Federated Learning for Crowd Counting in Smart Surveillance Systems

Yiran Pang, Zhen Ni, *Senior Member, IEEE*, and Xiangnan Zhong, *Member, IEEE*

***Abstract*—Crowd counting in smart surveillance systems plays a crucial role in Internet of Things (IoT) and smart cities, and can affect various aspects, such as public safety, crowd management, and urban planning. Using surveillance data to centrally train a crowd counting model raises significant privacy concerns. Traditional methods try to alleviate the concern by reducing the focus on individuals, but the concern still needs to be thoroughly resolved. In this work, we develop a horizontal federated learning (HFL) framework to train the crowd counting models which can preserve privacy simultaneously. This framework enables the smart surveillance system to learn from model aggregation without accessing the private data stored on local devices. Therefore, it eliminates the need for video data transmission, reduces communication costs, and avoids raw data leakage. Due to the lack of federated learning (FL) crowd counting data sets, we design four non-independent and identically distributed (non-IID) partitioning strategies, including feature-skew, quantity-skew, scene-skew, and time-skew, to simulate real-world FL scenarios. In addition, we present an efficient fully convolutional network (e-FCN) for each client to demonstrate the practical applicability of the proposed framework. The e-FCN adopts an encoder–decoder architecture with fewer parameters, making it communication-friendly and easier to train. This design can achieve competitive performance compared to more complex models in surveillance crowd counting in literature. Finally, we evaluate the proposed HFL framework with e-FCN under our skew strategies on multiple real-world data sets, including crowd surveillance, ShanghaiTech PartB, WorldExpo'10, FDST, CityUHK-X, UCSD, and MALL. Extensive experiments allow us to present our developed Federated Crowd Counting benchmark as a reference for future research and provide guidance for FL algorithm selection in smart surveillance system deployment.**

***Index Terms*—Convolutional neural networks (CNNs), crowd counting, data partition, federated learning (FL), non-independent and identically distributed (non-IID) partitioning benchmarks, smart surveillance system.**

I. Introduction

SMART surveillance systems play a critical role in smart city development and are widely recognized as important Internet of Things (IoT) applications [1], [2], [3]. Intelligent surveillance systems make use of cameras to monitor the presence and movement of crowd in public spaces. With the continued growth of urbanization, the importance of crowd counting in surveillance systems has grown significantly, especially in highly populated public areas, such as subway stations, stadiums, concerts, and exhibition centers [4]. Large gatherings of people can pose a significant risk of stampedes, which have resulted in tragic loss of life in recent years [5], [6]. These casualties are often caused by mechanical asphyxiation due to crowd crushing, which can occur rapidly depending on the pressure level [7]. One effective way to prevent such incidents is through the use of smart surveillance systems that monitor crowd density trends and take action before they reach critical levels. Given the significance of this issue, there has been a significant amount of research conducted on the development of automated crowd counting techniques using image and video analysis.

(a) (b)

Fig. 1. (a) Representative images of CS data set [11]. (b) Representative images of ground-truth of the density map. By integrating the density map, we can easily get the number of people in any region.

With the development of convolutional neural networks (CNNs), CNN-based architectures have demonstrated powerful automatic feature extraction capabilities [8], [9], [10], [11], [12], [13], [14], [15]. They have been widely adopted due to their superior performance compared to traditional methods. CNN-based methods in crowd counting typically use crowd density estimation, which involves predicting the density map of a crowd scene using CNN models. As shown in Fig. 1, the density map represents the number of people at any location within an image or video frame and the total number of people present. Compare with simply getting the number, the spatial information in a density map helps prevent crowd crush and stampede events because it identifies subregions with high crowd density and counts the people separately [5]. By analyzing the density map, we can proactively identify hotspots with high crowd density, take proactive measures to prevent crowd crush and stampede events, and ensure the safety of events and public places.

However, the effectiveness of CNN-based architectures is highly dependent on large-scale training in data centers [16].

Manuscript received 26 June 2023; accepted 8 August 2023. Date of publication 17 August 2023; date of current version 24 January 2024. This work was supported in part by the National Science Foundation under Grant 2047010, Grant 2047064, Grant 1947418, and Grant 1947419. *(Corresponding author: Xiangnan Zhong.)*

The authors are with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: ypang2022@fau.edu; zhenni@fau.edu; xzhong@fau.edu).

Digital Object Identifier 10.1109/JIOT.2023.3305933

Collecting a large number of surveillance videos as training data can raise serious privacy concerns. Because monitoring collects not only information about the number of people, but also personal faces, clothing, and body posture, which can be used for inappropriate purposes. In addition, the surrounding environment captured by surveillance can also be a risk of abuse.

To address such issues, some studies have tried to develop a privacy-friendly crowd counting method. Chan et al. [17] proposed a privacy-preserving system for crowd counting. They recognized that previous methods, such as detecting and tracking individuals over time to count the number, violated privacy. Instead, they extracted overall features based on segments, edges, and textures from images to estimate the overall number of people. Alleviate privacy concerns by minimizing the focus on personal identities. Similarly, semi-supervised [18], [19] and weakly supervised [20], [21] approaches can significantly reduce the annotation on people. Crowd counting tasks are usually labeled by pointing out the center coordinates of the person's head or using a bounding box to frame the head position. Reducing the number of annotations can reduce the focus on individuals. However, the given approach still uses crowd images for training in a data center. It cannot be considered a privacy-preserved approach. Synthetic data sets, such as CVCS [22] and GCC [23], provide a large amount of training data without privacy concerns. Unfortunately, synthetic data sets have discrepancies from the real world, and real-world data is still needed to fine-tune models to achieve adequate performance [24]. In addition to using visible light images, Tse et al. [25] designed a privacy-aware crowd-counting system using thermal cameras to classify and count people indoors. Moreover, there are some nonimage-based counting methods, such as those based on WiFi [26], [27] and IoT sensors [28]. However, these methods cannot effectively use existing surveillance equipment, and some incur additional costs. They also typically only provide a rough estimate of the crowd size over a large area.

To address the above challenge, we integrate crowd counting with the federated learning (FL) framework. FL is a promising solution that allows multiple parties to collaboratively train models while keeping their local data decentralized [29]. This approach is particularly beneficial in smart surveillance scenarios. By employing the FL method, original surveillance data does not need to be uploaded, effectively mitigating the risk of data leakage during transmission, and reducing transmission costs. However, non-independent and identically distributed (non-IID) data can negatively impact model performance, which is one of the key challenges in FL. Non-IID refers to a data distribution that deviates from the assumption that all samples in a data set are both statistically independent and drawn from the same probability distribution. In crowd counting, it is usually caused by variations in camera settings and crowd conditions.

While several studies have attempted to develop effective FL algorithms [29], [30], [31], [32] under non-IID conditions, systematic experimental research to understand their strengths and weaknesses remains lacking. This is due to the absence of real-world FL data sets. McMahan et al. [29] proposed FL data distribution strategies in image classification and text prediction tasks. LEAF [33] offered a partitioning strategy based on actual image or text data. Li et al. [34] provided several partition settings for classification tasks. OARF [35] and Senthilkumar et al. [36] suggested FL data sets by combining various real-world public data sets, but they did not offer algorithmic-level comparisons. The majority of existing research provides limited emphasis on the specific task of crowd counting in smart surveillance. While the skew settings come from image classification and text prediction tasks offer valuable insights, they do not address the unique complexities inherent to our task. Additionally, while merging multiple public data sets can create complex scenarios, this complexity primarily from the inherent heterogeneity of each data set. Quantifying the heterogeneity poses a significant challenge, complicating the creation of a controlled and reproducible research environment.

The fully convolutional network (FCN) [37], was designed specifically for pixel-wise tasks, such as semantic segmentation. The FCN is characterized by its replacement of the fully connected layer, typically seen in traditional CNNs, with a convolutional layer. This modification ensures that the network can accept images of arbitrary sizes and generate corresponding size outputs. The simplicity and effectiveness of FCN make it a popular choice for counting tasks in a wide range of applications [38]. To achieve higher accuracy on high-density data sets, deeper and more complex FCN architectures are often used [4], [13], [15], [23]. However, the practical application of these complex structures is limited by the constrained computing resources of client devices, and the inherently high communication costs associated with the FL environment. Concurrently, smart surveillance systems often operate in high-traffic scenarios where crowd density is relatively sparse.

Our main contributions are as follows.

1) This article develops a horizontal FL (HFL) framework for preserving privacy of crowd counting in smart surveillance systems. Unlike conventional methods that alleviate privacy concerns by reducing the focus on individuals, this framework protects client privacy by aggregating models without accessing sensitive data on local devices, It can effectively eliminate the need for raw data transmission, alleviate the burden of data transfer, and mitigate the risk of data leakage during the transmission process.
2) This article designs an efficient FCN (e-FCN), a streamlined architecture for crowd counting in smart surveillance systems. It is intended for deployment on each client device within an HFL framework. Comparing with the complex multicolumn or multitask structures which was usually used in the existing methods, our developed e-FCN simplifies the structure by integrating the deconvolution layers to restore spatial information. Given the limited resources of client devices, the simplified design, end-to-end training, and competitive performance of e-FCN make it a compelling choice for deployment in federated crowd counting systems.

3) Four non-IID partition strategies are designed in this article: a) feature skew; b) quantity skew; c) scene skew; and d) time skew. These strategies are applied across seven real-world surveillance data sets, which effectively emulate federated crowd counting scenarios and establish a reliable platform for the evaluation of FL algorithms. We recognize that existing methods for constructing non-IID scenarios fall short of reflecting the realities of crowd counting tasks. Therefore, the design of these new strategies is critical, as they facilitate more realistic non-IID construction and enable the improvement of variable control. This work establishes the Federated Crowd Counting benchmark, which can serve as a reference for future research. It also provides valuable insights for deploying smart surveillance systems.

We conducted extensive experiments on seven surveillance-perspective crowd counting data sets to evaluate the accuracy of four advanced FL algorithms. These data sets cover a range of smart surveillance systems application scenarios and provide insights into the development of privacy-preserving crowd counting in an FL framework under non-IID conditions. The experimental results also offer guidance for the practical deployment of the framework in various environments.

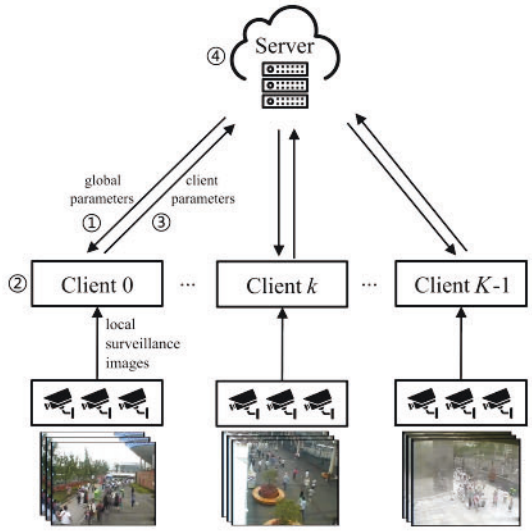

Fig. 2. Surveillance images are captured by Client and saved locally. Only the model parameters are involved in the communication with the server to protect the client's privacy. There are a total of K clients, k numbered starting from 0. The FL process includes four steps: ① distributing the global model to clients, ② training local models with the local surveillance data, ③ uploading the local models back to the server, and ④ updating the global model with the aggregated local models.

II. Proposed Counting Method

In this section, we first introduce our HFL-based surveillance crowd counting framework to address privacy concerns. Then we introduce our proposed surveillance crowd counting network for the client, which has a simpler architecture that is easy to train and communicates friendly. Finally, we introduce the corresponding HFL training method.

A. HFL Crowd Counting Framework

We seamlessly integrate the task of crowd counting in surveillance videos with the HFL framework. This fusion eliminates the need for raw video data transmission, significantly reduces communication costs and avoids data leakage during transmission.

As shown in Fig. 2, the framework includes a cloud server and multiple clients. The server maintains a global model but doesn't hold any data, while clients have their own data sets and models. In this framework, we consider all models to have the same structure. The server initiates communication by sending training requests to clients and clients accepting requests based on their training device availability. Once several requests are accepted, it signifies the start of a round of communication. First, the server sends global model parameters to all participating clients and waits for them to train their models. Then, clients initialize their counting models with global parameters and train the models using local surveillance data. After training, clients upload their model parameters to the server. It is important to note that the uploaded data does not include any raw data, significantly reducing the risk of privacy leaks. After all clients finish training and upload their weights, the server uses FL algorithms to aggregate and generate the global model. The communication cycle will repeat until the global model reaches satisfactory accuracy.

We discuss the impact of non-IID distribution and FL algorithm selection on surveillance crowd counting tasks in subsequent sections. Overall, our HFL-based framework improves the accuracy of the global model while ensuring the privacy of sensitive raw data. This is achieved by only uploading model weights to the central server. Although the uploading weights carry some risk of information leakage [39], our framework is still promising and useful, as it eliminates the need to share sensitive raw data. In actual operations, however, communicate conditions are often difficult to control and the computational power of the client is usually limited. We develop an efficient crowd counting network with fewer parameters and easy training.

B. e-FCN Architecture

To reduce training time, memory consumption and number of parameters, we use the truncated VGG-16 [40] as the front-end of our efficient FCN (e-FCN) for crowd counting, as shown in Fig. 3. We keep only the first 13 layers and remove the fully connected layer and some convolution and pooling layers. This results in our front-end having fewer parameters and a larger feature map (1/8 of input size), which retains as much spatial information as possible. For decoding high-level semantics and recovering spatial information, our back-end starts with a 3×3 convolutional layer that reduces channel number from 512 to 128. This is then followed by three consecutive 4×4 transposed convolutional layers with a stride of 2 and padding of 1. Each transposed layer halves the number of channels and doubles the size of the feature map. Finally, a 1×1 convolution layer transforms the feature map into a density map, reducing the number of channels from 16 to 1. All convolution layers are followed by ReLU activation. Our

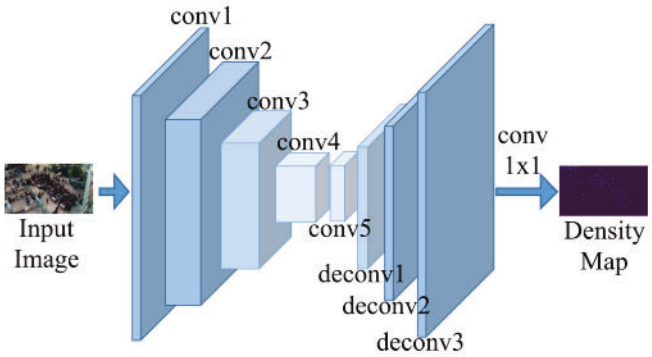

Fig. 3. e-FCN Network Architecture. Each cube represents a feature map of a convolutional layer. Conv1–4 represent the front-end, which correspond to the first 13 layers of VGG-16. Starting from conv5, the back-end structure is presented, where deconv1–3 represent 4 × 4 transposed convolutional layers with a stride of 2 and padding of 1. The final conv1 × 1 represents a 1 × 1 convolutional layer that transforms the feature map into a density map.

network architecture can be represented as follows:

$$F = D\big(E\big(\mathbf{I}; \Theta^E\big); \Theta^D\big) \tag{1}$$

where Θ^E and Θ^D denote the model parameters of the encoder E and decoder D, respectively. For a video frame $\mathbf{I}_i$

$$\mathcal{D}_i^P = F(\mathbf{I}_i; \Theta) \tag{2}$$

where $\mathcal{D}_i^P \in \mathbb{R}^{H\times W\times 1}$ representative the final predicted density map, has the same resolution as the input image $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$. Such structure allowing us to generate high-quality density maps.

C. Training Procedure

We use a straightforward approach on each client to train e-FCN as an end-to-end structure. In the first communication round, the server initialize the first ten convolution layers Θ^E as a pretrained VGG-16 [40] on ImageNet [41]. For Θ^D, the initial values come from a Gaussian initialization with a 0.01 standard deviation.

We deploy the Euclidean distance as the loss function in clients. These metrics are commonly used in crowd counting tasks and are known to be effective in measuring the accuracy of density maps

$$L = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathcal{D}_i^P - \mathcal{D}_i^G\right\|_2^2 \tag{3}$$

where N is the number of images, $\mathcal{D}_i^P$ represents the predicted density map, $\mathcal{D}_i^G$ is the ground truth (GT), i represents the ith sample, and $||\cdot||_2^2$ represents the Euclidean distance.

We use the Adam optimizer on the client side to minimize the loss function during training with learning rate of 5e-5. To further improve the performance of the network, we use data augmentation techniques during training, including random flipping and cropping of the input images. Data augmentation is a powerful tool that helps improve the network's generalization by creating new training samples from the existing ones. This can prevent overfitting and improve the network's performance on unseen data. This is especially helpful in FL in surveillance scenarios due to the limited number of scenes and samples held by each client.

TABLE I
SURVEILLANCE CROWD COUNTING DATA SETS

Dataset	Scene Attribute	Number of Samples	Average Crowd Counts
Crowd Surveillance [11]	Free	13,945	28
ShanghaiTech PartB [8]	Free	716	123
WorldExpo'10 [42]	108 Fixed	3,980	50
FDST [43]	13 Fixed	15,000	27
CityUHK-X [44]	55 Fixed	3,191	33
UCSD [17]	1 Fixed	2,000	25
MALL [45]	1 Fixed	2,000	31

III. PROPOSED PARTITIONING STRATEGIES

Our goal is to create a valuable FL benchmark by simulating real-world federated scenarios, aiming to facilitate research on the impact of non-IID settings on crowd counting. We first introduce the seven data sets used in our work for surveillance crowd counting. Then, we present four data partitioning methods based on the characteristics of the data sets.

We utilize real-world data sets and divide them into small subsets to create appropriate skew settings for FL crowd counting. Our approach allows researchers to easily manage imbalanced settings while independently studying the behavior of each algorithm under different scenarios, which is crucial for developing practical frameworks. We considered real-world scenarios in smart surveillance systems and identified feature, quantity, scene, and time distribution skew as possible non-IID data settings. We also discussed the potential for mixed types of skews.

We used seven real surveillance perspective crowd counting data sets, including two free scenes surveillance data sets crowd surveillance (CS) [11] and ShanghaiTech PartB [8], three fixed multiscenes WorldExpo'10 [42], FDST [43] and CityUHK-X [44] and two fixed single-scenes UCSD [17] and MALL [45]. Free scenes refer to data sets consisting of images captured from various surveillance cameras, as opposed to fixed scenes where images originate from a limited number of cameras. Among them, feature skew applies to all data sets, quantity skew applies to free scenes, scene skew applies to fixed multiscenes, and time skew applies to fixed single scenes. Table I shows the detailed information of each data set. For the IID setting, we shuffle all images in a data set and distribute them evenly to each client. Next, we describe our non-IID skew in detail.

Feature Distribution Skew: In feature distribution skewness, although the scene and time distribution are the same, the feature distribution varies among parties. In a surveillance scenario, camera sensors and network transmissions can all cause image noise. Thus, we design a feature imbalance distribution based on noise. The Gaussian noise is a common type of image noise that affects the details of the image [34], making it blurry and reducing contrast. In crowd counting tasks, it often leads to increased counting error [46]. Therefore, analyzing the performance of Gaussian noise on crowd counting algorithms

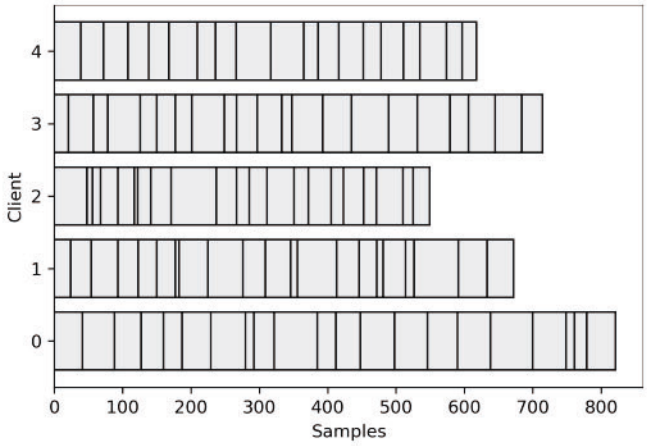

Fig. 4. Scenes skew distribution of WorldExpo'10 data set. Each rectangular segment represents an individual scene, and the training set has a total of 103 scenes. Each scene has a different number of video frames.

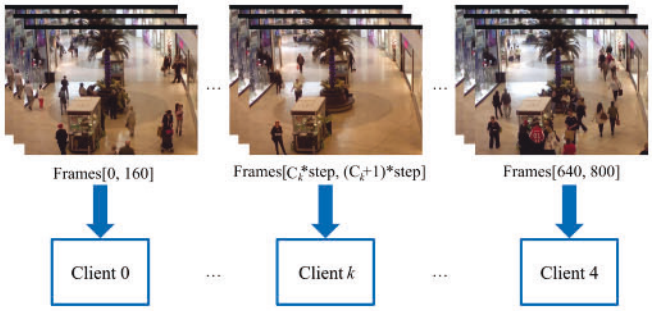

Fig. 5. Time skew distribution of MALL data set, organized by time sequence. Each client contains a consecutive time period, no overlap between clients.

in FL surveillance environments is crucial. We added different noise levels to each client k. $\mathbf{I} \sim \text{Gau}(\sigma \cdot k/K)$ for client C_k, where $\mathbf{I}$ is the video frame and $\text{Gau}(\sigma \cdot k/K)$ is a Gaussian distribution with mean 0 and variance $\sigma \cdot k/K$. We make the parties present characteristic differences by setting different σ for each client. By default, σ takes the value of 0.2. Due to the prevalence of noise in image data, this strategy is used for all data sets in this article.

Quantity Distribution Skew: In quantity skewness, each client has a different volume of local data set. To control variables, we assume that the data distribution remains consistent among parties and investigate the impact of quantity imbalance in FL for crowd counting in surveillance perspectives. We first shuffle all images and then assign different quantities of data samples to each party according to a log-normal distribution. Specifically, we sample from the shuffled data set $\mathbb{D}$ and assign the sampled $\mathbb{D}_i \sim \text{Log}_N(0, \sigma^2)$ as training data to the corresponding client C_i. Although quantity imbalance is a common scenario in many FL environments, in the context of surveillance videos, each client owns its camera. And, the data captured by one camera is usually stored in the same place. Ignoring this natural barrier by shuffling and randomly distributing all surveillance images does not align with reality. Thus, the simple quantity distribution skew is only applied to the CS and ShanghaiTech PartB data sets which have free-scenes.

Scene Distribution Skew: In Scene distribution skew, considering the distribution of surveillance cameras, we divide the training images among clients based on the scene captured by the cameras. As shown in Fig. 4, using WorldExpo'10 as an example, the training set has 103 fixed scenes. In this strategy, we first sort the data by scene labels, divide it into five sets of scene slices with sizes ranging from 19 to 21, and assign one set to each client. This distribution approach is more realistic and closer to the real-world surveillance scenarios. It includes the natural separation of video frames between different scenes, variations in camera equipment frame rates and angles, the complexity of the scene, and shifts in crowd density, making it a challenging data distribution skew. The skew was applied to the multifixed scenes data sets, including WorldExpo'10, FDST, and CityUHK-X.

Time Distribution Skew: In certain data sets, there is only one scene available, thus we divide the data based on time. The data is first sorted by time and then allocated to each client in sequence without any overlap in time. This skew is illustrated in Fig. 5. The data set contains N images. There are a total of K clients, k numbered starting from 0. The number of frames assigned to each client k is $[C_k \times \text{step}, (C_k + 1) \times \text{step}]$. For the same camera, data collected at different times may also suffer from independence due to factors, such as weather, lighting, and crowd flow. This skew is applied to surveillance crowd counting data sets containing only one scene, including UCSD and MALL.

Mixed Types of Skews: In practical applications, mixed biases may arise, leading to increased complexity. We discuss two categories of mixed skew in this study. The first category combines scene and feature skew, which is suitable for the data set consists of multiple fixed scenes. The second category combines quantity and feature skew and is applicable to the data set consists of free scenes. In real-world scenarios, each client typically holds video frames with similar image quality. Therefore, we first allocate images to clients and then apply feature skew separately for each client. By examining these two categories of mixed skew, we aim to better simulate real-world federated crowd counting environments and provide more accurate guidance for deploying federated crowd counting frameworks.

IV. Experiments

Our experimental goal is to learn an effective global model under settings that closely mimic real-world data distributions and provide guidance for algorithm selection in deploying the framework. We selected four popular FL algorithms. To investigate the effectiveness of existing FL algorithms on crowd counting in smart surveillance systems, we conduct extensive experiments on seven public surveillance perspective data sets, including CS [11], ShanghaiTech PartB [8], WorldExpo'10 [42], FDST [43], CityUHK-X [44], UCSD [17], and MALL [45].

FedAvg [29] is a widely used FL technique. The algorithm employs weighted averaging of locally computed weights from each client's model to generate the global model, where the

weights are proportional to the client data volume to ensure larger volumes contribute more significantly.

FedProx [30] improves FedAvg by introducing an *L2* regularization term in the local objective function to limit the distance between the local and global models, making the average model closer to the global optimal. The regularization weight is controlled by the hyperparameter μ, which need to carefully adjust. If μ is too small, regularization has little effect. If μ is too large, updates are small and convergence is slow.

FedNova [31] improves FedAvg by addressing the issue of different parties conducting different numbers of local steps during each round of FL. This can occur due to variations in computation power or local data set size. To ensure unbiased global updates, FedNova normalizes and scales the local updates of each client according to their number of local steps.

SCAFFOLD [32] improves FedAvg by applying variance reduction techniques. It introduces control variates for the server and each client to estimate the update direction of the global model and the update direction of each local model. The difference between these two update directions approximates the drift in the local training. SCAFFOLD corrects the local updates by adding this drift. Compared to the above three algorithms, SCAFFOLD doubles the communication size per round because of the additional control variates.

A. Implementation

The default number of participating clients is 5. Our ablation study shows that the client numbers have an influence on the allocation of images, which is a rise in client numbers can lead to the insufficiency of images for each client. This makes it difficult to accurately evaluate the algorithm performance. All parties participate in every round to eliminate the effect of randomness brought by party sampling [29]. The batch size is set to 16 and the number of local epochs is set to 5. We run all the studied algorithms for the same number of rounds for fair comparison. The number of communication rounds is set to 50 by default, except for the ShanghaiTech PartB and WorldExpo'10 data sets with 200 rounds.

The network directly converts the input image into a density map in training. The loss can be calculated as the mean absolute error (MAE) defined as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|C_i^P - C_i^G| \tag{4}$$

where N is the number of validation images, C_i^G is the GT of counting. C_i^P represents the predicted number by integrating the network output, which is defined as follows:

$$C_i^P = \sum_{h=1}^{H}\sum_{w=1}^{W}\mathcal{D}_i^P(h,w) \tag{5}$$

where (h, w) is a specific coordinate the predicted density map, while H and W show the total height and width. To unify the FL settings, we generated ground-truth density maps [9] using a Gaussian kernel with a fixed sigma of 4 for all data sets.

TABLE II
FEDERATED CROWD COUNTING BENCHMARK. THE MAE FROM DIFFERENT APPROACHES. FOR FEDPROX, WE TUNE μ FROM {0.001, 0.01, 0.1, 1} AND REPORT THE BEST ACCURACY. THE ABBREVIATION CS IN THE TABLE REFERS TO THE CS DATA SET, SHB REFERS TO THE SHANGHAITECH PARTB DATA SET, AND WE'10 REFERS TO THE WORLDEXPO'10 DATA SET

	Dataset	FedAvg	FedProx	FedNova	SCAFFOLD
IID	CS	7.3	8.7	7.1	**6.1**
	SHB	9.6	**9.2**	9.7	9.5
	WE'10	8.0	8.3	**6.4**	8.2
	FDST	1.67	1.66	1.76	**1.58**
	CityUHK-X	8.6	8.7	**8.1**	8.6
	UCSD	1.04	1.02	**0.97**	1.04
	MALL	1.56	**1.51**	1.54	1.52
Feature skew	CS	8.4	8.1	9.4	**7.4**
	SHB	9.7	10.1	10.3	**9.6**
	WE'10	8.1	8.3	**6.2**	7.3
	FDST	1.77	1.83	**1.46**	1.78
	CityUHK-X	8.7	8.5	**8.4**	8.5
	UCSD	1.08	1.07	**1.04**	1.11
	MALL	1.52	**1.50**	1.53	1.53
Quantity skew	CS	6.3	**5.6**	6.6	5.9
	SHB	9.4	9.3	10.9	**9.0**
Scene skew	WE'10	8.4	8.1	**6.1**	8.6
	FDST	3.4	3.7	**3.0**	3.2
	CityUHK-X	8.9	8.8	**8.6**	8.6
Time skew	UCSD	1.47	1.49	**1.13**	1.27
	MALL	1.58	1.56	1.57	**1.55**

B. Overall Accuracy Comparison

Table II shows the accuracy under different non-IID data settings. Although it is difficult to be met in real-world applications, we still provide results of the IID scenario (i.e., homogeneous partition) for comparison.

Comparison Among Different Non-IID Settings: First, the feature skew in the WorldExpo'10 and MALL data sets performed better with a lower MAE, and the impact on other data sets was relatively small. This may be because the random Gaussian noise creates a similar effect as data augmentation, improving the generalization ability of local models. Second, scene skew is considered to be the most challenging scenario. It resulted in twice MAE of the IID setting for the FDST data set. However, the WorldExpo'10 data set was less affected. This may be due to the difference in the number of scenes in the two data sets. The FDST training set contains 13 scenes, which results in each client only containing 2–3 scenes. The WorldExpo'10 training set contains 103 scenes, and each client contains over 20 scenes. The fewer independent scenes that each client holds, the greater the challenge. We also observed that the time skew was present challenges. The UCSD data set, with four time periods, was greatly affected, for each client contains only 1 to 2 independent time periods. While the MALL data set, with only one continuous period, had a lower accuracy but less impact. Third, in terms of quantity skew, FedProx and SCAFFOLD handled this scenario well by regularization and weighted average. Overall, limited scenes or time periods per client have a significant negative impact on crowd counting accuracy. Existing algorithms still have room for improvement in handling scene or time imbalance.

Comparison Under Mixed Types of Skews: The task of fixed scenes demonstrates heightened complexity attributed to mixed biases, whereas such an observation is not present in

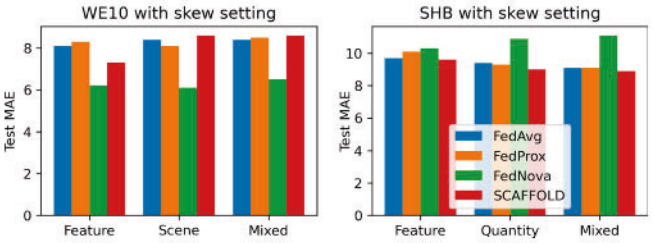

Fig. 6. Comparison of the impact of feature skew, scene skew, quantity skew, and mixed skew on the test MAE of four approaches for (left) WorldExpo'10 data set and (right) ShanghaiTech PartB data set.

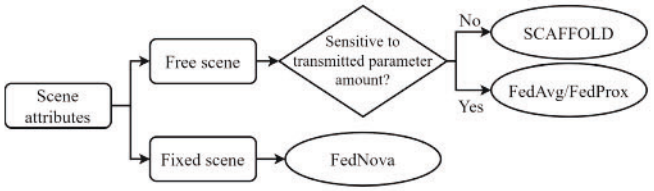

Fig. 7. Decision tree to recommends the FL algorithm given the non-IID setting.

free scenes. As depicted in Fig. 6, we compare the scenarios of solely adding noise, solely introducing scene/quantity skew, and mixed situations. The number of communication rounds is fixed at 50. Our observations reveal that in fixed scenes, the MAE of all methods increases in mixed situations, as both scene skew and feature skew present challenges during training. However, in ShanghaiTech PartB, the presence of mixed-type bias results in a lower MAE, excluding FedNova. This effect can be attributed to the inclusion of Gaussian noise, which relieves the overfitting of clients with fewer samples and leads to improved accuracy.

Comparison Among Different Algorithms: First, in multifixed scene data sets, FedNova has a significant advantage. It can cope well with feature skew and scene skew. FedNova normalizes and scales the local updates of each party according to their number of local steps before updating the global model, which plays a crucial role in the FL Crowd Counting task under multifixed scenes. However, in the free-scene data set, FedNova brings the worst performance. In such scenarios, SCAFFOLD typically achieves superior accuracy compared to other evaluated methods. FedProx exhibits marginally better performance than FedAvg. Nevertheless, FedAvg is always a reliable and straightforward choice for free-scene scenarios, considering the additional communication costs with SCAFFOLD and extra time costs of adjusting the hyper-parameter μ with FedProx.

Algorithm Selection: Based on our above observations, we draw a decision tree to summarize the appropriate FL algorithms for scene attribute on non-IID setting, as shown in Fig. 7. This decision tree helps users to select learning algorithms based on the characteristics of the smart surveillance system and research data sets. For example, if the local data set have scene distribution skew (e.g., multiple fixed scenes from different cameras), then FedNova may be the best algorithm for FL. If the local data set is from free scene, SCAFFOLD can be considered getting the lowest MAE, but SCAFFOLD introduces a doubling of the network transmission parameters.

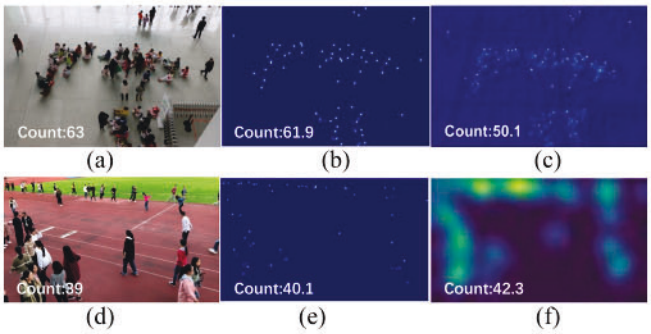

Fig. 8. Visualization of test examples from CS and FDST. (a) and (d) are the input samples, (b) and (e) represent estimated density maps by our e-FCN with FedNova under IID, and (c) and (f) show estimated density maps by centralized training PGCNet [11] and LSTN [43], respectively.

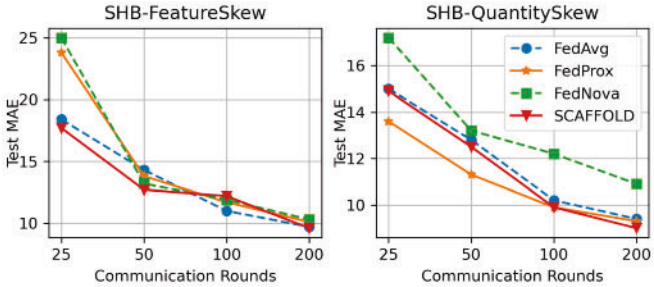

Fig. 9. Comparison of different approaches on ShanghaiTech PartB data set with feature skew (left) and quantity skew (right), where the curves show the change in test MAE as communication rounds increase.

Comparison With Model for Centralized Training: In large-scale data sets, our e-FCN combined with the FL algorithm outperforms state-of-the-art algorithms with centralized training, such as LSTN [43] (MAE 3.35) in FDST and PGCNet [11] (MAE 7.2) in CS. As shown in Fig. 8, our e-FCN structure, via the deconvolution layer, effectively restores spatial information. Compared to centralized training PGCNet [11] and LSTN [43], ours introduces less distortion and noise, while maintaining a higher spatial similarity to the GT. This not only provides us with more precise density map, but also results in a greater counting accuracy. In terms of accuracy, recent work [47] observes that fine-tuning models independently optimized from the same initialization fall into the same error basin in the error landscape. Our conjecture is that the averaging of locally trained models with distinct optimization directions in FL produces a favorable regularization effect that releases the global model from the confinement of error basins, thereby enhancing accuracy. This is particularly beneficial in large-scale data sets, as each client has sufficient local convergence to make a positive contribution to the global model.

C. Communication Efficiency

Fig. 9 shows the training curves of the studied FL algorithms on the ShanghaiTech PartB data set, including two settings: 1) feature skew and 2) quantity skew. In view of the potential instability of the convergence process of crowd counting, we opted to confine our analysis to a subset of

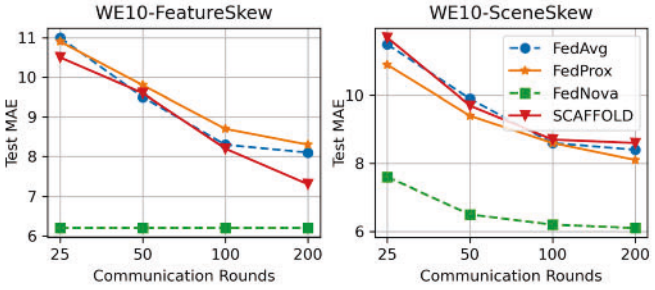

Fig. 10. Comparison of different approaches on WorldExpo'10 data set with feature skew (left) and quantity skew (right), where the curves show the change in test MAE as communication rounds increase.

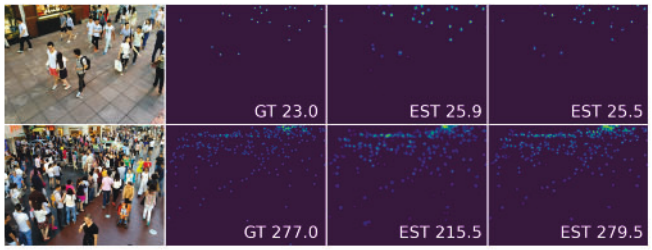

Fig. 11. Results of e-FCN for SHB data set optimized with FedProx under QuantitySkew setting. The first column is the original image; the second column is the GT density map; the third to sixth columns correspond to the estimated (EST) density maps of the 25th and 200th communication rounds, with MAE 13.6 and 9.3, respectively.

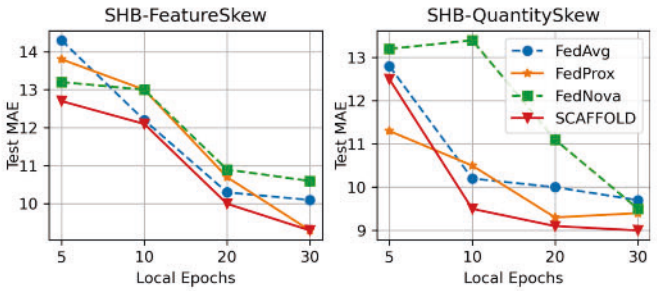

Fig. 12. Comparison of the impact of increasing the local epochs in each client on test MAE for four approaches under feature skew (left) and quantity skew (right) conditions on the ShanghaiTech PartB data set.

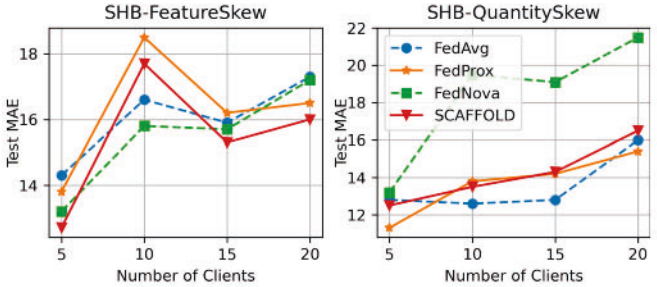

Fig. 13. Comparison of the impact of increasing the number of clients on test MAE for four approaches under feature skew (left) and quantity skew (right) conditions on the ShanghaiTech PartB data set.

rounds (i.e., the 25th, 50th, 100th, and 200th rounds) to mitigate the impact of such instability on our evaluation. In the feature skew setting, FedAvg and SCAFFOLD showed higher accuracy in the initial training stage. But as training progressed to the 50th round, the convergence speed and final accuracy of the four algorithms were very close. In the quantity skew setting, FedProx with regularization terms showed the best overall performance. At the 200th round, SCAFFOLD achieved the best accuracy, but only with a slight advantage. When compared in these two cases, our results showed that the presence of the Gaussian noise had a more significant impact on the final MAE than uneven quantity distribution in free scenes.

FedNova was designed to eliminate skew by equalizing the contributions of each participating node to the global model via the statistical number of mini-batches per client, but it did not perform well in handling free-scene quantity skew in the crowd counting task. However, the situation became very different in the WorldExpo'10 data set with multiple fixed scenes. As could be seen from Fig. 10, FedNova had an overwhelming advantage in both convergence speed and final accuracy. Experiments showed that, in an FL setting, FedNova's normalization and scaling of weights per client were crucial for surveillance crowd counting in multiple fixed scenes.

Fig. 11 shows the results of our e-FCN for the SHB data set optimized with FedProx under QuantitySkew setting. At the beginning of the training (at communication round 25), although there was a noticeable difference between the estimated numbers and GT in high-density scenarios, the density map was already able to reflect the overall distribution of the crowd. As the training progressed, we obtained a finer density map with more accurate numbers of people. It was necessary to select a suitable FL algorithm and set a reasonable expected communication round based on the task requirements, with a tradeoff between the number of communication rounds and the final accuracy.

D. Ablation Experiments

Increasing Local Updates: The number of local epochs can have a large effect on the accuracy of existing algorithms. We vary the number of local epochs from {5, 10, 20, 30} and report the final accuracy on ShanghaiTech PartB in Fig. 12. The number of communication rounds was fixed at 50. We observed that as the number of local epochs increased, the error of almost all algorithms decreased. However, excessive local epochs could greatly extend the local training time on the client and reduce FL system availability, particularly in the case of crowd counting tasks that had a large input dimension. This suggested that reasonable local epochs had to be set based on the requirements of the task and the availability of FL resources.

Increasing Participating Clients: As the number of clients increases, the MAE of all methods rises. We study the impact of the number of clients on the performance of our methods, as shown in Fig. 13. The number of participating clients ranges from {5, 10, 15, 20}. The number of communication rounds is fixed at 50. Our findings show that MAE significantly increases as the number of clients increases. This is particularly evident in the case of FedNova. Its strategy of scaling according to the number of training steps exhibits instability on small-scale local data sets. With more clients, the local data becomes smaller and is more prone to overfitting during the local training phase.

V. Conclusion

This article develops an HFL framework for crowd counting in smart surveillance systems. The proposed framework successfully tackles the challenge of preserving privacy in crowd counting while maintaining high accuracy. Our Federated Crowd Counting benchmarks guides FL algorithm selection while deploying smart surveillance systems. Our benchmark shows that the choice of FL algorithm depends on the specific scenarios of the task. For fixed scenes, FedNova demonstrates a dominant advantage. In contrast, for free scenes, the choice between FedAvg/FedProx and SCAFFOLD depends on whether the scenario is sensitive to communication parameters. Among our four skews, the scene skew is considered the most challenging when the number of scenes per client is limited. Time skew also presents challenges, while the quantity and feature skew can improve accuracy in some cases. Furthermore, our e-FCN demonstrates competitive performance compared to centralized training models on surveillance crowd counting data sets. The results highlight the potential of FL in preserving privacy and provide improved accuracy in crowd counting for smart surveillance systems.

References

[1] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with China perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, Aug. 2014.

[2] M. Handte, S. Foell, S. Wagner, G. Kortuem, and P. J. Marrón, "An Internet-of-Things enabled connected navigation system for urban bus riders," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 735–744, Oct. 2016.

[3] L. Hu and Q. Ni, "IoT-driven automated object detection algorithm for urban surveillance systems in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 747–754, Apr. 2018.

[4] M. A. Khan, H. Menouar, and R. Hamila, "Revisiting crowd counting: State-of-the-art, trends, and future perspectives," *Image Vis. Comput.*, vol. 129, Jan. 2023, Art. no. 104597.

[5] R. Zhao, D. Dong, Y. Wang, C. Li, Y. Ma, and V. F. Enríquez, "Image-based crowd stability analysis using improved multi-column convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5480–5489, Jun. 2022.

[6] W. Zhou, X. Min, Y. Zhao, Y. Pang, and J. Yi, "A multi-scale spatio-temporal network for violence behavior detection," *IEEE Trans. Biom, Behav., Ident.Sci.*, vol. 5, no. 2, pp. 266–276, Apr. 2023.

[7] R. S. Lee and R. L. Hughes, "Prediction of human crowd pressures," *Accid. Anal. Prevent.*, vol. 38, no. 4, pp. 712–722, 2006.

[8] 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.*, 2016, pp. 589–597.

[9] 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.*, 2018, pp. 1091–1100.

[10] 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. 734–750.

[11] Z. Yan et al., "Perspective-guided convolution networks for crowd counting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 952–961.

[12] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd counting using scale-aware attention networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2019, pp. 1280–1288.

[13] 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.

[14] J. T. Zhou et al., "Locality-aware crowd counting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3602–3613, Jul. 2022.

[15] Z.-Q. Cheng, Q. Dai, H. Li, J. Song, X. Wu, and A. G. Hauptmann, "Rethinking spatial invariance of convolutional networks for object counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19638–19648.

[16] Q. Wang, J. Gao, W. Lin, and X. Li, "NWPU-crowd: A large-scale benchmark for crowd counting and localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2141–2149, Jun. 2021.

[17] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–7.

[18] C. C. Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2256–2263.

[19] G. Olmschenk, J. Chen, H. Tang, and Z. Zhu, "Dense crowd counting convolutional neural networks with minimal data using semi-supervised dual-goal generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Learn. Imperfect Data Workshop*, 2019, pp. 1–8.

[20] M. V. Borstel, M. Kandemir, P. Schmidt, M. K. Rao, K. Rajamani, and F. A. Hamprecht, "Gaussian process density counting from weak supervision," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 365–380.

[21] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in CNNs by self-supervised learning to rank," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1862–1878, Aug. 2019.

[22] Q. Zhang, W. Lin, and A. B. Chan, "Cross-view cross-scene multi-view crowd counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 557–567.

[23] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8198–8207.

[24] Q. Wang, T. Han, J. Gao, and Y. Yuan, "Neuron linear transformation: Modeling the domain shift for crowd counting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3238–3250, Aug. 2022.

[25] R. Tse, T. Wang, M. Im, and G. Pau, "Privacy aware crowd-counting using thermal cameras," in *Proc. 12th Int. Conf. Digit. Image Process. (ICDIP)*, 2020, pp. 323–333.

[26] F. Wang, F. Zhang, C. Wu, B. Wang, and K. J. R. Liu, "Respiration tracking for people counting and recognition," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5233–5245, Jun. 2020.

[27] L. Zhang, Y. Zhang, B. Wang, X. Zheng, and L. Yang, "WiCrowd: Counting the directional crowd with a single wireless link," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8644–8656, May 2021.

[28] E. Cianca, M. De Sanctis, and S. D. Domenico, "Radios as sensors," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 363–373, Apr. 2017.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[30] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[31] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7611–7623.

[32] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," 2019, *arXiv:1910.06378*.

[33] S. Caldas et al., "LEAF: A benchmark for federated settings," 2018, *arXiv:1812.01097*.

[34] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, 2022, pp. 965–978.

[35] S. Hu, Y. Li, X. Liu, Q. Li, Z. Wu, and B. He, "The OARF benchmark suite: Characterization and implications for federated learning systems," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–32, 2022.

[36] R. Senthilkumar, S. Ritika, M. Manikandan, and B. Shyam, "Crowd counting using federated learning and domain adaptation," in *Proc. Int. Conf. Inf., Commun. Comput. Technol.*, 2022, pp. 97–111.

[37] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

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

[39] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[42] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 833–841.
[43] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-constrained spatial transformer network for video crowd counting," 2019, *arXiv:1907.07911*.
[44] D. Kang, D. Dhar, and A. Chan, "Incorporating side information by adaptive convolution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[45] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *Proc. BMVC*, vol. 1, 2012, p. 3.
[46] O. Elharrouss et al., "Drone-SCNet: Scaled cascade network for crowd counting on drone images," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 3988–4001, Dec. 2021.
[47] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 512–523.

Yiran Pang received the B.S. degree in computer science and technology and the M.S. degree in safety engineering from Chongqing University of Science and Technology, Chongqing, China, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree in computer science with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA.

His research interests include crowd counting and federated learning.

Zhen Ni (Senior Member, IEEE) received the Ph.D. degree from the University of Rhode Island, Kingston, RI, USA, in 2015.

He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. His research interests mainly include artificial intelligence, and computational methods, and machine learning.

Dr. Ni received the prestigious National Science Foundation (NSF) Faculty Early Career Development Program (CAREER) Award in 2021. He has been an Associate Editor of IEEE INTERNET OF THINGS JOURNAL since 2021, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2019, and *IEEE Computational Intelligence Magazine* since 2018.

Xiangnan Zhong (Member, IEEE) received the Ph.D. degree from the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA, in 2017.

She is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, USA. Her research interests include computational intelligence, reinforcement learning, cyber–physical systems, networked control systems, neural networks, and optimal control.

Prof. Zhong received the National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award in 2021 and the NSF CRII Award in 2019. She was a recipient of the International Neural Network Society (INNS) Aharon Katzir Young Investigator Award in 2021 and the INNS Doctoral Dissertation Award in 2019. She has been serving as an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2021.