Traffic Density Estimation by Distributed Proxy Model Learning for Internet of Vehicle

Qilei Li, Jing-an Cheng, Mingliang Gao, *Senior Member, IEEE*, Jinyong Chen, and Gwanggil Jeon

***Abstract*—Autonomous driving has been significantly advanced in today's society, which revolutionized daily routines and facilitated the development of the Internet of Vehicles (IoV). A crucial aspect of this system is understanding traffic density to enable intelligent traffic management. With the rapid improvement in deep neural networks (DNNs), the accuracy of density estimation has markedly improved. However, there are two main issues that remain unsolved. First, current DNN-based models are excessively heavy, characterized by an overwhelming number of training parameters (millions or even billions) and substantial computational complexity, indicated by a high number of FLOPs. These requirements for storage and computation severely limit the practical application of these models, especially on edge devices with limited capacity and computational power. Second, despite the superior performance of DNN models, their effectiveness largely depends on the availability of large-scale data for training. Growing privacy concerns have made individuals increasingly hesitant to allow their data to be publicly used for model training, particularly in vehicle-related applications that might reveal personal movements, which leads to data isolation issues. In this article, we address these two problems at once with a systematic framework. Specifically, we introduce the proxy model distributed learning (PMDL) model for traffic density estimation. PMDL model is composed of two main components. First, we introduce a proxy model learning strategy that transfers fine-grained knowledge from a larger master model to a lightweight proxy model, i.e., a proxy model. Second, we design a distributed learning strategy that trains multiple proxy models with privacy-aware local data and seamlessly aggregates these models via a global parameter server. This ensures privacy protection while significantly improving estimation performance compared to training models with limited, isolated data. We tested the proposed model on four major vehicle density analysis benchmarks and demonstrated its efficiency by outperforming other state-of-the-art competitors. The code is available at `https://github.com/jinyongch/DPML`.**

***Index Terms*—Distributed learning, Internet of Vehicle, knowledge distillation (KD), lightweight model, traffic analysis, vehicle counting.**

Received 27 July 2024; revised 12 October 2024; accepted 12 February 2025. Date of publication 21 February 2025; date of current version 9 June 2025. *(Corresponding author: Mingliang Gao.)*

Qilei Li is with the School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000, China, and also with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: q.li@qmul.ac.uk).

Jing-an Cheng, Mingliang Gao, and Jinyong Chen are with the School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000, China (e-mail: mlgao@sdut.edu.cn; 23404020560@stumail.sdut.edu.cn; jinyongch@outlook.com).

Gwanggil Jeon is with the Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, South Korea (e-mail: ggjeon@gmail.com).

Digital Object Identifier 10.1109/JIOT.2025.3542534

I. Introduction

TRAFFIC density estimation is a fundamental component of Internet of Vehicle (IoV) [1]. It also plays a crucial role in urban road monitoring and round-the-clock satellite surveillance. As shown in Fig. 1, this technique allows for the real-time tracking and analysis of vehicular traffic to improve traffic management efficiency. Furthermore, density estimation offers accurate ground traffic data in satellite monitoring, which is essential for urban planning and emergency management. These applications significantly influence road planning, intelligent route selection, traffic management, and the operation of vehicular networks [2].

With the rapid advancement of machine learning and deep learning technologies, significant progress has been made in traffic density estimation [3]. Nevertheless, current deep learning models [4], [5], [6] often possess a large number of parameters, which makes them less suitable for deployment on resource-constrained edge devices [7]. Particularly in scenarios demanding real-time processing, the lack of sufficient computational and storage capabilities on edge devices becomes evident. Most SOTA methods achieve high precision by incorporating numerous parameters, which severely restricts their real-time processing capacity. To balance parameter numbers and accuracy, knowledge distillation (KD) transfers knowledge from a well-trained teacher model to a simpler student model [8]. The student model achieves performance close to the teacher with fewer parameters and reduced computational needs [9]. This optimization makes it suitable for edge devices with limited resources. Moreover, researchers are diligently creating efficient and simplified lightweight models to the demands of edge computing environments [10]. These models excel in rapid responsiveness and reduced power consumption. They are ideal for real-time data processing on edge devices and meet the crucial needs of edge computing.

As privacy concerns grow and regulations become stricter, the challenge of efficiently estimating vehicle density while maintaining data privacy has emerged as a significant research topic. federated learning (FL) allows model training without centralizing data and effectively protects data privacy [11], [12], [13]. For example, Chen et al. [14] presented DLPTNet, which utilizes FL to provide accurate crowd counting without compromising user privacy. It trains models locally on each device and aggregates updates to the global model. This process avoids centralized storage and transmitting sensitive data, which reduces the risk of data breaches.

Based on the above context, we propose the proxy model distributed learning (PMDL) model for traffic density

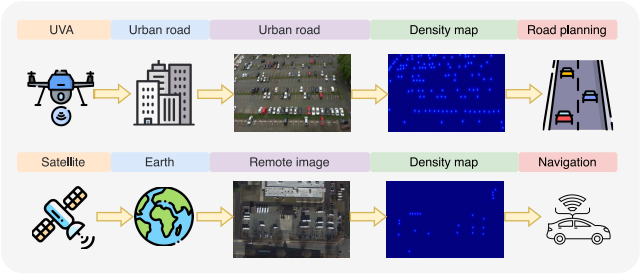

Fig. 1. Application of Vehicle Density Estimation in the IoV. Density estimation is crucial for urban road and satellite monitoring. It offers real-time vehicle flow analysis and precise traffic data to improve traffic management, urban planning, and emergency response.

estimation. The PMDL consists of three core components: 1) Master-Proxy KD; 2) federated proxy model optimization; and 3) mixture-of-expert inference. First, Master-Proxy KD involves a pretrained heavyweight master model and a lightweight proxy model. The proxy model learns from the master model and shares the final upsampling layer, which ensures precise vehicle density map generation while reducing the computational cost. Furthermore, FL is employed to preserve data privacy. Clients train local proxy models on their data and send only the backbone to a central server. The server aggregates these backbones into a global model, which is then redistributed to clients to enhance performance while maintaining privacy. Besides, the global backbone is integrated with multiple upsampling heads to form several expert models. These experts produce predictions that are averaged to improve reliability and estimate uncertainty, which is crucial for applications, such as autonomous driving.

The contributions of this article are three-fold.

1) The proposed PMDL model leverages a proxy learning method, transferring detailed knowledge from a comprehensive master model to a compact proxy model. This achieves a balance between parameter size and estimation accuracy.
2) We employ an FL framework that enables clients to train models locally using their own data and then aggregate the results to update the global model. This approach effectively addresses the issue of data silos while preserving data privacy.
3) We conducted extensive experiments on four vehicle benchmark datasets to validate the superior accuracy and robustness of the proposed PMDL model. Notably, this model achieves high performance with fewer parameters and lower FLOPs.

II. Related Works

A. Traffic Density Analysis

Traffic density analysis is a crucial component of modern intelligent transportation systems [15], which aims to optimize traffic management through accurate traffic density estimation. Early methods primarily relied on image processing approaches, such as background subtraction [16] and blob analysis [17]. However, these methods often face significant challenges in complex backgrounds. Sensor-based methods [18], which involve installing inductive loops and light detection and ranging devices (LIDAR) on roads to directly measure vehicle counts, yet have high installation and maintenance costs. Data-driven methods [19], on the other hand, predict future traffic density trends by analyzing historical traffic data, but they are heavily dependent on data quality and quantity and also raise privacy concerns. Currently, the mainstream approach is based on density map estimation methods [4], [20], [21], which use advanced machine learning algorithms to provide clear visualizations of traffic density and offer relatively accurate vehicle count estimates. However, these methods still face limitations in data collection and real-time performance. With the advent of IoV [22], a new solution for traffic density analysis has emerged. By leveraging real-time data obtained through IoV, traffic management systems can perform intelligent scheduling to optimize traffic flow. In this article, density map estimation and IoV are combined to expand traffic density analysis from static data processing to dynamic, real-time, and comprehensive traffic condition monitoring.

B. Distributed Deep Learning

Distributed deep learning (DDL) is a method of training deep learning models in parallel across multiple computing nodes, aimed at accelerating the training process and handling large-scale datasets [23]. The key goal of DDL is to effectively utilize computing resources and memory to improve the speed and performance of model training [24], [25]. FL, a significant form of distributed learning, maximizes the protection of each participant's data privacy. FL involves training machine learning models across multiple devices or nodes using local data, with model updates shared centrally without transferring raw data. Decentralized learning also distributes learning across nodes but without a central server, relying on peer-to-peer communication for model updates and coordination. Both aim to improve privacy and scalability but differ in their coordination methods. FL achieves distributed training and privacy protection by sharing only model parameters or gradients rather than the original data. McMahan et al. [26] introduced the basic concepts and framework of FL, proposing a federated averaging algorithm (FedAvg) that demonstrates how to train deep learning models on distributed devices efficiently. Li et al. [27] summarized the challenges, methods, and future directions of FL, including issues, such as privacy protection, communication efficiency, and system heterogeneity. Li et al. [28] studied the optimization of FL in heterogeneous networks, proposing a new algorithm to address differences in device performance and data quality. Abadi et al. [29] explored how to apply differential privacy techniques in deep learning to protect user data privacy while maintaining model performance.

C. Knowledge Distillation

KD is a technique where a large model (teacher) transfers its knowledge to a smaller model (student). This approach enables the student model to achieve performance comparable to the teacher by mimicking the teacher's output

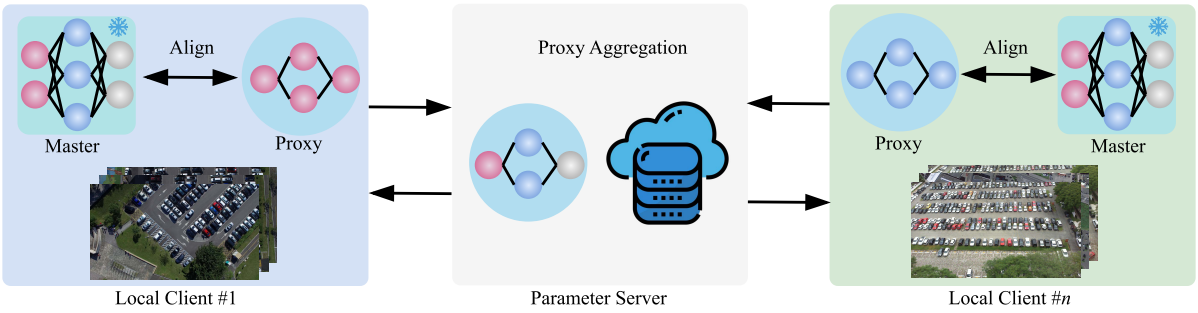

Fig. 2. Framework of the PMDL model. Local client n refers to n clients with identical network structures, training locally with their own data. Only model parameters are exchanged, while local data stays on the client. Align signifies the proxy model learning from the master model, which helps it learn fine-grained knowledge more effectively.

probabilities. Recent advances in KD include incorporating intermediate hints, attention transfer, and more sophisticated methods for enhancing student performance. For example, Tian et al. [30] introduced contrastive representation distillation, which focuses on aligning the students' representations with those of the teacher. Similarly, Cho and Hariharan [31] explored the efficacy of KD with early stopping, which demonstrates that early-stage teacher models can be effective for distillation. KD has facilitated the creation of lightweight models suitable for deployment on resource-constrained devices, which brings significant benefits to natural language processing, speech recognition, and computer vision. Jiao et al. [32] developed TinyBERT, a smaller and faster version of BERT, which achieves competitive performance on multiple NLP benchmarks. In speech recognition, Huang et al. [33] compared various KD techniques to compress acoustic models, which can enhance inference speed without compromising accuracy. In computer vision, KD has been particularly effective in creating efficient models for real-time applications. For example, Heo et al. [34] conducted an in-depth analysis of distillation techniques to reduce the computational burden of deep convolutional neural networks (CNNs). Additionally, recent methods like feature-map transfer and adversarial distillation have shown substantial improvements in tasks, such as image classification, object detection, and semantic segmentation. Gou et al. [9] conducted a comprehensive survey on these techniques, which emphasize the important role of KD in enabling high-performance models on edge devices with limited computational power.

D. KD for Internet of Vehicles

The application of KD in the IoV is particularly noteworthy. IoV systems require real-time data processing for tasks, such as autonomous driving, traffic management, and vehicle-to-vehicle communication. Lightweight models derived from KD can be deployed in vehicles to process visual data efficiently, and to enable advanced driver assistance systems (ADAS), as well as enhancing safety features. For instance, Chen et al. [35] demonstrated by using KD, one can create efficient perception models for autonomous driving, which can operate in real-time on edge devices within vehicles. Such models can accurately detect and classify objects, recognize traffic signs, and monitor driver behavior, all while maintaining low latency and high accuracy. These advancements in KD and lightweight model learning are crucial for supporting IoT, AIoT, and IoV scenarios, where computational resources are limited but real-time processing is essential. In this article, we applied KD to distil a lightweight from a larger master model to achieve real-time road traffic density analysis, which can provide a solid foundation for autonomous driving, and IoV.

III. Methodology

A. Overview of the Framework

In this article, we introduce a PMDL model to address computational complexity and data privacy in deep neural network (DNN)-based traffic density estimation. The overview of the framework is shown in Figs. 2 and 4, it is composed of three main components: 1) Master-Proxy KD; 2) federated proxy model optimizing; and 3) mixture-of-expert inference.

1) The Master-Proxy KD involves a pretrained heavyweight master model and a lightweight proxy model. The proxy model learns from the master model while sharing the final upsampling layer. This ensures accurate vehicle density map generation with reduced computational burden.
2) To maintain data privacy, we employ FL in the Federated Proxy Model Optimizing component. Clients train local proxy models on their data and send only the backbone to a central server. The server aggregates these to form a global backbone, which is then redistributed to the clients. This process enhances model performance while protecting data privacy.
3) In the Mixture-of-Expert Inference component, the global backbone is combined with multiple upsampling heads to create several expert models. Each expert generates predictions, which are averaged to improve reliability and estimate uncertainty. This is essential for high-stakes applications like autonomous driving.

B. Master-Proxy Knowledge Distillation

To achieve lightweight vehicle density map generation and counting, we design a Master-Proxy KD (MPKD) Framework. As demonstrated in Figure 3, this approach, heavyweight master model $M = \{\mathbf{B}_M, \mathbf{U}\}$, and a lightweight proxy model

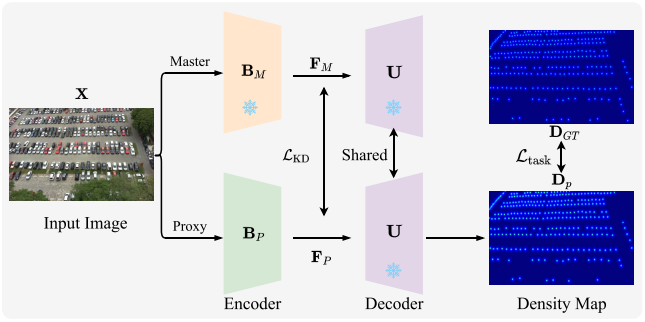

Fig. 3. Structure of the Master-Proxy KD framework. Both the master (top) and proxy (bottom) models are trained on local paired data. After pretraining the master model, the proxy model is then refined by fine grained knowledge from the master model supervised by the loss function $\mathcal{L}_{\text{KD}}$. Ultimately, the density map generated by the proxy model is aligned with the GT optimized by the loss function $\mathcal{L}_{\text{task}}$.

$P = \{\mathbf{B}_P, \mathbf{U}\}$. The final upsampling layer $\mathbf{U}$ for density map generation is shared among the master model and the proxy model. Once the master model is trained, it will remain frozen during the proxy model training. Let $\mathbf{F}_M$ and $\mathbf{F}_P$ represent the feature maps generated by the master and proxy backbones models, respectively, immediately before the shared upsampling layer $\mathbf{U}$ as

$$\mathbf{F}_{\text{M}} = \mathbf{B}_M(\mathbf{X}), \qquad \mathbf{F}_{\text{P}} = \mathbf{B}_P(\mathbf{X}). \tag{1}$$

The goal is to minimize the discrepancy between these feature maps $\{\mathbf{F}_{\text{M}}, \mathbf{F}_{\text{P}}\}$, and to ensure that the proxy model performs well on the vehicle density map generation task. The shared upsampling layer $\mathbf{U}$ transforms the feature maps $\{\mathbf{F}_M, \mathbf{F}_P\}$ into the respective density maps $\{D_{\text{M}}, D_{\text{P}}\}$

$$\mathbf{D}_M = \mathbf{U}(\mathbf{F}_M), \qquad \mathbf{D}_{\text{P}} = \mathbf{U}(\mathbf{F}_P). \tag{2}$$

The distillation process is jointly supervised by two learning objectives. The first one, KD Loss, denoted as $\mathcal{L}_{\text{KD}}$, calculates the mean square error (MSE) between the feature maps $\mathbf{F}_M$ and $\mathbf{F}_P$ as

$$\mathcal{L}_{\text{KD}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{F}_M^i - \mathbf{F}_P^i\right|^2} \tag{3}$$

where N is the number of elements in the feature maps. The second one, Task Loss, denoted as $\mathcal{L}_{\text{task}}$, calculates the MSE between the predicted density maps $\mathbf{D}_{\text{P}}$ and the ground truth (GT) density maps $\mathbf{D}_{\text{GT}}$ as

$$\mathcal{L}_{\text{task}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|\mathbf{D}_P^i - \mathbf{D}_{GT}^i\right|^2}. \tag{4}$$

The overall objective is to minimize the sum of these two losses

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{task}}. \tag{5}$$

By leveraging the pretrained knowledge of the master model and the efficiency of the proxy model, our framework ensures accurate vehicle density map generation while maintaining computational efficiency. The shared upsampling layer $\mathbf{U}$ acts as a conduit for transferring knowledge, so as to allow the proxy model to produce high-quality density maps without extensive additional training. This method effectively combines the strengths of both models, and achieves a balance between performance and efficiency.

C. Federated Proxy Model Optimizing

To address privacy concerns and data isolation issues in vehicle density estimation, we further integrate an FL strategy in our PMDL model, to achieve federated proxy model optimizing. This enables collaborative learning across multiple clients without exposing sensitive local data to a central server.

Assume there are K clients available, each client trains its local proxy model $\mathbf{P}_i$ on its dataset $\mathcal{D}_i = \{\mathbf{X}_i^j, \mathbf{Y}_i^j\}$, by iteratively optimizing the model's performance for vehicle density map generation with (5). Specifically, each client i locally trains its proxy model $\mathbf{P}_i$, and uploads only the backbone $\mathbf{B}_P^i$ (excluding the final upsampling layer $\mathbf{U_i}$) to the central server. The central server aggregates these backbones to form a global backbone $\mathbf{B}_P^g$, which is then sent back to the clients. This iterative model exchange process ensures that the model benefits from diverse data while maintaining data privacy. Under this FL context, the local training loss for client i, denoted as $\mathcal{L}_i$, is computed based on its dataset $\mathcal{D}_i$, and the learning objective for the client i is formulated as

$$\mathcal{L}_i = \sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i}\left|\mathbf{D}_{P_i}^j - \mathbf{D}_{GT_i}^j\right|^2} + \sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i}\left|\mathbf{F}_M^j - \mathbf{F}_P^j\right|^2} \tag{6}$$

where n_i is the number of samples in dataset $\mathbf{D}_i$, $\mathbf{D}_{P_i}$ is the predicted density map by the proxy model $\mathbf{P}_i$, and $\mathbf{D}_{GT_i}$ is the GT density map. After local training, each client sends its updated backbone to the central server. The global backbone $\mathbf{P}_B^g$ is computed as a weighted average of the local backbones

$$\mathbf{P}_B^g = \sum_{i=1}^{K}\frac{n_i}{N}\mathbf{P}_{B_i} \tag{7}$$

where K is the number of clients, n_i is the number of samples at client i, and $N = \sum_{i=1}^{K} n_i$ is the total number of samples across all clients. Each client then updates its local backbone $\mathbf{P}_{B_i}$ with the global backbone $\mathbf{P}_B^g$

$$\mathbf{P}_{B_i} \leftarrow \mathbf{P}_B^g. \tag{8}$$

This federated optimization process repeats for multiple communication rounds until convergence. By sharing only the backbones and not the final upsampling layer $\mathbf{U}$, the proposed PMDL framework maintains the consistency of the upsampling process across all clients, so as to lead to high-quality vehicle density maps. This method balances performance, efficiency, and privacy, thereby making it well-suited for real-world applications where data sensitivity is critical.

D. Mixture-of-Expert Inference

In inference stage, once the central parameter server aggregates the local models, we obtain a global backbone denoted as $\mathbf{P}_B^g$. Benefit from the disentangled design of the proxy model $M_i = \{B_{M_i}, \mathbf{U}_i\}$, during inference, we can combine $\mathbf{P}_B^g$ with the

TABLE I
STATISTICAL OVERVIEW OF THE EMPLOYED DATASET

Dataset	#Images	#Train	#Test	Avg. Resolution	#Total
Small-Vehicle [41]	280	222	58	2473 × 2339	148,838
Large-Vehicle [41]	172	108	64	1552 × 1573	16,594
CARPK [42]	1,448	989	459	720 × 1280	89,777
PUCPR+ [42]	125	100	25	720 × 1280	16,456
ShanghaiTech Part A [43]	482	300	182	589 × 868	241,677

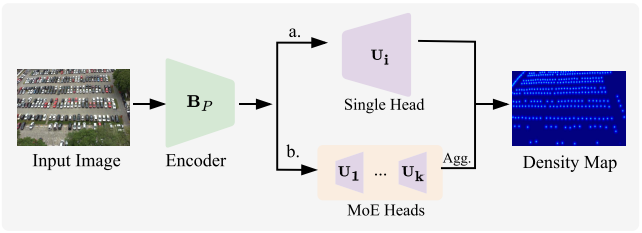

Fig. 4. Structure of the conditional model inference. (a) If the novel test data are allowed to access multiple clients, it can generate multiple predictions by utilizing the shared backbone and the accessible heads (experts). The final output is then determined through consensus among these predictions. (b) In contrast, if the novel test data are restricted to a single client, only one prediction will be generated using the corresponding available head.

upsampling heads $\{\mathbf{U}_i\}_{i=1}^{K}$ to form K complete experts. Each expert can then generate a prediction d_i for a new sample x as follows:

$$\{d_i\}_{i=1}^{K} = \{\mathbf{U}_i(\mathbf{P}_B^g(x))\}_{i=1}^{K}$$
$$D = \mathbb{E}\left(\sum_{i=1}^{K} d_i\right) = \sum_{i=1}^{K} \mathbb{E}(d_i)$$
$$\mathrm{Var}(D) = \sum_{i=1}^{K} \mathrm{Var}(d_i) + 2 \sum_{1\le i<j\le K} \mathrm{Cov}(d_i, d_j). \quad (9)$$

where $\mathbb{E}(\cdot)$ stands for the expected value, $\mathrm{Var}(\cdot)$ indicates the variance, and $\mathrm{Cov}(\cdot)$ represents the covariance. This can result in a set of K predictions, allowing us to achieve a more reliable final prediction by averaging these outputs D. This approach minimizes the risk of inaccurate predictions and provides an estimate of uncertainty by calculating the standard deviation. This capability is often missing in current DNN-based methods but is crucial for applications requiring high reliability, such as autonomous driving.

E. Ground Truth Generation

We generate the GT density maps as supervision by using the conventional focal inverse distance transform map method [36], which can achieve a precise representation of vehicle density by accounting for the distance of each pixel from the nearest annotated head location, thereby providing a more refined density estimation. The density map generation is formulated as follows:

$$F_{\mathrm{gt}} = \frac{1}{P(x,y)^{\alpha \times P(x,y)+\beta} + C} \quad (10)$$

where α and β are empirically set to 0.02 and 0.75, respectively, to align with the previous studies [14], [37]. The constant C is set to 1 to avoid division by zero and to ensure numerical stability. $P(x, y)$ denotes the Euclidean distance between a pixel at coordinates (x, y) and the closest annotated object location (x', y').

IV. EXPERIMENTS

A. Implementation Details

For the master model, we employed the larger `OSNetx1` [38], while for the proxy model, we utilized the more lightweight `OSNetx0.5` [38] variant. The master model has 4.66 M trainable parameters with the Flops of 57.61 G, and the lightweight proxy counterpart has 1.19 M trainable parameters with the Flops of 15.01 G. Once the master model was pretrained using the local data, it will be used to supervise the training of the proxy model. Both models were trained for 3000 epochs to ensure the convergence, with a learning rate of 1e-4, and optimized by the Adam algorithm [39] for parameter updates. During training, input images were randomly cropped to a spatial resolution of 576 × 768 and horizontally flipped for data augmentation. The batch size was set to 8. The coefficients for the distillation loss and regression loss were both initialized as 1, and optimized iteratively throughout the training process. All experiments were conducted using PyTorch [40] on an NVIDIA A100 80 GPU.

B. Datasets

We employed a wide range of dataset to benchmark the proposed PMDL model and compared against other state-of-the-art (SOTA) methods. The statistical overview of the datasets is shown in Table I.

CARPK dataset [42] originates from various regions in Hong Kong, such as parking lots and streets, and includes images of vehicles in different scenes and lighting conditions. It consists of 1448 drone view images, with 989 images for training and 459 images for testing.

PUCPR+ dataset [42] contains a large number of video sequences and images covering different driving scenarios and perspectives. It includes 125 images with a total of 16 456 annotations. Among these images, 100 are used for training, and 25 are reserved for testing.

Large-Vehicle dataset [41] is an open dataset specifically for the study and development of large vehicle detection, recognition, and tracking. The dataset comprises 172 remote sensing images, with each image having an average resolution of 1552 × 1573 pixels.

Small-Vehicle dataset [41] is another remote sensing vehicle counting dataset. It contains 280 high-resolution images with a

TABLE II
EFFICIENCY COMPARISON RESULTS OF VARIOUS METHODS

Methods	Params (M) ↓	FLOPs (G) ↓	Time(ms) ↓	FPS ↑
RAQNet [4]	28.30	250.80	35.20	28.30
SRRNet [5]	66.14	162.09	37.07	26.93
BL [44]	21.50	182.19	15.69	63.75
CSRNet [45]	16.26	182.69	15.07	66.35
CAN [46]	18.10	193.65	17.80	56.17
SASNet [6]	38.90	393.20	45.94	21.77
PMDL (Ours)	**4.38**	**44.28**	**11.87**	**84.28**

total of 148 838 small vehicles. Compared to the large vehicle dataset, it exhibits greater scale variation.

ShanghaiTech Part A [43] dataset mainly refers to datasets released by ShanghaiTech University for computer vision tasks. It is a well-known dataset for crowd counting and density estimation. It consists of 300 training images and 182 testing images.

C. Evaluation Metrics

We used mean absolute error (MAE) and root MSE (RMSE) are utilized as evaluation metrics. MAE is calculated as the mean of the absolute differences between the predicted values and the actual values across all test samples

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|x_i - \hat{x}_i\right|. \tag{11}$$

The RMSE value is computed as the square root of the average squared difference between the predicted values and the true values

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|x_i - \hat{x}_i\right|^2} \tag{12}$$

where N signifies the number of test images, and x_i and $\hat{x}_i$, respectively, represent the predicted and actual values for the ith image.

D. Efficiency Evaluation

A comparative analysis was performed to validate the efficiency of the proposed PMDL by benchmarking it against SOTA methods. The experiments utilize an RTX3090 device, with the output size specified as 576×768. PMDL's performance was assessed by evaluating the model's parameters, floating-point operations (FLOPs), inference time, and frames per second (FPS). The results of the experiments are detailed in Table II. PMDL has much fewer parameters compared to the other SOTA methods, with just 4.38M, and it requires only 44.28 FLOPs. This implies that PMDL is more compact and better suited for deployment in real-world situations with limited resources. Moreover, PMDL achieves the fastest inference time of 11.87 ms and the highest FPS of 84.28. This highlights PMDL's superior speed and efficiency for real-time processing.

TABLE III
COMPARISON RESULTS OF VARIOUS METHODS ON SMALL-VEHICLE AND LARGE-VEHICLE DATASETS

Method	Large Vehicle		Small Vehicle	
	MAE	RMSE	MAE	RMSE
SCAR [47]	62.78	79.46	497.22	1276.65
CMTL [48]	61.02	78.25	490.53	1321.11
MCNN [43]	36.56	55.55	488.65	1317.44
SPN [49]	36.21	50.56	445.16	1252.92
CAN [46]	34.56	49.63	457.36	1260.39
CSRNet [45]	34.10	46.42	443.81	1252.22
SFCN [50]	33.93	49.74	440.70	1248.27
SANet [51]	62.78	79.65	497.22	1276.66
ASPDNet [41]	31.76	40.14	433.23	1238.61
SFANet [52]	29.04	47.01	435.29	1284.15
SRRNet [5]	18.25	31.24	122.79	419.65
PMDL (Ours)	**12.95**	**24.03**	**104.09**	**416.78**

E. Comparison With State-of-the-Art Methods

Comparison on Vehicle Counting: Table III shows the results of the experiments performed on the large vehicle and small vehicle datasets. On the small vehicle dataset, PMDL has the lowest MAE of 104.09 and the lowest RMSE of 416.78. Compared to ASPDNet [41], PMDL improves the MAE by 75.97% and the RMSE by 66.35%. PMDL achieved the best performance on the large vehicle dataset with an MAE of 12.95 and an RMSE of 24.03. Although, ASPDNet [41] can capture multiscale features, its accuracy may be limited in small or dense vehicle scenarios. In contrast, PMDL leverages the teacher model to supervise the student model, thereby enabling the student model to learn more meaningful feature representations. Compared to SRRNet [5], PMDL improved the MAE by 29.04% and the RMSE by 23.08% while reducing the number of parameters by 98.2%. These results indicate that PMDL performs well in the vehicle counting task using remotely sensed datasets. The small size of the vehicles in these datasets highlights the applicability and effectiveness of PMDL for counting small objects. The accuracy of PMDL outperforms that of other methods, demonstrating its ability to effectively reduce the complexity of the model while maintaining a high level of accuracy.

Table IV displays the experimental results for the CARPK and PUCPR+ datasets. The proposed PMDL achieved an MAE of 9.91 and an RMSE of 12.74 on the CARPK dataset, both of which ranked second. PMDL shows a 14.2% and 13.8% increase in MAE and RMSE, respectively, compared to the top-performing SRRNet [5]. However, PMDL decreases the parameters by 98.2%, which greatly reduces model complexity with only a minimal decrease in accuracy. PMDL achieved the highest performance on the PUCPR+ dataset, with an MAE of 1.67 and an RMSE of 2.20. Compared to the third-ranked CSRNet [45], PMDL shows an 80.69% improvement in MAE and a 92.55% improvement in RMSE while reducing the parameter count by 92.7%. These results across both datasets indicate that PMDL offers a significant advantage for dense vehicle counting tasks from an overhead perspective, achieving a commendable balance between accuracy and efficiency. The subjective results on these two datasets are illustrated in

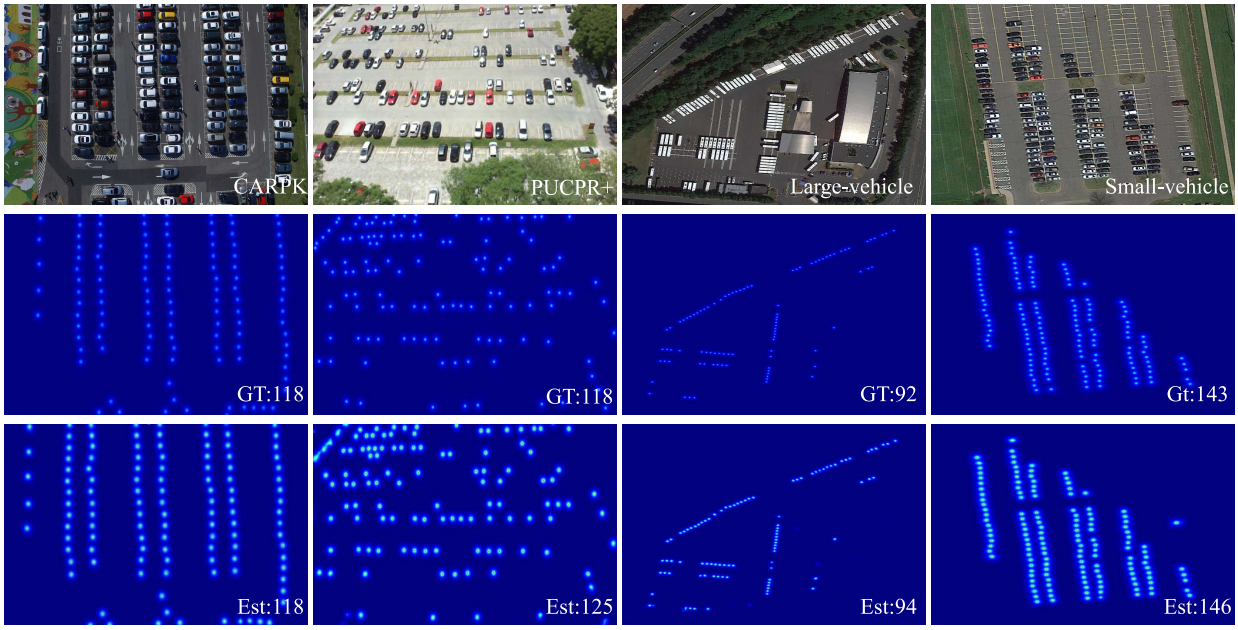

Fig. 5. Visual result on the benchmark vehicle datasets. The first row represents the input image, the second row is the GT, and the third row is the generated density map.

TABLE IV
COMPARISON RESULTS OF VARIOUS METHODS ON CARPK AND PUCPR+ DATASETS. THE BEST PERFORMANCE RESULTS ARE INDICATED IN BOLD, WHILE THE SECOND-BEST IS UNDERLINED

Method	CARPK		PUCPR+	
	MAE	RMSE	MAE	RMSE
Faster-RCNN [53]	103.48	110.64	156.76	200.59
YOLO [54]	102.89	110.02	156.72	200.54
SSD [55]	37.33	42.32	119.24	132.22
RetinaNet [56]	16.62	22.30	24.58	33.12
LPN [42]	23.80	36.79	22.76	34.46
One-look Regression [57]	59.46	66.84	21.88	36.73
MCNN [43]	39.10	43.30	21.86	29.53
LEP [58]	51.83	-	15.70	-
CSRNet [45]	11.48	13.32	8.65	29.53
SRRNet [5]	**8.50**	**10.98**	2.04	**2.79**
PMDL (Ours)	9.91	12.74	**1.67**	2.20

TABLE V
COMPARISON OF DIFFERENT METHODS ON SHANGHAITECH PART A DATASET

Method	MAE	RMSE
MCNN [43]	110.20	173.20
PCCNet [59]	73.50	124.00
SANet [51]	75.30	122.20
TDF-CNN [60]	97.50	145.10
LCNet [61]	93.30	149.00
CCNN [62]	88.10	141.70
1/4SAN+SKT [63]	78.00	126.60
MobileCount [64]	89.40	146.00
SRRNet [5]	60.80	103.00
RAQNet [4]	**59.00**	**101.20**
PMDL (Ours)	60.57	101.43

Fig. 5. The results demonstrate high accuracy across different datasets, particularly in the CARPK and small vehicle datasets, where the estimated number of vehicles is almost identical to the GT. From the distribution of the density maps, the generated vehicle locations closely match the v. Especially in dense vehicle scenarios, such as CARPK, the proposed PMDL accurately captures the distribution pattern of the vehicles.

Comparison on Crowd Counting: To assess the generalization ability of the proposed PMDL, we performed experiments on ShanghaiTech Part A datasets. The images in the Part A dataset come from the Internet and show a dense crowd distribution, which creates certain challenges for counting tasks. The Comparison results of PMDL are detailed in Table V.

The proposed PMDL demonstrated the second-best performance on the ShanghaiTech Part A dataset. Although compared to the best approach, RAQNet [4], the proposed method showed an increase of 2.59% in MAE and 0.23% in RMSE, it significantly improved model simplicity and efficiency by reducing parameters by 95.8%. Compared to the third-ranked SRRNet [5], the proposed PMDL improved MAE and RMSE by 0.38% and 1.52%, respectively, while reducing the number of parameters by 98.2%. The results from Part A datasets validate the effectiveness of the proposed PMDL model.

F. Ablation Study

Effectiveness of Master-Proxy Learning Framework: To validate the effectiveness of Master-Proxy KD, we conducted ablation experiments on three vehicle datasets. The results are presented in Table VI. Proxy refers to using only the student model, while Master-Proxy means using the teacher model to supervise the student model. The results demonstrate significant performance improvements across all three datasets, particularly on the PUCPR+ dataset, where the Master-Proxy KD method increased MAE and RMSE by 66.60% and 66.92%, respectively. These findings indicate

TABLE VI
ABLATION STUDY RESULTS OF MASTER-PROXY KD

Methods	Large vehicle		Small vehicle		PUCPR+	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Proxy	16.09	28.40	120.44	420.51	5.00	6.65
Master-Proxy	**12.95**	**24.03**	**104.09**	**416.78**	**1.67**	**2.20**

TABLE VII
ABLATION STUDY RESULTS OF FEDERATED TRAINING STRATEGY

Methods	$n = 2$		$n = 4$		$n = 8$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Split Data	123.59	**426.72**	131.24	426.61	150.38	433.86
PMDL (Ours)	**117.05**	427.90	**115.79**	**420.77**	**140.34**	**433.69**

that the proxy model, under the guidance of the master model, significantly enhances counting accuracy. Moreover, in practical applications, such design does not increase model parameters and computational complexity. This enhancement significantly increases the model's practical value and provides an efficient, accurate solution for vehicle counting in complex environments.

Effectiveness of Federated Training Strategy: To evaluate the effectiveness of the FL framework and its impact on model performance with varying client numbers, we conducted experiments on the Small-vehicle dataset. We randomly split the training set into n subsets and tested the local-trained model on the full test set. The final result is the average of n experiments. The PMDL method aggregates the models from n clients by v and evaluates the results. The experimental results are shown in Table VII. It is evident that our method outperforms the The split data baseline method is more stable. When training samples are sufficient, with n being 2 or 4, the performance remains balanced. In contrast, when data are insufficient, e.g., n being 8, the efficiency of the split training method decreases, which lead to approximately 10 MAE errors. This demonstrates the stability of PMDL, particularly in scenarios with limited local data.

V. CONCLUSION

In this article, we tackled the two main challenges in DNN-based traffic density estimation, namely computational complexity and data privacy, by the introduced PMDL model. The proposed PMDL model leverages a proxy model learning strategy to transfer knowledge from a larger master model to a lightweight proxy model. Additionally, we employed an FL strategy that allows multiple proxy models trained on privacy-aware local data to be aggregated via a global parameter server. This reduces the computational burden and maintains data privacy while improving model performance compared with per-local domain separate training. To validate the proposed PMDL model, we conducted extensive evaluations on four major vehicle density analysis benchmarks, as well as standard crowd-counting benchmark datasets. The experimental results demonstrated that the PMDL model outperforms state-of-the-art competitors by a large margin, which confirms its effectiveness and efficiency for real-world applications. Although, the federated training framework helps protect local data privacy by aggregating updates from connected local models, these updates remain susceptible to various malicious attacks targeting the model or gradients. An open gap for future research is the identification and mitigation of these attacks to maintain the integrity and security of PMDL framework.

REFERENCES

[1] H. Beenish, T. Javid, M. Fahad, A. A. Siddiqui, G. Ahmed, and H. J. Syed, "A novel Markov model-based traffic density estimation technique for intelligent transportation system," *Sensors*, vol. 23, no. 2, p. 768, 2023.

[2] Q. Cui et al., "Big data analytics and network calculus enabling intelligent management of autonomous vehicles in a smart city," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2021–2034, Apr. 2019.

[3] P. Kumar, S. Arkatkar, G. Joshi, and S. Easa, "Traffic density estimation methods for uninterrupted roadway facilities: Review and guidelines," *J. Transp. Eng., Part A, Syst.*, vol. 149, no. 2, 2023, Art. no. 3122004.

[4] W. Zhai, X. Xing, and G. Jeon, "Region-aware quantum network for crowd counting," *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 5536–5544, Aug. 2024.

[5] X. Guo, M. Gao, W. Zhai, Q. Li, and G. Jeon, "Scale region recognition network for object counting in intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15920–15929, Dec. 2023.

[6] Q. Song et al., "To choose or to fuse? Scale selection for crowd counting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 2576–2583.

[7] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proc. IEEE*, vol. 111, no. 1, pp. 42–91, Jan. 2023.

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

[9] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, 2021.

[10] C.-H. Wang, K.-Y. Huang, Y. Yao, J.-C. Chen, H.-H. Shuai, and W.-H. Cheng, "Lightweight deep learning: An overview," *IEEE Consum. Electron. Mag.*, vol. 13, no. 4, pp. 51–64, Jul. 2024.

[11] Y. Pang, Z. Ni, and X. Zhong, "Federated learning for crowd counting in smart surveillance systems," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 5200–5209, Feb. 2024.

[12] S. Zhang et al., "Federated learning in intelligent transportation systems: Recent applications and open problems," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3259–3285, May 2024.

[13] V. P. Chellapandi, L. Yuan, C. G. Brinton, S. H. Żak, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 119–137, Jan. 2024.

[14] J. Chen et al., "Privacy-aware crowd counting by decentralized learning with parallel transformers," *Internet Things*, vol. 26, Jul. 2024, Art. no. 101167.

[15] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[16] R. K. Chaurasiya, P. M. Gondane, B. Acharya, and M. I. Khan, "Automatic road traffic analyzer using background subtraction, blob analysis, and tracking algorithms," in *Proc. 7th Int. Conf. Comput. Appl. Electr. Eng.-Recent Adv. (CERA)*, 2023, pp. 1–6.

[17] S. Rajkumar, A. Hariharan, S. Girish, and M. Arulmurugan, "An efficient vehicle detection and shadow removal using Gaussian mixture models with blob analysis for machine vision application," *SN Comput. Sci.*, vol. 4, no. 5, p. 451, 2023.

[18] K. K. M. Shariff, M. Q. Z. M. Ali, A. H. Jahidin, M. S. A. M. Ali, and A. I. M. Yassin, "Video camera technology for vehicle counting in traffic census: Issues, strategies and opportunities," *Plan. Malay.*, vol. 21, pp. 61–72, Aug. 2023.

[19] W. Ma and S. Qian, "High-resolution traffic sensing with probe autonomous vehicles: A data-driven approach," *Sensors*, vol. 21, no. 2, p. 464, 2021.

[20] X. Guo, M. Gao, G. Zou, A. Bruno, A. Chehri, and G. Jeon, "Object counting via group and graph attention network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11884–11895, Sep. 2024.

[21] J. Chen, Q. Li, M. Gao, W. Zhai, G. Jeon, and D. Camacho, "Towards zero-shot object counting via deep spatial prior cross-modality fusion," *Inf. Fusion*, vol. 111, Nov. 2024, Art. no. 102537.

[22] J. Guo, M. Bilal, Y. Qiu, C. Qian, X. Xu, and K.-K. R. Choo, "Survey on digital twins for Internet of Vehicles: Fundamentals, challenges, and opportunities," *Digit. Commun. Netw.*, vol. 10, no. 2, pp. 237–247, 2024.

[23] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate edge and cloud computing with distributed deep learning for smart city Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8099–8110, Sep. 2020.

[24] Y. Wu, G. J. Mendis, and J. Wei, "DDLPF: A practical decentralized deep learning paradigm for Internet-of-Things applications," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9740–9752, Jun. 2021.

[25] W. Du et al., "Approximate to be great: Communication efficient and privacy-preserving large-scale distributed deep learning in Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11678–11692, Dec. 2020.

[26] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1–10.

[27] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[28] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, 2020, pp. 1–22.

[29] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2016, pp. 308–318.

[30] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–19.

[31] J. H. Cho and B. Hariharan, "Efficacy of knowledge distillation in neural network compression," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019.

[32] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," 2019, *arXiv:1909.10351*.

[33] Z. Huang, Z. Wang, G. Huang, and C. Xing, "A survey on knowledge distillation for neural network compression," 2021, *arXiv:2106.15254*.

[34] B. Heo, M. Lee, S. Yun, J. Choi, N. Kwak, and S. Choi, "A comprehensive overview of deep learning for image recognition," 2019, *arXiv:1904.01824*.

[35] G. Chen, W. B. Choi, X. Yu, T. Han, W. Xu, and Y. Gwon, "Distilling knowledge via knowledge review," 2020, *arXiv:2011.09161*.

[36] D. Liang, W. Xu, Y. Zhu, and Y. Zhou, "Focal inverse distance transform maps for crowd localization," *IEEE Trans. Multimedia*, vol. 25, pp. 6040–6052, Sep. 2022.

[37] W. Zhai, M. Gao, X. Guo, Q. Li, and G. Jeon, "Scale-context perceptive network for crowd counting and localization in smart city system," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18930–18940, Nov. 2023.

[38] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5056–5069, Sep. 2022.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[40] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.

[41] 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 Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 3642–3655, May 2021.

[42] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 4165–4173.

[43] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 589–597.

[44] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6141–6150.

[45] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1091–1100.

[46] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 5094–5103.

[47] J. Gao, Q. Wang, and Y. Yuan, "SCAR: Spatial-/channel-wise attention regression networks for crowd counting," *Neurocomputing*, vol. 363, pp. 1–8, Oct. 2019.

[48] V. A. Sindagi and V. M. Patel, "CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, 2017, pp. 1–6.

[49] X. Chen, Y. Bin, N. Sang, and C. Gao, "Scale pyramid network for crowd counting," in *Proc. WACV*, 2019, pp. 1941–1950.

[50] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Pixel-wise crowd understanding via synthetic data," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 225–245, 2021.

[51] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–17.

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

[53] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[54] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[55] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[56] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[57] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 785–800.

[58] T. Stahl, S. L. Pintea, and J. C. Van Gemert, "Divide and count: Generic object counting by image divisions," *IEEE Trans. Image Process.*, vol. 28, pp. 1035–1044, 2019.

[59] J. Gao, Q. Wang, and X. Li, "PCC Net: Perspective crowd counting via spatial convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3486–3498, Oct. 2020.

[60] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–8.

[61] X. Ma, S. Du, and Y. Liu, "A lightweight neural network for crowd analysis of images with congested scenes," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2019, pp. 979–983.

[62] X. Shi, X. Li, C. Wu, S. Kong, J. Yang, and L. He, "A real-time deep network for crowd counting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 2328–2332.

[63] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, and L. Lin, "Efficient crowd counting via structured knowledge transfer," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 2645–2654.

[64] 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, Sep. 2020.