DOPNet: Dense Object Prediction Network for Multi-Class Object Counting and Localization in Remote Sensing Images

Mingpeng Cui, Guanchen Ding, *Student Member, IEEE,* Daiqin Yang, *Member, IEEE,* and Zhenzhong Chen, *Senior Member, IEEE*

***Abstract*—Object counting and localization for remote sensing images are effective means to solve large-scale object analysis problems. Nowadays, most counting methods obtain the number of objects by employing convolutional neural network to regress a density map of objects. Even if these leading methods have achieved impressive performances, they simply focus on estimating the number of single-class objects, without providing location information and cannot support multi-class objects. To tackle these problems, a point-based network named Dense Object Prediction Network (DOPNet) is proposed for multi-class object counting and localization for remote sensing images. DOPNet differs from the conventional approach of predicting multiple density maps by incorporating category attributes into the predicted objects, enabling the accurate counting and localization of multi-class objects. Specifically, DOPNet adopts a multi-scale architecture to provide dense predictions of object proposals. A Scale Adaptive Feature Enhancement Module (SAFEM) is designed to predict scales of objects for the suppression of duplicate proposals. Given only point level annotations for training, a pseudo box generation algorithm is designed to find the most suitable pseudo box of each annotated object for the supervision of scale learning. Comprehensive experiments prove that DOPNet can achieve preferable performance on challenging benchmarks of counting while providing object locations. Code and pre-trained models are available at https://github.com/Ceoilmp/DOPNet.**

***Index Terms*—Dense object prediction network (DOPNet), multi-class object counting, localization, remote sensing, point-based network**

I. Introduction

COMPARED with other data acquisition technologies, remote sensing technology can observe large-scale areas from sky or space in a short period of time, and the observation area can reach tens of thousands of square kilometers. Based on this, the use of remote sensing imagery for object counting and object localization provides an effective solution to macroscopic and large-scale object analysis. Specifically, object counting and object localization are attractive and challenging vision tasks. They are important topics for advanced object analysis and have lots of practical applications, including video surveillance [1], safety monitoring [2], traffic analysis [3], and behavior modeling [4], [5]. Recently, with the development of deep learning, most state-of-the-art object counting methods estimate the number of objects by regressing a density map [6]–[8] of objects. These methods do not require annotated bounding box of object for training, and by using point-level annotations, they can estimate the number of single-class objects.

Although significant progress has been achieved, these density map based methods have some inherent drawbacks that need to be addressed. On the one hand, they only estimate the density map of objects and can not provide locations of the objects directly. In remote sensing image applications, locations of objects are important for many downstream applications, such as object tracking, object re-identification, and so on. On the other hand, most existing density map based methods are designed for the counting of single-class objects. In real-world scenarios, especially in remote sensing images, there are often multiple categories of objects that need to be counted simultaneously. Compared to single-class object counting methods, multi-class object counting methods require only one training process and can output the number of multi-class objects simultaneously, saving computational costs and speeding up inference.

Recently, there have been some related studies proposed for the above two issues [8]–[13]. To solve the problem of counting multi-class objects, the density map based method in [8] attempts to output a separate density map for each class of objects. Meanwhile, some researchers start to count and localize objects by generating point-level estimations. Specifically, some methods [9], [10] try to turn the counting problem into detection problem. They first use point-level annotations to generate pseudo bounding boxes, and later adopt a special training strategy to refine the bounding boxes and obtain the counting and localization results. Some other methods [11], [12] try to represent objects through blobs. They use specific loss function and architecture to output the blobs containing objects. Lately P2PNet [13] directly represents the object as a point to complete the counting task, without turning the counting problem into a detection problem. However, up to now there are few studies for counting and locating multi-class objects simultaneously.

To count and localize multi-class objects simultaneously, we propose a Dense Object Prediction Network (DOPNet) for remote sensing images. Inspired by [13], DOPNet is trained

This work was supported in part by National Natural Science Foundation of China under contract No. 62036005 and the Special Fund of Hubei Luojia Laboratory.

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

Z. Chen is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China and also with the Hubei Luojia Laboratory, Wuhan, China (e-mail: zzchen@whu.edu.cn).

Corresponding author: Daiqin Yang.

with point-level annotations and designed as a point-based network that generates estimation of objects directly. Total numbers, categories, and locations of objects are estimated from densely generated object proposals which are composed of four factors, including location, category, scale information, and a confidence score. Compared to natural images, objects in remote sensing images have distinct features, including large scale variations, significant aspect ratios, arbitrary orientations, and consistent background information [6], [14]. To accommodate the scale variation of objects in remote sensing images, DOPNet employs a multi-scale architecture for dense object proposal generation. However, without object scale annotations, it is difficult for the network to learn scale information for accurate suppression of duplicate proposals. To solve the problem, a Scale Adaptive Feature Enhancement Module (SAFEM) is designed to predict scale information for each object proposal, and an appropriate pseudo box generation algorithm is designed to find the most suitable pseudo box of object for the supervision of scale learning. SAFEM utilizes deformable convolutions [15] to extract feature information from semantic skeleton points of objects, thereby achieving high-quality feature extraction with consistent backgrounds, which enhances counting performance. Considering the large aspect ratio and arbitrary orientation of objects in remote sensing images, the pseudo-box generation algorithm uses positive and negative object proposals during the training process to predict the most suitable pseudo-box scales and orientations. Based on the above structures, DOPNet can achieve preferable performance in multi-class object counting as well as providing locations of objects.

The major contributions of this work are three-fold:

- DOPNet, a novel point-based network, is introduced for multi-class object counting and localization in remote sensing images. It can simultaneously predict object positions, categories, and scales, leading to enhanced object counting performance.
- A scale prediction module, SAFEM, is designed to extract critical semantic information, enhance object features, and generate scale predictions for target objects. A pseudo-box generation algorithm is also designed to provide essential supervision for the training of SAFEM with only point-level annotations.
- Comprehensive experiments prove that DOPNet can achieve excellent counting performance on satellite remote sensing images, making it well-suited for both counting and localization tasks.

The rest of the paper is organized as follows. The related work of object counting algorithms is briefly surveyed in Section II. The details of our proposed method are introduced in Section III. The experimental results and analysis are presented in Section IV. Finally, the conclusion is concluded in Section V.

II. Related work

In this section, recent object counting methods are reviewed. Notable, the multi-class object counting and the remote sensing object counting are discussed in Subsection II-B.

A. Detection based methods

1) Object detection: With the prosperity of Convolutional Neural Network (CNN), object detection networks [16]–[20] are widely used in object counting. Specifically, [21] proposed a network to get the number of objects by decoding an image into a set of people detections. DecideNet [22] generates the detection results and density maps separately. Then, it obtains the final counting result by adaptively assessing the reliabilities of the two types of estimations. However, these methods require detailed bounding box annotations and are difficult to effectively deal with the challenges faced for geospatial object analysis.

2) Remote sensing object detection: Object detection in remote sensing images has been a fundamental problem in the field of aerial and satellite image analysis [14]. Li *et al.* [14] proposed a double-channel feature fusion network that can learn local and contextual features along two independent pathway. Wang *et al.* [23] proposed a unified framework, which aggregates the context information both in multiple scales and the same scale feature maps. Cheng *et al.* [24] proposed a novel feature enhancement network for object detection in optical remote sensing images, which can capture global context cues and selectively strengthen class-aware features. In addition, there are some researches about anchor-free detector. For example, Cheng *et al.* [25] proposed a novel Anchor-free Oriented Proposal Generator, which can produce coarse oriented boxes and refine them into high quality oriented proposals.

B. Density map based methods

1) Object counting: Since [26] first introduced the method of predicting density map into object counting, density map based methods have gradually become the mainstream methods for object counting and have achieved impressive progress. These methods first convert the point annotations into a density map through a Gaussian kernel and then learn the generated density map. The final counting results are obtained by summing over the density map predicted by the network. Specifically, MCNN [27] proposed a simple but effective multi-column network, which utilizes filters with different receptive fields to adaptively process the image patches with different object densities. Switch-CNN [28] deliver the crowd scene patches to the best CNN regressor through a switch classifier, which can leverage the variation of crowd density within an image. CSRNet [29] replaces the pooling operations with the dilated kernels to enlarge the receptive fields. Therefore, the network can understand highly congested scenes and perform accurate counting estimation. SAAN [30] uses the attention mechanism to automatically focus on global and local scales. The network combines these global and local scale attention to achieve great performance. SASNet [7] automatically learns the internal correspondence between the scales and the feature levels and uses the Pyramid Region Awareness Loss (PRA Loss) to calculate the loss of the hardest sub-regions. MAN [31] incorporates global attention from vanilla transformer, learnable local attention, and instance attention into a counting

model. Although many advanced networks have been proposed, these networks are still difficult to implement for object localization and multi-class object counting.

2) Remote sensing object counting: In the past few years, there are some researches on object counting in remote sensing images. Mundhenk *et al.* [32] created a large diverse set of cars from overhead images and proposed a network that combines residual learning with Inception-style layers to count cars. GANet [33] uses weakly-supervised Background Attention (BA) between the background and objects to fuse different scales of feature maps and considers both the global and local appearance of the object to facilitate accurate localization. Hsieh *et al.* [34] presented a new large-scale car parking lot dataset (CARPK) and proposed LPNs for counting and localization simultaneously. Gao *et al.* [6] constructed a large-scale remote sensing object counting data set RSOC and proposed ASPDNet to attack the challenges of object counting in remote sensing images. Duan *et al.* [35] proposed MCFA that can effectively fuse context information from different receptive fields and improve representation learning without adding any additional supervision information. PSGCNet [36] combines Pyramid Scale Module (PSM) and Global Context Module (GCM) to adaptively capture the multi-scale information and select suitable scales in remote sensing images. Instead of learning the density map generated through a fixed Gaussian kernel, ADMAL [37] learns the density map generated based on the spatial features of objects and the performance achieves great improvement. Although these methods have achieved great performance, most of these methods focus on single-class object counting and are powerless on multi-class object counting and localization.

3) Multi-class object counting: Recently, DSACA [8] learns multiple density maps instead of a single density map to achieve multi-class object counting. DSACA uses the Dilated-Scale-Aware Module to capture the multi-scale information and the Category-Attention Module to generate the discriminative density maps. MOCSE [38] has established a synthetic data set and proposed a benchmark for multi-class object counting and scale estimation within a unified framework. LMCNet [39] introduces a lightweight multi-class counting network that employs the Ghost attention mechanism to obtain high-quality multi-channel density maps under low computational efficiency. Additionally, LMCNet [39] proposes the Focal-L2 loss function, mitigating issues associated with reduced counting network performance due to class imbalance. Nevertheless, these methods are still unable to directly complete the object localization.

C. Point-based methods

These methods usually obtain the number of objects by locating the objects first. In this way, point-based methods are inherently able to solve the localization problem that is difficult for the density map based methods. Specifically, PSDDN [9] mines the useful object scale information contained in the point-level annotations to initialize the pseudo bounding boxes. Then the network refines the pseudo bounding boxes during training. In this way, the counting problem is transformed into the detection-like problem. LC-FCN [11] designs a novel loss function that encourages the network to output a single blob for each object instance. In a similar way, Mohsen Zand *et al.* [12] proposed a novel multi-scale and multi-task architecture to output the blobs representing the objects in the scenes. Recently, Song *et al.* [13] innovatively proposed a purely point-based framework for object localization and object counting. In this framework, the Hungarian algorithm [40] is used to achieve the label assignment. Moreover, an end-to-end Crowd Localization TRansformer (CLTR) [41] views the object localization as a direct set prediction problem, taking extracted features and trainable embeddings as input of the transformer-decoder.

In general, point-based object counting methods remedy the shortcomings of density map based methods to a certain extent, but there are still few studies on multi-class object counting under the point-based framework.

III. METHODOLOGY

In this section, steps of object prediction with DOPNet will be introduced first. Then the detailed architectures of DOPNet are presented. After that, the training procedure of DOPNet will be depicted.

A. Object prediction with DOPNet

As shown in Fig. 1, the pipeline of our framework is composed of two procedures: the DOPNet and the Post-Processing. DOPNet consists of three main components, i.e., the backbone, the Scale Adaptive Feature Enhancement Module (SAFEM), and the Prediction Head. The output of DOPNet are three parallel feature maps denoted as R_d, where $d \in \{4, 8, 16\}$ represents the extent of downscaling. After that, a Non-Maximum Suppression (NMS) is performed on the output of the network to get the final results of counting and localization.

1) The output of DOPNet: For each input image I, DOPNet will generate three parallel feature maps, the resolution of which are $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ that of I respectively. Denoting these three feature maps as R_d, where $d \in \{4, 8, 16\}$ represents the extent of downscaling, each pixel in R_d can be mapped to a related tile of the input image I. As illustrated in Fig. 2, when there are N_{class} object categories to be identified, each pixel in R_d is a $(1 + 2 + N_{class} + 18)$ dimension vector, representing four factors of the object proposal in this related tile of the image. The first factor is a 1 dimension "object confidence" $conf$, indicating the possibility of whether does exist an object in this tile. The second factor is a 2 dimension "location offset" $(\Delta x, \Delta y)$ of the object proposal, depicting the coordinate offsets of the object proposal in this tile. The third factor is a N_{class} dimension one-hot vector cls, indicating the "category" of the object proposal. And the last factor represents the scale of the object proposal. It has 18 dimensions, including the coordinate offsets of nine skeleton points of the object proposal. Suppose there are M object proposals generated from R_d, the j-th object proposal can be denoted as $\hat{p}_j = \left(\hat{x}_j, \hat{y}_j, \hat{cls}_j, \hat{conf}_j, \hat{scale}_j\right), j \in \{1, \ldots, M\}$, in which $(\hat{x}_j, \hat{y}_j)$ is the location of the object proposal, $\hat{cls}_j$ is the category of the object proposal, $\hat{conf}_j$ is

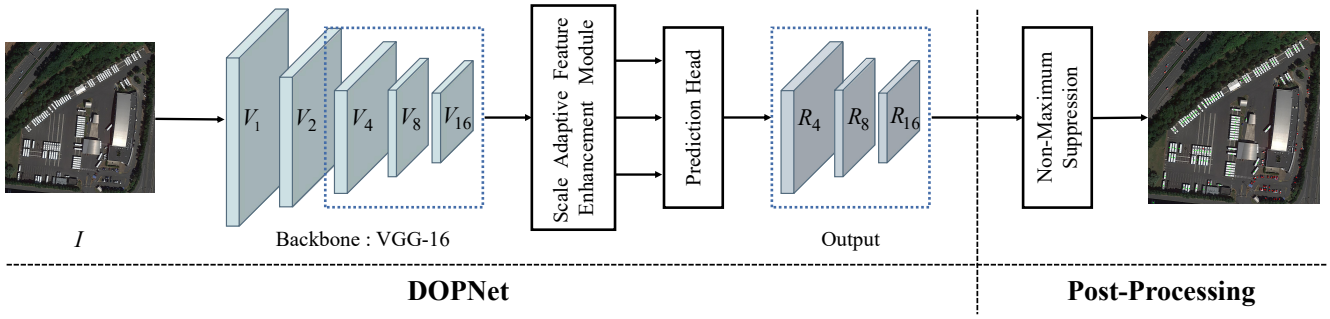

Fig. 1. The pipeline of our framework. For DOPNet, there are three main components: the backbone, the Scale Adaptive Feature Enhancement Module (SAFEM), and the Prediction Head. The output of DOPNet are three parallel feature maps and they are denoted as R_d, $d \in \{4, 8, 16\}$. In addition, Non-Maximum Suppression is performed to get the final results of counting and localization.

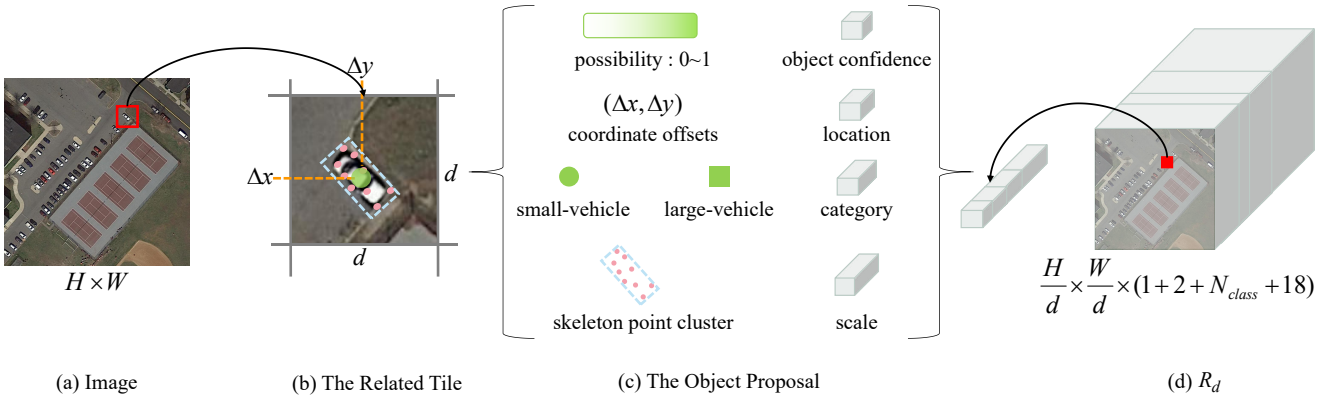

Fig. 2. Illustration for the output of the DOPNet. For each pixel in R_d, it represents an object proposal that is composed of four factors: the location, the object confidence score, the category, and the scale. Each pixel in R_d can be mapped to a related tile of the input image I. The location of the object proposal is strictly restricted to the related tile.

the object confidence score, and $\hat{scale}_j$ is the scale information of the object proposal. The set of object proposals is denoted as $\hat{\mathcal{P}} = \{\hat{p}_j \mid j \in \{1, .., M\}\}$.

2) Non-Maximum Suppression for getting the final predictions: In DOPNet, since an object could match one or more object proposals in each of the feature map R_d, $d \in \{4, 8, 16\}$, Non-Maximum Suppression (NMS) is performed to remove shadow object proposals from all the object proposals. Before performing NMS, the Minimum External Rotation Rectangle (MERR) of each object proposal is calculated with its nine skeleton points. During NMS, the object proposal with the highest object confidence score is first selected and taken out from the proposal list. Then, nearby object proposals having larger MERR Intersection Over Union (IOU) with the selected object proposal are suppressed and deleted from the list. By repeating the above two steps, the final predictions of multi-class object counting and localization can be obtained.

B. The architecture of DOPNet

As shown in Fig. 1, DOPNet consists of three main components. For the backbone network, the first 13 convolutional layers in VGG-16 [42] are used to extract features of images. In order to adapt to the large-scale variation of objects in remote sensing images and generate enough object proposals, three parallel feature levels with downsampling strides of 4, 8, and 16 are adopted for predicting object proposals. These feature layers are denoted as V_4, V_8, and V_{16} respectively. Next, through the process of SAFEM, deep features and shallow features are fused and enhanced. In addition, the coarse representations of object scale are predicted. Then, the Prediction Head will integrate the enhanced features and output the final results.

1) Scale Adaptive Feature Enhancement Module: Scale Adaptive Feature Enhancement Module (SAFEM) is designed to fuse and enhance features. In addition, it will attach the scale information to each object proposal. [43] and [44] have validated that a skeleton point cluster can be a fine-grained representation of the object scale information. Therefore, the structure of RepPoints [43] is adopted to generate the skeleton point cluster for each object proposal and extract key features of objects in the image. Generally speaking, a skeleton point cluster consisting of nine skeleton points is used to describe the scale information of the object proposal. For each skeleton point in the skeleton point cluster, two values of x and y are predicted to indicate the location offset of the skeleton point. Therefore, a total of 18 values will be predicted for each object proposal. Through the training of the network, the nine skeleton points spontaneously shift to semantically key locations of the object, and with the structure of RepPoints [43], key features of objects in the image can be extracted.

Specifically, as illustrated in Fig. 3, deep features and shallow features are fused through an FPN-like [45] structure. Next

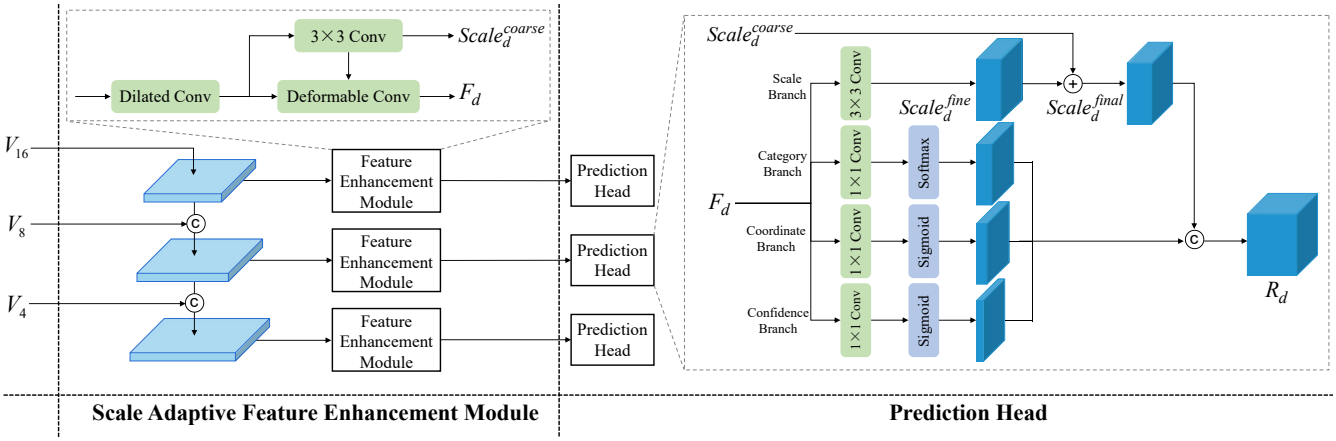

Fig. 3. The architectures of the Scale Adaptive Feature Enhancement Module and the Prediction Head.

the fused features are processed by a Feature Enhancement Module (FEM). In FEM, a dilated convolution [46] is first used to enlarge the receptive field. Then FEM generates a feature map with 18 channels through a 3×3 convolution. The feature map with 18 channels denoted as $Scale_d^{coarse}$ represents the skeleton point clusters that are used to coarsely represent the scale information of object proposals. Finally, $Scale_d^{coarse}$ is input as the offset into a deformable convolution [15]. By using the deformable convolution [15], features are further processed and enhanced. The further processed feature maps are denoted as F_d, $d \in \{4, 8, 16\}$. The final outputs of SAFEM consists of two parts: $Scale_d^{coarse}$, which coarsely represents the scale information of object proposals and the fused and enhanced feature map F_d.

2) The Prediction Head of DOPNet: As illustrated in Fig. 3, based on F_d, four branches are designed to refine the scale representation, classify the object proposals, regress the coordinates of the object proposal, and predict the object confidence score. A lightweight architecture is adopted to build the Prediction Head. For the scale representation branch, a 3×3 convolution is used to process F_d and a feature map with 18 channels denoted as $Scale_d^{fine}$ will be obtained to represent the fine scale information. For the classification branch, it outputs the class information of the object proposals with a softmax normalization. The coordinate branch is designed to predict the coordinate offsets of the object proposals in their related tiles. In the coordinate branch, a 1×1 convolution with a sigmoid activation function is used to process F_d and it will output a feature map with 2 channels. For the object confidence branch, it outputs the object confidence scores with a sigmoid normalization. Note that DOPNet generates two skeleton point cluster to represent the object scale information: $Scale_d^{coarse}$ and $Scale_d^{fine}$. The first skeleton point cluster $Scale_d^{coarse}$ is a coarse representation of the object scale information and the second skeleton point cluster $Scale_d^{fine}$ is the correction to $Scale_d^{coarse}$. $Scale_d^{coarse}$ and $Scale_d^{fine}$ will be added to get $Scale_d^{final}$ that is the final scale representation of object proposals. The final result of the Prediction Head is R_d that combines the results of four branches.

C. Training of DOPNet

In the training process, object proposals generated by DOPNet will be divided into positive samples that can match the object annotations and negative samples that can not match the object annotations. After that, losses are calculated to train DOPNet.

1) Label assignment strategy: For multi-class object counting, ground truth are locations of objects and categories of objects. Formally, for a given image I with N objects, the object annotation of the i-th object can be denoted as $p_i = (x_i, y_i, cls_i), i \in \{1, \ldots, N\}$, in which (x_i, y_i) is the location of the object annotation and cls_i is the category of the object annotation. The set of object annotations in the whole image is denoted as $\mathcal{P} = \{p_i \mid i \in \{1, \ldots, N\}\}$.

When object proposals $\hat{\mathcal{P}}$ are obtained, label assignment can be performed to assign positive samples to object annotations. Our label assignment strategy is divided into two steps: coarse screening and fine label assignment.

Step1: coarse screening: In this step, object proposals that are far away from the object annotations are set to negative samples. The distance threshold is denoted as ε. The goal of this step is to eliminate obvious negative samples and reduce computational burden for later fine label assignment. Since DOPNet generates object proposals at three downscaling resolutions, ε is set for each resolution respectively.

Through the above step, preliminary screening object proposals denote as $\hat{\mathcal{P}}_{ini}$ are obtained.

Step2: fine label assignment: Based on SimOTA [47], there are two steps to complete fine label assignment. Firstly, pair-wise matching cost for each proposal-annotation pair is calculated. The pairwise matching cost between $\hat{\mathcal{P}}_{ini}$ and $\mathcal{P}$ is denote as $\mathcal{D} \in \mathbb{R}^{M1\times N}$, where M_1 is the number of proposals in $\hat{\mathcal{P}}_{ini}$. Location distance, classification distance, and the object confidence score are combined to calculate $\mathcal{D}$. Formally, the cost matrix $\mathcal{D}$ is calculated as follows:

$$\mathcal{D}(\hat{\mathcal{P}}_{ini}, \mathcal{P}) = \Big(\mathcal{L}_{cls}(\hat{conf}_j \times \hat{p}_j, p_i) + \mathcal{L}_{loc}(\hat{p}_j, p_i)\Big)_{j\in M_1, i\in N} \tag{1}$$

where $\hat{conf}_j$ is the object confidence score of $\hat{p}_j$. $\mathcal{L}_{cls}(\hat{conf}_j \times \hat{p}_j, p_i)$ is the classification distance between

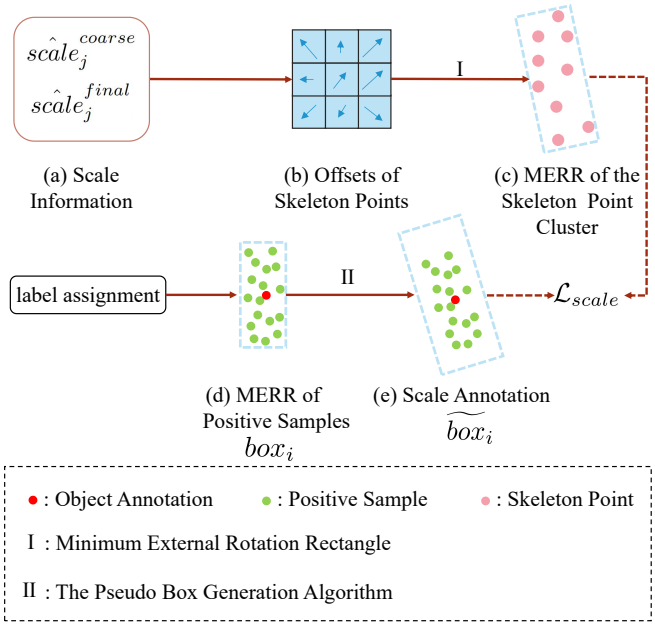

Fig. 4. The loss function for scale information. The MERR of the skeleton point cluster represents the scale prediction. The pseudo box generated from the pseudo box generation algorithm represents the scale annotation. The scale loss is calculated between the scale prediction and the scale annotation.

proposal $\hat{p}_j$ and annotation p_i and it is calculated by cross-entropy. $\mathcal{L}_{loc}(\hat{p}_j, p_i)$ is the location distance between proposal $\hat{p}_j$ and annotation p_i and it is calculated based on the Euclidean Distance.

For each object annotation p_i, its top k matching proposals with the least matching costs are assigned as its positive samples. The rest proposals are then assigned as negative samples. The value k is set through Dynamic k Estimation strategy [47]. Supposing there are totally K positive proposals, they can be denoted as $\hat{\mathcal{P}}_{pos} = \{\hat{p}_j \mid j \in \{1, \ldots, K\}\}$ and the negative proposals can be denoted as $\hat{\mathcal{P}}_{neg} = \{\hat{p}_j \mid j \in \{K+1, \ldots, M\}\}$. For each object proposal in $\hat{\mathcal{P}}_{pos}$, label assignment finds a specific object annotation corresponding to it. Formally, for positive proposal $\hat{p}_j \in \hat{\mathcal{P}}_{pos}$, the corresponding object annotation can be denoted as $p_{l(j)}$.

Note that for the positive proposals, location loss, classification loss, object confidence loss, and scale loss will be used for network training. But for negative proposals, only the object confidence loss will be used for network training.

2) Loss functions for location, classification and object confidence score: In DOPNet, the classification loss $\mathcal{L}_{cls}$ is calculated through cross-entropy loss and is defined as follows:

$$\mathcal{L}_{cls} = -\frac{1}{K}\sum_{j=1}^{K}\sum_{n=1}^{N_{class}} cls_{l(j),n} \ln \hat{cls}_{j,n} \tag{2}$$

where K is the number of positive proposals, N_{class} is the number of object categories contained in I, $cls_{l(j),n}$ is the n-th class label of $p_{l(j)}$, and $\hat{cls}_{j,n}$ is the n-th class label of $\hat{p}_j$.

The location loss $\mathcal{L}_{loc}$ is calculated based on Euclidean Distance and is defined as follows:

$$\mathcal{L}_{loc} = -\frac{1}{K}\sum_{j=1}^{K} \ln\left(\frac{2}{\pi}\arctan\frac{\varepsilon}{\sqrt{(\hat{x}_j - x_{l(j)})^2 + (\hat{y}_j - y_{l(j)})^2}}\right) \tag{3}$$

where K is the number of positive proposals, ε is a coefficient defined in Subsection III-C1.

For positive proposals, the ground truth for object confidence score is 1. For negative proposals, the ground truth for object confidence score is 0. The loss of object confidence score $\mathcal{L}_{conf}$ is defined as follows:

$$\mathcal{L}_{conf} = \mathcal{L}_{conf}^{pos} + \mathcal{L}_{conf}^{neg} \tag{4}$$

where $\mathcal{L}_{conf}^{pos}$ is the object confidence loss for positive samples, and $\mathcal{L}_{conf}^{neg}$ is the object confidence loss for negative samples. They are calculated as follows:

$$\mathcal{L}_{conf}^{pos} = \sqrt{\frac{1}{K}\sum_{j=1}^{K}(\hat{conf}_j - 1)^2} \tag{5}$$

$$\mathcal{L}_{conf}^{neg} = \sqrt{\frac{1}{M-K}\sum_{j=K+1}^{M}(\hat{conf}_j - 0)^2} \tag{6}$$

where K is the number of positive proposals, M is the number of object proposals.

3) Loss function for scale information: In DOPNet, the coarse scale loss and the final scale loss of the positive object proposal are calculated to supervise the prediction of scale information. The entire flow is illustrated in Fig. 4. $\hat{scale}_j^{coarse}$ and $\hat{scale}_j^{final}$ represent the coarse scale information and the final scale information of the positive object proposal $\hat{p}_j \in \hat{\mathcal{P}}_{pos}$ respectively.

Scale annotation: Since there is no scale information in the ground truth of counting task, pseudo box generation algorithm is designed to find a pseudo box for each object annotation representing its ground truth of scale information.

Specifically, oriented rectangles are used to represent scale annotations of objects. An oriented rectangle is composed of five factors: the x and y coordinates of the rectangle's center point, the width and height of the rectangle, and the rotation angle of the rectangle. Obviously, the center point of the rectangle is the location of the object annotation. So, the pseudo box generation algorithm just needs to find the scale and the rotation angle of the rectangle. The algorithm can be divided into two parts: finding the scale of the pseudo box and finding the rotation angle of the pseudo box. Given an annotated object p_i, the positive proposals assigned to p_i are denoted as $\hat{\mathcal{P}}_S^i$, the positive proposals not assigned to p_i are denoted as $\hat{\mathcal{P}}_N^i$. The set of the object annotations in the image is denoted as $\mathcal{P}$. The entire flow is described as follows:

Step1: obtain the basic pseudo box. Based on the label assignment strategy outlined in Subsection III-C1 (as depicted in the second image of Fig. 5), the positive proposals assigned to an object can partially reflect the object's scale. In the case of object annotation p_i, the center of its scale corresponds to the annotation's location. Consequently, a central symmetry operation is performed on $\hat{\mathcal{P}}_S^i$ with respect to p_i, resulting in

Fig. 5. Positive proposals assigned to the object annotations and the Minimum External Rotation Rectangle (MERR) of positive proposals. The first image displays the positive proposals in the initial epoch of the first training phase. The second image showcases the positive proposals in the final epoch of the first training phase. Lastly, the third image illustrates the pseudo scale annotations generated from the positive proposals depicted in the second image.

$\hat{\mathcal{P}}_S^{i'}$. Subsequently, the Minimum External Rotation Rectangle (MERR) encompassing $\hat{\mathcal{P}}_S^i$ and $\hat{\mathcal{P}}_S^{i'}$, referred to as box_i, is employed as the fundamental pseudo box, characterized by a rectangle with a center point at p_i.

Step2: scale the pseudo box with the pre-defined scaling factor $factor_{scale}$**.** Objects in remote sensing images frequently exhibit significant variations in aspect ratios. Due to the utilization of Euclidean distance for computing distances between object proposals and the corresponding objects in the label assignment strategy, it often becomes challenging to categorize object proposals located along the longer side of the object as positive proposals. As illustrated in the second image of Fig. 5, positive proposals assigned to large vehicles often encompass only the shorter side, leading to difficulties in predicting their scale along the longer side. To address this issue, the basic pseudo box box_i is scaled using a pre-defined scaling factor $factor_{scale}$.

Step3: rotate the scaled pseudo box with the pre-defined rotation factor $factor_{rotate}$**.** In remote sensing images, objects often exhibit arbitrary orientations. To acquire more precise object scale information, the scaled pseudo box undergoes rotation and fine-tuning utilizing a predefined rotation factor of $factor_{rotate}$.

Step4: find the rotation angle of the pseudo box. A well-oriented pseudo box should only contain $\hat{\mathcal{P}}_S^i$ and should not include $\hat{\mathcal{P}}_N^i$. Therefore, among pseudo boxes with the same scaling factor but different rotation angles, the box that contains the fewest instances of $\hat{\mathcal{P}}_N^i$ is selected as the most suitable box in the group.

Step5: find the scale of the pseudo box. A well-defined pseudo box should only encompass a single object annotation. Thus, among pseudo-boxes with optimal rotation angles but different scaling factors, the box that contains the fewest $\mathcal{P}$ annotations is selected as the final scale annotation for p_i. The final scale annotation for p_i is denoted as $\widetilde{box}_i$.

Scale prediction: As both $\hat{scale}_j^{coarse}$ and $\hat{scale}_j^{final}$ are clusters of nine skeleton points, their Minimum External Rotation Rectangles $MERR(\hat{scale}_j^{coarse})$ and $MERR(\hat{scale}_j^{final})$ are used as the scale predictions.

Loss for scale information: For positive proposal $\hat{p}_j \in \hat{\mathcal{P}}_{pos}$, $p_{l(j)}$ is its matching object annotation and $\widetilde{box}_{l(j)}$ is the corresponding scale annotation. The coarse scale loss and final scale loss are defined as follows:

$$\mathcal{L}_{scale}^{coarse} = \frac{1}{K}\sum_{j=1}^{K}(smooth_{L_1}(MERR(\hat{scale}_j^{coarse}), \widetilde{box}_{l(j)})) \tag{7}$$

$$\mathcal{L}_{scale}^{final} = \frac{1}{K}\sum_{j=1}^{K}(smooth_{L_1}(MERR(\hat{scale}_j^{final}), \widetilde{box}_{l(j)})) \tag{8}$$

where K is the number of positive proposals.

The scale loss of DOPNet is defined as follows:

$$\mathcal{L}_{scale} = \mathcal{L}_{scale}^{coarse} + \mathcal{L}_{scale}^{final} \tag{9}$$

4) Training strategy: As depicted in the first image of Fig. 5, during the initial epochs of the first training phase, the positive proposals assigned to each object tend to be both inaccurate and limited in quantity. Consequently, predicting the scale of the object becomes exceedingly challenging in these early stages. To address this issue, the training process is divided into two distinct phases.

In the first phase, illustrated in the first image of Fig. 5, when DOPNet fails to generate a sufficient number of assigned positive proposals for each object, the total loss is designed as follows:

$$\mathcal{L}_{total}^{phase1} = \mathcal{L}_{conf} + \mathcal{L}_{cls} + \mathcal{L}_{loc} \tag{10}$$

In the second phase, as depicted in the second image of Fig. 5, as more positive proposals emerge, DOPNet incorporates the prediction of object scale information into the network. During this phase, the total loss is designed as follows:

$$\mathcal{L}_{total}^{phase2} = \mathcal{L}_{conf} + \mathcal{L}_{cls} + \mathcal{L}_{loc} + \delta\mathcal{L}_{scale} \tag{11}$$

where δ is a balancing coefficient and it is set to $1/10$ here.

IV. IMPLEMENTATION AND EXPERIMENTS

In this section, the data sets, implementation details, and evaluation metrics are introduced first. Then, the multi-class object counting results of DOPNet on two data benchmarks are reported. Finally, ablation experiments are conducted and the experiment results are discussed to evaluate the effectiveness of DOPNet.

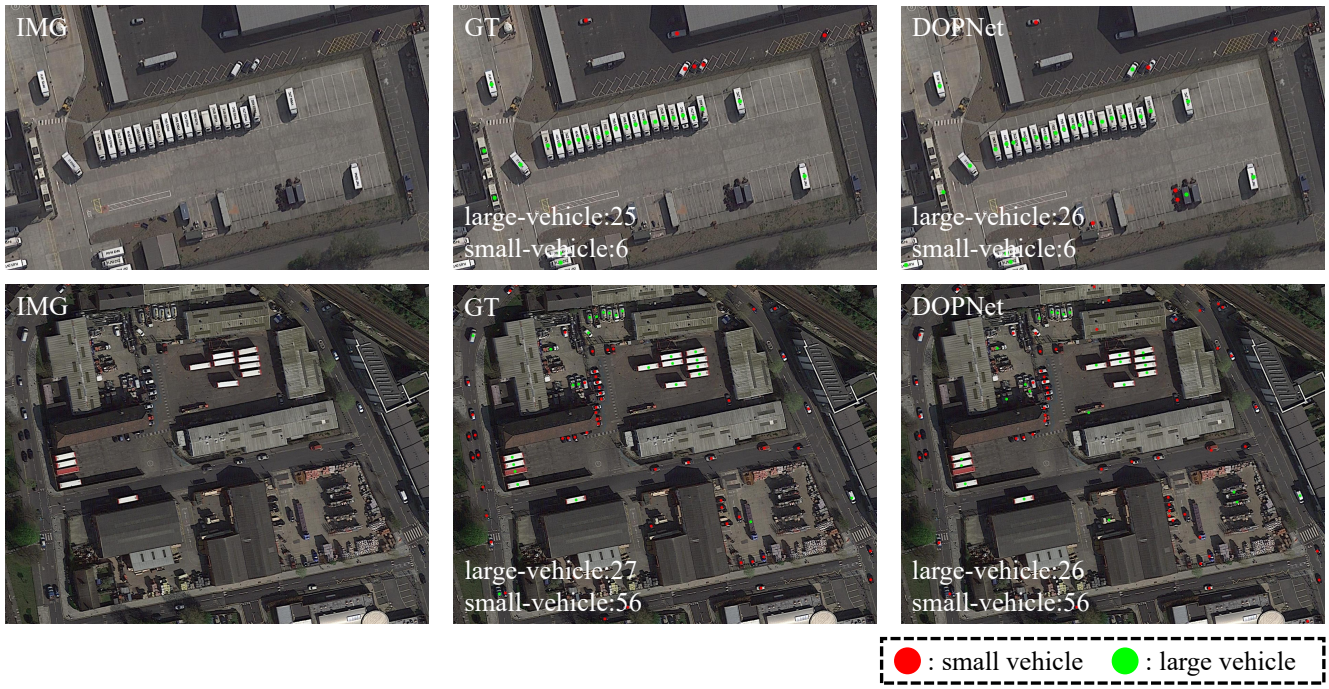

Fig. 6. Visualization results of our method for multi-class object counting on the RSOC data set. The first column represents the original images. The second column represents the object annotations. The third column represents the results predicted by our method.

A. Data sets

There is no publicly available data set on multi-class object counting. Following the work of [8], the single-class object counting data set RSOC [6], which has four categories of objects, and the object detection data set VisDrone-DET [48] are used to evaluate the performance of DOPNet.

1) RSOC data set: The RSOC data set is a large-scale remote sensing object counting data set. It consists of 3057 images and four object categories: buildings, ships, small vehicles, and large vehicles. In this data set, small vehicles and large vehicles always present in a same image. Therefore, following the work of [8], a multi-class object data set is composed by the two sub-sets of small vehicles and large vehicles and adopted in the experiments.

2) VisDrone-DET data set: This benchmark data set consists of 10209 static images, which are captured by various drone-mounted cameras. For the task of object detection, there are ten categories annotated: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. To increase the density of objects in the image, following the work of [8], pedestrian and people are combined as a same category, and tricycle and awning-tricycle are combined as a same category. Since annotations of object detection data set are bounding boxes, centers of bounding boxes are used as point annotations of seven categories except people. For the category of people, the top of the bounding boxes, which are close to the heads, are taken as the annotations for counting.

B. Implementation details

1) Training strategy: The Adam [54] optimizers with a learning rate of 10^{-6} and 10^{-4} are used to optimize the backbone network parameters and other network parameters respectively. The weight decay of the optimizer is set to 10^{-4}. As is mentioned in Subsection III-C4, the training process is divided into two phases. In the first phase, DOPNet is trained for 200 epochs for the RSOC data set and 50 epochs for the VisDrone-DET data set respectively. In the second phase, DOPNet is also trained for 200 and 50 epochs for the RSOC data set and the VisDrone-DET data set respectively. The training was performed with an NVIDIA TITAN X GPU.

2) Training data augment: Each training image is first scaled with a scaling factor randomly selected from $[0.5, 1.5]$. Then, the images are randomly cropped with the size of 256×256 and 512×512 for the RSOC data set and the VisDrone-DET data set respectively. And each cropped image patch is flipped with a probability of 0.5.

C. Evaluation metrics

*1) **Counting metrics**:* Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are adopted to measure the counting performance of each object category. The metrics are defined as follows:

$$MAE = \frac{1}{N_{image}} \sum_{i=1}^{N_{image}} |\hat{P}_i - P_i| \tag{12}$$

$$RMSE = \sqrt{\frac{1}{N_{image}} \sum_{i=1}^{N_{image}} |\hat{P}_i - P_i|^2} \tag{13}$$

where N_{image} is the number of images, $\hat{P}_i$ is the predicted number in the i-th image, and P_i indicates the ground-truth number in the i-th image.

TABLE I
QUANTITATIVE COMPARISONS OF DIFFERENT STATE-OF-THE-ART METHODS ON THE RSOC DATA SET. THE RED AND BLUE FONTS RESPECTIVELY REPRESENT THE FIRST AND SECOND PLACE.

Type	Task	Method	Year & Venue	**Mean**		Small-Vehicle		Large-Vehicle		FLOPs	Param.
				MAE	**RMSE**	MAE	RMSE	MAE	RMS		
Single-Class	Counting	MCNN [27]	2016 CVPR	142.98	488.81	266.23	947.77	19.73	29.85	-	-
		CSRNet [29]	2018 CVPR	75.83	294.49	138.70	568.44	12.95	20.54	-	-
		SFCN [49]	2019 CVPR	108.11	444.12	201.91	865.17	14.30	23.07	-	-
		CAN [50]	2019 CVPR	119.19	469.15	222.39	911.80	15.99	26.50	-	-
		ASPDNet [6], [51]	2020 ICASSP/TGRS	125.49	437.60	236.62	849.99	14.36	25.21	-	-
		PSGCNet [36]	2022 TGRS	40.33	95.72	69.24	169.13	11.41	22.30	8.3G	27.5M
		ADMAL [37]	2022 TGRS	57.75	217.31	101.75	412.43	13.75	22.19	-	-
		MAN [31]	2022 CVPR	36.09	85.26	58.96	147.08	13.22	23.43	6.6G	31.0M
	Localization	P2PNet [13]	2021 ICCV	38.21	115.66	63.48	206.14	12.94	25.17	6.5G	19.2M
	Localization	CLTR [41]	2022 ECCV	41.04	93.46	67.04	160.11	15.03	26.08	6.8G	40.9M
Multi-Class	Counting	MCNN† [27]	2016 CVPR	105.60	425.03	186.91	814.10	24.29	35.96	-	-
		SANet† [52]	2018 ECCV	143.67	500.23	246.83	939.16	40.51	61.30	-	-
		CSRNet† [29]	2018 CVPR	87.54	361.73	154.90	697.10	20.18	26.35	-	-
		BL† [53]	2019 ICCV	107.21	297.61	172.88	534.46	41.54	60.75	-	-
		CAN† [50]	2019 CVPR	58.99	137.24	94.77	235.97	23.20	38.50	-	-
		DSACA [8]	2021 SPL	42.43	127.44	65.40	223.58	19.47	31.31	18.4G	26.7M
	Localization	DOPNet (ours)	-	37.46	93.95	62.43	167.76	12.50	20.14	5.3G	15.5M

In addition, the MAE and RMSE results of all categories are averaged to get MAE_{mean} and $RMSE_{mean}$, which indicate the performance of multi-class object counting. Specifically, MAE_{mean} and $RMSE_{mean}$ are defined as follows:

$$MAE_{mean} = \frac{1}{N_{class}} \sum_{i=1}^{N_{class}} MAE_i \tag{14}$$

$$RMSE_{mean} = \frac{1}{N_{class}} \sum_{i=1}^{N_{class}} RMSE_i \tag{15}$$

where N_{class} is the number of object categories.

*2) **Localization metrics**:* In this work, the Precision, Recall, and F1-measure are used as the localization metrics, following [35], [55]. If the distance between a predicted point and GT point is less than the predefined distance threshold σ, this predicted point will be treated as True Positive (TP). For the RSOC data set, the average width of objects are analysed and the average width of small vehicles and large vehicle are 17px and 48px respectively. Therefore, a fixed threshold $\sigma = 17$ is used for small vehicle and $\sigma = 48$ is used for large vehicle.

D. Comparisons on the RSOC and VisDrone-DET data sets

The compared methods are divided from two aspects. The first aspect distinguishes between methods designed for multi-class object counting and those intended for single-class object counting. The second aspect differentiates whether the method incorporates object localization capabilities. For single-class object counting methods, the state-of-the-art methods in the field of remote sensing object counting, including ASPDNet [6], PSGCNet [36] and ADMAL [37] are compared with our method. In addition, the state-of-the-art methods in the field of crowd counting, including P2PNet [13], CLTR [41] and MAN [31], where P2PNet [13] and CLTR [41] can also perform object localization, are compared with our method. For multi-class object counting methods, DSACA [8] is compared with our method. Moreover, following the work of [8], the output channel of single-class object counting network can be modified to output multiple density maps simultaneously. These methods denoted as ending with † are also compared with our method.

TABLE II
COMPARISON OF NETWORK PARAMETERS, FLOPS, AND EXECUTION PERFORMANCE.

Method	Param.	FLOPs	Execution Time (ms)		
			Small-Vehicle	Large-Vehicle	Total
PSGCNet [36]	8.3 G	27.5 M	48.10	49.74	97.84
MAN [31]	6.6 G	31.0 M	47.91	45.56	93.47
P2PNet [13]	6.5 G	19.2 M	49.97	50.61	100.58
CLTR [41]	6.8 G	40.9 M	44.27	47.04	91.31
DSACA [8]	18.4 G	26.7 M	-	-	104.33
DOPNet (ours)	5.3 G	15.5 M	-	-	65.95

Specifically, PSGCNet [36], ADMAL [37], P2PNet [13], CLTR [41], and MAN [31] are retarined with their source code. The results of DSACA [8] and methods modified from single-class object counting methods are cited from [8]. Other methods are retrained with C-3 framework [56]. For methods that can accomplish object localization, the performance of average precision, average recall and F-measure is also compared.

*1) **Performance on the RSOC data set**:* The performance of object counting on the RSOC data set is presented in Table I. Our method demonstrates state-of-the-art performance when compared to both multi-class object counting methods and localization methods. While the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for small vehicles are worse in our model compared to MAN [31], it is

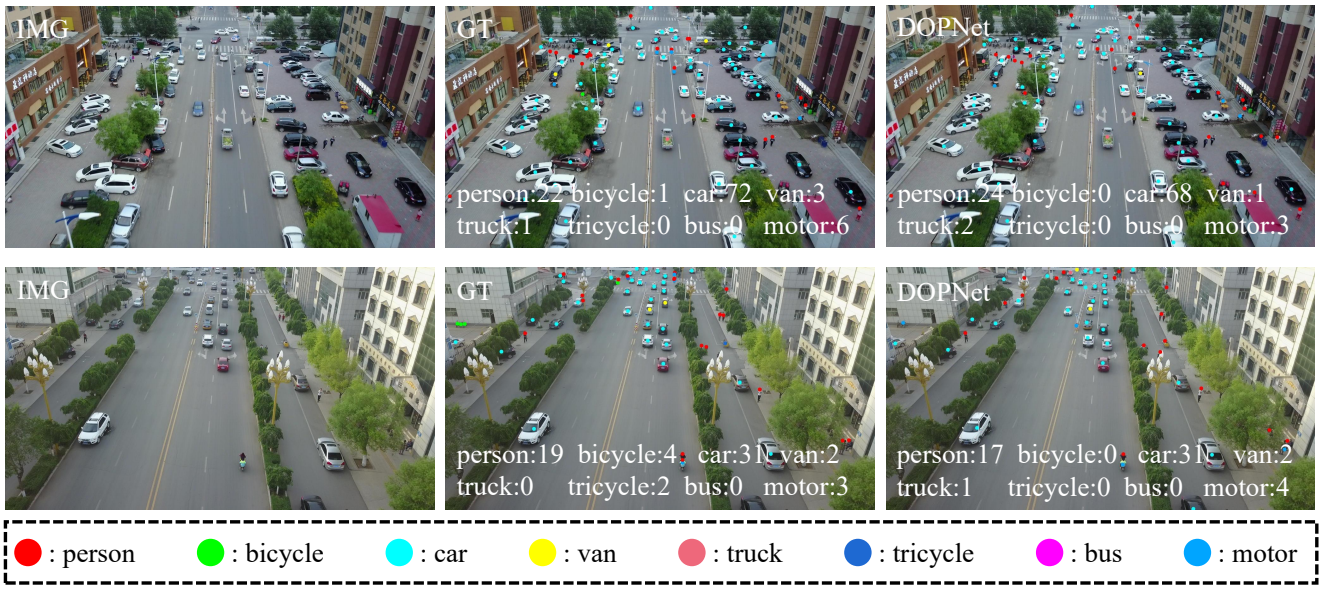

Fig. 7. Visualization results of our method for multi-class object counting on the VisDrone-DET data set. The first column represents the original images. The second column represents the object annotations. The third column represents the results predicted by our method.

TABLE III
LOCALIZATION PERFORMANCE OF DIFFERENT METHODS ON THE RSOC DATA SET.

Network	**Macro**			Small-Vehicle			Large-Vehicle		
	P	R	F	P	R	F	P	R	F
P2PNet [13]	72.19%	70.02%	71.09%	74.61%	76.99%	75.78%	69.76%	63.04%	66.23%
CLTR [41]	68.47%	53.15%	59.85%	60.96%	53.34%	56.90%	75.97%	52.96%	62.41%
DOPNet (ours)	71.15%	71.24%	71.19%	70.54%	72.22%	71.37%	71.76%	70.25%	71.00%

important to highlight that our model possesses the capability to simultaneously count and localize multiple classes of objects. The number of parameters, FLOPS, and execution time for the six methods that exhibited the best performance on the RSOC data set were assessed. The results, which are shown in Table II, indicate that our method has fewest parameters and fewest FLOPs. Furthermore, to evaluate the model's execution time, the RSOC data set is partitioned into 512×512 patches and utilized for testing. An analysis of the execution time specifically for small and large vehicles is performed. The results, as depicted in Table II, demonstrate that the shortest overall execution time is exhibited by our model. In object counting, as shown in the first row of Fig. 6, since there will be false positives and missed positives, it is incomplete to evaluate network performance only by counting results. Therefore, object localization performance is also compared and the results are shown in Table III. Aside from a better counting performance, the localization performance of our method is also comparable to state-of-the-art methods such as P2PNet [13] and CLTR [41]. Selected results are visualized in Fig. 6. From qualitative and quantitative results, it can be seen that DOPNet is effective for multi-class object counting.

*2) **Performance on the VisDrone-DET data set:*** The performance of object counting on VisDrone-DET data set is shown in Table IV. DOPNet achieves the second place on MAE_{mean} and $RMSE_{mean}$ among multi-class object counting methods. Although the MAE_{mean} and $RMSE_{mean}$ of our model are not the best ones, it processes other advantages such as less parameters and faster inference speed as shown in Table II. Compared with DSACA [8], our method can also provide localization information of objects. Visualized results are shown in Fig. 7. From qualitative and quantitative results, it can be seen that DOPNet is also effective for drone-based multi-class object counting.

*3) **Comparison of the two data sets:*** DOPNet achieves different performances on the above two data sets. The RSOC data set comprises images captured from an overhead view, resulting in no object overlap. In contrast, images in the VisDrone-DET data set are captured from an oblique view and contain numerous instances of object overlap. Specifically, as depicted in the second column of Fig. 8, when there are actions such as a person riding a bicycle or a motor, the proximity between the person and the bicycle, or between the person and the motor, is much closer. Due to the overlapping of the person, bicycle, and motor, it becomes challenging for DOPNet to accurately predict scale information and effectively suppress duplicate predictions. In addition, as shown in Table V and the third column in Fig. 8, the average object size

TABLE IV
QUANTITATIVE COMPARISONS OF DIFFERENT STATE-OF-THE-ART METHODS ON THE VISDRONE-DET DATA SET. THE RED AND BLUE FONTS RESPECTIVELY REPRESENT THE FIRST AND SECOND PLACE.

Method	**Mean**		Person		Bicycle		Car		Van		Truck		Tricycle		Bus		Motor	
	MAE	**RMSE**	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN [27]	5.77	8.79	15.87	25.91	2.19	3.59	14.85	20.36	2.92	4.58	1.32	2.19	2.48	3.68	0.43	1.20	6.09	8.83
CSRNet [29]	5.74	8.80	15.87	25.57	1.99	3.83	14.85	20.37	2.76	4.69	1.38	2.18	2.34	3.72	0.66	1.13	6.10	8.87
SFCN [49]	5.75	8.80	15.87	25.58	1.99	3.82	14.85	20.31	2.91	4.58	1.41	2.18	2.40	3.73	0.43	1.20	6.16	8.97
CAN [50]	5.27	7.90	11.85	19.37	2.14	3.59	15.13	19.76	2.74	4.76	1.31	2.18	2.40	3.72	0.52	1.15	6.06	8.69
ASPDNet [6], [51]	5.57	8.57	14.30	23.94	2.04	3.64	14.87	20.40	2.77	4.66	1.34	2.18	2.40	3.73	0.73	1.13	6.12	8.90
PSGCNet [36]	3.70	5.88	7.25	12.03	2.07	3.64	4.68	7.40	2.92	4.66	1.39	2.50	2.47	3.97	0.52	1.18	8.29	11.63
ADMAL [37]	3.99	6.65	6.46	12.46	2.35	4.34	5.86	9.73	3.60	5.90	1.37	2.70	2.88	4.61	0.46	1.22	8.92	12.24
MAN [31]	4.05	6.52	7.96	13.31	2.35	4.34	4.88	7.87	3.60	5.90	1.37	2.70	2.88	4.61	0.46	1.22	8.92	12.24
P2PNet [13]	3.21	5.34	7.47	12.66	2.03	3.84	5.21	8.16	2.62	4.46	1.33	2.59	2.26	3.58	0.46	1.22	4.28	6.21
CLTR [41]	3.47	5.86	8.20	13.15	2.07	4.08	5.54	8.89	2.54	4.69	0.99	1.93	2.15	3.77	0.43	1.17	5.84	9.16
MCNN† [27]	5.67	8.30	12.27	18.69	2.35	4.32	17.89	23.59	2.82	4.18	1.39	2.36	2.93	4.63	0.43	1.22	5.26	7.42
SANet† [52]	7.55	10.83	25.48	35.37	2.38	4.54	15.27	20.26	3.61	5.90	1.37	2.70	2.90	4.76	0.42	0.87	8.99	12.24
CSRNet† [29]	4.60	5.97	9.10	11.68	2.49	3.44	8.50	10.92	5.96	6.48	1.83	2.47	2.82	3.61	0.80	1.12	5.27	8.03
BL† [53]	5.46	6.60	11.88	14.48	2.84	3.58	11.49	13.29	6.22	6.89	2.83	2.59	2.88	3.69	0.78	1.07	4.74	7.22
CAN† [50]	6.86	10.30	9.14	12.47	6.67	11.23	8.77	11.66	8.88	12.02	8.75	15.97	5.99	8.89	2.23	3.30	4.48	6.82
DSACA [8]	3.43	5.36	5.04	7.65	2.35	4.33	3.98	6.02	2.54	4.51	1.32	2.59	2.88	4.61	0.42	0.97	8.90	12.23
DOPNet (ours)	3.48	5.60	8.63	13.05	2.34	4.34	5.49	8.65	2.57	4.57	1.36	2.25	2.54	4.21	0.45	1.14	4.48	6.55

TABLE V
STATISTICS OF THE OBJECT COUNTING DATA SETS.

Dataset	Sensor	View	Average Size	Overleap
RSOC	Satellite	Overhead	704.0 px	False
VisDrone-DET	Drone	Oblique	2448.4 px	True

in the VisDrone-DET data set is significantly bigger than that in the RSOC data set. Bigger object sizes make it difficult for DOPNet to generate accurate pseudo-boxes with point-level annotations. It consequently downgrades the performance of SAFEM and brings challenges to the suppression of the redundant predictions of objects. Moreover, with overlaps between objects, the point annotations in the VisDrone-DET data set are prone to be inaccurate. As shown in the first image of the third column in Fig. 8, the annotations within the red rectangle are incorrectly marked on the adjacent objects. These inaccurate annotations have a much greater impact on point-based methods than on density-based methods.

In summary, when there is no overlap between objects and when objects are not very large, positive proposals generated by the label assignment strategy of DOPNet can more accurately cover the objects. Consequently, more precise pseudo-boxes can be generated for the training of SAFEM. The distinct performance variations of DOPNet on the two data sets indicate that DOPNet is better suited for object

E. Ablation of DOPNet

In this subsection, ablation studies are conducted to verify the effectiveness of individual modules or strategies of DOPNet. All experiments are conducted on the RSOC data set.

1) Effectiveness of the Multi-Scale architecture (MS) and the Scale Adaptive Feature Enhancement Module (SAFEM): As shown in Fig. 3, DOPNet consists of three branches, each of which contains a SAFEM that predicts the object scale and a prediction head. To evaluate the effectiveness of the Multi-Scale architecture (MS) and Scale Adaptive Feature Enhancement Module (SAFEM), a set of experiments is conducted on the RSOC data set. Firstly, the backbone network (VGG-16 [42]) and a prediction head are utilized to compose the baseline model. Since the baseline model does not use SAFEM, it cannot predict the scale of the object, and thus cannot complete non-maximum suppression. To solve the problem, a strategy that adaptively learns the distance between objects is used for non-maximum suppression. Specifically, it is to predict the distance between each object and its nearest neighbor, and then use this distance as the radius to form a circle that is the non-maximum suppression area. Based on the baseline model, Multi-Scale architecture and SAFEM are added to the network. Specifically, for Multi-Scale architecture, three feature layers of different resolutions output by the backbone network are respectively connected to a prediction head. The strategy of non-maximum suppression is same as the baseline model. SAFEM is a module designed based on Multi-Scale structure. After adding SAFEM to the network, non-maximum suppression is performed by the prediction of object scale. The results are shown in Table VI. When Multi-Scale modules and SAFEM are added to the network at the same time, the performance of the network will be greatly improved.

To further investigate how SAFEM enhances network performance and the impact of the Pseudo Box Generation Algorithm on scale prediction, an additional set of ablation experiments is conducted. Specifically, to investigate whether the improvement in network performance by SAFEM is due to

Fig. 8. Visualization of the VisDrone-DET data set. The first column and the third column are some images sampled from VisDrone-DET data set. The second column is the content of the red rectangle in the first column.

an increase in network parameters, the first set of experiments utilizes the complete network structure and the same experimental parameters, but supervision on scale prediction within the SAFEM is removed. In other words, in the first set of experiments, the offset parameters of deformable convolutions [15] in SAFEM are trained and learned solely through the counting task. To validate the effectiveness of the Pseudo Box Generation Algorithm, the second set of experiments uses the complete network structure and the same experimental parameters but removes the Pseudo Box Generation Algorithm. Instead, the second set of experiments directly uses the Minimum External Rotation Rectangle of positive proposals assigned to annotated objects during the training process as the supervision of the scale prediction.

The experimental results in Table VII demonstrate that SAFEM significantly enhances the performance of the network in object counting, and this improvement is not solely attributed to an increase in network parameters. Additionally, the Pseudo Box Generation Algorithm makes a substantial contribution to the network's counting performance. The relevant visualization results are shown in Fig. 9. From the second column of Fig. 9, it can be observed that without supervising the scale prediction of SAFEM, the offset parameters of deformable convolutions [15] are relatively arbitrary and cannot represent the scale information of the object. In addition, as shown in the third column of Fig. 9, predicting the scale of remote sensing objects that have extreme aspect ratios and arbitrary orientations is very difficult without using the Pseudo Box Generation Algorithm to generate more precise pseudo-scale annotations. From the fourth column of Fig. 9, it can be seen that the best results are achieved when using the Pseudo Box Generation Algorithm and supervising the scale prediction.

TABLE VI
ABLATION STUDY OF DIFFERENT MODULES ON THE RSOC DATA SET

Network	Mean		Small-Vehicle		Large-Vehicle	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
baseline	53.07	219.93	90.19	413.37	15.94	26.49
baseline + MS	47.93	172.11	80.61	318.87	15.25	25.34
baseline + MS + SAFEM	37.46	93.95	62.43	167.76	12.50	20.14

TABLE VII
ABLATION STUDY OF SAFEM AND THE PSEUDO BOX GENERATION ALGORITHM ON THE RSOC DATA SET. W/O SCALE LOSS AND W/O SCALE ANNOTATION REPRESENT WITHOUT SCALE LOSS FOR SUPERVISION DURING TRAINING, AND WITHOUT PSEUDO SCALE ANNOTATION GENERATED BY THE PSEUDO BOX GENERATION ALGORITHM DURING TRAINING, RESPECTIVELY.

Strategy	Mean		Small-Vehicle		Large-Vehicle	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o Scale Loss	61.93	227.07	101.97	417.28	21.89	36.85
w/o Scale Annotation	51.74	123.82	87.41	221.82	16.07	25.83
DOPNet (ours)	37.46	93.95	62.43	167.76	12.50	20.14

Fig. 9. Visualization of predicted objects and scales for multi-class object counting on the RSOC data set. w/o Scale Loss and w/o Pseudo Scale Annotation represent without scale loss for supervision during training, and without pseudo scale annotation generated by the Pseudo Box Generation Algorithm during training, respectively.

TABLE VIII
ABLATION STUDY OF $factor_{scale}$ AND $factor_{rotate}$ ON THE RSOC DATA SET.

$factor_{scale}$	$factor_{rotate}$	Mean		Small-Vehicle		Large-Vehicle	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
0.5, 1, 2, 3		37.46	93.95	62.43	167.76	12.50	20.14
0.5, 1, 3, 4	30°	42.82	111.23	71.15	198.52	14.49	23.94
0.5, 1, 5, 6		45.56	124.69	75.62	222.30	15.50	27.08
	10°	41.95	154.69	72.25	289.71	11.65	19.68
	20°	40.81	131.68	69.51	243.37	12.10	19.89
0.5, 1, 2, 3	30°	37.46	93.95	62.43	167.76	12.50	20.14
	40°	39.90	121.04	66.67	220.49	13.15	21.59
	50°	39.83	111.60	66.44	220.51	13.22	22.68

TABLE IX
ABLATION STUDY OF TRAINING STRATEGY ON THE RSOC DATA SET.

Training Strategy	Phase	Mean		Small-Vehicle		Large-Vehicle	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
One-Phase	-	42.77	117.12	71.71	210.74	13.82	23.49
Two-Phase	Phase 1	62.04	258.09	108.24	490.65	15.84	25.53
	Phase 2	37.46	93.95	62.43	167.76	12.50	20.14

2) Ablation experiments of $factor_{scale}$ and $factor_{rotate}$ in scale annotation generation: To obtain object scale information and facilitate training for SAFEM, two parameters, $factor_{scale}$ and $factor_{rotate}$, need to be predefined. In order to explain the sensitivity of these parameters to the network, a series of ablation experiments are conducted. Specifically, $factor_{scale}$ is a predefined scaling factor in scale annotation generation, consisting of four terms. The first two terms fix one side of the pseudo box while reducing or fixing the other side. The third and fourth terms are used to enlarge the sides of the pseudo-box. In our experiments, the same $factor_{scale}$ is set for all object categories to test the method's robustness. The results of these experiments are presented in Table VIII, where DOPNet achieves the best performance on the RSOC data set when $factor_{scale}$ is set to 0.5, 1, 2, 3, and $factor_{rotate}$ is 30°.

3) Effectiveness of the two-phase training strategy: In DOPNet, the training process is divided into two phases. In the first phase, DOPNet is trained to generate enough positive samples for each annotation. In the second phase, DOPNet predicts the scale information of objects based on the accurately assigned positive samples. To verify the effectiveness of the training strategy, two sets of experiments are conducted. The first experiment uses a two-phase training strategy. DOPNet is

TABLE X
ABLATION STUDY OF LOCATION LOSS FUNCTION

mse	ε	*sigmoid*	*arctan*	Mean		Small-Vehicle		Large-Vehicle	
				MAE	RMSE	MAE	RMSE	MAE	RMSE
✓				44.86	137.47	77.10	253.80	12.61	21.14
✓	✓			44.11	133.87	76.07	248.22	12.16	19.53
✓		✓		40.38	114.50	68.77	208.63	11.99	20.36
✓			✓	39.65	110.80	67.33	200.90	11.97	20.70
✓	✓	✓		42.36	138.18	71.84	255.27	12.87	21.10
✓	✓		✓	37.46	93.95	62.43	167.76	12.50	20.14

first trained for 200 epochs with $\mathcal{L}_{conf}$, $\mathcal{L}_{cls}$, and $\mathcal{L}_{loc}$. Then DOPNet integrates $\mathcal{L}_{scale}$ into the total loss and is retrained for another 200 epochs. The second experiment directly uses all losses to train DOPNet for 400 epochs. The results in Table IX show that the two-phase training strategy is necessary and can improve the performance of DOPNet.

4) Effectiveness of the distance loss function: When calculating the distance loss, DOPNet utilizes a modified loss function based on the Mean Squared Error (MSE) loss. As shown in Equation 3, in contrast to the MSE loss, the adopted loss function introduces two key modifications: the inclusion of a scale factor ε and the utilization of the arctan activation function for normalization. In DOPNet, the inclusion of the scale factor ε serves the purpose of aligning candidate points generated from multiple-scale feature layers for loss function calculation at the same scale. The application of the activation function for loss normalization in DOPNet is aimed at ensuring that distance loss, classification loss, and confidence loss are updated for network parameters consistently at the same scale. To verify the effectiveness of the distance loss function, a set of experiments is conducted. At first, the MSE loss is chosen as the baseline for the entire set of experiments. Then, based on the MSE loss, the scale factor ε and activation functions (including sigmoid and arctan) are separately introduced into the distance loss function. Finally, both the scale factor ε and activation function are simultaneously introduced on the top of the MSE loss. The results are shown in Table X. As shown in Table X, introducing a scale factor ε on top of the MSE loss function or using an activation function (whether sigmoid or arctan) both result in improved performance for object counting. When both a scale factor ε and an activation function are simultaneously introduced on top of the baseline, using arctan as the activation function yields better performance. Relative to the sigmoid function, the gradient descent with arctan is faster and less prone to vanishing gradients. In Equation 3, the input to the activation function is $\frac{\varepsilon}{\sqrt{(\hat{x}_j - x_{l(j)})^2 + (\hat{y}_j - y_{l(j)})^2}}$, which is greater than 1. In this case, experimental results show that the arctan activation function provides more informative gradient information, leading to the best experimental performance.

V. CONCLUSION

In this paper, a point-based network named DOPNet is proposed for multi-class object counting and localization in remote sensing images. Different from previous object counting methods, DOPNet generates object proposals instead of density maps to obtain the number of objects. By attaching class labels and scale information to object proposals, DOPNet can predict the number of multi-class objects simultaneously. Besides these, effective training strategy is designed to generate scale information from point annotations. Comprehensive experiments demonstrate the effectiveness of DOPNet on multi-class object counting and localization in remote sensing images.

However, it is important to acknowledge the limitations of DOPNet. One limitation is that DOPNet may be sensitive to the predefined hyperparameters used in the scale annotation generation. In future research, we may consider to design a label assignment strategy that are appropriate for the objects with large aspect ratios and arbitrary orientations. For the problem of scarcity of multi-class object counting data set in remote sensing images, we will try to design a specific remote sensing multi-class object counting data set.

REFERENCES

[1] Y. Wang, J. Hou, and L.-P. Chau, "Object Counting in Video Surveillance using Multi-Scale Density Map Regression," in IEEE International Conference on Acoustics, Speech and Signal Processing, 2019, pp. 2422–2426.

[2] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan, "A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance," IEEE Transactions on Industrial Electronics, vol. 58, pp. 2030–2044, 2010.

[3] 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, pp. 1408–1422, 2018.

[4] C. Dupont, L. Tobias, and B. Luvison, "Crowd-11: A Dataset for Fine Grained Crowd Behaviour Analysis," in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 9–16.

[5] 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, pp. 1048–1061, 2016.

[6] G. Gao, Q. Liu, and Y. Wang, "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, pp. 3642–3655, 2020.

[7] Q. Song, C. Wang, Y. Wang, Y. Tai, C. Wang, J. Li, J. Wu, and J. Ma, "To Choose or To Fuse? Scale Selection for Crowd Counting," in AAAI Conference on Artificial Intelligence, 2021, pp. 2576–2583.

[8] 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.

[9] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, Box out: Beyond Counting Persons in Crowds," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6469–6478.

[10] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, and R. V. Babu, "Locate, Size, and Count: Accurately Resolving People in Dense Crowds via Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, pp. 2739–2751, 2020.

[11] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the Blobs: Counting by Localization with Point Supervision," in European Conference on Computer Vision, 2018, pp. 547–562.

[12] M. Zand, H. Damirchi, A. Farley, M. Molahasani, M. Greenspan, and A. Etemad, "Multiscale Crowd Counting and Localization By Multitask Point Supervision," in IEEE International Conference on Acoustics, Speech and Signal Processing, 2022, pp. 1820–1824.

[13] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu, "Rethinking Counting and Localization in Crowds: A Purely Point-Based Framework," in IEEE/CVF International Conference on Computer Vision, 2021, pp. 3365–3374.

[14] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-Insensitive and Context-Augmented Object Detection in Remote Sensing Images," IEEE Transactions on Geoscience and Remote Sensing, vol. 56, pp. 2337–2348, 2017.

[15] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," in IEEE/CVF International Conference on Computer Vision, 2017, pp. 764–773.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems, vol. 28, 2015.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

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

[19] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," arXiv preprint arXiv:2107.08430, 2021.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in European Conference on Computer Vision, 2016, pp. 21–37.

[21] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-End People Detection in Crowded Scenes," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 2325–2333.

[22] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, "DecideNet: Counting Varying Density Crowds through Attention Guided Detection and Density Estimation," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5197–5206.

[23] P. Wang, X. Sun, W. Diao, and K. Fu, "FMSSD: Feature-Merged Single-Shot Detection for Multiscale Objects in Large-Scale Remote Sensing Imagery," IEEE Transactions on Geoscience and Remote Sensing, vol. 58, pp. 3377–3390, 2019.

[24] G. Cheng, C. Lang, M. Wu, X. Xie, X. Yao, and J. Han, "Feature Enhancement Network for Object Detection in Optical Remote Sensing Images," Journal of Remote Sensing, vol. 2021, 2021.

[25] G. Cheng, J. Wang, K. Li, X. Xie, C. Lang, Y. Yao, and J. Han, "Anchor-Free Oriented Proposal Generator for Object Detection," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–11, 2022.

[26] V. Lempitsky and A. Zisserman, "Learning to Count Objects in Images," Advances in Neural Information Processing Systems, vol. 23, 2010.

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

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

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

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

[31] H. Lin, Z. Ma, R. Ji, Y. Wang, and X. Hong, "Boosting Crowd Counting via Multifaceted Attention," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19 628–19 637.

[32] 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, 2016, pp. 785–800.

[33] 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," in 28th ACM International Conference on Multimedia, 2020, pp. 709–717.

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

[35] 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, vol. 60, pp. 1–12, 2021.

[36] G. Gao, Q. Liu, Z. Hu, L. Li, Q. Wen, and Y. Wang, "PSGCNet: A Pyramidal Scale and Global Context Guided Network for Dense Object Counting in Remote-Sensing Images," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–12, 2022.

[37] G. Ding, M. Cui, D. Yang, T. Wang, S. Wang, and Y. Zhang, "Object Counting for Remote-Sensing Images via Adaptive Density Map-Assisted Learning," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–11, 2022.

[38] Z. Liu, Q. Wang, and F. Meng, "A Benchmark for Multi-Class Object Counting and Size Estimation using Deep Convolutional Neural Networks," Engineering Applications of Artificial Intelligence, vol. 116, 2022.

[39] L. Zhang, X. Wei, H. Yu, and L. Jin, "LMCNet: A Lightweight Multi-class Counting Network with Ghost Attention Mechanism and Focal-L2 Loss," 2023.

[40] H. W. Kuhn, "The Hungarian Method for The Assignment Problem," Naval Research Logistics Quarterly, vol. 2, pp. 83–97, 1955.

[41] D. Liang, W. Xu, and X. Bai, "An End-to-End Transformer Model for Crowd Localization," in European Conference on Computer Vision, 2022, pp. 38–54.

[42] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in International Conference on Learning Representations, 2015, pp. 1–14.

[43] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point Set Representation for Object Detection," in IEEE/CVF International Conference on Computer Vision, 2019, pp. 9657–9666.

[44] W. Li, Y. Chen, K. Hu, and J. Zhu, "Oriented RepPoints for Aerial Object Detection," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1829–1838.

[45] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[46] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," arXiv preprint arXiv:1511.07122, 2015.

[47] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "OTA: Optimal Transport Assignment for Object Detection," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 303–312.

[48] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and Tracking Meet Drones Challenge," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, pp. 7380–7399, 2021.

[49] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning From Synthetic Data for Crowd Counting in the Wild," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8198–8207.

[50] W. Liu, M. Salzmann, and P. Fua, "Context-Aware Crowd Counting," in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5099–5108.

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

[52] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale Aggregation Network for Accurate and Efficient Crowd Counting," in European Conference on Computer Vision, 2018, pp. 734–750.

[53] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian Loss for Crowd Count Estimation with Point Supervision," in IEEE/CVF International Conference on Computer Vision, 2019, pp. 6142–6151.

[54] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.

[55] Q. Wang, J. Gao, W. Lin, and X. Li, "NWPU-Crowd: A Large-Scale Benchmark for Crowd Counting and Localization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 6, pp. 2141–2149, 2020.

[56] 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.