This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

A LightWeight Object Counting Network Based on Density Map Distillation

Zhilong Shen, *Student Member, IEEE*, Guoquan Li, *Member, IEEE*, Hongying Meng, *Senior Member, IEEE*

***Abstract*—In recent years, remote sensing object counting has received widespread attention from academia and industry due to its potential value in urban traffic management, public safety, and agriculture planting. However, object counting tasks still have many technical challenges for computer vision because of the scale changes, uneven object density, and complex background noise in remote sensing images. Although the latest research shows that CNN are technically feasible for object counting, most current CNN-based methods rely on complex network architectures, which limits their deployment in practical application scenarios. In response to the above issues, a lightweight object counting method named EdgeCount aims to balance interface speed and object counting accuracy more effectively. Specifically, we construct a network architecture based on density map knowledge distillation, allowing lightweight student models to learn object density distribution from teacher models. The teacher and student models are composed of an encoder-decoder structure. In the encoding stage, we use the MobileViT as the backbone. The student model only uses the first four layers of the backbone, effectively reducing the number of parameters and computational burden of the model. In addition, we introduce channels and spatial attention module to enhance the ability of feature extraction. In decoding, a low parameter weighted multi-scale feature fusion module (LW-MMFFM) is designed to improve the model's ability to recognize and segment minor structural differences in multi-scale features. Finally, this article conducted experiments on multiple remote sensing object counting datasets (RSOC, CARPK, PUCPR, DroneCrowd) and dense population counting datasets (ShanghaiTech, UCF-QNRF), and the experimental results demonstrated the effectiveness and superiority of the EdgeCount method.**

***Index Terms*—Object Counting, distillation, lightweight, multiscale**

I. Introduction

The object counting task, including crowd counting [1] [2], vehicle counting [3] and general object counting [4], aims to estimate the number of target instances in the image [5] or video [6] in a specific scene. In recent years, with the rapid development of computing power, the object counting method [7] [8] based on the convolutional neural network has made significant progress with the help of a large number of label data. The counting task has practical use for dense scenes, especially in remote sensing scenes. For example, TasselNetV3 [9] is proposed to calculate the number and distribution of plants from aerial images. However, the object counting of remote sensing images is more challenging than standard scenes due to the small object scale of the remote sensing image itself and the complex background interference. It is difficult to extract the characteristics of the target and affect the counting performance. Meanwhile, in remote sensing object counting, the processing of multi-scale information needs higher granularity, and the requirement of a receptive field is also more significant. Therefore, when the scale change is involved, the scale difference of the remote-sensing object is more significant. The domain gap between the scene in the dataset and the natural world scene also limits the use of the counting algorithm.

In [10], the multi-column convolution network is used for multi-scale feature extraction to extract more discriminative object features and rich semantic information from remote sensing images. It extracts multi-scale features by aggregating multiple branches with different receptive fields, which gives the network strong multi-scale representation fusion ability and has achieved encouraging performance. However, they usually aggregate features from different layers in a scale-agnostic manner, which may lead to inconsistent mapping between feature levels and object scales. In addition, these methods bring network parameters and computational burden. In the era of mobile computing and edge computing, improving the model's accuracy while maintaining a low reasoning delay to meet the limited computing capacity of edge devices or embedded systems is crucial. At present, [11] [12] has proposed several lightweight counting networks to improve the operation efficiency. Although they have some advantages, they still have some limitations in remote sensing images with more complex scenes. Firstly, the feature extraction ability of lightweight network structures is limited, which makes it challenging to eliminate complex backgrounds in remote sensing images. Second, the existing lightweight methods improve the scale representation ability by extracting or fusing multi-scale information. However, the existing networks pay more attention to the extraction of scale information rather than the fusion of scale information. In general, the above methods do not balance the counting accuracy and running speed, that is, on the basis of meeting the real-time requirements, the counting ability can still be improved.

In addition to optimizing the network structure, model lightweight can be constructed through model pruning [13], low-rank factorization [14], model quantization [15], and knowledge distillation [16]. The first three model compression methods mainly focus on simplifying the network architecture and reducing parameters and rarely consider knowledge transformation and transmission. Knowledge distillation makes

Zhilong Shen and Guoquan Li are with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: D230101019@stu.cqupt.edu.cn; ligq@cqupt.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

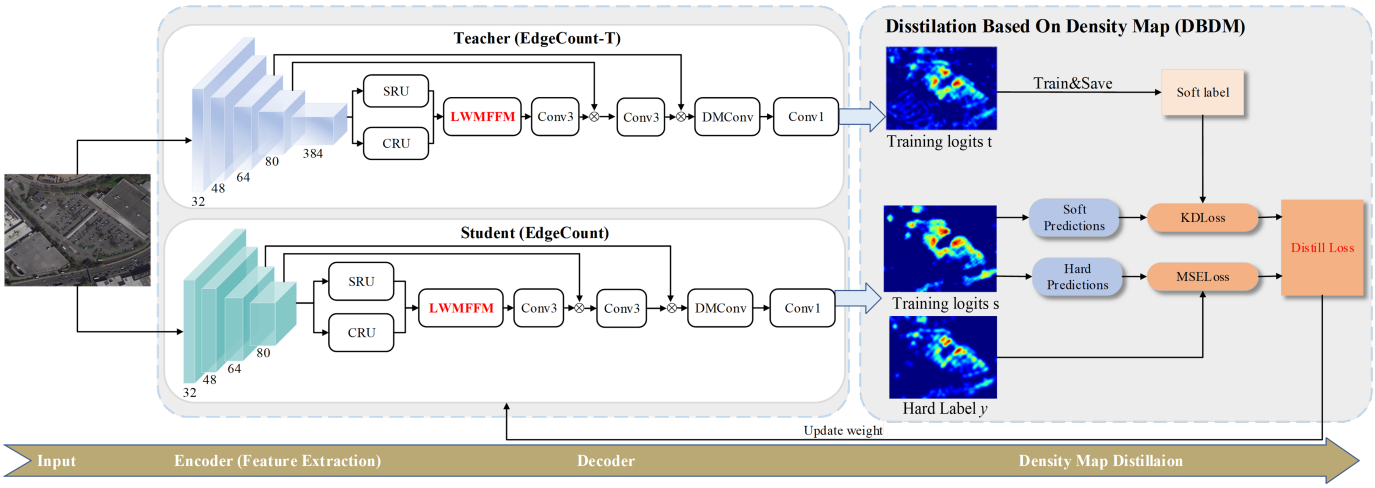


Fig. 1: Framework of our proposed EdgeCount based on density map knowledge distillation. The student model EdgeCount learned knowledge from the teacher model EdgeCount-T through knowledge extraction and soft labeling, which enhanced the counting ability of the student model and the quality of density map generation. In addition, a low parameter weighted multi-scale feature fusion module (LW-MMFFM) is designed to improve the model's ability to recognize and segment minor structural differences in multi-scale features, which is illustrated in Fig. 3.

up for this defect. Knowledge distillation was first proposed by Hinton et al. [17] in 2015. Different from the above three methods, it not only concerns the performance and efficiency of the model but also focuses on improving the performance of lightweight networks through knowledge transfer. In the process of knowledge distillation, the knowledge in the teacher network is transferred to a smaller student network by matching the output probability distribution of the teacher network and the student network so that the student network can imitate the knowledge distribution. It emphasizes how to retain and transfer the knowledge of the original model while compressing the model to make the lightweight model closer to its teacher model in performance.

This paper proposes a lightweight object counting network based on density map knowledge distillation to be applied to remote sensing scenes. The teacher network is named edgecount-t, and the student network is named edgecount. In the encoder stage, MobileViT [18] is used as the backbone network, in which the teacher network uses L1-L5 layer network structure to extract high-level target feature information, and the student network only uses the first four layers. In addition, spatial and channel attention mechanisms are used to enhance the detail feature capture ability of lightweight networks. In the coding phase, a light multi-scale correlation learning module, which consists of two parts: multi-scale feature extraction module and cross-scale learning mechanism, in which the multi-scale feature extraction module is composed of 1x1 convolution modules with different expansion rates, and the cross-scale learning module establishes the correlation of varying scale features. In addition, the knowledge content in the teacher model is transferred to the student model through distillation learning. The main contributions of our work are summarized as follows:

1) We design a lightweight object counting method Edgecount for remote sensing, which can realize a better balance between the accuracy and the interference speed of the network to meet the real-time requirements.
2) We design a object counting trainning structure based on density map knowledge distillation. The student model EdgeCount learned knowledge from the teacher model EdgeCount-T through knowledge extraction and soft labeling, which enhanced the counting ability of the student model and the quality of density map generation. This method effectively reduced the number of parameters, model size and Glflops.
3) We designed a low parameter weighted multi-scale feature fusion module to learn the relationship and interaction between different scale features.

Many experiments on the remote sensing object counting datasets(RSOC, Dronecrowd) and dense crowd datasets (ShanghaiTech, QNRF) show the effectiveness of the proposed method. In particular, our approach can achieve similar or even better detection results than the optimal algorithm.

II. RELATED WORK

This section briefly reviews the relevant object counting methods, especially the counting methods based on density maps. Then we introduce the application of lightweight networks in computer vision.

A. Object counting

In order to fully capture the feature details in images, many researchers have adopted complex multi column convolutional structures. Whether it is universal object counting or specific target counting, these complex network architectures have been adopted as the basic skeleton to address common challenges in the field of counting, such as multi-scale changes and interference from complex backgrounds. However, in both general target counting and specific target counting, complex

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

network architectures are used as backbone to address general challenges in the field of counting, such as scale changes and background interference. SFANet [19], ASPDNet [10], [20], PSGCNet [21], and CAN [22] have been proven to be effective against VGG-16. Similarly, SFCN [23] achieved good counting performance on ResNet-101. For example, in order to accurately process and estimate counting tasks in highly congested scenarios and obtain high-quality density maps, a congested scene recognition network (CSRNet [24]) was proposed. In addition, in order to cope with pedestrians or other objects of different sizes, the scale pyramid network (SPN [25]) was designed, which uses a shared single deep column structure and combines multi-scale feature extraction, utilizing the scale pyramid module to obtain advanced information. In [23], spatial fully convolutional networks (SFCN) are specifically designed to utilize a large amount of synthesized data for crowd counting. Multi column convolutional neural network (MCNN) is a multi column structured network designed to solve scale change problems, where each column is constructed based on different filter kernels.

B. Lightweight Networks

In the past few years, lightweight neural architecture [26] has been an active research field, aiming to achieve the best balance between accuracy and efficiency in object counting. Most state-of-the-art and efficient networks are designed based on an efficient CNN architecture. Mobilecount [12] introduces MobileNetV2 [27] as the backbone for the first time to significantly reduce FLOPs at a little cost of performance drop. Lw-count proposes an effective, lightweight encoding-decoding crowd counting network through a refined ghost block and a scale regression module to reduce the error details and chessboard effect. LMSFFNet [3] achieves object counting in remote sensing scenarios by constructing a lightweight multi-scale fusion network. LEDCrowdNet [28] achieves an optimal trade-off between counting performance and running speed for edge applications of IoVT. Model lightweight is also achieved via knowledge transformation and transmission except to optimize network structure, Knowledge distillation is a way to solve this problem.

Knowledge distillation was originally proposed in [17]. It can transfer knowledge from a large network to a small network. In knowledge distillation, a small student network mimics the intermediate output of a large teacher network. In [29] and [30], the teacher network and student network imitate between layers of the same dimension. [31] proposes a new structured knowledge transfer (SKT) framework. SKT can derive structured knowledge from well-trained teacher networks to generate a lightweight but efficient student network. [32] studied unsupervised crowd counting by transferring knowledge from tagged data to unlabeled data. Unlike traditional and distorted knowledge distillation, Hou et al. [29] proposed self attention distillation (SAD). Sad does not require a teacher network. Distillation is the knowledge of attention layer by layer and spreads between layers.

III. Proposed Method

This section provides a detailed introduction to the overall framework of EdgeCount, network structure based on density map distillation, low parameter weighted scale feature fusion module, and loss function.

A. Overview of the Network Architecture

Fig. 1 depicts the overall framework of the proposed method described in this paper. The process begins by passing the data through the teacher model (EdgeCount-T) for training. This step aims to capture the logits and soft labels produced by the teacher model. Subsequently, training the EdgeCount model facilitates the student model's learning from the teacher network's soft labels, thereby establishing a knowledge transfer between density maps. Furthermore, the KDLoss function assesses the gap in density map distributions between the student and teacher models. At the same time, MSELoss quantifies the discrepancy between the student model's training outputs and actual values. Ultimately, these measurements are integrated into DistillLoss for parameter updates. The subsequent sections detail the specifics of the proposed method.

B. Disstilation Based On Density Map

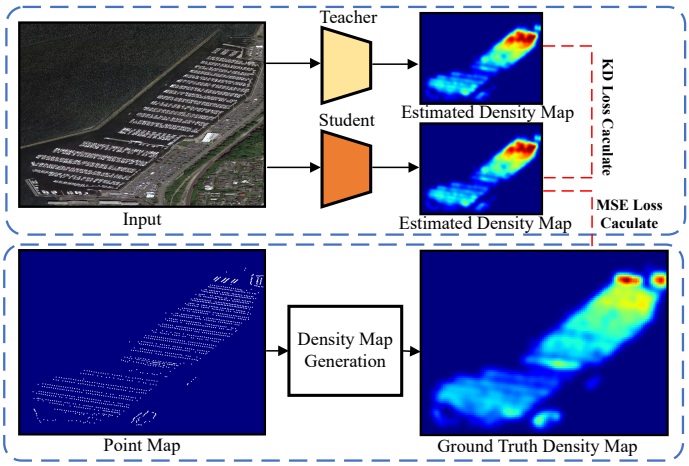


Fig. 2: Illustration of the general framework of density map object counting algorithms based Knowledge Disstilation.

Most existing methods mainly reduce the amount of computation and parameters by optimizing the network structure but rarely consider using the existing network for knowledge transformation. However, with the reduction of network layers and channels, the network cannot extract richer crowd features, resulting in a significant decline in counting accuracy. At the same time, the network structure design often needs experimental data to verify its effectiveness, which brings expensive time costs. To solve the above issues, we designed a network architecture based on density map distillation, as shown in Fig. 2. The student model can learn the density map distribution information from the teacher model by constructing a knowledge distillation, where the density map represents the distribution of each target on the pixel unit.

TABLE I: TECHNICAL SPECIFICATION OF NVIDIA EMBEDDED PLATFORMS

	NVIDIA Jetson Xavier NX	NVIDIA Jetson Orin NX	NVIDIA Jetson Nano	NVIDIA Jetson Orin Nano
AI Performance	21 TOPS[Int8]	100 TOPS	472 GFLOPs	40TOPS
GPU	384-core NVIDIA Volta	1024-core NVIDIA Ampere	128-core NVIDIA Maxwell	1024-core NVIDIA Ampere
CPU	6-core NVIDIA Carmel ARM	8-core NVIDIA Cortex ARM	4-core NVIDIA Quad-core	6-core NVIDIA Cortex ARM
Storage	16GB eMMC5.1	512GB	microSD	512GB NVMe
RAM	8GB 128-bit LPDDR4	16GB	4GB 64-bit LPDDR4 25GB/s	8GB

Algorithm 1: EdgeCount training pseudocode algorithms

Input: Train Dataset; Training epochs n
Output: Lightweight deep learning detection model based on knowledge distillation

procedure: Train Teacher Model $EdgeCount\text{-}T$
  Input train dataset into the teacher network for training
  Use mse_loss as loss function
  Save logits c
  Save Teacher model as the teacher model
return Teacher model
Load $EdgeCount\text{-}T$
count=0
λ**=0.01**
procedure: EdgeCount Model
Input train dataset into the student network for training
for epoch in epochs **do**
  $teacher\_logits\ t \leftarrow$ the training logits of EdgeCount-T
  $student\_logits\ s \leftarrow$ the training logits of EdgeCount
  $soft\_label\ q^{\mathrm{t}} \leftarrow \sum_{i=1}^{H}\sum_{j=1}^{W} c_{ij}$
  $soft\_label\ p^{\mathrm{s}} \leftarrow \sum_{i=1}^{H}\sum_{j=1}^{W} s_{ij}$
  calculate $distill\_loss$ based on q^{t} and p^{s}
  $KDLoss \leftarrow \sum_{i=1}^{N} p^{\mathrm{s}} \cdot \left(\log\left(p^{\mathrm{s}}\right) - \log\left(q^{\mathrm{t}}\right)\right)$
  calculate mse_loss based on $ground\_truthy_i$ and $student\_logits$
  $MSELoss = \frac{1}{2N}\sum_{i=0}^{N}\left(y - p^{\mathrm{s}}\right)^2$
  $DistillLoss = MSELoss + \lambda * KDLoss$
  Update the weight and bias of student model by $DistillLoss$
  $best\_count \leftarrow getCount(ground\_truth, student\_logits)$
end for
return EdgeCount Model

1) Teacher model: The teacher model consists of encoder and decoder. The encoder is the L1-L5 layer of MobileViT, which is mainly used to extract the target features in the sample, and the decoder is further used to extract the subtle features of the segmented target. In particular, the teacher model itself is also lightweight. Since the teacher network has rich target feature information, it can be trained into a highly accurate model. Then, the sample features can be converted into Logits learning tags to introduce the student model through knowledge distillation.

To investigate the efficacy of the teacher model, we incorporate spatial attention mechanisms SRU and CRU to amplify the network's feature extraction capabilities. In the decoder, the previous strategy generally grasps more conceptual spatial features through a multi-layer architecture. Nevertheless, the multi-layer architecture's potential to detect intricate changes between scales may be more resilient, simplifying the identification of the fine segmentation structure. A scale-weighted enhancement module has been developed, incorporating the expansion convolution group and cross-scale connection operation. The convolution group is expanded to increase the number of scales that can be represented by a single output layer. Next, the scale information is analyzed hierarchically from coarse to fine in order to capture multi-scale information at a finer granularity. The model can gain a deeper understanding and analysis of the input feature information through cross-scale connections. Additionally, cross-scale connections allow the model to further comprehend and analyze input feature information. The required number of channels to generate a new feature map varies due to different convolution kernel parameters, optimizing the fullest extent of feature space information. The teacher model undergoes training with mselos and Adam optimizers until its performance no longer improves. Once trained, the model generates an output logits value c, which is a density map of logits. The trained teacher model can be utilized to prepare the student model edgecount.

2) Student model: The student model is a lightweight model proposed in this paper, with a structure similar to the teacher model. In the middle encoder of the student network, only MobileViT L1-L4 layers are utilized. Consequently, the number of channels present in the last layer of the encoder is only 80, which is less than 1/4 of that of the teacher model. This reduction in channels effectively lessens the total computation and parameters, and significantly reduces the size of the model and flops in finer grained multi-scale operations.

The distillation process of the student model is as follows: first, the teacher's logit c is obtained by edgecount-t with the pre-training weight, and then the teacher's model and its pre-training weight c are loaded before training the student model. In each epoch, the training data is input into the teacher and student models, respectively. The decoder outputs the target probability at the pixel position to obtain the logits t and s, respectively. Then the soft tags (i.e., the number of targets) are generated by summing the pixel values in the matrix, which are q^{t} and p^{s} respectively. The formula can express as follows:

$$q^{\mathrm{t}} = \sum_{i=1}^{H}\sum_{j=1}^{W} c_{ij}, p^{\mathrm{s}} = \sum_{i=1}^{H}\sum_{j=1}^{W} s_{ij} \tag{1}$$

Then, the distribution difference between the student density map and the teacher density map is calculated by Kloss, and mselos calculates the difference between the student model and the GT. Finally, the student model is optimized by constructing distillloss by combining the above losses, so that the student network can learn the target distribution and counting performance similar to the teacher network as much as possible.

C. Low-parameter Weighted Multi-scale Feature Fusion Module(LW-MFFM)

To effectively address the challenges posed by varying target scales in images, researchers have developed various

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

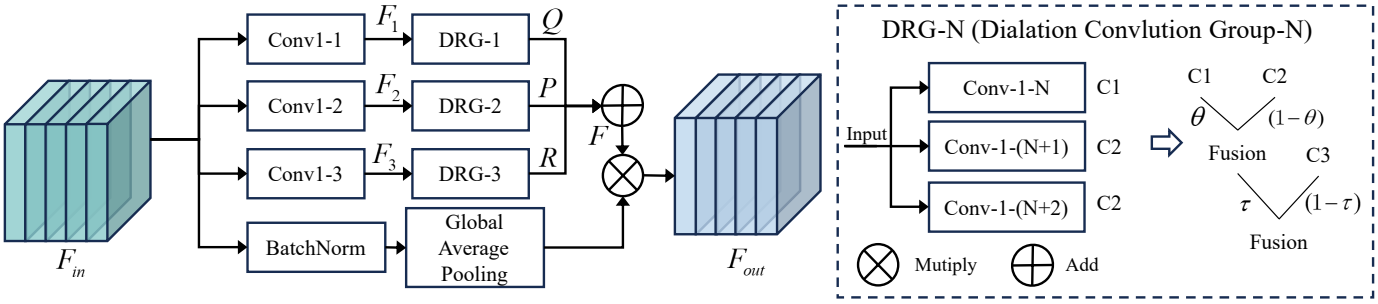


Fig. 3: Details of LW-MFFM architecture. Multiscale features are captured by four parallel different receptive fields. The DGR-N are the group of CNNs, and N means .All convolutional layers maintain the previous size by using padding and followed by the softmax layer.

Algorithm 2: Low-parameter Weighted Multi-scale Feature Fusion Module(LW-MFFM)

Input: The input feature map $F_{in} \in \mathbb{R}^{C\times H\times W}$.
Output: Merged feature map F_{out}.
1: Define 1×1 convolution layer D_i, dilation rate=$\{1,2,3\}$.
2: Define dialation convolution groups l_2 (drg-1, drg-2, drg-3), with rates $j \in \{\{1,2,3\},\{3,4,5\},\{5,6,7\}\}$.
3: Generate two Randomly parameters θ, τ.
4: **for** (D_i) in (D) **do**
5: Apply D_i block to feature map F_{in}.
6: Obtain dilated feature map F_i.
7: **end for**
8: l_2=$D_j\,(W_j, F_i)$
9: Fine-grained sub-features q_i, p_i, r_i.
10: Calculate the weight value Q, P, R by Eqs.3 and 4.
11: Get F by Eq.5.

methods to capture rich multi-scale features, utilizing multi-level, multi-branch, or multi-column feature fusion techniques. Nevertheless, current technologies exhibit limitations in capturing nuanced variations between scales, particularly in accurately identifying and segmenting targets with minor structural differences. In response to this, we propose a lightweight scale-weighted enhancement module (LW-MMFFM), designed to capture detailed changes in channel direction and spatial dimensions. Its infrastructure is depicted in Fig. 3.

The LW-MMFFM module comprises three core stages: multi-scale feature extraction, fine-grained feature analysis, and cross-scale connection operations. Initially, the module broadens the scale range that a single output layer can represent through the use of dilation convolutional groups. It employs hierarchical methods to meticulously analyze scale information, capturing multi-scale features from broad to fine, including smaller levels. Subsequently, a cross-scale connectivity mechanism was constructed to ensure robust performance across diverse scenarios, enhancing the model's ability to comprehend and analyze input feature information. This mechanism effectively associates and parses critical features. Furthermore, drawing inspiration from the cited work [], the cross-scale connection module utilizes a comparable organizational approach for arranging expanded convolutional network groups, thereby facilitating the reuse of computational results and markedly reducing computational costs. The following will provide a detailed introduction to the three critical stages within this module.

During the multi-scale feature extraction phase, the input feature F_{in} is sent into three parallel 1×1 expansion convolution D_i, whose expansion rate is i $\in \{1,2,3\}$, and its operation process can be expressed as $l_1 = D_i\,(W_i, F_{\text{in}})$, and then get three multi-scale feature maps F_1, F_2, F_3. Then, we designed three dialation convolution groups (DCG) with dense connections (dcg-1, dcg-2, dcg-3) to refine them deeply. The feature is decomposed into three sets of sub-node features, and then the dialation convolution D_j of multiple subgroups with expansion rates $j \in \{\{1,2,3\},\{3,4,5\},\{5,6,7\}\}$, which can be expressed as:

$$l_2 = D_j\,(W_j, F_i)\,, j \in \{\{1,2,3\},\{3,4,5\},\{5,6,7\}\} \quad (2)$$

The feature map involving rich multi-scale information is connected to the next layer. The specific operation is as follows: first, the feature map $f_1$will be input into DRG-1, where DRG-1 is composed of three expansion convolutions with different expansion rates, and then Three fine-grained sub-features q_1, q_2, q_3. Then by randomly generating two parameters θ, τ to establish the linear relationship of fine-grained characteristics at different scales, the specific operations are as follows:

$$\begin{cases} Q_1 = q_1 \\ Q_2 = Q_1 \cdot \theta + (1-\theta)\cdot q_2 \\ Q = Q_2 \cdot \tau + (1-\tau)\cdot q_3 \end{cases} \quad (3)$$

$$\begin{cases} P = (p_1 \cdot \theta + (1-\theta)\cdot p_2)\cdot\tau + (1-\tau)\cdot p_3 \\ R = (r_1 \cdot \theta + (1-\theta)\cdot r_2)\cdot\tau + (1-\tau)\cdot r_3 \end{cases} \quad (4)$$

Finally, the cross-scale weighted connection operation is performed to obtain the final weighted multi-scale fine-grained fusion feature map F. The specific operations are as follows:

$$F = [Q\cdot\theta + (1-\theta)\cdot P]\cdot\tau + (1-\tau)\cdot R \quad (5)$$

However, due to the limited feature extraction ability of the lightweight architecture, background noise is still inevitable in the feature map input to the module. Specifically, a C is generated through the global average pooling and sigmoid activation function and multiplied by F. In this way, the number of parameters of the network structure is reduced, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

the local and global information are thoroughly combined so the network can better extract the critical knowledge of the target object. The specific formula is as follows:

$$\omega_g = \text{Sigmoid}(GAP(F)) \tag{6}$$

Where F represents the input characteristic graph, a weight is obtained by global average pooling and multiplied by F. The formula is:

$$F_{\text{out}} = \omega_g \otimes F \tag{7}$$

Where $\otimes$ represents multiplication, in this method all layers have the same number of channels but use different expansion rates to capture multi-scale information. Therefore, four different receptive fields are used to construct a pyramidal multi-scale structure that can extract rich feature information. Then, the global weight coefficient is multiplied by a multi-scale feature map to obtain the global information, which reduces the number of parameters and computations without changing the scale. Although the tree structure looks complex for complexity analysis, our scale enhancer is lightweight due to the ingenious organization of various expansion convolutions. The standard convolution usually contains a single kernel type with kernel size k^2 and depth D. The standard computing block consists of a conv layer of 1×1 and two layers with the same depth. The kernel size is 32, and the depth is d. The parameter quantity (P) and flops can be calculated as:

$$P = D^2 + 2 \times 9D^2 = 19D^2, FLOPs = 19D^2WH \tag{8}$$

Where $W$and Hrepresent the spatial width and height of the feature graph. Similarly, parameters and flops can be expressed as:

$$P = D^2 + 3 \times 9\left(\frac{D}{3}\right)^2 + 9 \times 9\left(\frac{D}{9}\right)^2 = 5D^2 \tag{9}$$

$$FLOPs = 5D^2WH \tag{10}$$

D. Loss Function

Distillloss is composed of MSEloss and KDLoss. KDLoss is used to help students learn the knowledge distribution of the teacher model. Its formula is as follows:

$$L_{kd} = y \cdot \log \frac{y}{y_{\text{pred}}} = y \cdot (\log y - \log y_{\text{pred}}) \tag{11}$$

Where y represents the density map distribution of the teacher model output, y_{pred} represents the density map of the predicted result of the student model. The distribution difference between the teacher and student models can be calculated using the above formula. However, this paper also uses Euclidean distance to estimate the gap between the predicted and actual values. The Euclidean distance is adopted as a loss function to evaluate the difference between the expected density maps and the GT. The loss function l_c is defined as follows:

$$L_{mse} = \frac{1}{2M}\sum_{i=1}^{M}\left\|F\left(C_i;\Theta\right) - F_i^{GT}\right\|_2^2 \tag{12}$$

where M represents the number of pixels in the density maps, C_i represents the input image, θ represents the training parameters, and $F\left(C_i;\Theta\right)$ and F_i^{GT} represents the estimated number and the GT, respectively. The final loss L is defined as follows:

$$L = L_{mse} + \lambda * L_{kd} \tag{13}$$

IV. EXPERIMENTS

This section presents experiments to evaluate our proposed methods. We first describe the implementation details, including the dataset, parameter settings, and evaluation metrics. Then, the performance of the proposed method is demonstrated on remote sensing datasets and dense crowd datasets. In addition, visualization results were presented to illustrate the effectiveness of EdgeCount and EdgeCount-T.

A. Implementation Details

1) Datasets: We carry out experiments on four remote sensing datasets (RSOC [10], CARPK [33], PUCPR+ [33], and DroneCrowd [42]) and dense crowd datasets (ShanghaiTech Part_ A/B [35] and UCF-QNRF [36]) to verify the generalization ability and robustness of the model, as described in detail in Table III.

2) Parameter Settings: EdgeCount and EdgeCount-T are implemented by the PyTorch toolbox, and comparable experiments are conducted on one NVIDIA RTX6000 GPU. Speed experiments conduct on the edge development board. The Adam optimizer is used, and the initial learning rate is 1e-4 without learning rate decay. Hyperparameters λ are set to 0.001. For all subsets, we adopt a batch size of 6. The training process can converge within 500 epochs. The images with resolutions higher than 768 $\times$ 1024 are downscaled to 768 $\times$ 1024. Random cropping and horizontal flipping are applied for augmentation to improve the training and avoid overfitting. Specifically, the crop size is 256×256 for the RSOC_Building and 512×512 for the others.

3) Evaluation Metrics: Two widely used metrics in object counting, mean absolute error (MAE) and RMSE, are adopted to measure the performance of each algorithm. They are defined as follows,

$$\text{MAE} = \frac{1}{N}\sum_{i=0}^{N}|y_i - \hat{y}_i| \tag{14}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}|y_i - \hat{y}_i|^2} \tag{15}$$

where N represents the number of samples, y_i and $\hat{y}_i$ represent the ground truth and the predicted value, respectively.

B. Object Counting

1) Performance On RSOC: The RSOC dataset includes four categories: buildings, small vehicles, large vehicles, and ships. In these categories, different types of images have different resolutions and object sizes. For example, the resolution of small vehicle images is 2473 × 2339, with an average of approximately 531 object instances per image.

According to Table IV, it can be shown that EdgeCount-T achieved the lowest MAE on the Building, Ship, and S-Vehicle datasets. Compared with the state-of-the-art method

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

TABLE II: REMOTE SENSING OBJECT COUNTING DATASETS USED FOR EVALUATION

Datasets		Platform	Images	Training/test	Average Resolution	Annotation Format	Count Statistics			
							Total	Min	Average	Max
RSOC [10]	Building	Satellite	2468	1205/1263	512×512	center point	76,215	15	30.88	142
	Small-vehicle	Satellite	280	222/58	2473×2339	oriented bounding box	148,838	17	531.56	8531
	Large-vehicle	Satellite	172	108/64	1552×1573	oriented bounding box	16,594	12	96.48	1336
	Ship	Satellite	137	97/40	2558×668	oriented bounding box	44,892	50	327.68	1661
CARPK [33]		Drone	1448	989/459	1280×720	bounding box	89,777	1	62	188
PUCPR+ [33]		Camera	125	100/25	1280×720	bounding box	16,915	0	135	331
DroneCrowd [34]		Drone	33,600	24,600/9,000	1920×1080	bounding box	4,864,280	25	144.8	455

TABLE III: CROWD COUNTING DATASETS USED FOR EVALUATION

Datasets	Avg.	Max.	Total	Numbers of Image	Average Resolution
ShanghaiTech A [35]	501	3,139	241,667	482	589 × 868
ShanghaiTech B [35]	123	578	88,488	716	768 × 1024
UCF-QNRF [36]	815	12,865	1,251,642	1,525	2013 × 2902

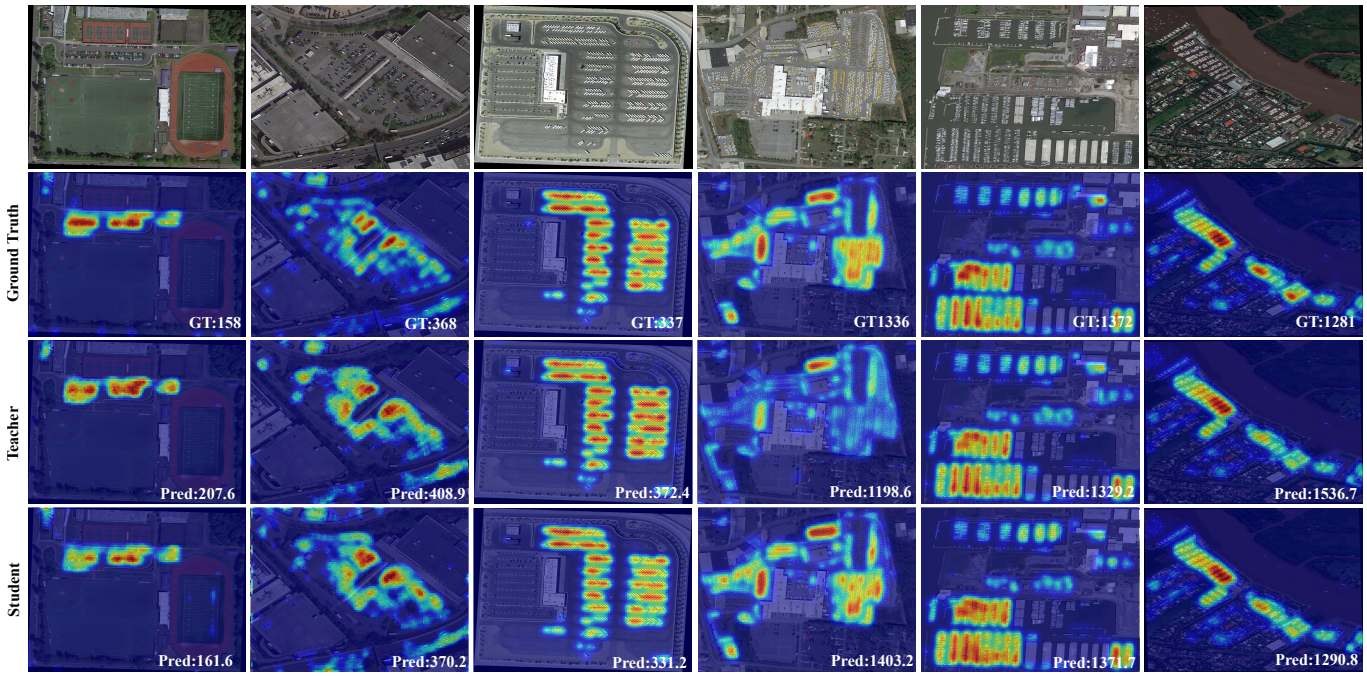


Fig. 4: Visualization density map on the RSOC dataset. The first row represents the original images, the second row represents the predictions of EdgeCount-T, and the third row represents the predictions of EdgeCount.

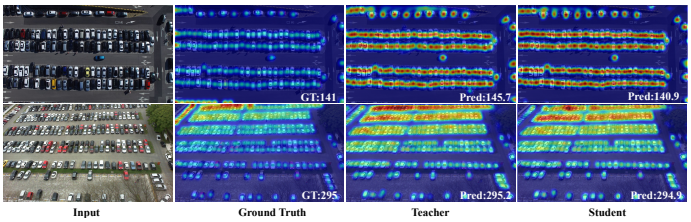


Fig. 5: Visualization results on the CARPK and PUCPR+ dataset. The first column shows the original image and the ground-truth counts, and second column shows the density maps. The last two column represents density map generate by EdgeCount-T and EdgeCount respectively.

ADMAL, EdgeCount-T is not better in terms of MAE and MSE, but its network structure complexity is challenging to apply in practice. Compared with the second group of methods, EdgeCount-T performs better than all methods and has the least GFLOPs. Therefore, the Teacher model EdgeCount-T can achieve the best counting performance through less parameter consumption. Meanwhile, for the student model, the EdgeCount achieves the lowest number of parameters, GFlops, and superior performance in Ship datasets. EdgeCount also achieves better MAE and RMSE on S-Vehicle and L-Vehicle than complex methods such as PSGCNet and eFreeNet, except for ADMAL. However, EdgeCount has only 0.12M parameters and 1.951GFlops overall, which can achieve similar or even better results than ADMAL. This proves that EdgeCount can effectively balance detection accuracy and running speed. Fig. 4 shows some visualization results of our proposed method in RSOC. The density map generated by EdgeCount is almost the same as the ground truth, and the predicted number is less different from the actual number, which proves that EdgeCount is robust to the scale change and an uneven distribution of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

TABLE IV: QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE RSOC DATASET. The FLOPS on 1024 × 768 INPUTS ARE REPORTED.

Methods	Year	Params(M)	GFlops	RSOC-Building		RSOC-S-Vehicle		RSOC-L-Vehicle		RSOC-Ship	
				MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CSRNet [24]	2018	16.26	325.3	8.00	11.78	443.72	1252.22	34.10	46.42	240.01	394.81
CAN [22]	2019	18.18	-	9.12	13.38	457.36	1260.39	34.56	49.63	282.69	423.44
SFANet [19]	2019	-	-	8.18	11.75	435.29	1284.15	29.04	47.01	201.61	332.87
ASPDNet [10], [20]	2021	22.702	465.75	7.59	10.66	433.23	1238.61	18.76	31.06	193.83	318.95
MCFA [5]	2022	-	-	7.93	11.82	238.46	625.90	12.94	20.25	50.45	65.24
TransCrowd-Token [37]	2022	-	-	8.88	12.48	370.96	1209.74	33.53	40.06	108.43	190.05
TransCrowd-GAP [37]	2022	-	-	8.58	12.51	382.06	1209.55	31.54	37.26	95.56	164.49
PSGCNet [21]	2022	27.51	385.79	7.54	10.52	157.55	245.31	11.00	17.65	74.91	112.11
eFreeNet [38]	2023	-	-	5.62	7.69	195.86	463.62	14.55	19.77	65.34	85.45
ADMAL [39]	2023	-	-	5.55	**7.73**	115.61	210.77	11.68	**17.34**	45.07	64.78
MCNN [7]	2016	0.13	21.17	13.65	16.56	488.65	1317.44	36.56	55.55	263.91	412.30
CMTL [40]	2017	2.454	95.55	12.78	15.99	490.53	1321.11	61.02	78.25	251.17	403.07
SANet [41]	2018	1.39	71.45	29.01	32.96	497.22	1276.66	62.78	79.65	302.37	436.91
MobileCount [12]	2020	3.40	6.15	7.72	11.90	316.02	598.58	18.5	30.4	73.2	100.2
LMSFFNet [3]	2023	4.58	14.9	6.52	70.0	141.7	273.0	12.74	27.13	49.47	85.03
LEDCrowdNet [28]	2023	2.02	-	6.81	10.22	237.18	621.80	22.01	32.84	82.26	131.52
EdgeCount-T	-	1.32	2.724	**5.49**	8.05	**106.53**	**203.14**	**11.05**	20.25	37.94	54.91
EdgeCount	-	**0.12**	**1.951**	6.27	9.46	136.61	260.6	11.34	18.37	**36.38**	**52.44**

TABLE V: QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS ON THE CARPK AND PUCPR+ DATASETS

Methods	CAPRK		PUCPR+	
	MAE	RMSE	MAE	RMSE
LPN [43]	23.80	36.79	22.76	34.46
RetinaNet [44]	16.62	22.30	24.58	33.12
LEP [45]	51.83	-	15.17	-
MCNN [7]	39.10	43.30	21.86	29.53
CSRNet [24]	11.48	13.32	8.65	10.24
BL [8]	9.58	11.38	6.54	8.13
ASPDNet [10]	7.81	10.16	-	-
PSGCNet [21]	8.15	10.46	5.24	7.36
ADMAL [39]	**5.12**	**7.05**	-	-
LMSFFNet [3]	7.05	9.03	4.49	6.21
EdgeCount-T	6.11	8.51	3.56	**4.72**
EdgeCount	7.12	9.44	**3.39**	5.15

TABLE VI: PERFORMANCE COMPARISON ON THE DRONECROWD DATASET [34]

Methods	Params(M)	FPS	MAE	RMSE
MSCNN [46]	-	1.76	58.0	75.2
Switch-CNN [47]	15.3	0.01	66.5	77.8
StackPooling [48]	-	0.73	68.8	77.2
DA-Net [49]	-	2.52	36.5	47.3
CSRNet [24]	16.26	3.92	19.8	25.6
CAN [22]	18.18	7.12	22.1	33.4
DM-Count [50]	21.5	10.04	18.4	27.0
STNNet [34]	-	3.41	**15.8**	**18.7**
PSGCNet [21]	27.51	6.79	24.7	31.9
CMTL [40]	2.45	2.31	56.37	65.9
LMSFFNet [3]	4.58	5.75	23.85	30.69
ACSCP [51]	5.1	1.58	48.1	60.2
EdgeCount-T	1.32	15.51	24.65	29.12
EdgeCount	**0.12**	**18.52**	20.48	28.91

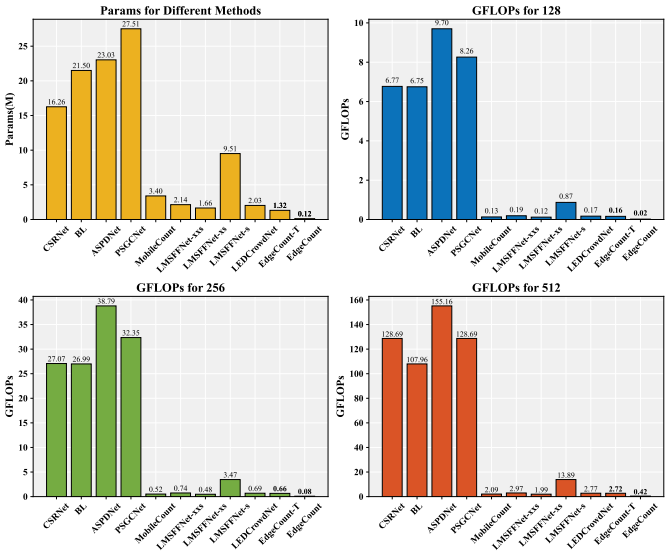


Fig. 6: The visualization shows the comparison results of the proposed method and other methods in terms of parameter size and computational complexity at different scales.

object.

2) Performance on CRUPK+ and PUCPR: CARPK is a large-scale UAV vehicle counting dataset. It consists of 1448 pictures containing about 17000 vehicle labels, of which 989 are used as training sets and 459 as test sets. PUCPR+ is also a vehicle counting dataset, which includes 125 pictures and about 17000 car labels, of which 100 images are used as training sets and the rest as test sets. The results are shown in Table V, which indicates that EdgeCount-T can achieve the lowest MAE and RMSE values on PUCPR+, followed closely by EdgeCount. In CAPRK, in addition to ADMAL, EdgeCount-T can achieve better accuracy than other models, and EdgeCount can surpass other models except for ADMAL and LMSFFNet in Kyoto. Fig.5 shows the density map generation effect of EdgeCount and EdgeCount-T on CARPK and PUCPR+.

3) Performance on DroneCrowd: We also evaluate our method on a more challenging dataset, DroneCrowd, which contains 112 video clips with 33,600 high-resolution frames (1920×1080) captured in 70 different scenarios. The results are shown in table ref table: dronecrowd. The table showed that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

TABLE VII: COMPARISON OF INFERENCE SPEED ON DIFFERENT PLATFORMS

Methods	Params	Size	FLOPs	Latency				FPS			
				NX	ORIN-NX	NANO	ORIN-NANO	NX	ORIN-NX	NANO	ORIN-NANO
CSRNet [24]	16.26	128	6.77G	34.13	19.05	156.11	12.73	29.30	52.50	6.41	78.57
		256	27.07G	154.24	51.36	675.08	33.30	6.48	19.47	1.48	30.03
		512	128.69G	555.57	208.56	-	136.34	1.80	4.79	-	7.33
BL [8]	21.5	128	6.75G	42.06	23.46	133.04	15.21	23.78	42.62	8.51	65.75
		256	26.99G	164.40	54.90	536.00	34.49	6.08	18.21	1.90	29.00
		512	107.96G	483.64	191.87	-	126.86	2.07	5.21	-	7.88
ASPDNet [10]	23.03	128	9.70G	90.69	59.80	379.05	44.40	11.03	16.72	2.68	22.52
		256	38.79G	273.14	133.39	1188.51	99.69	3.66	7.50	0.84	10.03
		512	155.16G	970.21	513.61	-	353.71	1.03	1.95	-	2.83
PSGCNet [21]	27.51	128	8.26G	92.28	32.46	173.26	20.23	10.84	30.81	5.77	49.44
		256	32.35G	265.45	68.11	642.34	44.25	3.77	14.68	1.56	22.60
		512	128.69G	628.93	251.90	-	171.50	1.59	3.97	-	5.83
LMSFFNet [3]	4.58	128	117.76M	71.22	45.10	187.57	44.47	14.04	22.17	5.33	22.48
		256	476.55M	73.47	51.67	246.51	48.15	13.61	19.35	4.06	20.77
		512	1.99G	137.65	121.35	719.00	83.73	7.26	8.24	1.39	11.94
LMSFFNet-S [3]	9.51	128	868.49M	76.49	41.50	187.68	39.35	13.07	24.10	5.33	25.42
		256	3.47G	82.60	50.38	289.71	44.85	12.11	19.85	3.45	22.30
		512	13.89G	178.50	133.83	871.45	91.33	5.60	7.47	1.15	10.95
LEDCrowdNet [28]	2.06	128	173.26M	56.48	34.48	142.00	32.83	17.71	29.00	7.37	30.46
		256	692.96M	59.81	35.23	172.70	34.32	16.72	28.39	5.79	29.14
		512	2.77G	67.19	42.11	260.37	34.58	14.88	23.75	3.80	28.91
EdgeCount-T	1.32	128	163.10M	90.40	63.43	234.84	62.51	11.06	15.77	4.25	16.00
		256	658.06M	99.08	69.29	318.96	66.82	10.09	14.43	3.14	14.97
		512	2.72G	154.73	141.48	795.15	94.57	6.46	7.07	1.25	10.57
EdgeCount	0.12	128	19.87M	57.44	41.07	150.42	40.46	17.41	24.35	6.65	24.72
		256	84.75M	64.10	43.33	196.14	41.90	15.60	23.08	5.10	23.87
		512	423.91M	107.47	94.72	558.18	66.45	9.30	10.56	1.79	15.05

TABLE VIII: PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON CROWED DATASET

Methods	Par.(M)	SHT-A		SHT-B		UCF-QNRF	
		MAE	MSE	MAE	MSE	MAE	MSE
CP-CNN [52]	68.40	73.6	106.4	20.1	30.1	-	-
Switch-CNN [47]	15.30	90.4	135.0	21.6	33.4	-	-
CSRNet [24]	16.26	68.2	115.0	10.6	16.0	135.4	207.4
BL	21.50	61.5	103.2	7.5	12.6	87.7	158.1
UOT [53]	21.50	58.1	95.9	6.5	10.2	83.3	**142.3**
SUA-Fully [54]	15.85	66.9	125.6	12.3	17.9	119.5	213.3
UEPNet [55]	26.21	54.6	91.2	6.4	10.9	-	-
P2PNet [56]	18.34	**52.7**	**85.1**	**6.3**	**9.9**	85.3	154.5
DKPNet [57]	30.63	55.6	91.0	6.6	10.9	**81.4**	147.2
MCNN [7]	0.13	110.2	173.2	26.4	41.3	277.0	426.0
CMTL [40]	2.47	101.3	152.4	20.0	31.1	252.0	514.0
ACSCP [51]	5.10	75.7	102.7	17.2	27.4	-	-
TDF-CNN [58]	0.13	97.5	145.1	20.7	32.8	-	-
SANet [41]	0.91	75.3	122.2	10.5	17.9	152.6	247.3
PCC-Net [59]	0.55	73.5	**102.7**	11.0	19.0	148.7	247.3
LCNet [60]	0.86	93.3	149.0	15.3	25.2	-	-
C-CNN [61]	**0.073**	88.1	141.7	14.9	22.1	-	-
LMSFFNet [3]	4.58	85.85	139.9	9.2	15.1	112.8	201.6
EdgeCount-T	1.32	**68.95**	118.64	**8.07**	13.72	111.43	189.16
EdgeCount	0.12	74.04	119.35	8.58	**13.38**	**107.3**	**183.34**

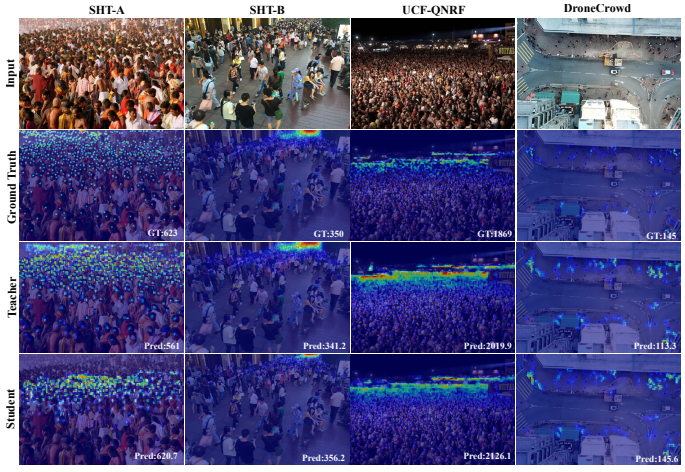


Fig. 7: Visualization results of our method for crowd counting on ShanghaiTech Part_A, ShanghaiTech Part_B, UCF-QNRF and DroneCrowd respectively.

EdgeCount can achieve the fastest FPS detection speed and the smallest parameter quantity, while MAE and RMSE can surpass networks such as LMSFFNet, CMTL, and PSGCNet. Our method can achieve comparable counting performance to the best approach on a large dataset with a few parameters and the fastest detection speed.

C. Comparisons on Dense Crowd Counting Datasets

We compared the proposed method with various state-of-the-art methods on the SH-T A/B and UCF-QNRF datasets, and the specific details are presented in Table VIII. In comparison to the first group on the SHT dataset, our method outperforms CP-CNN and Switch CNN, and it also requires fewer parameters than both. In the second group, EdgeCount-T recorded the lowest MAE on SHT-A and SHT-B, while EdgeCount notched up the lowest MSE and RMSE on UCF-QNRF, attributable to the LW-MMFFM module's enhanced ability to aggregate multi-scale and fine-grained features. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

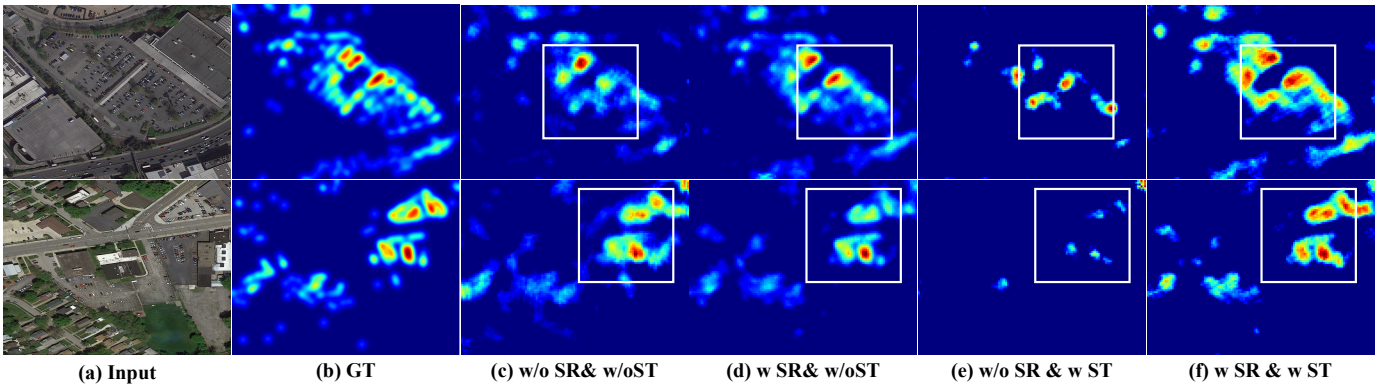


Fig. 8: Density maps generated from each stage of our proposed method EdgeCount-T.

visualization results are depicted in Fig.7.

D. Speed comparison

To rigorously assess the performance of the proposed EdgeCount method in terms of parameters, GFLOPs, FPS, and latency on different edge devices, we conducted comparative experiments at multiple resolutions (128, 256, and 512) and benchmarked them against current state-of-the-art methods. Parameters and GFLOPs at different input resolutions are depicted in Fig.6. EdgeCount exhibits the lowest GFLOPs at each scale, 19.87M, 84.75M, and 423.91M, while maintaining a meager parameter count (merely 0.12M). Furthermore, EdgeCount demonstrates a faster processing speed and higher FPS on all four test devices compared to CSRNet, BL, ASPDNet, and PSGCNet. EdgeCount attains superior latency and FPS performance on three different input resolutions relative to other lightweight models. Particularly on the NX and ORIN-NX devices, the latency experienced by EdgeCount is comparable to that of LEDCrowdNet at a resolution of 128. However, on the Nano and Orin-Nano devices, LEDCrowdNet outperforms EdgeCount, attributable to its single-column network structure and parallel multi-core hole convolutional module for feature extraction, which integrates MobileNetV2 and Vision Transformer. Although EdgeCount does not record the fastest FPS on all test devices, its capacity to effectively extract rich, high-semantic information from teacher models via distillation learning enables it to realize comparable or superior detection performance relative to more complex networks, with minimal inference speed loss. These results illustrate that EdgeCount can effectively balance detection accuracy and inference speed.

V. ABLATION STUDY AND ANALYSIS

A. Ablation on EdgeCount-T

In this section, ablation studies are conducted on the EdgeCount-T. The detailed results are presented in Table IX. The results indicate that EdgeCount-T achieves relatively excellent counting performance while maintaining low parameters and GFLOPs, even without additional modules. Following the introduction of SRConv, the parameters and GFLOPs saw increases of 0.27M and 1.23 GFlops, respectively, resulting in significant improvements across the four subsets. Notably, the MAE and RMSE for small vehicles (SV) experienced a slight increase. As depicted in Fig.8, integrating SRConv allows the network to capture richer target information, resulting in a density map that better aligns with the ground truth (GT).

B. Albation on EdgeCount

1) Contribution of module: Experiments were conducted to examine the impact of SRConv and LW-MFFM on EdgeCount's counting results and density map quality, as illustrated in Table X, with visual results presented in the figures. Fig.9 (a) depicts the density map generation without any modules, which, relative to the Ground Truth (GT), lacks significant target information, and the resulting density map quality is relatively indistinct. With the introduction of SRConv, despite a minimal increase in parameters and computational cost (0.01M and 0.04 GFLOPs, respectively), the network's overall performance is significantly enhanced. The visual results in figures a-b show that incorporating SRConv enables the network to extract more abundant target feature information. The addition of LW-MFFM alone results in a noticeable improvement in network performance, as observed in Fig.9 (a-c), where the network captures more detailed information. Integrating both modules results in observable improvements in the quality of network-generated images, as depicted in Fig.9 (d). Consequently, as the number of integrated modules increases, there is a slight increment in parameters and GFLOPs by 0.05M and 0.29 GFLOPs, respectively, corresponding with a significant improvement in target counting performance. Notably, in the Small Vehicle (SV) dataset, EdgeCount's Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) showed significant improvements of 26% and 47%, respectively.

2) Contribution of Disstilation: Importantly, the EdgeCount student network exhibits improved performance in four categories of the RSOC dataset through the implementation of the knowledge distillation (KD) step, as opposed to scenarios without KD. This enhancement is attributed to the network's ability to inherit vital information from the teacher network, which boasts richer representation capabilities, during the knowledge distillation process. In the remaining three experimental settings, it was noted that knowledge distillation effectively enhances the network's target counting ability without incurring additional parameters and computational overhead.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

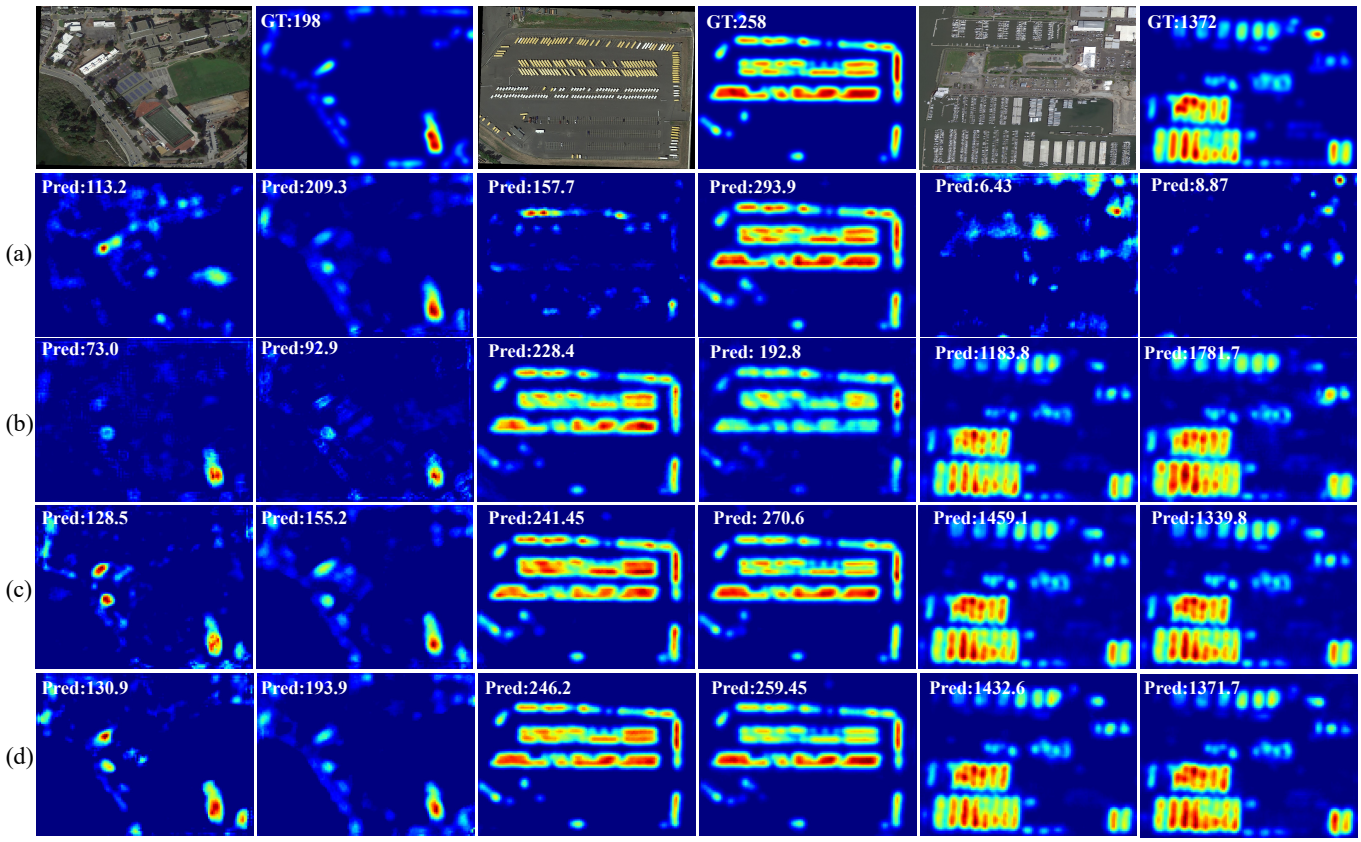


Fig. 9: Density maps generated from each stage of our proposed method EdgeCount.In each sample, the first row are images and ground truth; Starting from the second row, the first column of each sample represents the density map without distillation learning, and the second column represents the density map after distillation learning. (a-d) mean density maps generated from baseline, Baseline+SCConv, Baseline+LW-MFFM, Baseline+SCConv+LW-MFFM. respectively.

TABLE IX: ABLATION STUDY OF EdgeCount-T ON RSOC DATASET

Method	Param	Gflops	Building		Ship		Lage-Vechile		Small-Vehicle	
			MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o SCConv & w/o LW-MFFM	308.787K	2.644G	6.25	9.15	53.51	77.19	12.42	22.52	142.09	298.83
w SCConv & w/o LW-MFFM	576.435K	3.468	5.73	8.17	48.97	67.32	12.2	26.27	147.3	380.73
w/o SCConv & w LW-MFFM	1.049M	8.085	6.35	9.09	38.9	58..50	14.591	41.79	188.97	376.14
EdgeCount-T	1.316M	8.91G	5.49	8.05	37.91	54.81	11.05	18.37	106.53	136.61

C. *Effect of the Hyperparameter* λ

To verify the effectiveness of the loss function, we conduct experiments under the condition of different λ. As can be observed from Table XI, when λ = 0.001, we can obtain the best performance.

VI. CONCLUSIONS

In this work, we propose a lightweight object counting method named EdgeCount aims to balance interface speed and object counting accuracy more effectively. Specifically, we construct a network architecture based on density map knowledge distillation, allowing lightweight student models to learn object density distribution from teacher models. The teacher and student models are composed of an encoder-decoder structure. In the encoding stage, we use the MobileViT as the backbone. The student model only uses the first four layers of the skeleton network, effectively reducing the number of parameters and computational burden of the model. In addition, we introduce channels and spatial attention mobile to enhance the ability of feature extraction. In decoding, a low parameter weighted multi-scale feature fusion module (LW-MMFFM) was designed to improve the model's ability to recognize and segment minor structural differences in multi-scale features. Finally, this article conducted experiments on multiple remote sensing object counting datasets (RSOC, CARPK, PUCPR, DroneCrowd) and dense population counting datasets (ShanghaiTech, UCF-QNRF), and the experimental results demonstrated the effectiveness and superiority of the EdgeCount method.

REFERENCES

[1] X. Wei, Y. Qiu, Z. Ma, X. Hong, and Y. Gong, "Semi-supervised crowd counting via multiple representation learning," *IEEE Transactions on Image Processing*, 2023.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI10.1109/TCSVT.2024.3469933,

TABLE X: ABLATION STUDY OF EdgeCount ON RSOC DATASET

Method	Param	Gflops	KD	Building		Ship		Lage-Vechile		Small-Vehicle	
				MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
w/o SCConv & w/o LW-MFFM	0.07	1.66	-	7.34	10.29	49.01	67.96	18.81	28.29	220.18	649.85
			✓	6.16	8.68	41.54	61.01	14.57	36.24	195.15	687.11
w SCConv & w/o LW-MFFM	0.08	1.70	-	7.25	10.57	55.33	74.6	14.07	27.7	184.39	396.28
			✓	6.38	9.54	43.58	64.78	17.52	28.76	168.29	342.29
w/o SCConv & w LW-MFFM	0.1	1.92	-	6.22	8.94	50.62	72.41	13.72	30.28	167.3	333.37
			✓	6.07	8.81	38.66	55.4	13.56	36.74	157.4	444.01
w SCConv & w LW-MFFM	0.12	1.95	-	5.89	8.69	46.92	61.83	17.36	30.87	158.94	300.01
			✓	5.71	8.45	38.23	56.58	11.72	31.91	141.63	314.31

TABLE XI: ABLATION STUDY ON λ

λ	Building		Ship		Large-Vehicle		Small-Vehicle	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
0	6.25	8.92	43.68	61.88	13.43	29.51	153.88	302.36
0.1	6.2	8.92	41.58	61.41	13.7	29.47	166.94	370.62
0.01	6.27	9.46	36.38	52.44	11.34	18.37	136.61	260.6
0.001	5.71	8.45	38.23	54.62	11.71	25.16	163.17	333.92

[2] J. Yi, Y. Pang, W. Zhou, M. Zhao, and F. Zheng, "A perspective-embedded scale-selection network for crowd counting in public transportation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2023.

[3] J. Yi, Z. Shen, F. Chen, Y. Zhao, S. Xiao, and W. Zhou, "A lightweight multiscale feature fusion network for remote sensing object counting," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[4] G. Sun, Z. An, Y. Liu, C. Liu, C. Sakaridis, D.-P. Fan, and L. Van Gool, "Indiscernible object counting in underwater scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 791–13 801.

[5] 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.

[6] Y. Hou, S. Zhang, R. Ma, H. Jia, and X. Xie, "Frame-recurrent video crowd counting," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[7] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 589–597.

[8] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020.

[9] 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*, vol. 60, pp. 1–15, 2021.

[10] 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, no. 5, pp. 3642–3655, 2020.

[11] Y. Liu, G. Cao, H. Shi, and Y. Hu, "Lw-count: An effective lightweight encoding-decoding crowd counting network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[12] P. Wang, C. Gao, Y. Wang, H. Li, and Y. Gao, "Mobilecount: An efficient encoder-decoder framework for real-time crowd counting," *Neurocomputing*, vol. 407, pp. 292–299, 2020.

[13] Y. Zhang and N. M. Freris, "Adaptive filter pruning via sensitivity feedback," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[14] B. Sokal, P. R. Gomes, A. L. De Almeida, B. Makki, and G. Fodor, "Reducing the control overhead of intelligent reconfigurable surfaces via a tensor-based low-rank factorization approach," *IEEE Transactions on Wireless Communications*, 2023.

[15] Y. Shang, Z. Yuan, B. Xie, B. Wu, and Y. Yan, "Post-training quantization on diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1972–1981.

[16] L. Zhang, R. Dong, H.-S. Tai, and K. Ma, "Pointdistiller: Structured knowledge distillation towards efficient and compact 3d detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 21 791–21 801.

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[18] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[19] L. Zhu, Z. Zhao, C. Lu, Y. Lin, Y. Peng, and T. Yao, "Dual path multi-scale fusion networks with attention for crowd counting," *arXiv preprint arXiv:1902.01115*, 2019.

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

[21] 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.

[22] 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.

[23] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8190–8199.

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

[25] 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 (WACV)*, 2019, pp. 1941–1950.

[26] Z. Wang, Y. Zhang, Y. Liu, D. Zhu, S. A. Coleman, and D. Kerr, "Elwnet: An extremely lightweight approach for real-time salient object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[28] J. Yi, F. Chen, Z. Shen, Y. Xiang, S. Xiao, and W. Zhou, "An effective lightweight crowd counting method based on an encoder-decoder network for the internet of video things," *IEEE Internet of Things Journal*, 2023.

[29] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning to steer by mimicking features from heterogeneous auxiliary networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8433–8440.

[30] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

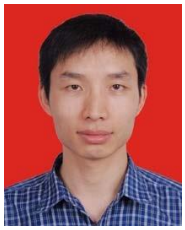
[31] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, and L. Lin, "Efficient crowd counting via structured knowledge transfer," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 2645–2654.

[32] Y. Liu, Z. Wang, M. Shi, S. Satoh, Q. Zhao, and H. Yang, "Towards unsupervised crowd counting via regression-detection bi-knowledge transfer," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 129–137.

[33] 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 international conference on computer vision*, 2017, pp. 4145–4153.

[34] L. Wen, D. Du, P. Zhu, Q. Hu, Q. Wang, L. Bo, and S. Lyu, "Detection, tracking, and counting meets drones in crowds: A benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7812–7821.
[35] 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 conference on computer vision and pattern recognition*, 2016, pp. 589–597.
[36] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 532–546.
[37] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, "Transcrowd: weakly-supervised crowd counting with transformers," *Science China Information Sciences*, vol. 65, no. 6, p. 160104, 2022.
[38] Y. Huang, Y. Jin, L. Zhang, and Y. Liu, "Remote sensing object counting through regression ensembles and learning to rank," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.
[39] 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.
[40] 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 (AVSS)*, 2017, pp. 1–6.
[41] 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.
[42] 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 international conference on computer vision*, 2017, pp. 4145–4153.
[43] 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 international conference on computer vision*, 2017, pp. 4145–4153.
[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
[45] T. Stahl, S. L. Pintea, and J. C. Van Gemert, "Divide and count: Generic object counting by image divisions," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 1035–1044, 2018.
[46] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 465–469.
[47] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Computer Vision & Pattern Recognition*, 2017.
[48] S. Huang, X. Li, Z.-Q. Cheng, Z. Zhang, and A. Hauptmann, "Stacked pooling: Improving crowd counting by boosting scale invariance," *arXiv preprint arXiv:1808.07456*, 2018.
[49] Z. Zou, X. Su, X. Qu, and P. Zhou, "Da-net: Learning the fine-grained density distribution with deformation aggregation network," *IEEE Access*, vol. 6, pp. 60 745–60 756, 2018.
[50] B. Wang, H. Liu, D. Samaras, and M. H. Nguyen, "Distribution matching for crowd counting," *Advances in neural information processing systems*, vol. 33, pp. 1595–1607, 2020.
[51] S. Zan, X. Yi, B. Ni, M. Wang, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.
[52] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1861–1870.
[53] Z. Ma, X. Wei, X. Hong, H. Lin, Y. Qiu, and Y. Gong, "Learning to count via unbalanced optimal transport," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 2319–2327.
[54] Y. Meng, H. Zhang, Y. Zhao, X. Yang, X. Qian, X. Huang, and Y. Zheng, "Spatial uncertainty-aware semi-supervised crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 549–15 559.
[55] C. Wang, Q. Song, B. Zhang, Y. Wang, Y. Tai, X. Hu, C. Wang, J. Li, J. Ma, and Y. Wu, "Uniformity in heterogeneity: Diving deep into count interval partition for crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3234–3242.
[56] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3365–3374.
[57] B. Chen, Z. Yan, K. Li, P. Li, B. Wang, W. Zuo, and L. Zhang, "Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 065–16 075.
[58] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," 2018.
[59] J. Gao, Q. Wang, and X. Li, "Pcc net: Perspective crowd counting via spatial convolutional network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
[60] X. Ma, S. Du, and Y. Liu, "A lightweight neural network for crowd analysis of images with congested scenes," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 979–983.
[61] X. Shi, X. Li, C. Wu, S. Kong, J. Yang, and L. He, "A real-time deep network for crowd counting," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2328–2332.

Zhilong Shen (Student Member, IEEE) received the M.S. degrees from Chongqing University of Science and Technology, in 2023. He is currently pursuing the Ph.D. degree in communication and information engineering, with the School of the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include object counting, object detection, and image recognition.



Guoquan Li (Member, IEEE) received the M.S. and Ph.D. degrees in circuits and systems from Chongqing University, Chongqing, China, in 2006 and 2012, respectively. From 2009 to 2010, he was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA. He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing. His current research interests include image processing, machine learning, and MIMO wireless networks.

Hongying Meng (Senior Member, IEEE) received the Ph.D. degree in communication and electronic systems from Xi'an Jiaotong University, Xi'an, China. He is currently a Reader at the Department of Electronic and Electrical Engineering, Brunel University London, U.K. He has authored over 170 publications, including IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, IEEE TRANSACTIONS ON COGNI TIVE AND DEVELOPMENTAL SYSTEMS, ICASSP, and CVPR. His research interests include digital signal processing, affective computing, machine learning, human–computer interaction, and computer vision. He is an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS.