IOP Publishing Measurement Science and Technology

Meas. Sci. Technol. 36 (2025) 026213 (11pp) https://doi.org/10.1088/1361-6501/ada6f1

Efficient vehicular counting via privacy-aware aggregation network

Jing-an Cheng[1,3], Qilei Li[2,3], Jinyong Chen[1] and Mingliang Gao[1,*]

[1] School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000, People's Republic of China
[2] School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, United Kingdom

E-mail: mlgao@sdut.edu.cn

Received 8 October 2024, revised 25 December 2024
Accepted for publication 7 January 2025
Published 27 January 2025

Abstract

Vehicle counting is crucial for effective road planning and traffic management. Despite significant advancements that have been achieved with the development of deep learning technology, current counting models rely on large-scale parameters and substantial computational resources, which limits their practical application. Additionally, these methods are typically trained on large centralized datasets, which may result in inefficiencies for resource-constrained devices. Furthermore, inadequate privacy protection poses potential risks of personal information leakage. To address these issues, we introduce a lightweight counting network, privacy-aware aggregation network (PANet) for real-world application in this paper. In PANet, a pyramid feature enhancement module is built to aggregate multi-scale information and enhance key representation, while also optimizing the channel-wise output of the model to reduce computational complexity. Furthermore, a federated learning framework is implemented to distribute the computational load and safeguard user privacy. Experimental results on a wide range of counting benchmarks demonstrate the superior efficiency and accuracy of PANet. The code is available at https://github.com/sdut-jacheng/PANet.

Keywords: vehicle counting, lightweight network, federated learning, scale variation

1. Introduction

Vehicle counting is a crucial task in contemporary traffic management and urban planning. It aims at inferencing the number of vehicles in static images or videos. With the progress in deep learning technologies, research on vehicle counting has garnered increased attention from scholars and the accuracy of vehicle counting has markedly improved [1]. In particular, the use of convolutional neural networks (CNNs) for object detection and recognition has led to unprecedented advancements in vehicle counting [2, 3].

Although these methods have shown notable performance improvements, there are still some unresolved technical issues. The first issue is the contradiction between calculation amount and calculation accuracy. Figure 1 illustrates a comparison of parameters and counting accuracy for several state-of-the-art (SOTA) methods on the PUCPR+ dataset [4]. The results shown in figure 1 reveal a common trend that models with higher prediction accuracy generally consists of more parameters. Specifically, these models often require substantial computational resources, especially on resource-constrained edge devices [5]. Thus, the key challenge in vehicle counting is how to reduce model complexity while maintaining high accuracy [6].

The second issue in vehicle counting tasks is scale variation, where the size of vehicles in the same scene changes significantly due to factors such as camera angle, height, and traffic density [13]. This problem of scale variation decreases the accuracy and robustness of vehicle detection and counting, especially in high-density and complex traffic environments. To overcome this challenge, various solutions have

[3] Jing-an Cheng and Qilei Li contributed equally.
[*] Author to whom any correspondence should be addressed.

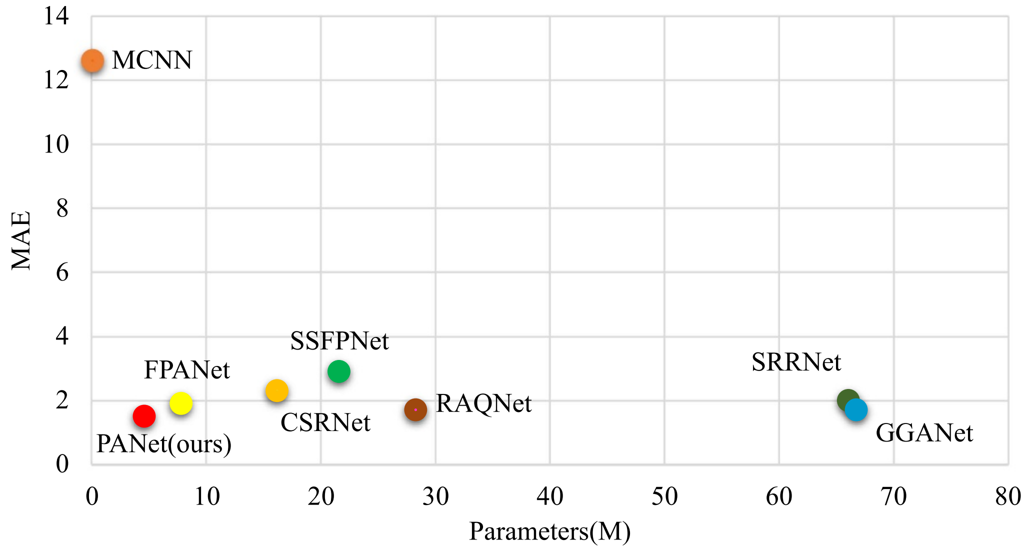

Figure 1. Comparison of the number of parameters and accuracy within the SOTA counting models on the PUCPR+ dataset. A lower mean absolute error (MAE) indicates a higher counting accuracy, and a larger number of parameters reflects a heavier network. These SOTA models include: MCNN [7], CSRNet [8], RAQNet [9], SRRNet [3], GGANet [10], FPANet [11], SSFPNet [12]. The proposed approach ensures high counting precision with a minimal number of parameters.

been proposed. These solutions mainly include pyramid network structures and multi-scale feature fusion. For example, Zhai *et al* [11] proposed a crowd counting model termed feature pyramid attention network (FPANet). They built a multi-scale aggregation module to aggregate information from different scales to address the problem of scale variation. Chen *et al* [12] proposed a selective spatial frequency pyramid network (SSFPNet) in which a hybrid feature pyramid module is developed to aggregate multi-scale information.

The third issue is data privacy and security. Conventional centralized training often necessitates centralizing all data on a server for processing. This process requires significant computational power and raises concerns about potential data leakage. To overcome this issue, decentralized learning has become a promising solution. Federated learning is a widely used decentralized learning technique [14–16]. It trains models across multiple clients and aggregates their updates. It helps to prevent data leakage while alleviating the computational pressure on individual clients. In this regard, Chen *et al* [15] proposed a federated learning-based network termed DLPTNet, which achieves accurate crowd counting while ensuring user privacy. Pang *et al* [17] developed a horizontal federated learning framework. The framework updates the global model by aggregating parameters from local models. It does not require sharing local data, which ensures data privacy. However, federated learning requires frequent transmission of model parameters among clients and the central server. The size of the model directly affects communication overhead. Lightweight networks can reduce the volume of packages transmitted, which increases communication efficiency and overall performance.

Based on the aforementioned background, we propose a privacy-aware aggregation network (PANet). It employs a meticulously designed pyramid feature enhancement (PFE) module to deal with the problem of scale variation while reducing the computational load. Moreover, it combines a federated learning framework to achieve balanced computational loads while ensuring data privacy. Meanwhile, the proposed lightweight PANet reduces communication overhead in federated learning by limiting the number of transmitted parameters. Overall, the contributions of this work are summarized as follows.

(i) We present PANet, a lightweight network to improve vehicle counting accuracy with fewer parameters. Specifically, it contains a well-designed PFE module to capture multi-scale vehicle features, which is beneficial for addressing scale variation.
(ii) We employ a federated learning framework, which addresses the issue of high computational pressure on single clients while upholding data privacy. Furthermore, it mitigates the forgetting effect during client-side updates, which enables efficient and accurate vehicle counting.
(iii) We perform extensive experiments on five vehicle benchmark datasets to showcase the superior accuracy and robustness of the proposed PANet. More importantly, the model achieves high performance with a much smaller number of parameters and floating-point operations (FLOPs).

2. Related work

2.1. Vehicle counting

In recent years, deploying surveillance cameras in urban settings has significantly enhanced the application of vision-based techniques for assessing traffic density [1]. These techniques are typically divided into two principal categories. The first includes traditional methods that rely on frames-based method [18, 19], detection-based method [20, 21], and motion-based method [22]. These methods often face performance issues in urban surveillance due to the impact of perspective shifts and uneven density distribution. The second category comprises methods that use CNN models to generate vehicle density maps, which are then used to analyze traffic flow. Yi *et al* [23] developed a multi-scale feature fusion network. It employs a series of channel-space attention mechanisms, multi-scale context fusion modules, and count-scale pooling modules to boost feature extraction and identify subtle features in target objects. Zhai *et al* [9] proposed a region-aware quantum network, which employs cascaded object region awareness modules to extract local information and quantum-driven calibration modules to capture global information. This design effectively mitigates background interference and significantly improves counting accuracy. Guo *et al* [3] developed the scale region recognition network, which incorporates scale-aware perception and object region recognition modules. By encoding features at multiple scales and minimizing background noise, it enhances the accuracy of counting. Chen *et al* [12] proposed a SSFPNet. It integrates pyramid attention and hybrid feature pyramid modules to gather multi-scale information, and precisely extract object region features.

2.2. Federated learning

Federated learning involves training distributed models across multiple local data sources to achieve data-distributed learning. This approach provides a robust solution for mitigating data privacy and security issues [24]. Arapakis *et al* [25] proposed P4L, a method for enhancing privacy protection. It utilizes a privacy-preserving peer-to-peer (P2P) learning framework across different devices and employs partial homomorphic encryption to ensure the confidentiality of shared gradients. Zhou *et al* [26] designed a privacy-aware asynchronous federated learning framework based on P2P. This framework develops a communication mechanism based on secret sharing to secure the encrypted P2P FL process and introduces a Gaussian mechanism to ensure the anonymity of local model updates. Nevertheless, due to communication efficiency, the frequent exchange of model updates or parameters between clients and the central server or other clients can impact overall model performance. To reduce communication overhead while preserving privacy, this issue has become a key research focus in federated learning in recent years [27]. Wang *et al* [28] proposed a communication-efficient adaptive federated learning technique. It prioritizes compressing one-way communication from clients to the central server, which reduces communication overhead. Wang *et al* [29] developed an efficient asynchronous federated learning approach. It allows edge nodes to select and update parts of the model from the cloud based on local data distribution. This technique significantly decreases both computational and communication loads, which improves the efficiency of federated learning.

2.3. Lightweight network

To streamline networks and enhance computational efficiency, lightweight network models have received extensive attention in the research community [30]. Howard *et al* [31] proposed MobileNet, which reduces model parameters by using depthwise separable convolutions instead of standard convolutions. Zhang *et al* [32] developed ShuffleNet, which employs group convolutions to minimize parameter count and channel shuffling to facilitate information exchange between different groups. Han *et al* [33] designed GhostNet, which identifies redundant feature maps extracted by convolutions and developed the Ghost module to reduce feature redundancy. Tang *et al* [34] proposed GhostNetV2, which enhances GhostNet by incorporating hardware-friendly attention into convolutions to improve the effectiveness of inexpensive operations. This approach boosts network performance and maintains its lightweight characteristics.

3. Methodology

3.1. Overall framework

As illustrated in figure 2, the proposed network consists of three main components: an encoder, a PFE module, and a decoder. The encoder uses OSNet [35] for feature extraction, the PFE module captures multi-scale information and enhances key features, and the decoder uses several transposed convolutions to upsample the density map to match the input dimensions.

3.2. PFE module

To capture multi-scale features and precisely extract vehicle information, we designed the PFE module, as depicted in figure 2. It comprises two main components: the multi-scale feature perception (MFP) unit and the feature enhancement (FE) unit, which collaboratively enhance the counting capabilities of the network.

For the input feature $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$, it is initially distributed across j distinct branches. In each branch, depth-wise separable dilated convolutions (DSDConvs) with different dilation rates (rate $= 1,2,...,p$) are applied to broaden the receptive field without the additional computational cost, which enhances the capture of extensive contextual information. After the DSDConvs, each output feature map is processed through a 3×3 convolution to refine the features. These processed feature maps are combined with the next branch using element-wise addition to promote information fusion among features. Ultimately, the feature maps from all branches are merged into a unified multi-scale feature map $\mathcal{X}_1$ through concatenation. This process is formulated as,

$$Y_k = \mathrm{DSDConv}_{r_1}(X), k = 1 \tag{1}$$

$$Y_k = \mathrm{DSDConv}_{r_i}(X + \mathrm{Conv2d}(Y_{k-1})), \quad i \in \{2,\ldots,p\}, k \in \{2,\ldots,j\} \tag{2}$$

$$\mathcal{X}_1 = \mathrm{Concat}(Y_1, Y_2, \ldots, Y_j), \tag{3}$$

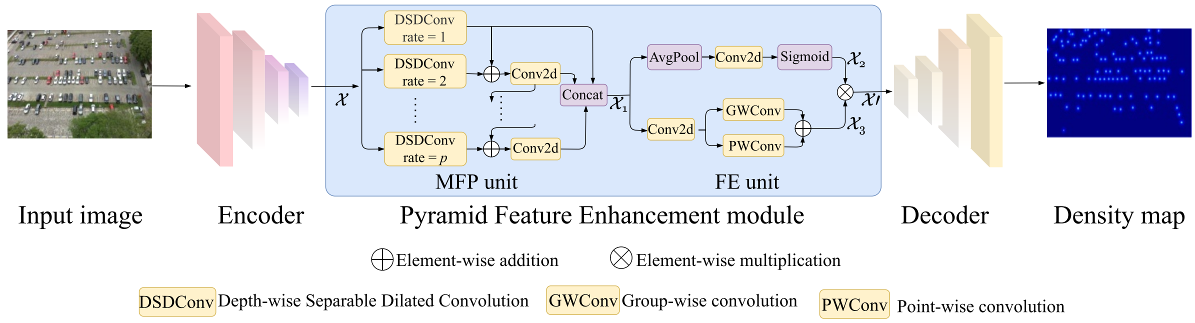

Figure 2. The pipeline of the PANet for vehicle counting.

where DSDConvr_i represents DSDConvs with varying dilation rates, Conv2d signifies a 3×3 convolution, and Y_k is defined as the kth branch.

The output feature map $\mathcal{X}_1$ from the multi-feature resolution unit is directed into two distinct processing branches in the FE unit. On the one hand, $\mathcal{X}_1$ is processed through average pooling to minimize dimensions and abstract basic information, then further refined through a 3×3 convolution layer. After that, a Sigmoid function is used to distill high-level information. On the other hand, the feature map is compressed through a 1×1 convolution to simplify its complexity. The compressed features are then refined through group-wise and point-wise convolutions and combined using element-wise addition. Finally, the feature maps from both branches are multiplied element-wise to produce an enhanced output feature map $\mathcal{X}\prime$. This process is formulated as,

$$\mathcal{X}_2 = \mathrm{Sigmoid}\left(\mathrm{Conv2d}\left(\mathrm{AvgPool}\left(\mathcal{X}_1\right)\right)\right), \tag{4}$$

$$\begin{aligned}\mathcal{X}_3 = \mathrm{Concat}\left(\mathrm{GWConv}\left(\mathrm{Conv2d}\left(\mathcal{X}_1\right)\right),\right.\\ \left.\mathrm{PWConv}\left(\mathrm{Conv2d}\left(\mathcal{X}_1\right)\right)\right),\end{aligned} \tag{5}$$

$$\mathcal{X}\prime = \mathcal{X}_2 \bigotimes \mathcal{X}_3, \tag{6}$$

where GWConv refers to group-wise convolution. PWConv stands for point-wise convolution. $\bigotimes$ represents element-wise multiplication.

3.3. Federated learning framework

In light of the need to manage distributed computing resources efficiently while safeguarding data privacy, this paper introduces a federated learning framework to balance the computational load. The framework leverages local data from various contributors to train machine learning models, with the updated local models being centrally aggregated without sharing the original datasets. This approach significantly reduces global loss and ensures satisfactory performance on participating devices.

As shown in figure 3, the federated learning process starts with downloading the initial global model from the central server to local devices. Subsequently, each client updates this model using their respective local data. A proximal term is incorporated into each client's objective function during the update process to address data heterogeneity and reduce local model bias. This proximal term ensures that local updates align closely with the initial model. This process is formulated as follows,

$$\theta_i^{t+1} = \theta_i^t - \eta\left(\nabla_{\theta_i}\mathcal{L}\left(\theta_i^t\right) + \mu\left(\theta_i^t - \theta_g\right)\right), \tag{7}$$

where θ_i^t indicates the local model parameters during the tth iteration, and θ_i^{t+1} indicates the parameters in the $(t+1)$th iteration. The gradient of the loss function with respect to the local model parameters at the tth iteration is represented by $\nabla_{\theta_i}\mathcal{L}(\theta_i^t)$. The term $\mu(\theta_i^t - \theta_g)$ represents the proximal term gradient at the tth iteration.

The updated local model parameters are then uploaded to the central server. Finally, to create the global model, the central server aggregates the weights submitted by each client and applies an arithmetic averaging method. It is formulated as,

$$W_{\mathrm{global}} = \frac{1}{n}\sum_{i=1}^{n} W_{\mathrm{client}_i}, \tag{8}$$

where W_{global} denotes the aggregate global weight of the PANet, and W_{client_i} signifies the local weight for each client. This method enhances computational efficiency through parallel computing. It alleviates the computational burden on individual clients and significantly reduces training time. Additionally, the design addresses potential privacy breaches associated with data sharing in traditional machine learning and reduces the reliance of the model on a single data source, which improves model generalization.

3.4. Ground truth (GT) generation

The density map is generated using the method of focal inverse distance transform map [36]. It is formulated as,

$$F_{\mathrm{gt}} = \frac{1}{P\left(x,y\right)^{\alpha\times P(x,y)+\beta} + C}, \tag{9}$$

where α and β are defined as 0.02 and 0.75 based on previous approaches [3, 15]. A constant $C = 1$ is utilized to avert division by zero errors, and $P(x,y)$ quantifies the Euclidean distance between a pixel at coordinates (x,y) and the closest annotated head location $(x\prime, y\prime)$.

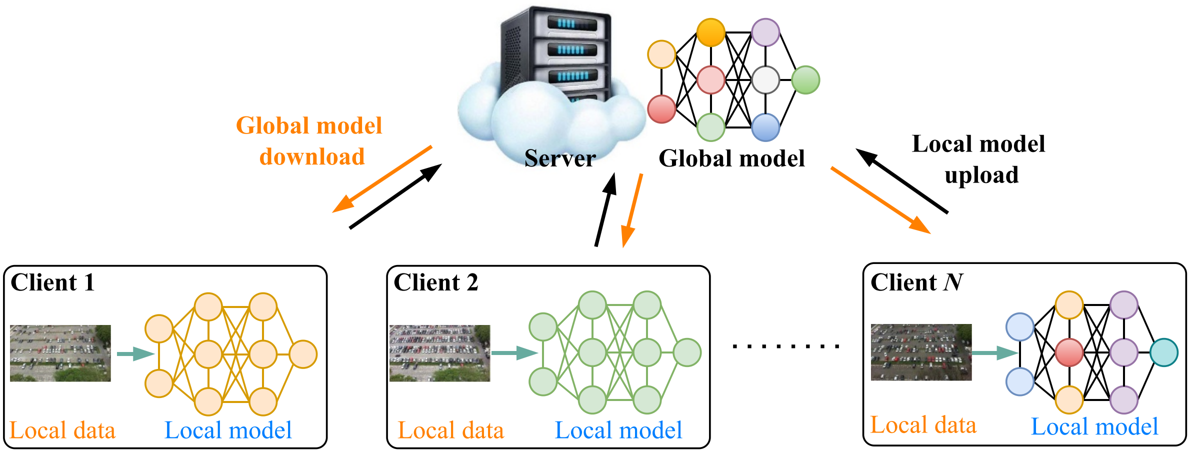

Figure 3. The overview of the federated learning framework.

3.5. Loss function

The Euclidean loss is adopted to measure the pixel-wise difference between the predicted map and the GT. It is formulated as,

$$\text{loss} = \frac{1}{K}\sum_{i=1}^{K} \|F(I_i) - G_i\|_2^2, \tag{10}$$

where K represents the batch size, and $F(I_i)$ denotes the predicted density map. G_i denotes the associated density map of the GT.

4. Experimental results and analysis

4.1. Datasets

In this paper, we evaluated the proposed methods on seven benchmarks to comprehensively demonstrate the effectiveness over the existing SOTA methods. The benchmarks include:

CARPK dataset [4] consists of 1448 drone-view images from four different parking lots, with a total of 89 777 annotations. It is divided into 989 images for training and 459 images for testing.

PUCPR+ dataset [4] is an extensive resource for vehicle counting that includes various weather conditions. It contains 125 images with a total of 16 456 annotations. Among these, 100 images are used for training, and 25 are reserved for testing.

Large-vehicle dataset [37] contains 172 remote sensing images, each with an average resolution of 1552×1573 pixels. The primary focus of the annotations is on large vehicles within these images.

Small-vehicle dataset [37] is another remote sensing vehicle counting dataset. It comprises 280 high-resolution images with a total of 148 838 small vehicles. Compared to the Large-vehicle dataset, it shows greater scale variation.

TRANCOS dataset [38] contains 1244 images from congested traffic environments, each accompanied by a mask.

ShanghaiTech Part A [7] dataset comprises 300 training images and 182 testing images. These images are sourced from the internet and display a relatively dense crowd distribution.

UCF_CC_50 [39] dataset comprises 50 images with diverse resolutions, each averaging 1280 individuals. In total, 63 075 individuals are annotated, with the number of individuals per image varying from 94 to 4543, which indicates substantial variations among the images. The statistics of these datasets is shown in table 1.

4.2. Implementation details

We use OSNet [35] as the backbone, which employs a lightweight structure and has effective feature extraction capabilities. During the training stage, the samples are randomly cropped to a size of 256 × 256 and horizontally flipped for data augmentation. The batch size and the number of epochs are set to 8 and 3000, respectively. The Adam algorithm [40] is employed for optimization, with a learning rate of 1e-4 and a weight decay of 5e-4. To assess efficiency, the input size is configured to 576 ×768, without involving specific datasets. To ensure a fair comparison, all the parameters and the architectures of the comparison methods are obtained from the authors' publicly available codes. All the experiments are conducted on the same hardware with PyTorch [41] on an RTX 3090 GPU.

4.3. Evaluation metrics

The mean absolute error (MAE) and root mean squared error (RMSE) are employed to assess the precision and stability of the counting task. They are defined as,

Table 1. Statistics of the benchmarking datasets.

Dataset	# Images	Train	Val	Test	Average resolution	Min	Max	Avg	Total
CARPK [4]	1448	989	—	459	720 × 1280	—	—	—	89 777
PUCPR+ [4]	125	100	—	25	720 × 1280	—	—	—	16 456
Large-vehicle [37]	172	108	—	64	1552 × 1573	—	—	—	16 456
Small-vehicle [37]	280	222	—	58	2473 × 2339	—	—	—	16 456
TRANCOS [38]	1244	403	420	421	—	—	—	—	46 796
Part A [7]	482	300	—	182	589 × 868	33	3139	501	241 677
UCF_CC_50 [39]	50	40	—	10	—	94	4543	1280	63 705

Table 2. Comparison of different methods in efficiency.

Methods	Params (M)↓	FLOPs (G)↓	Time (ms)↓	FPS↑
CSRNet [8]	16.26	182.69	15.07	66.35
CAN [42]	18.10	193.65	17.80	56.17
SASNet [43]	38.90	393.20	45.94	21.77
BL [44]	21.50	182.19	15.69	63.75
SRRNet [3]	66.14	162.09	37.07	26.93
RAQNet [9]	28.30	250.80	35.20	28.30
PANet (Ours)	**4.61**	**27.36**	**11.03**	**90.64**

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|s_i - \hat{s}_i|, \tag{11}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \hat{s}_i)^2}, \tag{12}$$

where N represents the total image count, s_i is the GT count, and $\hat{s}_i$ is the predicted value for the i-th image.

4.4. Efficiency evaluation

To validate the efficiency of the proposed PANet, a comparative analysis is conducted against the SOTA methods. The performance of PANet was assessed by analyzing the parameters, FLOPs, inference time, and frames per second (FPS) of the model. The experimental results are presented in table 2. Considering that some lightweight networks have no public code, we were unable to conduct a detailed analysis of their FLOPs, FPS, and inference time on a unified experimental platform. Therefore, we compared the number of parameters and the counting accuracy on the ShanghaiTech Part A dataset. The experimental results are shown in table 3.

Compared to the SOTA methods listed in table 2, PANet has the fewest parameters at 4.61 M and requires only 27.36G FLOPs. This indicates that PANet is more compact and easier to deploy in the real world with limited resources. Additionally, PANet achieves the fastest inference time of 11.03 ms and the highest FPS of 90.64. This shows the superior speed and capability of PANet for real-time processing.

The comparison results between the proposed PANet and other lightweight networks in terms of Params, MAE, and RMSE are shown in table 3. It indicates that the MoibleCount has the minimum number of parameters (3.4 M), but the accuracy is the worst. The proposed PANet has 4.61 M parameters which ranks in the mid-range in terms of parameters among the compared methods. However, PANet achieves the best performance in accuracy, with 58.42 and 91.7 in MAE and RMSE, respectively.

Table 3. Comparison of different lightweight methods in Params and counting accuracy. The best results are presented in **bold**.

Methods	Params (M)↓	MAE↓	RMSE↓
MobileCount [45]	**3.40**	89.4	146.0
Repmobilenet [46]	3.41	84.2	127.5
LMSFFNet [23]	4.58	85.9	139.9
ACSCP [47]	5.10	75.7	102.7
MDCount [48]	5.33	84.2	130.7
FPANet [11]	7.80	70.9	120.6
PSCC+DCL [49]	8.96	65.0	108.0
PANet (Ours)	4.61	**58.42**	**91.7**

Table 4. Comparison of different methods on CATRK and PUCPR+ datasets. Results are shown in **bold** for the best performance and <u>underlined</u> for the second-best.

	CARPK		PUCPR+	
Method	MAE	RMSE	MAE	RMSE
YOLO [50]	102.89	110.02	156.72	200.54
Faster-RCNN [51]	103.48	110.64	156.76	200.59
SSD [52]	37.33	42.32	119.24	132.22
LEP [53]	51.83	—	15.70	—
One-look regression [54]	59.46	66.84	21.88	36.73
LPN [55]	23.80	36.79	22.76	34.46
RetinaNet [56]	16.62	22.30	24.58	33.12
MCNN [7]	39.10	43.30	21.86	29.53
CSRNet [8]	11.48	13.32	8.65	29.53
SRRNet [3]	8.50	10.98	2.04	2.79
RAQNet [9]	**5.38**	**7.83**	<u>1.71</u>	<u>2.54</u>
PANet (Ours)	<u>5.94</u>	<u>8.23</u>	**1.46**	**2.03**

4.5. Performance evaluation

4.5.1. Comparison on vehicle counting. The experimental results on the CARPK and PUCPR+ datasets are shown in table 4. On the CARPK dataset, the proposed PANet achieved an MAE of 6.25 and an RMSE of 8.58, which both ranked second. Compared to the top-performing RAQNet [9], PANet shows an increase in MAE and RMSE by 9.4% and 4.9%, respectively. Nevertheless, PANet reduces the parameter count by 83.7%, which significantly reduces model complexity with only a slight decrease in accuracy. Unlike RAQNet [9], which primarily focuses on addressing background interference, PANet leverages efficient MFP unit and FE unit to capture multi-scale features and precisely extract vehicle information.

On the PUCPR+ dataset, PANet achieved an MAE of 1.46 and an RMSE of 2.03, the best among all methods. Compared to the third-placed SRRNet [3], PANet shows a 28.4% and 27.2% improvement in MAE and RMSE, respectively, while reducing the parameter count by 93%. The results indicate that the proposed PANet achieves superior efficiency and precision in multi-scale information extraction compared to SRRNet [3], which also focuses on scale variation challenges. By leveraging the PFE module to optimize the representation and integration of multi-scale features, PANet can alleviate the effect of scale variations. This improvement further validates its practicality and robustness in complex scenarios.

The results of the experiments on the large-vehicle and small-vehicle datasets are presented in table 5. For the small-vehicle dataset, the proposed PANet achieved the lowest MAE of 118.76 and the second-lowest RMSE of 424.57. Compared to ASPDNet [37], PANet improved MAE and RMSE by 72.6% and 65.7%, respectively. The results show that the proposed PANet achieves notable performance improvements compared to ASPDNet [37], which also addresses multi-scale challenges in remote sensing tasks. Through the incorporation of the PFE module, PANet can capture diverse scale features of vehicles in remote sensing data. Furthermore, the integration of global and local features enhances the discrimination of feature representations. On the large-vehicle dataset, PANet achieved the best performance in both MAE and RMSE, with scores of 15.66 and 31.13, respectively. Compared to SANet [57] which also addresses scale variation problems, PANet improved MAE by 75.06% and RMSE by 60.92%. SANet [57] relies on fixed convolutional kernels for multi-scale feature extraction and employs a relatively simple feature fusion strategy, which limits its ability to handle the diverse and variable scales of targets in complex scenarios. In contrast, PANet incorporates the PFE module, which substantially enhances multi-scale feature extraction and representation. Specifically, the MFP unit employs adjustable dilated convolutions to flexibly capture multi-scale information, while the FE unit integrates global and local features for deeper feature fusion. This approach enables PANet to focus more on variable targets.

In addition, validation experiments were conducted on the Trancos dataset, as shown in table 6. Although the MAE of PANet was 23% higher than SRRNet, it achieved a 93% reduction in parameters. Compared to other SOTA methods, PANet improved the counting accuracy. It highlights the effectiveness of the proposed PFE module in aggregating multi-scale features and enhancing key representations. It helps the model to better cope with traffic congestion in diverse environments and adapt to varying lighting conditions and crowd densities.

Figure 4 illustrates the qualitative results across five vehicle datasets, including CARPK, PUCPR+, Small vehicle, Large vehicle, and TRANCOS. The predicted counts (Est) align closely with the GT in all scenarios. This demonstrates the robustness and adaptability of PANet across different vehicle densities and perspectives.

Table 5. Comparison of different methods on small and large vehicle datasets. The best results are presented in **bold**, while the second-best results are highlighted in <u>underline</u>.

	Small vehicle		Large vehicle	
Method	MAE	RMSE	MAE	RMSE
MCNN [7]	488.65	1317.44	36.56	55.55
CSRNet [8]	443.81	1252.22	34.10	46.42
SCAR [58]	497.22	1276.65	62.78	79.46
ASPDNet [37]	433.23	1238.61	31.76	40.14
SFCN [59]	440.70	1248.27	33.93	49.74
SFANet [60]	435.29	1284.15	29.04	47.01
CMTL [61]	490.53	1321.11	61.02	78.25
CAN [42]	457.36	1260.39	34.56	49.63
SPN [62]	445.16	1252.92	36.21	50.56
SANet [57]	497.22	1276.66	62.78	79.65
SRRNet [3]	<u>122.79</u>	**419.65**	<u>18.25</u>	<u>31.24</u>
PANet (Ours)	**118.76**	<u>424.57</u>	**15.66**	**31.13**

Table 6. Comparison of different methods on TRANCOS vehicle dataset. **Bold** indicates the best results, and <u>underlined</u> highlights the second-best results.

Methods	MAE
SANet [63]	17.77
Lempitsky *et al* [64]	13.76
Guerrero-Gómez-Olmedo *et al* [38]	13.29
CCNN [65]	10.99
Zhang *et al* [66]	5.31
SRRNet [3]	**3.89**
PANet (Ours)	<u>5.05</u>

4.5.2. Comparison on crowd counting. To validate the generalization ability of the proposed PANet, cross-domain experiments are conducted on two crowd datasets (ShanghaiTech Part A and UCF_CC_50). The comparison with several SOTA methods is presented in table 7.

On the ShanghaiTech Part A dataset, PANet achieves the best results in MAE and RMSE. Compared to RAQNet [9], PANet improved MAE by 1.0%, with a decrease of 9.4% in RMSE, and an 83.7% reduction in parameter. In the UCF_CC_50 dataset, PANet achieved the best performance across all metrics. Compared to MobileCount [45], a lightweight network, PANet improved MAE and RMSE by 63.01% and 61.51%, respectively. These results demonstrate that PANet performs well in both dense and highly variable scenes. Additionally, the cross-domain experiment results indicate that PANet maintains high counting accuracy and robustness across different types of datasets, which further proves its broad applicability in practical scenarios. The subjective results on these two datasets are shown in figure 5. The result clearly shows that the generated density maps closely resemble the GT, and the predicted values are also very close to the

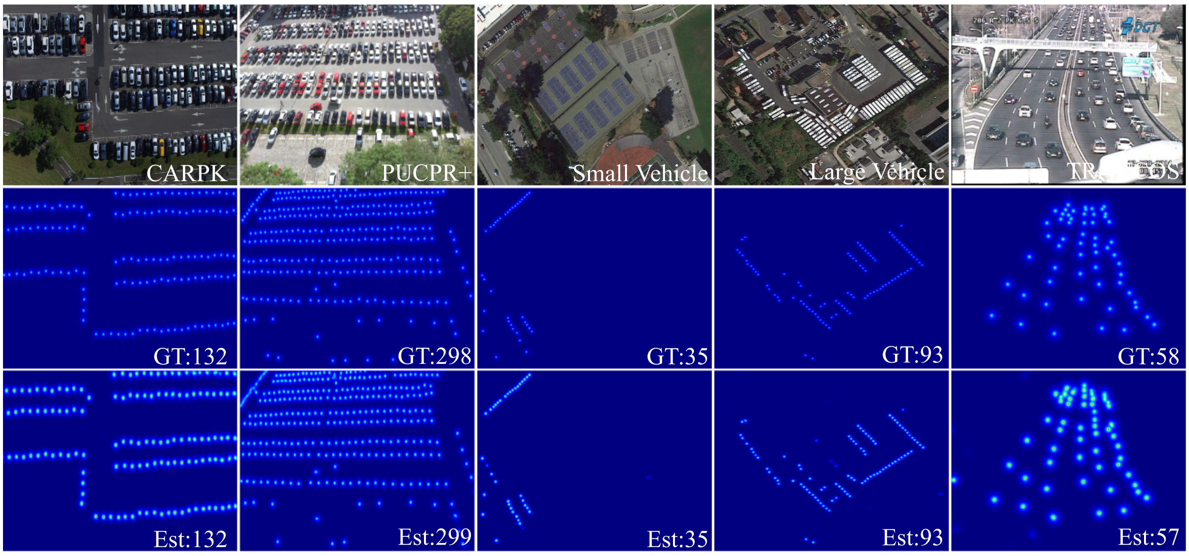

Figure 4. The visual result on five vehicle datasets. The first row represents the input image, the second row is the ground truth, and the third row is the generated density map.

Table 7. Comparison of different methods on ShanghaiTech Part A and UCF_CC_50 crowd datasets.

	ShanghaiTech Part A		UCF_CC_50	
Method	MAE	RMSE	MAE	RMSE
MCNN [7]	110.20	173.20	377.60	509.10
TDF-CNN [67]	97.50	145.10	354.70	491.40
LCNet [68]	93.30	149.00	326.70	430.60
MobileCount [45]	89.40	146.00	284.80	392.80
CCNN [69]	88.10	141.70	—	—
1/4SAN+SKT [70]	78.00	126.60	—	—
SANet [57]	75.30	122.20	358.40	334.90
PCCNet [71]	73.50	124.00	240.00	315.50
SRRNet [3]	60.80	103.00	172.90	256.30
RAQNet [9]	59.00	101.20	177.10	247.60
PANet (Ours)	**58.42**	**91.70**	**105.36**	**151.19**

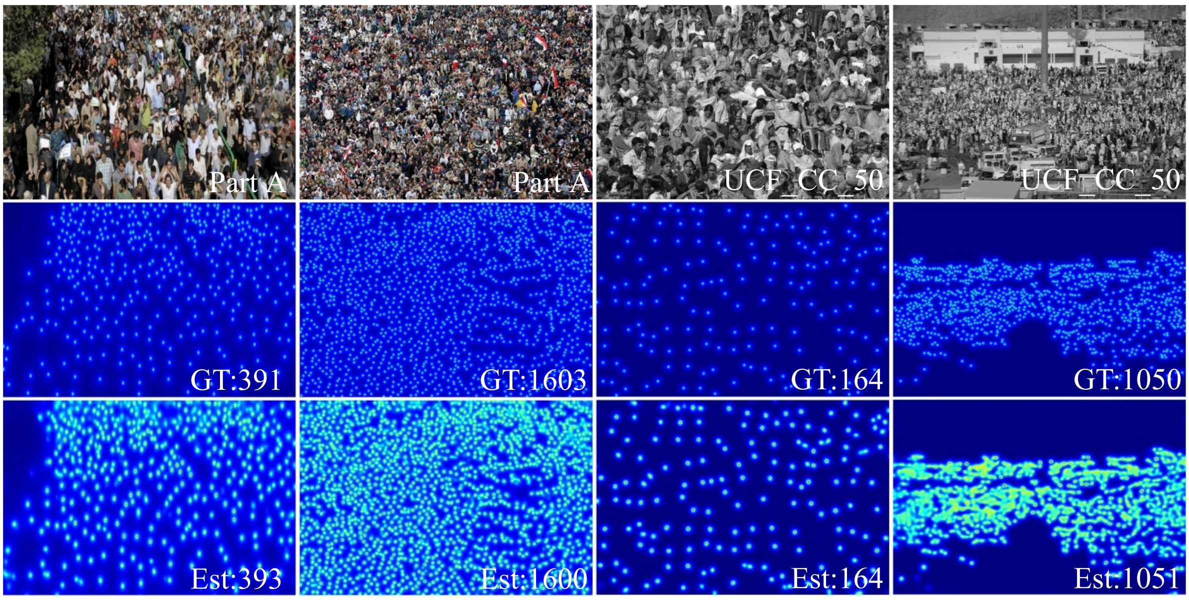

Figure 5. The visual result on ShanghaiTech PartA and UCF_CC_50 datasets.

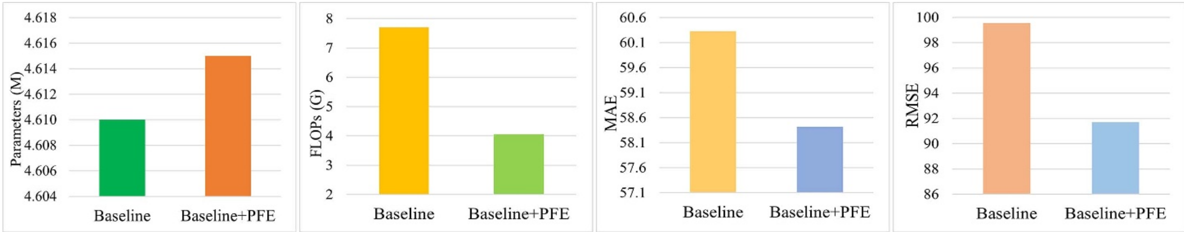

Figure 6. Ablation study on key modules of the PFE module.

Table 8. Ablation study of the federated learning framework.

	$n=2$		$n=4$		$n=8$	
Clients	MAE	RMSE	MAE	RMSE	MAE	RMSE
Avg.	71.07	119.84	122.70	233.60	169.56	360.03
PANet (Ours)	**61.90**	**101.52**	**63.59**	**104.90**	**70.40**	**120.22**

actual values. This indicates that PANet is effective and superior across various scenarios.

4.6. Ablation study

To evaluate the effectiveness of the PFE module and the federated learning framework, we conducted ablation studies on the ShanghaiTech Part A dataset. The results are displayed in figure 6 and table 8. The baseline refers to a network containing only the encoder and decoder.

Figure 6 shows the baseline scores of 60.32 and 99.56.0 in MAE and RMAE, respectively. When the PFE module is incorporated, MAE decreases by 3.15%, and RMSE decreases by 7.89%. In the PFE module, we adjusted the baseline channels from 512 to 256. This adjustment reduced FLOPs by 47.4% with only a 0.11% increase in parameters and simultaneously improved the performance of the PANet.

To evaluate the impact of the number of clients on model performance and the effectiveness of the federated learning framework, we conducted experiments on the ShanghaiTech PartA dataset. The Avg method randomly splits the training set of the Part A dataset into n equal parts, uses one part for training, and tests on the entire test set. This experiment was repeated 8 times, and the average result was calculated. In the federated learning framework, n represents the number of clients. The results for different numbers of clients (n = 2, 4, 8) using both the Avg method and the proposed PANet with federated learning framework are shown in table 8.

As the number of clients increases, the MAE and RMSE values for both methods also increase, which indicates a decline in model performance. This decline is due to the more pronounced non-independent and identically distributed data characteristics as the number of clients increases, which poses greater challenges for global model aggregation. Across all client numbers, our method consistently outperforms the Avgdataset method in terms of MAE and RMSE values, which further validates the effectiveness of the federated learning framework.

5. Conclusion

In this paper, we presented a lightweight PANet for vehicle counting to achieve efficient performance on edge devices. Additionally, the federated learning framework alleviates the pressure on individual devices processing data from various sources while protecting data privacy. Specifically, the PFE module improves network accuracy while optimizing network output to reduce computational load. Moreover, the proposed federated learning framework distributes computational load, which enhances model training efficiency and reduces the burden on individual nodes. Additionally, it ensures data privacy by aggregating parameters without sharing local datasets. Experimental results from five vehicle datasets and two crowd datasets show that PANet demonstrated significant advantages in effectiveness and accuracy.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iD

Mingliang Gao https://orcid.org/0000-0001-7273-7499

References

[1] Tituana D E V, Yoo S G and Andrade R O 2022 Vehicle counting using computer vision: a survey *2022 IEEE 7th Int. Conf. for Convergence in Technology (I2CT)* (IEEE) pp 1–8

[2] Xu H, Cai Z, Li R and Li W 2022 Efficient CityCam-to-edge cooperative learning for vehicle counting in ITS *IEEE Trans. Intell. Transp. Syst.* **23** 16600–11

[3] Guo X, Gao M, Zhai W, Li Q and Jeon G 2023 Scale region recognition network for object counting in intelligent transportation system *IEEE Trans. Intell. Transp. Syst.* **24** 15920–29

[4] Hsieh M-R, Lin Y-L and Hsu W H 2017 Drone-based object counting by spatially regularized regional proposal network *Proc. IEEE Int. Conf. on Computer Vision* (*Venice, Italy, 22–29 October 2017*) pp 4165–73

[5] Chaudhuri Y, Kumar A, Phukan O C and Buduru A B 2024 A lightweight feature fusion architecture for resource-constrained crowd counting (https://doi.org/10.48550/arXiv.2401.05968)

[6] Venkatesh S and Sankara Babu B 2022 A survey: vehicle detection and counting *2022 13th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)* (*Kharagpur, India, 3–5 October 2022*) (IEEE) pp 1–5

[7] Zhang Y, Zhou D, Chen S, Gao S and Ma Y 2016 Single-image crowd counting via multi-column convolutional neural network *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*Las Vegas, NV, USA, 27–30 June 2016*) pp 589–97

[8] Li Y, Zhang X and Chen D 2018 Csrnet: dilated convolutional neural networks for understanding the highly congested scenes *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*Salt Lake City, UT, USA, 18–23 June 2018*) pp 1091–100

[9] Zhai W, Xing X and Jeon G 2024 Region-aware quantum network for crowd counting *IEEE Trans. on Consumer Electronics* (https://doi.org/10.1109/TCE.2024.3378166)

[10] Guo X, Gao M, Zou G, Bruno A, Chehri A and Jeon G 2023 Object counting via group and graph attention network *IEEE Trans. Neural Netw. Lear. Syst.* (https://doi.org/10.1109/TNNLS.2023.3336894)

[11] Zhai W, Gao M, Li Q, Jeon G and Anisetti M 2023 FPANet: feature pyramid attention network for crowd counting *Appl. Intell.* **53** 19199–216

[12] Chen J, Gao M, Guo X, Zhai W, Li Q and Jeon G 2023 Object counting in remote sensing via selective spatial-frequency pyramid network *J. Softw. Pract. Exper.* **54** 1754–73

[13] La H-P, Ha M-T, Nguyen H-L and Nguyen M-T 2020 Vehicle counting: survey and experiments *2020 7th NAFOSTED Conf. on Information and Computer Science (NICS)* (*Ho Chi Minh City, Vietnam, 26–27 November 2020*) (IEEE) pp 350–5

[14] Zhou X, Ye X, Kevin I, Wang K, Liang W, Nair N K C, Shimizu S, Yan Z and Jin Q 2023 Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications *IEEE Trans. Comput. Soc. Syst.* **10** 1742–51

[15] Chen J *et al* 2024 Privacy-aware crowd counting by decentralized learning with parallel transformers *Internet Things* **26** 101167

[16] Zhou X, Liang W, Kevin I, Wang K and Yang L T 2020 Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations *IEEE Trans. Comput. Soc. Syst.* **8** 171–8

[17] Pang Y, Ni Z and Zhong X 2023 Federated learning for crowd counting in smart surveillance systems *IEEE Internet Things J.* **11** 5200–9

[18] Tsai C-M and Yeh Z-M 2013 Intelligent moving objects detection via adaptive frame differencing method *Intelligent Information and Database Systems: 5th Asian Conf., ACIIDS 2013, Proceedings Part I (Kuala Lumpur, Malaysia, 18–20 March 2013)* vol 5 (Springer) pp 1–11

[19] Cucchiara R, Grana C, Piccardi M and Prati A 2000 Statistic and knowledge-based moving object detection in traffic scenes *ITSC 2000. 2000 IEEE Intelligent Transportation Systems. Proc. (Cat. No. 00TH8493)* (*Dearborn, MI, USA, 1–3 October 2000*) (IEEE) pp 27–32

[20] Zheng Y and Peng S 2012 Model based vehicle localization for urban traffic surveillance using image gradient based matching *2012 15th Int. IEEE Conf. on Intelligent Transportation Systems* (*Anchorage, AK, USA, 16–19 September 2012*) (IEEE) pp 945–50

[21] Toropov E, Gui L, Zhang S, Kottur S and Moura J M F 2015 Traffic flow from a low frame rate city camera *2015 IEEE Int. Conf. on Image Processing (ICIP)* (*Quebec City, QC, Canada, 27–30 September 2015*) (IEEE) pp 3802–6

[22] Chen Z, Ellis T and Velastin S A 2012 Vehicle detection, tracking and classification in urban traffic *2012 15th International IEEE Conf. on Intelligent Transportation Systems* (*Anchorage, AK, USA, 16–19 September 2012*) (IEEE) pp 951–6

[23] Yi J, Shen Z, Chen F, Zhao Y, Xiao S and Zhou W 2023 A lightweight multiscale feature fusion network for remote sensing object counting *IEEE Trans. Geosci. Remote Sens.* **61** 1–13

[24] Qi P, Chiaro D, Guzzo A, Ianni M, Fortino G and Piccialli F 2023 Model aggregation techniques in federated learning: a comprehensive survey *Future Gener. Comput. Syst.* **150** 272–93

[25] Arapakis I, Papadopoulos P, Katevas K and Perino D 2023 P4l: privacy preserving peer-to-peer learning for infrastructureless setups (arXiv:2302.13438)

[26] Zhou X, Liang W, Kevin I, Wang K, Yan Z, Yang L T, Wei W, Ma J and Jin Q 2023 Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems *IEEE Wirel. Commun.* **30** 82–89

[27] Woisetschläger H, Isenko A, Wang S, Mayer R and Jacobsen H-A 2024 A survey on efficient federated learning methods for foundation model training (https://doi.org/10.24963/ijcai.2024/919)

[28] Wang Y, Lin L and Chen J 2022 Communication-efficient adaptive federated learning *Int. Conf. on Machine Learning* (PMLR) pp 22802–38

[29] Wang Q, Li Q, Wang K, Wang H and Zeng P 2021 Efficient federated learning for fault diagnosis in industrial cloud-edge computing *Computing* **103** 2319–37

[30] Wang C-H, Huang K-Y, Yao Y, Chen J-C, Shuai H-H and Cheng W-H 2022 Lightweight deep learning: an overview *IEEE Consum. Electron. Mag.* **13** 51–64

[31] Howard A G, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M and Adam H 2017 Mobilenets: efficient convolutional neural networks for mobile vision applications (https://doi.org/10.48550/arXiv.1704.04861)

[32] Zhang X, Zhou X, Lin M and Sun J 2018 Shufflenet: an extremely efficient convolutional neural network for mobile devices *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 6848–56

[33] Han K, Wang Y, Tian Q, Guo J, Xu C and Xu. C 2020 Ghostnet: more features from cheap operations *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (*Seattle, WA, USA, 13–19 June 2020*) pp 1580–9

[34] Tang Y, Han K, Guo J, Xu C, Xu C and Wang Y 2022 Ghostnetv2: enhance cheap operation with long-range attention *Advances in Neural Information Processing Systems* **vol 35** pp 9969–82

[35] Zhou K, Yang Y, Cavallaro A and Xiang T 2021 Learning generalisable omni-scale representations for person re-identification *IEEE Trans. Pattern Anal. Mach. Intell.* **44** 5056–69

[36] Liang D, Xu W, Zhu Y and Zhou Y 2022 Focal inverse distance transform maps for crowd localization *IEEE Trans. Multimedia* (https://doi.org/10.1109/TMM.2022.3203870)

[37] Gao G, Liu Q and Wang Y 2021 Counting from sky: a large-scale data set for remote sensing object counting and a benchmark method *IEEE Trans. Geosci. Remote Sens.* **59** 3642–55

[38] Guerrero-Gómez-Olmedo R, Torre-Jiménez B, López-Sastre R, Maldonado-Bascón S and Onoro-Rubio D 2015 Extremely overlapping vehicle counting *Pattern Recognition and Image Analysis: 7th Iberian Conf., IbPRIA 2015, Proc. 7 (Santiago de Compostela, Spain, 17–19 June 2015)* (Springer) pp 423–31

[39] Idrees H, Saleemi I, Seibert C and Shah M 2013 Multi-source multi-scale counting in extremely dense crowd images *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*Portland, OR, USA, 23–28 June 2013*) pp 2547–54

[40] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (https://doi.org/10.48550/arXiv.1412.6980 Focus to learn more)

[41] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Adv. Neural Inf. Process. Syst.* **32** 8024–35

[42] Liu W, Salzmann M and Fua P 2019 Context-aware crowd counting *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (*Long Beach, CA, USA*, *15–20 June 2019*) pp 5099–108

[43] Song Q, Wang C, Wang Y, Tai Y, Wang C, Li J, Wu J and Ma J 2021 To choose or to fuse? Scale selection for crowd counting *Proc. AAAI Conf. on Artificial Intelligence* vol 35 pp 2576–83

[44] Ma Z, Wei X, Hong X and Gong Y 2019 Bayesian loss for crowd count estimation with point supervision *Proc. IEEE/CVF Int. Conf. on Computer Vision* (*Seoul, Korea (South)*, *27 October –2 November 2019*) pp 6142–51

[45] Wang P, Gao C, Wang Y, Li H and Gao Y 2020 Mobilecount: an efficient encoder-decoder framework for real-time crowd counting *Neurocomputing* **407** 292–9

[46] Lin C and Hu X 2024 Efficient crowd density estimation with edge intelligence via structural reparameterization and knowledge transfer *Appl. Soft Comput.* **154** 111366

[47] Shen Z, Xu Y, Ni B, Wang M, Hu J and Yang X 2018 Crowd counting via adversarial cross-scale consistency pursuit *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*Salt Lake City, UT, USA*, *18–23 June 2018*) pp 5245–54

[48] Meng X and Ren Z 2021 MDCount: a lightweight encoder-decoder architecture for resource-saving crowd counting *J. Phys.: Conf. Ser* 2024 012031

[49] Wang Q, Lin W, Gao J and Li X 2020 Density-aware curriculum learning for crowd counting *IEEE Trans. Cybern.* **52** 4675–87

[50] Redmon J, Divvala S, Girshick R and Farhadi A 2016 You only look once: unified, real-time object detection *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*Las Vegas, NV, USA*, *27–30 June 2016*) pp 779–88

[51] Ren S, He K, Girshick R and Sun J 2016 Faster R-CNN: towards real-time object detection with region proposal networks *IEEE Trans. Pattern Anal. Mach.* **39** 1137–49

[52] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y and Berg A C 2016 SSD: single shot multibox detector *Computer Vision–ECCV 2016: 14th European Conf., Proceedings Part I (Amsterdam, The Netherlands, 11–14 October 2016),* vol 14 (Springer) pp 21–37

[53] Stahl T, Pintea S L and Van Gemert J C 2018 Divide and count: generic object counting by image divisions *IEEE Trans. Image Process.* **28** 1035–44

[54] Nathan Mundhenk T, Konjevod G, Sakla W A and Boakye K 2016 A large contextual dataset for classification, detection and counting of cars with deep learning *Computer Vision–ECCV 2016: 14th European Conf., (Proceedings Part III) (Amsterdam, The Netherlands, 11–14 October 2016)* vol 14 (Springer) pp 785–800

[55] Hsieh M-R, Lin Y-L and Hsu W H 2017 Drone-based object counting by spatially regularized regional proposal network *Proc. IEEE Int. Conf. on Computer Vision* pp 4145–53

[56] Lin T-Y, Goyal P, Girshick R, He K and Dollár P 2017 Focal loss for dense object detection *Proc. IEEE Int. Conf. on Computer Vision* pp 2980–8

[57] Cao X, Wang Z, Zhao Y and Su F 2018 Scale aggregation network for accurate and efficient crowd counting *Proc. European Conf. on Computer Vision (ECCV)* pp 734–50

[58] Gao J, Wang Q and Yuan Y 2019 SCAR: spatial-/channel-wise attention regression networks for crowd counting *Neurocomputing* **363** 1–8

[59] Wang Q, Gao J, Lin W and Yuan Y 2021 Pixel-wise crowd understanding via synthetic data *Int. J. Comput. Vis.* **129** 225–45

[60] Zhu L, Zhao Z, Lu C, Lin Y, Peng Y and Yao T 2019 Dual path multi-scale fusion networks with attention for crowd counting (https://doi.org/10.48550/arXiv.1902.01115)

[61] Sindagi V A and Patel V M 2017 Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting *2017 14th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE) pp 1–6

[62] Chen X, Bin Y, Sang N and Gao C 2019 Scale pyramid network for crowd counting *2019 IEEE Winter Conf. on Applications of Computer Vision (WACV)* (*Waikoloa, HI, USA*, *7–11 January 2019*) (IEEE) pp 1941–50

[63] Fiaschi L, Köthe U, Nair R and Hamprecht F A 2012 Learning to count with regression forest and structured labels *Proc. 21st Int. Conf. on Pattern Recognition (ICPR2012)* (IEEE) pp 2685–8

[64] Lempitsky V and Zisserman A 2010 Learning to count objects in images *Proc. 23rd Int. Conf. on Neural Information Processing Systems* vol 1 pp 1324–32

[65] Onoro-Rubio D and López-Sastre R J 2016 Towards perspective-free object counting with deep learning *Computer Vision–ECCV 2016: 14th European Conf., (Proceedings Part VII) (Amsterdam, The Netherlands, October 11–14 2016)* vol 14 (Springer) pp 615–29

[66] Zhang S, Wu G, Costeira J P and Moura J M F 2017 Understanding traffic density from large-scale web camera data *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*Honolulu, HI, USA*, *21–26 July 2017*) pp 5898–907

[67] Babu Sam D and Venkatesh Babu R 2018 Top-down feedback for crowd counting convolutional neural network *Proc. AAAI Conf. on Artificial Intelligence* vol 32

[68] Ma X, Du S and Liu Y 2019 A lightweight neural network for crowd analysis of images with congested scenes *2019 IEEE Int. Conf. on Image Processing (ICIP)* (*Taipei, Taiwan*, *22–25 September 2019*) (IEEE) pp 979–83

[69] Shi X, Li X, Wu C, Kong S, Yang J and He. L 2020 A real-time deep network for crowd counting *ICASSP 2020-2020 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (*Barcelona, Spain*, *4–8 May 2020*) (IEEE) pp 2328–32

[70] Liu L, Chen J, Wu H, Chen T, Li G and Lin L 2020 Efficient crowd counting via structured knowledge transfer *Proc. 28th ACM Int. Conf. on Multimedia* pp 2645–54

[71] Gao J, Wang Q and Li X 2019 Pcc net: perspective crowd counting via spatial convolutional network *IEEE Trans. Circuits Syst. Video Technol.* **30** 3486–98