Object Counting for Remote Sensing Images via Adaptive Density Map Assisted Learning

Guanchen Ding, *Student Member, IEEE,* Mingpeng Cui, Daiqin Yang, *Member, IEEE,* Tao Wang, Sihan Wang, and Yunfei Zhang

***Abstract*—Object counting has attracted a lot of attention in remote sensing image analysis. In density map based object counting algorithms, the ground truth density maps generated by fix-sized Gaussian kernels ignore the spatial features of the objects. In this paper, an Adaptive Density Map Assisted Learning algorithm (ADMAL) is proposed, which taps into spatial features of the objects from the beginning phase of ground truth density map generation. ADMAL consists of two networks: a Contexture Aware Density Map Generation (CADMG) network and a Transformer-based Density Map Estimation (TDME) network. The CADMG network is designed to generate a ground truth density map from each annotated point map. Comparing with Gaussian convolved density maps, the ground truth density maps generated by CADMG will be tailored according to the texture and neighborhood relationship among objects, which can promote the learning effect of the TDME network. TDME is the core network for object counting. The backbone of the TDME network adopts a Swin transformer structure, the self-attention mechanism of which possesses a larger receptive field for effective feature extraction in remote sensing images. Comprehensive experiments prove that the ground truth density map generated by CADMG can help various density map estimation networks achieve better training effects, among which TDME achieves the best performances. Moreover, the ADMAL algorithm can achieve preferable object counting performances for both satellite-based image and drone-based image. Code and pre-trained models are available at https://github.com/gcding/ADMAL-pytorch.**

***Index Terms*—Object Counting, Contexture Aware Density Map Generation Network, Transformer, Self-attention, Remote Sensing**

I. Introduction

With the rapid advancement of computer vision, object counting has led to a surge of research interest in recent years. Object counting aims to count a specific category in an image or video [1]–[3]. It can be applied in numerous fields, such as crowd behavior understanding [4] and traffic analysis [5]. Recent studies adopt Convolutional Neural Network (CNN) to regress a density map from an input image and get the counting result by summing up the density map. These density map based methods have made significant progress and become the mainstream in object counting.

This work was supported in part by grants from the National Natural Science Foundation of China under Grant 61771348, and Tencent.

G. Ding, M. Cui and D. Yang are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: gcding@whu.edu.cn; Ceoilmp@whu.edu.cn; dqyang@whu.edu.cn).

T. Wang, S. Wang and Y. Zhang are with Tencent, Shenzhen 518000, China (email: tuckerwang@tencent.com; lovingwang@tencent.com; yanniszhang@tencent.com).

Corresponding author: Daiqin Yang.

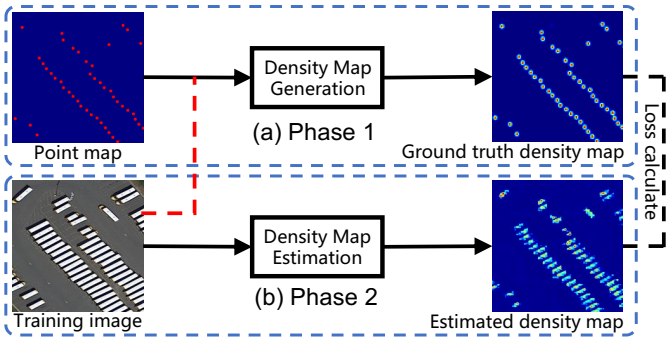

Fig. 1. Illustration of the general framework of density map based object counting algorithms. The red dashed line represents the fundamental difference between our proposed DMG method and the mainstream DMG methods.

To avoid regression from continuous image to discrete point map, the training process of density map based methods usually consists of two phases: generating ground density maps with a Density Map Generation (DMG) module and training a Density Map Estimation (DME) module with the ground truth density map just generated. Currently, most DMG modules use Gaussian convolution to generate continuous ground truth density maps from discrete point maps as shown in part (a) of Fig. 1 and the DME module usually adopts a CNN-based regression model for density map estimation as shown in part (b) of Fig. 1.

Most existing methods focus on optimizing phase 2. Phase 1, which also plays an important role, tends to be overlooked by researchers. Wang et al. [6] tried to develop an adaptive density map generation network to fuse multiple density maps for surveillance images. The network outputs masks for density maps generated by different Gaussian kernels through a self-attention mechanism. This method essentially deals with scale variation of objects by fusing density maps of different Gaussian kernels. However, Gaussian shaped densities may not be the best choice for object characterization and network learning. For phase 2, the size of the receptive field is directly related to the performance of deep neural network based estimation. Generally, CNN-based networks usually use deformable convolution [7] or dilated convolution [8] to expand their receptive field as in ASPDNet [9], [10] and CSRNet [11]. However, even with these two methods, the receptive fields of CNN-based networks are still limited.

In this paper, an Adaptive Density Map Assisted Learning (ADMAL) algorithm is proposed to deal with these aforementioned issues. It integrates the two phases into one

optimization framework, in which the ground truth density maps are generated toward the learning effect of the regression model. ADMAL consists of two networks: a Contexture Aware Density Map Generation (CADMG) network and a Transformer-based Density Map Estimation (TDME) network. As shown in Fig. 1, by using the image as additional input, the proposed CADMG can generate ground truth density maps adaptively by combining the contextual information of objects with their precise locations. The TDME network is designed to regress an input image to an estimated density map. It adopts the transformer structure, which is a sequence-to-sequence prediction method with a global receptive field, to enlarge the receptive field for more accurate contextual information extraction. The transformer structure was originally designed for Nature Language Processing (NLP) and direct use of the transformer in image feature extraction can lead to unbearable computational complexity. To make a balance between complexity and the size of receptive field, the Swin transformer [12] is chosen and adopted as the backbone of TDME.

The training process of ADMAL includes three steps. In the first step, the CADMG network is trained to generate ground truth density maps with the input of annotated point maps and their related training images. Then the TDME network is trained by the training images and the ground truth density maps generated by CADMG. In the last step, the CADMG and TDME networks are trained jointly to both generate learning-oriented ground truth density maps and refine the regression model for better prediction. In the testing stage, only the TDME network is kept to predict a density map from an input image for the counting result, so the design of the CADMG network will not affect the inference speed. Comprehensive experiments demonstrate that ADMAL outperforms state-of-the-art methods on satellite-based data sets and is competitive with state-of-the-art methods on drone-based data set. The contributions of this paper are summarized as follows:

- For the DMG module, a generic CADMG network is proposed to tap into the spatial features of the objects so as to generate better ground truth density maps in terms of promoting the learning effect of the DME module.
- For the DME module, a TDME network using the transformer structure is proposed to achieve accurate context information extraction with larger receptive fields. And the proposed ADMAL algorithm integrates the CADMG network and the TDME network into a whole framework for better training performance of the TDME network.
- Comprehensive experiments prove that our CADMG network can achieve excellent performance improvement for different DME modules and the performance of the proposed ADMAL algorithm is better than the state-of-the-art methods.

Compared with the previous conference version [13], this article has the following extensions and improvements: (1) The transformer structure, instead of the CNN structure, is used as the backbone for density map estimation. The enlarged receptive field of the transformer is particularly important for large-scale remote sensing images. (2) Comprehensive experiments are conducted, including performance comparison with different algorithms on various counting objects. Ablation studies for different modules are also conducted. (3) A brief review of the related work and a more detailed introduction to the proposed network are presented to give readers a comprehensive understanding of our work.

II. Related Work

In this section, we review the recent object counting methods. In addition, we also discuss the related topic of the visual transformer.

A. Object counting

The development of object counting mainly has the following three stages: (1) detection and count; (2) direct regression; (3) regression density map and count.

1) Detection and count: At first, Moranduzzo et al. [14] use keypoint extraction and support vector machine classification methods to detect and count UAV image vehicles. Later, with the development of neural networks, CNN-based detection algorithms [15]–[18] are widely used in object counting. However, in dense scenes or when objects in the image are severely occluded, detection and count methods are often difficult to give satisfactory counting results.

2) Direct regression: In order to solve this problem, later methods [19], [20] directly use regression to get the total number of objects in the image. However, the counting result given by this method has an upper bound, which is obviously unreasonable.

3) Regression density map and count: Lempitsky et al. [21] first propose the method of regression density map and count. This method obtains the density map by convolving the point map with the fixed Gaussian kernel and then regressing it to the generated density map. The accumulation of the predicted density map is the final count result. Since the introduction of this method, the method of regressing density map and count gradually became the mainstream. MCNN [22] proposes a method of using three columns of different convolution kernels to regress the density map in parallel to solve the image target scale variation. Switch-CNN [23] generates multiple density map predictions at a time and then uses a switch to determine the optimal option as the final result. CSRNet [11] proposes to use dilated convolution to increase the receptive field of the network so as to make the network perform better and more robust. SAAN [24] proposes to use the attention mechanism forcing the model to automatically focus on certain global and local scales appropriate for the image. Liu et al. [25] further injected the attention of crowd understanding into deformable convolution and proposed ADCrowdNet to solve the problem of accuracy degradation in high congested noisy scenes. Guo et al. [26] proposed the Dilated-Attention-Deformable ConvNet (DADNet) model combining dilated convolution, attention mechanism and deformable convolution. DADNet is not only good at capturing the rich spatial context of salient and tiny regions of interest simultaneously, but it also maintains robustness to background noise (such as partially occluded objects). However, these algorithms only focus on how to

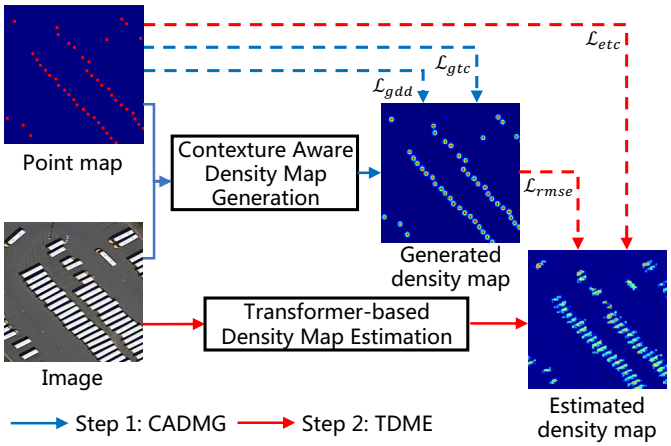

Fig. 2. Architecture of the Adaptive Density Map Assisted Learning (ADMAL) algorithm. The blue dotted line and red dotted line represent the loss functions of CADMG and TDME, respectively.

better regress the density map, without considering whether the generated density map is reasonable and whether it is the most suitable for network learning.

B. Remote Sensing Object Counting

In the past few years, there are also some literature on drone-based image and satellite-based image object counting. Mundhenk et al. [27] propose a neural network, named ResCeption, combining residual learning with inception-style layers to count cars. Qiang et al. [28] propose a Guided Attention Network (GANet), which considers semantic feature representation and object appearance through weakly supervised background attention and foreground attention. Hsieh et al. [29] propose Layout Proposal Networks (LPN) and spatial kernels use spatial layout information to count and localize targets simultaneously. Gao et al. proposed a remote sensing image object counting data set RSOC containing four types of targets in [9], and proposed ASPDNet [9] for remote sensing image object counting. Gao et al. [10] later expanded the data set and improved the algorithm. DSACA [30] designs a multi-mask structure to suppress negative interactions between objects to achieve multi-object counting. PSGCNet [31] combines Pyramid Scale Module (PSM) and Global Context Module (GCM) to alleviate the problem of large scale in remote sensing image object counting.

It could be found that most of the existing algorithms have two problems: (1) overlook the importance of a more suitable and reasonable ground truth density map; (2) the ways and effects of increasing the receptive field of deep neural network based networks are limited. To solve the first problem, a Contexture Aware Density Map Generation (CADMG) network is proposed, which can be trained end-to-end. To solve the second problem, the transformer structure, which is popular in NLP and contains a larger receptive field is introduced into density map estimation. Background information about the transformer structure will be introduced in the next subsection.

TABLE I
A LIST OF NOTATIONS MAINLY USED IN THIS PAPER.

Symbol	Definition
x_i, y_j	Image for training and image for testing
p_i	Annotated point map for training image x_i
d_i^{dmg}, d_i^{dme}	Density map generated by DMG and predicted by DME
σ_i	Gaussian kernel size
$C_i, \hat{C}_i$	Counting result of point map and density map
$\mathcal{L}$	loss function
H, W	Height and weight of image
N, c	Batch size and the number of channels
λ_i	Hyper parameters

C. Visual transformer

The transformer [32], which is popular in NLP [33], [34], uses the (multi-head) self-attention mechanism to capture the global dependency between input and output. DETR [35] regards target detection as a set prediction problem, uses CNN to extract basic features, sends them to transformer for relationship modeling, and obtains the detection results. Based on DETR, Deformable-DETR [36] uses multi-scale deformable attention instead of self-attention in Encoder and cross-attention in Decoder. The pioneering work of ViT [37] directly uses a pure transformer architecture to classify small non-overlapping image blocks. Similar to ViT, SETR [38] uses transformer for semantic segmentation from sequence to sequence perspective. However, the high performance of Vit and its follow-ups are based on large-scale training data sets. At the same time, these methods have a large computational complexity when processing high-resolution images. Swin [12] reduces the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ by grouping N tokens of the native transformer. At the same time, through hierarchical and shifted windows, the network can see a larger range of information and strengthen the information interaction between groups.

III. Methodology

In this section, an overview of ADMAL is depicted first. Then, the structure of the Contexture Aware Density Map Generation (CADMG) network and the Transformer-based Density Map Estimation (TDME) network are introduced, respectively. Finally, the method of joint training is briefed.

A. ADMAL Overview

As shown in Fig. 2, ADMAL comprises two networks: the Contexture Aware Density Map Generation (CADMG) network and the Transformer-based Density Map Estimation (TDME) network. For each training image x_i, it has an annotated ground truth point map p_i. As p_i is a sparse binary matrix, it is hard for the TDME network to learn a proper regression function between x_i and p_i. Therefore, ADMAL first trains the CADMG network to generate a continuous ground truth density map d_i^{dmg} from each p_i and then utilizes these ground truth density maps to construct a regression function with TDME through supervised learning. After these

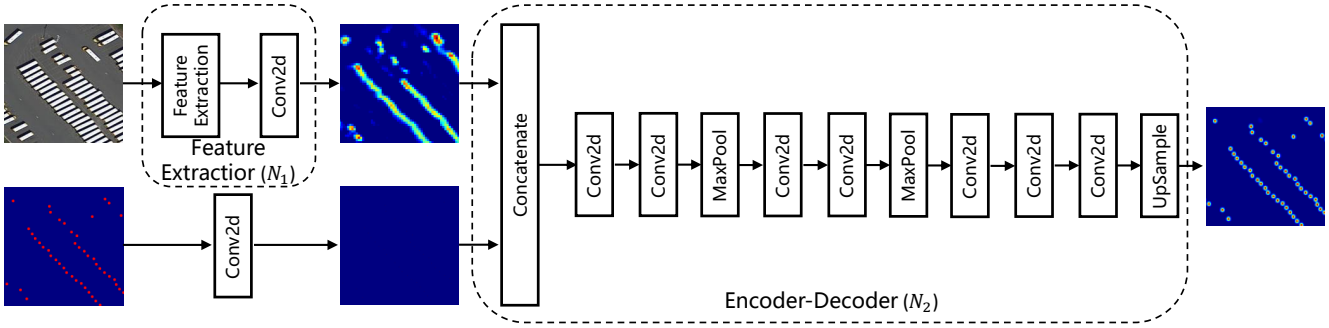

Fig. 3. Architecture of the Contexture Aware Density Map Generation (CADMG) network. The point map and the training image serve as the input of CADMG to generate a ground truth density map. All the Conv2d in the figure are followed by a ReLU activation function, which is omitted from the figure.

two coarse learning steps, CADMG and TDME are then jointly trained together, encouraging information exchange between them. The learning rate of the joint training is set to a lower level, imposing a fine-tuning to the parameters. The whole training process of ADMAL can be summarized into the following three steps.

Step1: Training of the Contexture Aware Density Map Generation network. The CADMG network takes the training image x_i and its point map p_i as an input pair and generates ground truth density map d_i^{dmg} for each p_i. The network is trained with a combined density distribution loss and total counting loss.

Step2: Training of the Transformer-based Density Map Estimation network. Taking both the generated density map d_i^{dmg} and the point map p_i as ground truth, the TDME network is trained to predict a density map d_i^{dme} for each training image x_i. The network is trained with a combined density error loss and total counting loss.

Step3: Joint training of the CADMG and TDME networks. After CADMG and TDME are individually and sequentially trained, they are then trained jointly by summing up the losses from the two networks and using a lower learning rate to fine-tune both of the two networks together.

A list of notations used in this paper are summarized in Table I.

B. The Contexture Aware Density Map Generation network

The DMG module is designed to turn discrete annotated point maps into continuous density maps for the learning of proper regression functions. The proposed CADMG network is an embodiment of the DMG module. To compare with existing DMG algorithms, the widely used Gaussian based density map generation method is described first.

1) Gaussian based density map generation: Lempitsky et al. [21] first propose to use a density map for crowd counting. Subsequently, density map based methods have gradually become the mainstream for object counting. Most of the methods directly generate ground truth density maps through a Gaussian convolution function of:

$$d_i = G_{\sigma_i}(p_i) \tag{1}$$

where p_i is the point map, d_i is the density map generated from p_i, and $G_{\sigma_i}(\cdot)$ represents Gaussian convolution with a standard deviation σ_i. Most of the existing methods empirically adopt the Gaussian kernel with $\sigma_i = 15$.

2) CADMG network: With different contextual and neighborhood relationships among objects, a fix-sized Gaussian can not optimally reflect the diverse situations of individual objects in the ground truth density map. And a Gaussian-shaped density map may not even be the best choice for the sake of DME's learning effect. Therefore, the CADMG network is designed to tap into image features of individual objects by integrating the training images into the generated ground truth density maps. At the same time, it will fine-tune the generated ground truth density map according to the learning effect of the TDME network.

As shown in Fig 3, the CADMG network takes both point map p_i and its corresponding training image x_i as input. Image x_i is first processed by a feature extraction sub-network $\mathcal{N}_1$, which adopts the pre-trained VGG19 [39] model and uses the output of the relu4_4 layer as the feature map. The feature map together with the point map are then fed into an estimation sub-network $\mathcal{N}_2$, which adopts a simple encoder-decoder architecture. The encoder uses five convolution layers with ReLU activation functions and two max-pooling to extract features. The decoder uses two convolution layers with ReLU activation functions and the bicubic upsampling to get the density map.

3) Loss function: Taking the point maps as ground truth, the loss function for the training of CADMG consists of two portions, the total counting loss and the density distribution loss. The total counting loss function measures the discrepancy between the total counting number of the generated density map d_i^{dmg} and the point map p_i. Let C_i denote the ground truth count calculate from p_i:

$$C_i = \sum_{h=1}^{H}\sum_{w=1}^{W} p_i(h,w) \tag{2}$$

where H and W are the height and width of p_i, $p_i(h,w)$ represents the value of p_i at position (h,w). The total counting loss is defined as:

$$\mathcal{L}_{gtc} = \frac{1}{N}\sum_{i=1}^{N} |C_i - \hat{C}_i^{dmg}| \tag{3}$$

where N is the number of training images in each training batch and $\hat{C}_i^{dmg}$ is the counting result calculate from d_i^{dmg} as in Eq. 2.

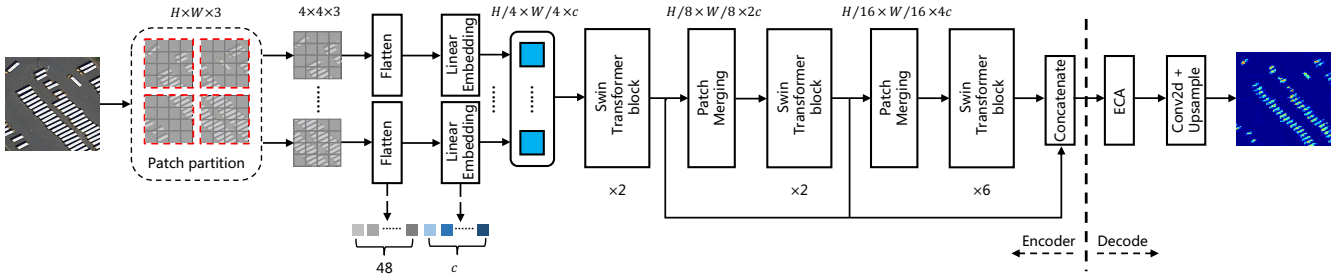

Fig. 4. Architecture of the Transformer-based Density Map Estimation (TDME) network. The Swin transformer is used as the backbone of the TDME network. Each blue rectangular block with a black border represents the c channel output of a Linear Embedding.

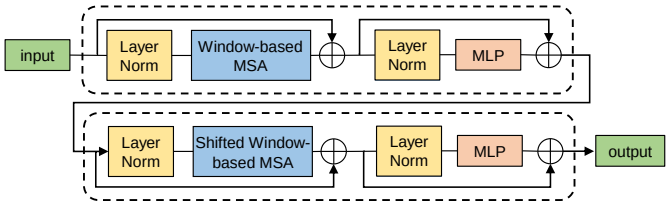

Fig. 5. Layer structure of a Swin transformer block.

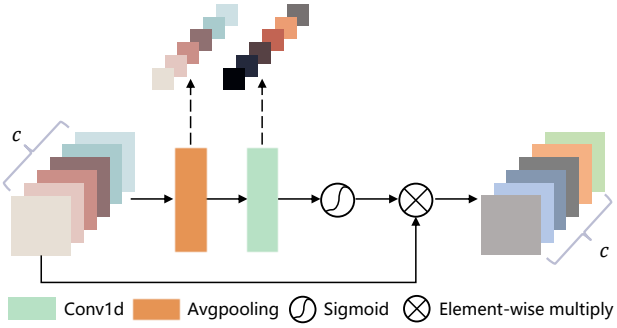

Fig. 6. Architecture of the Efficient Channel Attention (ECA) module in detail.

In addition to an accurate sum of each generated ground truth density map, the distribution of the generated densities should also conform to the distribution of annotated points in the point map. The density distribution loss is thus defined as the following formula:

$$\mathcal{L}_{gdd} = -\frac{1}{N \times H \times W} \sum_{i=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (p_i(h,w) \times d_i^{dmg}(h,w)) \tag{4}$$

As p_i is a sparse binary matrix, in which annotated points are 1s and others are 0s, $-\mathcal{L}_{gdd}$ measures the similarity between the generated ground truth density map d_i^{dmg} and the point map p_i by calculating the average density value at the positions of the annotation points. A smaller $\mathcal{L}_{gdd}$, a larger $-\mathcal{L}_{gdd}$ in turn, means that density peaks of d_i^{dmg} conform to the annotation positions.

As the total counting number of d_i^{dmg} should also be as close as possible to the actual total number in p_i. The total loss of CADMG network is thus summarized as:

$$\mathcal{L}_{dmg} = \lambda_1 \mathcal{L}_{gtc} + \lambda_2 \mathcal{L}_{gdd} \tag{5}$$

C. The Transformer-based Density Map Estimation network

In the TDME network, the Swin transformer structure [12], instead of the commonly used CNN structure, is adopted as the encoder. With its larger reception field, the transformer-based encoder can obtain stronger feature expression capability for large-scale remote sensing images. The transformer structure has shown its powerful capacity for representation learning. As its computational complexity is proportional to the square of image's pixel number, it is difficult to directly apply the transformer structure to tasks with larger input images. The Swin transformer improves the computational efficiency of the transformer by proposing shifted windows with cross-window connections. Detailed network architecture of TDME is depicted in Fig. 4.

1) Transformer based encoder: As shown in Fig. 4, the input image is split into non-overlapping patches by patch partition and the patch size is set to 4×4. After that, each patch is flattened and converted into the required dimension of c. Three Swin transformer blocks are then applied on these patches to extract features.

Fig. 5 shows the layer structure of a Swin transformer, which consists of Layer Norm, Residual Connections, (Shifted) Window-based Multi-head Self-Attention and MLP blocks. Compared with standard transformer structure, the Swin transformer replaces Multi-head Self-Attention (MSA) with (Shifted) Window-based Multi-head Self-Attention ((S)W-MSA). Through W-MSA, the Swin transformer limits the self-attention mechanism to a window area. The computational complexity is thus changed from $\mathcal{O}(P^2)$ to $\mathcal{O}(PM^2)$, where P is the number of patches and M is the size of window. When $M \ll P$, the computational complexity of window self-attention based Swin is much lower than that of the global self-attention based standard transformer. The window size is set to 7×7. Meanwhile, to maintain a larger receptive field, connections between windows are established through window shifting.

To generate a hierarchical representation, the number of patches is reduced as the network deepens, and the dimension of patches is increased through patch merging layers. The output size of each patch merging layer is indicated in Fig. 4. Features extracted by the three Swin transformer blocks are all scaled to 1/4 size of the input image and cascaded together at the end of the encoder.

2) Decoder: Since the energies of different feature channels are not the same, an Efficient Channel Attention (ECA) mechanism is adopted to pay more attention to prominent channels. The structure of ECA module is shown in Fig. 6. A 1D convolution is conducted first. The size of the convolution kernel k adapts with the channel number c, so as to strike a balance between performance and complexity. Following the same setting in [40], k is calculated by:

$$k = \left| \frac{\log_2(c)}{\gamma} + \frac{b}{\gamma} \right|_{\text{odd}} \tag{6}$$

where $|x|_{odd}$ indicates the nearest odd number of x, c is the number of feature channels, γ and b are hyper parameters, setting to 2 and 1, respectively. With element-wise multiply, features with different energies are given different weights when fused.

After ECA, a 3×3 convolution is used to fuse the features to a single channel, which is then upsampled to the size of the input image and output as the predicted density map.

3) Loss function: Taking both the point map p_i and the generated density map d_i^{dmg} as ground truth, the loss function for the training of TDME also consists of two portions. One is the Root Mean Square Error $\mathcal{L}_{rmse}$. It evaluates the difference between the density maps predicted by TDME and the ground truth density maps generated by DMG. The $\mathcal{L}_{rmse}$ loss function is formulated as follows:

$$\mathcal{L}_{rmse} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |d_i^{dme} - d_i^{dmg}|^2} \tag{7}$$

As the counting result of the predicted density map d_i^{dme} should be close to the actual counting result of the annotated point map p_i. Similar as in DMG, the other part of the loss is defined as:

$$\mathcal{L}_{etc} = \frac{1}{N}\sum_{i=1}^{N} |C_i - \hat{C}_i^{dme}| \tag{8}$$

where the meaning of the symbol is similar as in Eq. 3.

The total loss function of the TDME network is then defined as:

$$\mathcal{L}_{dme} = \lambda_3 \mathcal{L}_{rmse} + \lambda_4 \mathcal{L}_{etc}. \tag{9}$$

D. Joint training

After sequential and independent coarse training of CADMG and TDME, these two networks are then trained together to promote information exchange and fine adjustment. The CADMG network is encouraged to generate ground truth density maps that are more suitable for TDME's learning effect. And the TDME network is encouraged to further refine its parameters for a more accurate prediction. In this last phase of training, a lower learning rate is used for a fine-tune of the whole network. The loss function in this step is defined as:

$$\mathcal{L} = \mathcal{L}_{dmg} + \mathcal{L}_{dme} \tag{10}$$

IV. Experiments

In this section, the performance of ADMAL is compared with state-of-the-art algorithms [1]. Implementation details are introduced first, including data sets, parameter settings, and evaluation metrics. Then, the object counting results on satellite-based data sets and drone-based data sets are reported. In addition to quantitative comparisons, visualized results are also presented to demonstrate the effectiveness of ADMAL.

A. Implementation Details

1) Data sets: To prove the effectiveness of ADMAL, performances are evaluated on two public data sets, RSOC [10], CARPK [29], and a data set proposed by us.

RSOC [10] data set is the largest available satellite-based object counting data set. The four subsets of RSOC are labeled as RSOC_Building, RSOC_Small-Vehicle, RSOC_Large-Vehicle and RSOC_Ship, respectively. CARPK is a classic drone-based data set, which contains nearly 90000 cars with bounding box annotations. In addition to the above two public data sets, a Tree data set is proposed in this paper and the download link for the data set can be found in our code repository [2]. Automatic counting of trees can be used for tracing ecological changes. The Tree data set contains nearly 28,000 annotations in different scenarios. The statistics of these data sets are detailed in Table II.

2) Parameter settings: ADMAL is implemented by PyTorch [48] under the C-3 framework [49] and uses NVIDIA 1080TI GPUs for network training and testing. Parameters are set empirically. For the three steps mentioned in section III-A, the Adam [50] algorithm is used to optimize the network. Hyper parameters λ_1, λ_2, λ_3 and λ_4 are set to 0.001, 0.01, 1 and 0.001. The learning rate of step1 and step2 is set to e^{-5}, and the learning rate of step3 is set to e^{-6}. Following the settings of ASPDNet [9], [10], the batch size is set to 1 and the images with resolutions higher than 768×1024 are downscaled to 768×1024. For the RSOC_Building and CARPK, a batch size of 2 is adopted due to memory limitation of the NVIDIA 1080TI. If the side of an image is not divided by 16, zero paddings are inserted to the edges of the image. All training will end at the 500th epoch.

3) Evaluation metrics: Two widely used metrics in object counting, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), are adopted to measure the performance of each algorithm. They are defined as follows:

$$MAE = \frac{1}{K}\sum_{j=1}^{K} |\hat{C}_j - C_j| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{K}\sum_{j=1}^{K} |\hat{C}_j - C_j|^2} \tag{12}$$

where K represents the number of test images, $\hat{C}_j$ is the predicted count for testing image y_i, and C_j is the ground truth count calculate from y_j's point map p_j.

[1] Some results are obtained from public papers and quoted in the table.
[2] https://github.com/gcding/ADMAL-pytorch

TABLE II
STATISTICS OF THE TESTED DATA SETS.
TOTAL, MIN, AVERAGE AND MAX REPRESENT THE TOTAL NUMBER, MINIMUM NUMBER, AVERAGE NUMBER AND MAXIMUM NUMBER OF OBJECTS IN THE DATA SETS, RESPECTIVELY.

Data set	Object Category	Sensor	Train/Test	Average Resolution	Count Statistics			
					Total	Min	Average	Max
RSOC	Building	Satellite	1205 / 1263	512 × 512	76,215	15	30.88	142
RSOC	Small-Vehicle	Satellite	222 / 58	2473 × 2339	148,838	17	531.56	8531
RSOC	Large-Vehicle	Satellite	108 / 64	1552 × 1573	16,594	12	96.48	1336
RSOC	Ship	Satellite	97 / 40	2558 × 2668	44,892	50	327.68	1661
CARPK	Car	Drone	989 / 459	1280 × 720	89,777	1	62	188
Tree	Tree	Satellite	971 / 212	512 × 512	28,053	0	23.71	278

TABLE III
QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE RSOC DATA SET. THE INFERENCE SPEED IS ACHIEVED ON TREE DATA SET USING ONE NVIDIA 1080TI GPU. THE BEST RESULTS ARE IN RED.

Method	Year & Venue	Building		Small-Vehicle		Large-Vehicle		Ship		Tree		Speed
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	FPS
MCNN [22]	2016 CVPR	13.65	16.56	488.65	1317.44	36.56	55.55	263.91	412.30	5.87	9.06	180.89
CMTL [41]	2017 AVSS	12.78	15.99	490.53	1321.11	61.02	78.25	251.17	403.07	7.79	11.30	60.00
CSRNet [11]	2018 CVPR	8.00	11.78	443.72	1252.22	34.10	46.42	240.01	394.81	5.36	8.26	42.99
SANet [42]	2018 ECCV	29.01	32.96	497.22	1276.66	62.78	79.65	302.37	436.91	6.25	11.16	35.20
SFCN [43]	2019 CVPR	8.94	12.87	440.70	1248.27	33.93	49.74	240.16	394.81	5.66	9.27	15.54
SPN [44]	2019 WACV	7.74	11.48	445.16	1252.92	36.21	50.65	241.43	392.88	5.20	9.02	33.64
SCAR [45]	2019 NC	26.90	31.35	497.22	1276.65	62.78	79.64	302.37	436.92	4.73	8.1	37.73
CAN [46]	2019 CVPR	9.12	13.38	457.36	1260.39	34.56	49.63	282.69	423.44	5.72	9.21	33.37
ASPDNet [9], [10]	2020 ICASSP/TGRS	7.59	10.66	433.23	1238.61	18.76	31.06	193.83	218.95	4.70	7.84	17.93
MCFA [47]	2021 TGRS	7.93	11.82	238.46	625.90	12.94	20.25	50.45	65.24	-	-	-
ADMAL (ours)	-	5.55	7.73	115.61	210.77	11.68	17.34	45.07	64.78	4.45	7.42	38.25

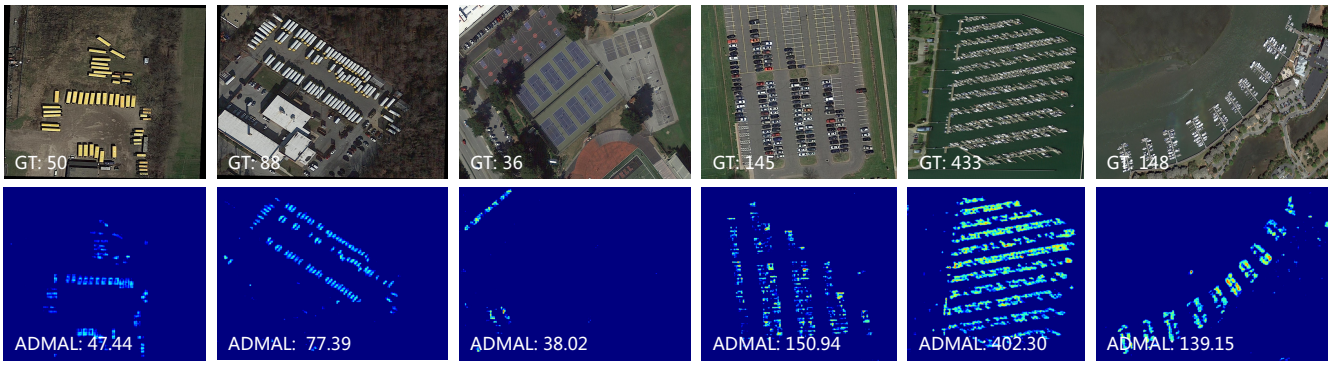

Fig. 7. Visualization results of ADMAL on RSOC data set. The first row is the testing image. The second row is the density map predicted by ADMAL.

TABLE IV
QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE CARPK DATA SET.

Method	Year & Venue	CARPK	
		MAE	RMSE
MCNN [22]	2016 CVPR	10.19	14.18
CSRNet [11]	2018 CVPR	7.80	9.76
GAP [51]	2018 arXiv	7.88	9.30
GSP [52]	2019 CVPRW	5.46	-
UFCNN [53]	2019 TIE	5.42	7.38
ASPDNet [9], [10]	2020 ICASSP/TGRS	7.81	10.16
ADMAL (ours)	-	5.12	7.05

B. *Satellite-based object counting*

ADMAL is compared with the state-of-the-art algorithms on the RSOC and Tree data sets. To better verify the effectiveness of ADMAL, performances of some crowd counting methods are also presented for comparison and their results are quoted from ASPDNet [10]. As shown in table III, ADMAL achieves the best performance on all of the four subsets in RSOC, in terms of both MAE and RMSE. At the same time, our algorithm has the fastest inference speed among the top three algorithms in terms of counting performance. Some qualitative results are visualized in Fig. 7 and Fig. 8. These quantitative and qualitative results demonstrate that ADMAL has strong object counting performance on satellite-based data sets.

TABLE V
ABLATION STUDY ON RSOC DATA SET.

DMG variations	DME variations	Building		Small-Vehicle		Large-Vehicle		Ship	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
$G_{\sigma_i}(\cdot)=5$	TDME	5.85	8.51	127.52	294.85	13.52	18.87	51.67	73.28
$G_{\sigma_i}(\cdot)=15$	TDME	5.84	8.39	128.10	256.81	12.31	17.83	59.80	77.74
$G_{\sigma_i}(\cdot)=25$	TDME	5.84	8.31	134.31	294.84	12.28	18.80	60.40	79.96
CADMG	VGG-16 backbone	6.11	9.26	232.04	494.20	15.85	25.43	53.27	72.83
CADMG	ResNet-50 backbone	8.88	13.14	257.18	735.72	12.32	18.34	76.80	113.48
CADMG	MCNN backbone	9.12	13.72	345.78	1080.35	33.45	53.51	104.32	131.12
CADMG	CSRNet backbone	5.84	9.19	220.94	491.39	13.86	19.59	57.92	87.71
CADMG w/o $\mathcal{N}_1$	TDME	5.58	8.02	132.03	244.87	11.78	17.86	52.64	71.75
CADMG w/o $\mathcal{L}_{gdd}$	TDME	5.70	7.93	126.30	225.64	12.01	17.76	49.71	69.61
CADMG	TDME w/o ECA	5.62	7.92	139.18	281.16	12.29	17.69	49.87	68.97
CADMG	TDME	5.55	7.73	115.61	210.77	11.68	17.34	45.07	64.78

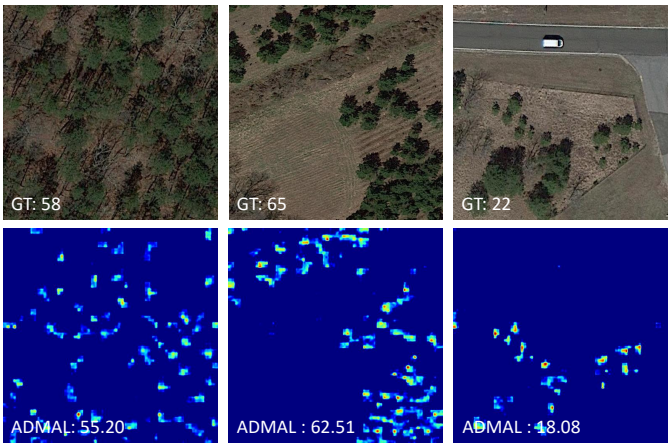

Fig. 8. Visualization results of ADMAL on Tree data set. The first row is the testing image. The second row is the density map predicted by ADMAL.

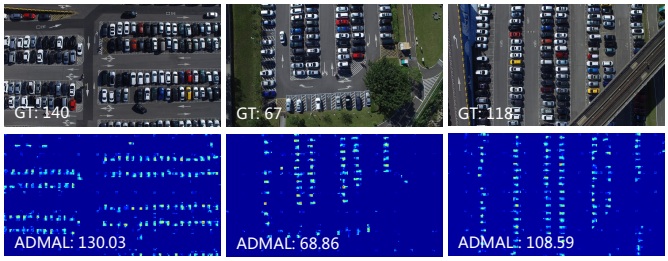

Fig. 9. Visualization results of ADMAL on CARPK data set. The first row is the testing image. The second row is the density map predicted by ADMAL.

C. Drone-based object counting

To further validate the performance of ADMAL, MAE and RMSE are evaluated on the drone-based datasets CARPK. ADMAL is compared with the state-of-the-art methods including crowd counting methods (MCNN [22], CSRNet [11]), car counting methods (GAP [51], GSP [52], UFCNN [53]) and a remote sensing object counting method (ASPDNet [9], [10]).

The results shown in Table IV demonstrate that ADMAL outperforms comparing methods on the CARPK dataset. The images sampled from CARPK are visualized in Fig. 9. It also demonstrates that ADMAL has good counting performance and strong localization ability.

V. ABLATION OF ADMAL

In this section, ablation experiments are conducted and experimental results are discussed to evaluate contributions of CADMG, TDME, and their different sub-modules.

A. Effectiveness of CADMG

In order to prove the effectiveness of the CADMG network, the results of three other versions of TDME networks (with the same network structure and initial status but trained by different ground truth density maps) were compared. Experimental results are listed in the first three rows of Table V. The performances of ADMAL are listed in the last row of the table. The results in Table V first imply that the optimal kernel size of Gaussian convolution varies with tasks. For example, kernel size of $\sigma_i = 5$ generates the best performances for ship counting. But for building counting, $\sigma_i = 25$ is preferable. Besides, ADMAL achieves the best performance in terms of both MAE and RMSE. This indicates that the ground truth density maps generated by CADMG are more suitable for the training of TDME.

The ground truth density maps generated by Gaussian kernel of $\sigma = 25$ and CADMG are drawn in Fig. 11. Although the density map generated by fix-sized Gaussian kernel looks more consistent and neat, it does not convey any contexture information for the objects. As shown in the figure, the density map generated by CADMG does not present the shape of Gaussian. It exhibits a smoother connection between adjacent objects. For pairs with special contextual structures, as the ship pairs beside piers, their densities tend to merge into whole pieces. For isolated objects, their peak responses in the generated density map are affected by their distances to neighboring objects as well as the number of near by objects. In addition, because the contextual information are considered, the density of the isolated object is smaller than the density of grouped objects. These phenomenons show that the CADMG network can adaptively combine location, neighborhood relationships, and contextual information of

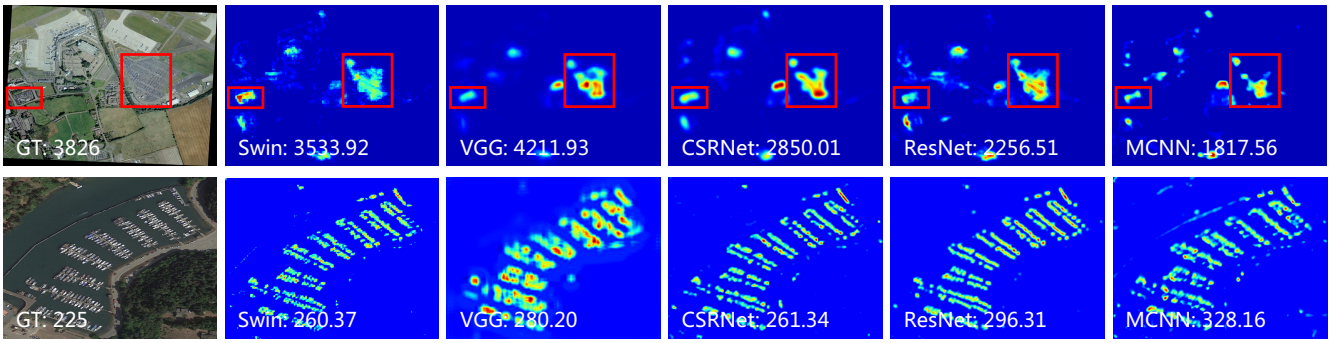

Fig. 10. Visualization results of ADMAL with different DME backbones on the RSOC dataset. The first column is the testing image. The following columns are the predicted density maps of DMEs with different backbones.

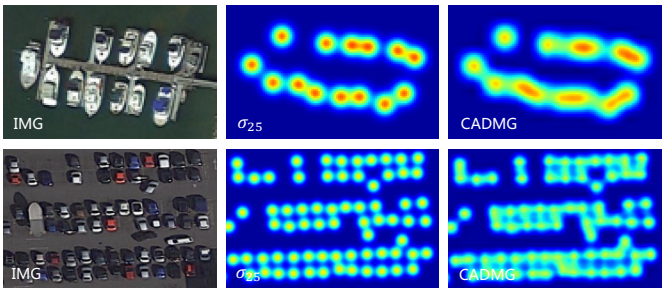

Fig. 11. Visualization of the generated ground truth density maps. The two image patches are cropped from the RSOC training set. The first column is the training image. The second column is the density map generated by Gaussian kernel size of $\sigma = 25$. The third column is the density map generated by CADMG.

objects into the generated density maps, which can provide a smoother learning process and promote the learning effect of the TDME network.

B. Contribution of $\mathcal{N}_1$ and $\mathcal{L}_{gdd}$ in CADMG

To verify the contribution of the feature extraction network $\mathcal{N}_1$ and the proposed $\mathcal{L}_{gdd}$ loss in the CADMG network, the following two experiments are designed: a) CADMG w/o $\mathcal{N}_1$: The $\mathcal{N}_1$ network is removed from the CADMG network and the training image is directly concatenated with the point map for encoding. As the output of $\mathcal{N}_1$ has a different number of channels compared with the training image, the number of input channels for the first convolution layer in $\mathcal{N}_2$ is changed accordingly. b) CADMG w/o $\mathcal{L}_{gdd}$: The density distribution loss $\mathcal{L}_{gdd}$ is removed from the total loss of CADMG during the whole training process.

Quantitative results are shown in the 8-9 rows of Table V. As shown in the table, both $\mathcal{N}_1$ and the $\mathcal{L}_{gdd}$ loss can help the ADMAL achieve better results on the RSOC data set. It can be explained as that the $\mathcal{N}_1$ sub-network can extract deep features from the image and provide $\mathcal{N}_2$ richer representative information for ground truth density map generation. The $\mathcal{L}_{gdd}$ loss can enforce the local peaks in the density map generated by CADMG to arise near the annotated points, so as to better reflect the actual density distribution of the objects. From this point of view, $\mathcal{L}_{gdd}$ can help reduce the influence of non-object features in the image.

C. Different network backbones for DME

In order to prove the effectiveness of using Swin as the backbone of DME, DMEs with other backbones are tested for ablation study. There are two types of networks tested in the experiments. The first type is the general CNN-based networks, represented by VGG16 [39] and ResNet50 [54]. The second type is the networks proposed for object counting, represented by MCNN [22] and CSRNet [11].

As shown in the 4-5 and 6-7 rows of Table V, Swin backbone outperforms other backbones, and on the subset of RSOC_Small-Vehicle, the performance improvements are impressive. These are attributed to the powerful representation capabilities of the larger receptive field in Swin transformer structure. It is thus capable of handling situations with complex background and object textures, as in the RSOC_Small-Vehicle subset.

Fig. 10 gives a visualized comparison of Swin and other networks serving as the backbone of DME. The areas with complex backgrounds and crowded objects are marked with red rectangles. It can be observed that DME with the Swin backbone obtains more accurate density maps than other networks. These results prove that the transformer structure can help the DME network generate more accurate density maps in complex scenes.

D. Contribution of ECA in TDME

The TDME network without the ECA module is tested to evaluate the contribution of the ECA module. As shown in the tenth row of Table V, the performance of ADMAL experiences a significant improvement after adding the ECA module, especially in the RSOC_Small-Vehicle subset. This is because the features of the images in RSOC_Small-Vehicle are complex. The ECA module can thus promote the performances by paying more attention to prominent channels.

VI. Conclusion

In this paper, we have presented a novel Adaptive Density Map Assisted Learning algorithm named ADMAL for object counting in remote sensing images. Different from previous object counting algorithms, ADMAL focuses on generating adaptive density maps for better learning of the regression model. Its Contexture Aware Density Map Generation (CADMG) network works together with its Transformer-based Density Map Estimation (TDME) network to generate

learning oriented density maps as well as construct a more representative regression model for density map estimation. The Swin transformer used in the TDME network strikes a balance between the size of the receptive field and the computational complexity. It helps ADMAL achieve better performance with acceptable complexity. The performance of ADMAL is evaluated on three public data sets, and the proposed method presents competitive performance to other state-of-the-art methods.

REFERENCES

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

[2] D. Wang, D. Zhang, G. Yang, B. Xu, Y. Luo, and X. Yang, "SSRNet: In-Field Counting Wheat Ears Using Multi-Stage Convolutional Neural Network," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[3] H. Lu, L. Liu, Y.-N. Li, X.-M. Zhao, X.-Q. Wang, and Z.-G. Cao, "TasselNetV3: Explainable Plant Counting With Guided Upsampling and Background Suppression," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[4] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang, "Data-Driven Crowd Understanding: A Baseline for a Large-Scale Crowd Dataset," *IEEE Transactions on Multimedia*, vol. 18, no. 6, p. 1048–1061, jun 2016.

[5] D. Kang, Z. Ma, and A. B. Chan, "Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1408–1422, 2018.

[6] J. Wan and A. Chan, "Adaptive Density Map Generation for Crowd Counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2019, pp. 1130–1139.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 764–773.

[8] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[9] G. Gao, Q. Liu, and Y. Wang, "Counting Dense Objects in Remote Sensing Images," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 4137–4141.

[10] ——, "Counting From Sky: A Large-Scale Data Set for Remote Sensing Object Counting and a Benchmark Method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 3642–3655, 2020.

[11] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.

[12] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2021, pp. 10 012–10 022.

[13] J. Huang, G. Ding, Y. Guo, D. Yang, S. Wang, T. Wang, and Y. Zhang, "Drone-Based Car Counting via Density Map Learning," in *2020 IEEE International Conference on Visual Communications and Image Processing*. IEEE, 2020, pp. 239–242.

[14] T. Moranduzzo and F. Melgani, "Automatic Car Counting Method for Unmanned Aerial Vehicle Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1635–1647, 2013.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016.

[17] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[18] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[19] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature Mining for Localised Crowd Counting," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2012, pp. 21.1–21.11.

[20] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep People Counting in Extremely Dense Crowds," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015, pp. 1299–1302.

[21] V. Lempitsky and A. Zisserman, "Learning To Count Objects in Images," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1324–1332, 2010.

[22] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[23] D. Babu Sam, S. Surya, and R. Venkatesh Babu, "Switching Convolutional Neural Network for Crowd Counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5744–5752.

[24] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd Counting Using Scale-Aware Attention Networks," in *2019 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2019, pp. 1280–1288.

[25] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "ADCrowdNet: An Attention-Injective Deformable Convolutional Network for Crowd Understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3225–3234.

[26] D. Guo, K. Li, Z.-J. Zha, and M. Wang, "DADNet: Dilated-Attention-Deformable ConvNet for Crowd Counting," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1823–1832.

[27] 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 *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 785–800.

[28] C. YuanQiang, D. Du, L. Zhang, L. Wen, W. Wang, Y. Wu, and S. Lyu, "Guided Attention Network for Object Detection and Counting on Drones," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 709–717.

[29] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-Based Object Counting by Spatially Regularized Regional Proposal Network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 4145–4153.

[30] W. Xu, D. Liang, Y. Zheng, J. Xie, and Z. Ma, "Dilated-Scale-Aware Category-Attention ConvNet for Multi-Class Object Counting," *IEEE Signal Processing Letters*, vol. 28, pp. 1570–1574, 2021.

[31] G. Gao, Q. Liu, L. Li, Q. Wen, and Y. Wang, "PSGCNet: A Pyramidal Scale and Global Context Guided Network for Dense Object Counting in Remote Sensing Images," *arXiv preprint arXiv:2012.03597*, 2020.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[34] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[35] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[36] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *International Conference on Learning Representations*, 2021.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[38] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking Semantic Segmentation From a Sequence-to-Sequence Perspective With Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6881–6890.

[39] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[40] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.
[41] V. A. Sindagi and V. M. Patel, "CNN-Based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2017, pp. 1–6.
[42] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale Aggregation Network for Accurate and Efficient Crowd Counting," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.
[43] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning From Synthetic Data for Crowd Counting in the Wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8198–8207.
[44] X. Chen, Y. Bin, N. Sang, and C. Gao, "Scale Pyramid Network for Crowd Counting," in *2019 IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2019, pp. 1941–1950.
[45] J. Gao, Q. Wang, and Y. Yuan, "SCAR: Spatial-/channel-wise attention regression networks for crowd counting," *Neurocomputing*, vol. 363, pp. 1–8, 2019.
[46] W. Liu, M. Salzmann, and P. Fua, "Context-Aware Crowd Counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.
[47] Z. Duan, S. Wang, H. Di, and J. Deng, "Distillation Remote Sensing Object Counting via Multi-scale Context Feature Aggregation," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8026–8037, 2019.
[49] J. Gao, W. Lin, B. Zhao, D. Wang, C. Gao, and J. Wen, "C^3 Framework: An Open-source PyTorch Code for Crowd Counting," *arXiv preprint arXiv:1907.02724*, 2019.
[50] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[51] S. Aich and I. Stavness, "Improving Object Counting with Heatmap Regulation," *arXiv preprint arXiv:1803.05494*, 2018.
[52] ——, "Global Sum Pooling: A Generalization Trick for Object Counting with Small Datasets of Large Images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.
[53] 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*, vol. 66, no. 12, pp. 9651–9662, 2019.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.