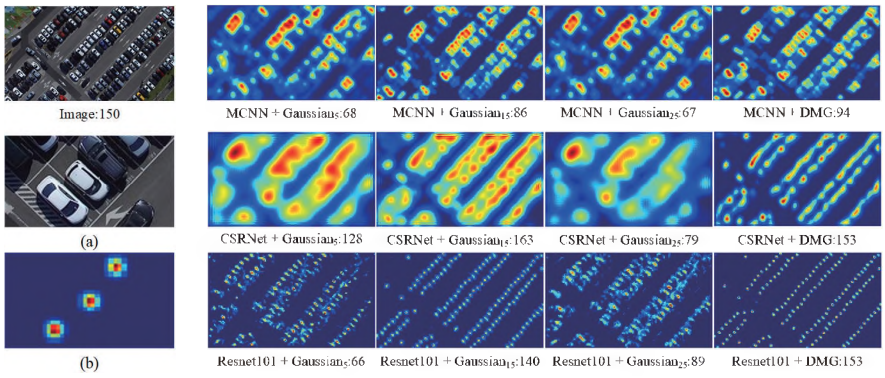

Fig. 2. Image and density maps. (a) and (b) show enlarged local area of the image and its density map generated by DMG. The second to fifth columns show density maps estimated by different counters. Subscripted numbers indicate kernel sizes of Gaussian. Numbers after the colon indicate the counts of density maps.

TABLE II
THE COMPERISON OF DIFFERENT METHODS

Method	MCNN	CSRNet	Resnet101	GAP	GSP	UFCNN	Ours
MAE	10.19	7.80	6.05	7.88	5.46	5.42	**5.13**
MSE	14.18	9.76	8.16	9.30	–	7.38	**6.75**

Bold mumbers depict the best preformance.

their source code, the results of our proposed are compared with what they provided in their paper.

As shown in Table II, our proposed method is lower than the optimal method by 0.29 on MAE. On MSE, this data is 0.63. Compared with the most classic algorithm MCNN in counting, the proposed method has improved 5.06 and 6.43 on MAE and MSE. On the most challenging dataset in drone-based car counting, our method outperforms the state-of-the-art methods.

V. CONCLUSITON

In this paper, we propose a method for drone-based counting. The network consists of a DMG subnetwork and a DME subnetwork. DMG learns to generate proper density maps. And these density maps are then used to supervise the training of DME. Both DMG and DME are trained jointly. With the trained DME, density maps can be estimated from images. Comparing with traditional Gaussian kernel-based density maps, the density maps generated by learning can adapt to local differences among individual cars and improve the performance of counting networks. And as demonstrated by the experimental results, our method outperforms state-of-the-art methods on drone-based images.

REFERENCES

[1] Z. Zhang, M. Wang, and X. Geng, "Crowd counting in public video surveillance by label distribution learning," *Neuro Computing*, vol. 166, pp. 151–163, 2015.

[2] M. Marsden, K. McGuinness, S. Little, C. E. Keogh, and N. E. O'Connor, "People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8070–8079.

[3] J. Paul Cohen, G. Boucher, C. A. Glastonbury, H. Z. Lo, and Y. Bengio, "Count-ception: Counting by fully convolutional redundant counting," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 18–26.

[4] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 270–285.

[5] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *AAAI conference on artificial intelligence*, 2018.

[6] J. Wan and A. Chan, "Adaptive density map generation for crowd counting," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 1130–1139.

[7] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 785–800.

[8] Y. Cai, D. Du, L. Zhang, L. Wen, W. Wang, Y. Wu, and S. Lyu, "Guided attention network for object detection and counting on drones," *arXiv preprint arXiv:1909.11307*, 2019.

[9] G. Gao, Q. Liu, and Y. Wang, "Counting dense objects in remote sensing images," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4137 4141.

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

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

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

[13] M. R. Hsieh, Y. L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4145–4153.

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

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

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

[17] ——, "Global sum pooling: A generalization trick for object counting with small datasets of large images," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 73–82.

[18] W. Li, H. Li, Q. Wu, X. Chen, and K. N. Ngan, "Simultaneously detecting and counting dense vehicles from drone images," *IEEE Transactions on Industrial Electronics (TIE)*, vol. 66, no. 12, pp. 9651–9662, 2019.