ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA251489

Federated Learning for Cross-Dataset Generalization in Litter Detection

Luciano Baresi[a], Simone Bianco[b], Livia Lestingi[a] and Iyad Wehbe[a]

[a]Politecnico di Milano, 20133 Milano, Italy
[b]University of Milano-Bicocca, 20126 Milan, Italy

Abstract. The accumulation of litter in natural environments poses significant ecological and social challenges, motivating the development of automated solutions for litter detection. However, collecting and centrally aggregating large-scale annotated datasets for training object detectors often raises privacy and ownership concerns. In this work, we propose a Federated Learning (FL) framework to train a lightweight litter detection model based on the YOLO architecture, which enables collaborative model development without requiring centralized access to raw data. Each participating client locally trains the model on site-specific datasets collected in the wild, and only model updates are shared with a central server for aggregation. We compare and contrast different FL process configurations involving mixed and heterogeneous training datasets built starting from two commonly used benchmark datasets collected across different locations and having very different visual data distributions, i.e. TACO and PlastOPol. Experimental results show that the federated model, trained across these non-IID data distributions, achieves superior generalization in cross-dataset evaluation compared to the corresponding centrally trained models.

1 Introduction

Littering constitutes a serious environmental and social issue in many urban and rural areas around the world. Inappropriate waste disposal not only pollutes ecosystems, but also threatens wildlife and diminishes the aesthetic and ecological value of natural spaces.

Automated systems for detecting litter offer an effective solution by enabling rapid and precise identification of waste in a variety of settings. These systems can operate continuously, cover broad geographic regions, and minimize the need for manual labor, making them a vital asset in contemporary waste management frameworks.

An increasingly effective way to support large-scale litter detection is through citizen science, which encourages public participation in scientific data collection. With smartphones now widely available, people can contribute by taking images of littered environments and reporting incidents. This crowd-sourced approach helps generate extensive datasets over time and across locations, greatly enhancing environmental monitoring efforts.

COmmunity-Based Organized Littering (COBOL) [2] aims to develop a flexible, scalable, and privacy-preserving framework to manage the entire waste disposal process, with a strong emphasis on citizen participation, Federated Learning (FL), and self-adaptive technologies. COBOL integrates computer vision, model-driven engineering, and decentralized data processing to support real-time litter detection, classification, and disposal in diverse urban and rural communities. Citizens can contribute both explicitly (e.g., reporting via mobile apps) and implicitly (e.g., automated litter detection through background media analysis on personal devices). The system leverages federated learning to collaboratively train detection models without centralizing sensitive data, enhancing privacy and scalability. Among the key innovations, there are lightweight object detectors that can be deployed in edge devices [3]. However, collecting and aggregating at a central level the annotated datasets, to be used to train them, raises privacy and ownership concerns.

This paper presents a FL framework for training a lightweight litter detection model built on the YOLO architecture. The proposed approach facilitates collaborative model training across multiple sites without the need to centralize raw image data. Instead, each client independently trains the model on its local data set, captured in diverse real-world environments, and transmits only the learned model updates to a central server for aggregation.

Although the proposed FL framework is applicable in principle to other domains, both the experimental setup and the design choices are specifically tailored to the garbage detection problem. The datasets used in our evaluation consist of annotated litter images collected from urban contexts, reflecting the diversity and noise of real-world waste detection. Moreover, the choice of lightweight object detection models is motivated by the requirement to have the detector perform inference on citizens' mobile devices. Finally, the definition of community-level clusters in the federation mirrors how data is naturally generated and shared in this application domain (e.g., by municipalities or local communities), which distinguishes our approach from more generic FL settings.

The paper is structured as follows: Section 2 surveys related work; Section 3 presents the application in detail; Section 4 describes the experimental setup devised to develop the application; Section 5 presents the experimental results; Section 6 concludes the paper.

2 Related Work

This section briefly surveys state-of-the-art approaches to litter detection and FL applied to image processing tasks.

2.1 Litter detection

The current literature on automated litter detection presents a wide variety of approaches, differing in the model architectures employed, the datasets used, and the formulation of the task, ranging from object detection to semantic segmentation. All these works train a single centralized model that has access to all the data.

Proença and Simões introduce the TACO (Trash Annotations in Context) dataset[1] and use Mask R-CNN as a baseline detector. Wang et al. [28] propose the MJUWASTE dataset and evaluate several large-scale segmentation models such as FCN-8s, PSPNet, CCNet, and DeepLabv3, each with different network backbones. Patrizi et al. [21] present a data augmentation strategy in which isolated waste items photographed against white backgrounds were combined into more complex scenes to enrich the diversity of the dataset. Córdova et al. [4] conduct a benchmark study comparing various modern convolutional neural network detectors, including Faster R-CNN, Mask R-CNN, EfficientDet, RetinaNet, and different YOLO-v5 variants, on two datasets. Majchrowska et al. [16] aggregate images from multiple open-access sources and introduced a two-stage approach that combines EfficientDet-D2 for localization and EfficientNet-B2 for classification. Jalal et al. [9] evaluate the performance of multiple versions of YOLO-v5 (from YOLO-v5s to YOLO-v5x) on a custom-built dataset. Similarly, Das et al. [5] explore YOLO-v5 variants and integrated test-time augmentation (TTA) to boost inference performance, trading off for increased computational overhead. Lastly, Mandhati et al. [17] conduct evaluations of YOLO-v5l and Faster R-CNN on a mixture of datasets, including TACO, PlastOPol [4], UAV-DB [8], and UAVVaste [11]. Bianco et al. [3] compare the most efficient YOLO-v5 and v8 variants showing that with their training procedure the efficient variants were able to surpass the performance achieved by larger models in the state of the art on TACO.

2.2 *Federated Learning for Image Processing*

In recent years, several studies have examined FL from different angles.

Li et al. [13] provide a foundational survey on FL, addressing how constraints such as limited on-device resources, non-IID distributions, and user privacy shape the need for novel optimization schemes. Their work highlights how FL deployment in real-world applications involves balancing privacy, convergence accuracy, and computational feasibility.

Rahman et al. [22] provide a broad, high-level survey of the challenges and design aspects of federated learning, such as communication overhead, data partitioning, and system heterogeneity.

As for image processing specifically, FL has been applied in several domains [10], mainly in medical image analysis [7]. In this field, protecting patients from data leakage is a major concern; at the same time, training is often performed with scarce (or scarcely labeled) data. As a result, FL configurations often gravitate towards partial weight sharing and data synthesis techniques [7].

Previous studies also specifically address the object detection task. FedVision [15] proposes a platform to train object detection models in a federated setting, also providing a user interface to specify the model to train and configuration parameters (e.g., the number of rounds). Zhang et al. [29] present FedVisionBC, a blockchain-based FL system with additional protection measures against external privacy attacks, tested on the YOLO-v5[2] model. The latest developments in this area are presented by Alahdal et al. [1] that exploit the FedAVG algorithm to train YOLO-v8[3].

The work presented in this paper builds on previous results by training YOLO-v8 in a federated setting with realistic heterogeneous dataset splits. Furthermore, unlike previous studies dealing with object detection model training with a federated approach, we exploit a refinement of the FedAVG aggregation method that takes into account the size of each local dataset and the loss resulting from a training round. These measures result in a FL solution that better suits heterogeneous data splits between clients as expected in our target application. In addition, we conduct a systematic evaluation of heterogeneity levels by exploring multiple federated configurations involving 2, 4, 8, and 16 clients, and considering both random and visual-similarity-based data partitioning schemes. This experimental design enables an empirical analysis of the impact of increasing client fragmentation and inter-client variability on both object presence detection (recall) and localization performance (mAP@50).

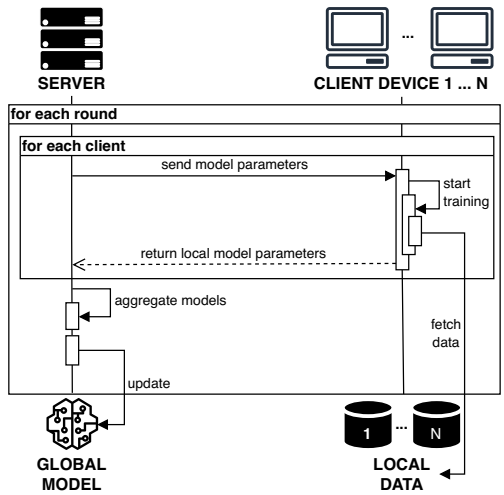


Figure 1: Schematic overview of the FL process.

3 COBOL

COBOL envisages a data-driven approach to detect and dispose of waste in urban and suburban areas [2]. The approach envisages citizens of a community reporting the presence and location of litter in two ways. Firstly, they can actively submit a new report by taking a geolocalized picture through a mobile app. Secondly, mobile devices run an automated detector that analyzes media to detect the presence of litter. In both cases, the picture is analyzed to detect the presence of litter and, if so, to classify its size and material. Based on the identified attributes, a notification is sent to the local authority to trigger the proper waste removal process.

Application-specific Challenges. In COBOL, the images under analysis are stored on mobile devices of citizens and must include geolocalization for the waste management process to be effective. Besides the geographical location, identifying the correct removal authority also calls for a classification of the object size (e.g., bulky waste removal might be entrusted to dedicated bodies) and material (e.g., asbestos requires ad-hoc abatement processes to prevent health-related risks).

This kind of image processing task requires state-of-the-art object detection models whose training process requires significant computational resources and a rich training dataset [14]. Although apparently less sensitive compared to other domains such as healthcare, raw images may still contain identifiable surroundings and location cues, making straightforward anonymization techniques insufficient. Therefore, having all citizens share their data with a centralized

[1] http://tacodataset.org
[2] https://github.com/ultralytics/yolov5
[3] https://github.com/ultralytics/ultralytics

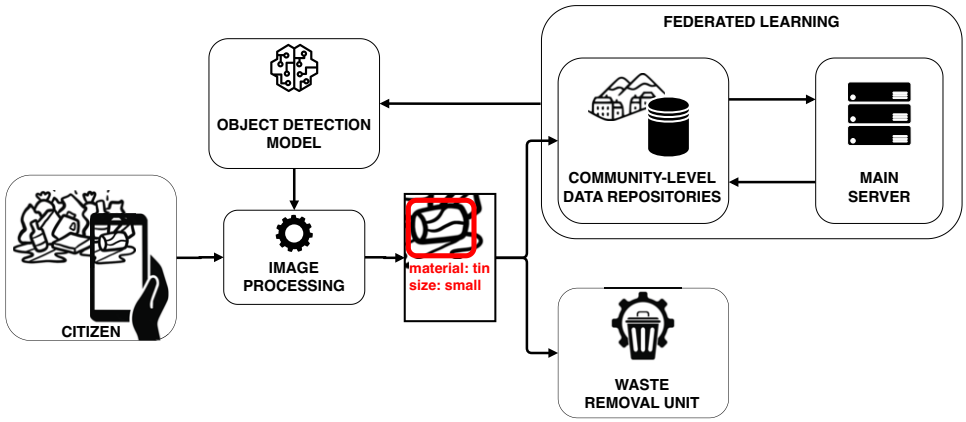


Figure 2: Schematic overview of the COBOL approach.

server to contribute to the training process would expose them to a considerable risk of security and privacy violations. Moreover, given that the application implies tight interplay with different public administration entities, compliance with data regulations is pivotal and potentially threatened by raw data transfers from citizens' phones to a centralized repository.

Federated Learning. As opposed to traditional solutions, FL decentralizes the Machine Learning (ML) training process across several *client* devices orchestrated by a *server* [19]. Fig. 1 provides an overview of the FL process. Training occurs in rounds. At the beginning of a round, the server shares the latest model parameters with the clients. Each client stores data locally that are never shared with the server or other clients, thus protecting it from privacy leaks. Upon receiving the global model parameters, each client performs a new round of learning —for a given amount of epochs— with the locally available data. When training ends, the clients return the parameters of the model trained locally to the server. The server aggregates the parameters, then updates and evaluates the global model. Learning terminates after a set number of rounds or when the global model fulfills set performance requirements (e.g., a target level of accuracy).

Since the server always has visibility exclusively over the model parameters, not the data, FL significantly reduces the risk of data leaks. FL also potentially leads to faster convergence compared to centralized training by diluting the computational load of training across the clients rather than having it concentrated on a single machine. On the other hand, FL performance varies significantly with the number of clients and their available resources [13]. Performance may also decay in the presence of heterogeneous data, that is, non-IID data from various geographically distributed devices [13]. As a result, the FL solution must be strategically tuned to meet application-specific requirements.

Proposed FL-based solution. Given the application-specific challenges, COBOL incorporates a FL phase in the approach summarized in Fig. 2. Decentralizing the object detection training process to a federation significantly reduces the chances of exposing contributing citizens to data leaks and compromising compliance with local or national data regulations. Moreover, informing citizens that the application is structured to protect their privacy and sensitive data fosters trust toward the service encouraging greater participation from the community. However, while the litter detector is meant to run on mobile devices for the inference phase, these do not provide sufficient resources to participate in the training directly.

COBOL addresses this problem by introducing the notion of *community-level data repository*. Different communities—such as small neighboring towns—share data with each other in an anonymized fashion, constitute a cluster. Different clusters never share data with each other nor with higher-level administration bodies. The resulting cluster network then serves as the federation in the FL setting, where each cluster participates as a client in the training and incremental improvement of the litter detection model. While pooling data at the community level may introduce new privacy trade-offs, we envisage community clusters to be sufficiently large to minimize the risk of re-identification (e.g., multiple towns rather than multiple households). Additional safeguard measures such as differential privacy and secure aggregation may be introduced in the FL setup to further reduce the risk of leakage.

4 Experimental setup

The following describes the adopted detector, the FL configurations, the selected datasets, and how we split them across the clients.

4.1 Litter detector

COBOL aims to develop a litter detector that can perform inference on pictures taken by citizens on the type of devices typically employed in citizen science activities, e.g., smartphones with low processing capabilities. Therefore, we select the YOLO object detector [23] as base detector model since it is a state-of-the-art, real-time object detection algorithm, belonging to the category of one-stage detectors [30]. In particular, we consider its most popular version, implemented in the Ultralytics library, i.e. YOLO-v8 focusing in particular on the most efficient *nano* size (YOLO-v8n). The chosen base detector uses input images of size 640×640 and is trained with the same default hyperparameters in all experiments with automatic batch size selection and automatic optimizer selection.

4.2 Federated Learning Setup

FL was performed using the Flower[4] framework, which provides efficient and user-friendly tools for simulation. Experiments were designed to evaluate how different data distributions across multiple

[4] https://flower.ai

nodes impact the performance of the model trained with the federated approach, simulating real-world scenarios where multiple communities collaborate to develop a global litter detection model. We specifically consider federations with 2, 4, 8, and 16 clients. For this work, we select two well-known benchmark datasets for the training. When the federation consists of two nodes, each dataset is dataset is assigned as a whole to a node. For all federations with 4 to 16 clients, we devise two schemes to distribute data between nodes:

- *Randomly distributed data*, which simulates homogeneous conditions across different nodes (i.e., the municipalities).
- *Heterogeneously distributed data*, with data distributed based on visual background similarity, simulating heterogeneous and realistic conditions across different communities.

We refer to the selection of a number of clients and a data distribution scheme as a FL *scenario*. Each scenario was run for 10 rounds, with each round comprising 10 local training epochs per client, leading to a total of 100 epochs. This structure ensures comparability between federated and centralized training approaches.

The aggregation method applied on the server was based on an adapted Federated Averaging (FedAvg) strategy. Unlike the standard FedAvg [19] , which typically weights the model updates only by the number of training samples at each node, the selected method accounts for both the improvement in training loss and the number of training samples as aggregation weights. The aggregation weight of each client is proportional to the product of its dataset size and the improvement in training loss, then normalized across clients. The dataset size reflects the statistical reliability of the local update (larger datasets provide more robust training against heterogeneous data distributions), while the improvement term prioritizes clients that actually reduce their loss, providing effective updates rather than noisy or unhelpful ones. The combination balances representativeness and progress across heterogeneous clients. This choice reflects the complexity of object detection tasks, where each data sample can contain multiple objects of varying sizes, thus requiring a loss-based weighting to better reflect model performance improvements. This approach is inspired by previous work such as FedNolowe [12], FedAWA [25], and adaptive aggregation weights for federated pancreas MRI segmentation [20], where loss-based and adaptive weighting schemes have proven effective in handling heterogeneous data scenarios.

Formally, the aggregation weight for each client i was calculated as $\text{weight}_i = \text{diff}_i \times N_i$, where N_i is the number of training samples at node i and diff_i represents the improvement in the training loss of node i, computed as:

$$\text{diff}_i = \begin{cases} \text{loss}_i^{\text{initial}} - \text{loss}_i^{\text{final}}, & \text{if } |\text{loss}_{\text{train}}| \geq 2 \\ \text{loss}_i^{\text{initial}}, & \text{otherwise} \end{cases}$$

This aggregation method ensures that nodes contributing more substantial improvements have a proportionally higher influence on the global model, thereby improving overall model robustness and performance across heterogeneous data distributions.

4.3 Datasets and dataset split across FL clients

In this work we selected two commonly used benchmark datasets collected across different geographical locations and having very different visual data distributions, as can be seen in Figures 3 and 4.

The first dataset considered is TACO, which contains 1500 images of waste captured in diverse real-world settings, including woods, roads, and beaches, and are labeled with a total of 4784 annotations. TACO objects are labeled into 60 fine-grained categories which belong to 28 super categories (from "Plastic bag & wrapper", being the most present category, to "Battery", being the least present one). To mitigate the class balance problem, the authors of TACO also propose a 1-class subdivision, which is the most frequently used annotation [4, 5, 3] where only one class is considered, i.e. the *litter* class, and is the one used here.

Figure 3: Some examples of images contained in the two datasets considered: TACO (top) and PlastOPol (bottom). Images are reshaped to square shape for better visualization.

Figure 4: Visual data distribution of the two datasets considered. Visual appearance is computed extracting neural features from an Inception v3 pretrained on ImageNet, that are then mapped into 2D space with t-SNE. [27]. For each image the nearest neighbor image belongs to the same dataset in just about 27% of the cases.

The second dataset considered is PlastOPol [4] comprising 2418 images annotated with a total of 5300 instances of litter. The images were collected via the Marine Debris Tracker, capturing diverse real-world environments such as urban areas, beaches, forests, and flint fields. PlastOPol is structured as a one-class dataset, where all the data corresponds to the *litter* class as its super-category. This makes the dataset fully compatible with the TACO version considered in this paper for both centralized and federated learning.

The two datasets considered are then split across a varying number of clients when performing FL. The first split considers only two clients, each having access to just one of the datasets. This configuration simulates the real case where each client represents a different municipality that collects data from its own citizens and is not al-

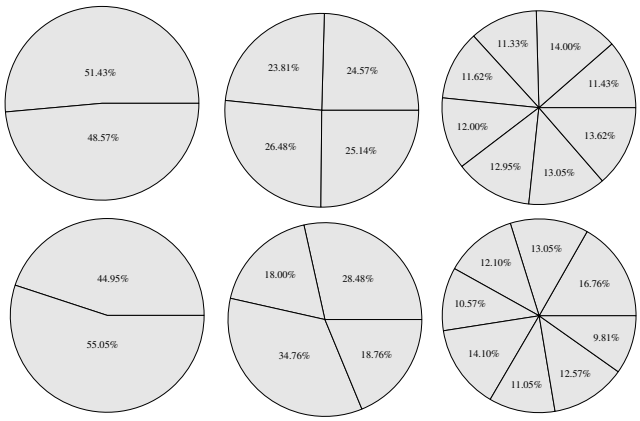


Figure 5: Ratio of training samples owned by each client on the TACO dataset. Left to right: 2, 4, and 8 clients. Split method: random (top), clustering by visual similarity (bottom).

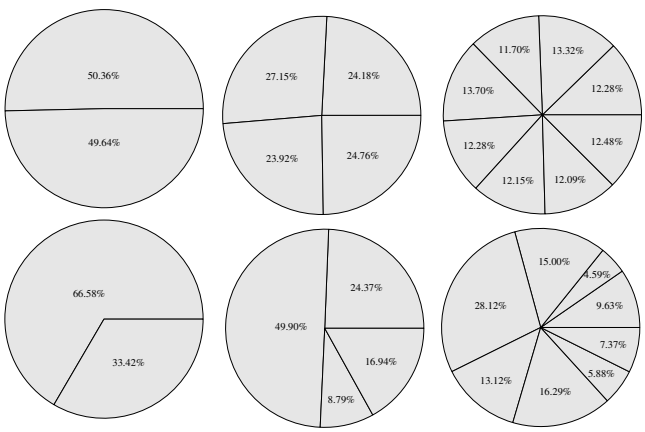


Figure 6: Ratio of training samples owned by each client on the PlastOPol dataset. Left to right: 2, 4, and 8 clients. Split method: random (top), clustering by visual similarity (bottom).

lowed to share it with other communities. We then consider a growing number of clients: 4 clients (2 for each dataset), 8 clients (4 for each dataset), and 16 clients (8 for each dataset). These configurations simulate the real case where we have a growing number of citizens and they do not want to share their data. In the case of two or more clients for each dataset, we have to decide how to split the data across clients. Here we consider two different scenarios: in the first scenario we randomly divide the data across clients, corresponding to a low data heterogeneity between clients, i.e. citizens of the same municipality. In the second scenario, we divide the data across clients on the basis of their visual similarity. For each image in the dataset we extract the neural features from the *avg_pool* layer of an ImageNet pretrained Inception v3 image classification model [26], resulting in a 2048-dimensional descriptor for each image. These descriptors are then normalized to the unitary norm and clustered with $k-$means (with $k = 2, 4, 8$) associating a single cluster to each of the clients considered. We highlight that in the first scenario the different clients are also homogeneous in the number of training samples they have. In the second scenario, instead, this is not guaranteed. We show the ratio of training data that each client has in Figure 5 for TACO and in Figure 6 for PlastOPol.

5 Experimental results

The empirical validation of the proposed FL solution addresses the following research questions:

RQ1. How accurate is the FL-trained model in detecting the *presence* of an object compared to the model trained with centralized solution?

RQ2. How accurate is the FL-trained model in detecting the presence *and position* of an object compared to the centralized approach?

To address both questions, we trained YOLO-v8n under seven federated configurations as per the setup in Section 4 (2, 4 (random and heterogeneous), 8 (random and heterogeneous), and 16 (random and heterogeneous) clients) and tested the resulting models on three distinct evaluation sets: a subset of TACO, a subset of PlastOPol, and a combined set of both (no model has seen any of these testing sets during the training). Each scenario was replicated 5 times. In both cases, our selected baseline is the accuracy of YOLO-v8n trained through the traditional centralized approach for 100 epochs. For the centralized approach, we select for training either a subset of TACO, a subset of PlastOPol, or a combination of both. The resulting models are then tested on the same test datasets selected for the FL-trained models for a fair comparison.

Building networks for federated learning simulations is computationally demanding due to the complexity of coordinating multiple nodes and processing heterogeneous data distributions. Although Flower provides user-friendly and effective methods to simplify simulations, considerable computational resources remain necessary to carry out the envisaged tasks.

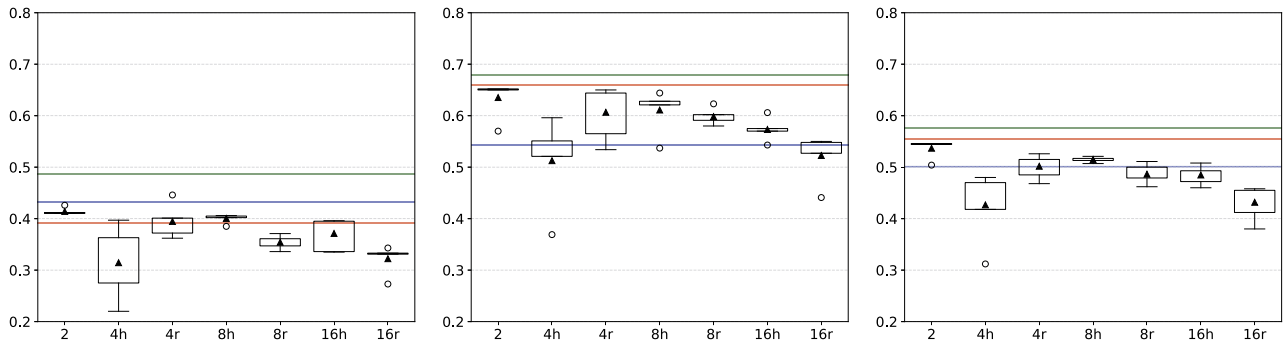
Experiments are carried out adopting the setup described in Section 4 that mimics a possible deployment environment of the COBOL approach[5]. All experiments were performed on a Linux-based commodity machine equipped with 256 GB of memory, two Intel Xeon Gold CPUs, providing a total of 24 physical cores, and two NVIDIA A40 GPUs, each with 48 GB of dedicated memory, leveraging CUDA version 12.5 for GPU-accelerated computations. For each experiment, two out of the available CPUs are assigned to the server node, while the rest are split between the clients (thus, at most 11 per client). We recall that, in the proposed solution, training occurs at community level, not on citizens' mobile devices. Therefore, the deployment environment mimics a situation in which each community is equipped with a midtier commodity machine.

5.1 RQ1: Evaluating the Sensitivity of the Model

To assess the model's accuracy in detecting the presence of an object, we computed the *recall* metric (also known as true positive rate (TPR) or sensitivity), defined as the ratio $\mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, where TP denotes true positives and FN false negatives. Recall is particularly informative in scenarios where missed detections are critical, as in litter detection systems. Results are presented in Fig. 7, where horizontal lines represent the recall achieved by the centralized model on the same test dataset trained as indicated by the label (i.e., the baseline).

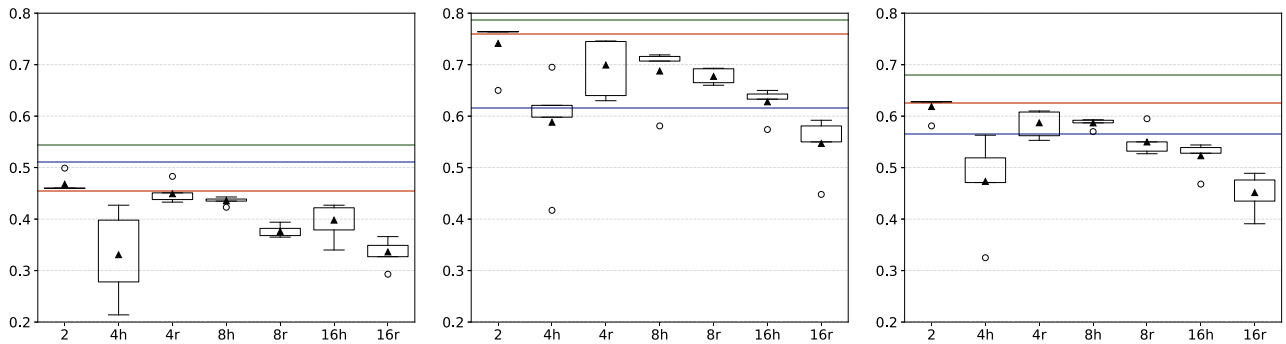
Across all test sets, the 2-client FL configuration—where each node has access to an entire dataset achieves the highest recall values, comparable to centralized training baselines. This confirms that combining two models trained on distinct datasets enhances generalization by capturing complementary features. However, increasing the

[5] Code available here: https://doi.org/10.5281/zenodo.15355714.



(a) TACO. (b) PlastOPol. (c) TACO and PlastOPol.

Figure 7: RQ1 results in terms of recall on different test datasets: TACO (a), PlastOPol (b), TACO and PlastOPol (c). Horizontal lines represent the baseline performance when trained on TACO (blue), PlastOPol (red), or both (green).



(a) TACO (b) PlastOPol (c) TACO and PlastOPol.

Figure 8: RQ2 results in terms of mAP@50 on different test datasets: TACO (a), PlastOPol (b), TACO and PlastOPol (c). Horizontal lines represent the baseline performance when trained on TACO (blue), PlastOPol (red), or both (green).

number of clients, particularly under heterogeneous (non-IID) data distributions, leads to reduced recall. This performance drop indicates that deeper training than the centralized scenario (more rounds or epochs) may be needed for the global model to converge and fully exploit the fragmented knowledge contributed by each client.

These findings suggest that flat aggregation strategies may not scale well with increasing data fragmentation. Instead, a *hierarchical aggregation* strategy—first aggregating models within clusters of similar nodes, and then merging these intermediate models—could improve convergence and robustness. This layered approach could allow local models to stabilize before contributing to the global model, potentially preserving inter-client consistency and leading to better overall performance.

5.2 RQ2: Evaluating Litter Localization Capability of the Model

The accuracy metric that also accounts for the position of the object detected in the image is the mAP@50 (mean Average Precision at an Intersection over Union of 0.50) [6], a widely used performance metric in object detection tasks. It evaluates how well a model detects and classifies objects by considering both the precision and recall across all classes. Average Precision (AP) measures the area under the precision-recall curve for a specific class, while Intersection over Union (IoU) is a threshold that determines whether a predicted bounding box is a true positive or a false positive. mAP@50 computes the AP at the IoU threshold of 0.50 for each class, then takes the mean across all classes. Results are shown in Fig. 8.

The 2-node FL setup where each node corresponds to a full dataset (TACO or PlastOPol), consistently achieves the highest mAP@50 across all test sets: combined, TACO-only, and PlastOPol-only. This configuration benefits from full data diversity at each client and stable aggregation, effectively bridging the performance gap with centralized training, suggesting that the performance can be achieved by aggregating models from 2 different data centers.

We can observe that in general federated configurations based on visual heterogeneity (i.e., 8h, 16h) tend to outperform those with random splits (i.e., 8r, 16r). A more homogeneous data distribution across randomly partitioned clients leads to more consistent model updates and improved aggregation. The results on the TACO test dataset constitute an exception since 4r achieves better performance than 4h. This is due to the imbalance of data that each client has, making the model trained by the client with less data much worse than the one trained by the client with much more data. With more clients, this is less likely to happen due to the fact that we have more models to aggregate. In the 4-client setting, one local model trained on a highly biased local dataset can have a big influence on the global model, drifting it toward that bias. This effect decreases as the num-

ber of clients increases (for example, 8 or 16).

When evaluating models on the combined test set (i.e., the union of TACO and PlastOPol), the 2-client federated model achieved the highest average mAP, closely approaching the performance of the centralized model trained on both datasets and outperforming all other FL configurations. These results indicate that aggregating models trained on just two distinct nodes—each with a complete, non-overlapping dataset—can effectively retain inter-dataset diversity and lead to good generalization. However, as the number of clients increases, each with access to a smaller subset of the data, the overall performance tends to degrade. This decline is particularly evident in random configurations (i.e., 4r, 8r, 16r), where the combination of reduced data volume per node and high data variability makes it harder to aggregate consistent and meaningful model updates.

The advantage of aggregating models trained on TACO and PlastOPol is especially notable across different test sets. This can be attributed to the complementary nature of the datasets: TACO contains fewer samples and predominantly smaller litter objects, which are generally more difficult to detect, whereas PlastOPol offers a larger number of samples with more prominent objects. As a result, the combined training enhances the global model's robustness (especially when tested over a combination of the 2 datasets).

In summary, FL-trained models demonstrate strong generalization, particularly when client data is well-structured and sufficiently diverse. In several scenarios, especially with 2-node or random-split setups, FL performance closely follows or slightly trails the centralized baselines. On the other hand, increasing the number of clients negatively affects performance, particularly under heterogeneous (non-IID) settings. This is due to smaller data volumes per client and increased variability, which together hinder stable convergence and reduce the representativeness of individual local models during aggregation.

5.3 Threats to Validity

To mitigate external validity threats and reduce the risk of obtaining results by chance, we replicate all scenarios 5 times. Preliminary evaluations show that the selected budget of clients and rounds is sufficient to achieve a plateau in accuracy.

Given the limited amount of replications, we do not resort to statistical tests (such as the Mann-Whitney U test [18]) whose reliability is limited with small sample sizes nor tests designed for small-sample scenarios which assume independent and non-aggregated outcomes [24], thus mitigating threats to conclusion validity. Our conclusions are instead based on confidence intervals whose credibility does not change with sample size.

6 Conclusions

We present a FL approach to training a litter detection model for the COBOL application case. The nature of the application requires particular focus to the distribution of the data among different clients of the federation (i.e., the different municipalities), which are expected to be heterogeneous in terms of background environment and litter objects detected in the image. The FL approach is particularly suited to this application since it allows citizens to share their pictures only at a municipality level, thus protecting their privacy.

Experimental results show that object detection models trained using the FL paradigm potentially perform comparably to centralized model training in terms of accuracy. On the other hand, results show that, without varying the number of training rounds, FL performance decays as the number of clients grows. Therefore, a key challenge to address in the future is the design, within the COBOL approach, of a self-adaptation component to adapt the FL configuration (e.g., in terms of training rounds or deployed resources) in response to changes in the federation itself (e.g., when new municipalities join or when some repositories are momentarily unreachable due to network connectivity issues).

Acknowledgements

This work has been supported the COmmunity-Based Organized Littering (COBOL) national research project, which has been funded by the MUR under the PRIN 2022 PNRR program (contract nr. P20224K9EK).

References

[1] N. M. Alahdal, F. Abukhodair, L. H. Meftah, and A. Cherif. YOLO meets FedAVG: A privacy-preserving approach to autonomous vehicles object detection. In *International Conference on Advanced Innovations in Smart Cities*, pages 01–06. IEEE, 2025.

[2] L. Baresi, S. Bianco, A. Di Salle, L. Iovino, L. Mariani, D. Micucci, L. B. R. dos Santos, M. T. Rossi, and R. Schettini. COBOL: COmmunity-based Organized Littering. In *Euromicro Conference on Software Engineering and Advanced Applications*, pages 511–517. IEEE, 2024.

[3] S. Bianco, E. Gaviraghi, and R. Schettini. Efficient deep learning models for litter detection in the wild. In *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*, pages 601–606. IEEE, 2024.

[4] M. Córdova, A. Pinto, C. C. Hellevik, S. A.-A. Alaliyat, I. A. Hameed, H. Pedrini, and R. d. S. Torres. Litter detection with deep learning: A comparative study. *Sensors*, 22(2):548, 2022.

[5] D. Das, K. Deb, T. Sayeed, P. K. Dhar, and T. Shimamura. Outdoor trash detection in natural environment using a deep learning model. *IEEE Access*, 2023.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[7] H. Guan, P.-T. Yap, A. Bozoki, and M. Liu. Federated learning for medical image analysis: A survey. *Pattern Recognition*, page 110424, 2024.

[8] R. Hann and J. Wallisch. UAV Database, 2020. URL https://doi.org/10.18710/L41IGQ.

[9] S. I. Jalal, H. A. Ahmed, and M. H. Ahmed. Design a robust real-time trash detection system using yolov5 variants. In *2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, pages 1–6. IEEE, 2023.

[10] F. A. KhoKhar, J. H. Shah, M. A. Khan, M. Sharif, U. Tariq, and S. Kadry. A review on federated learning towards image processing. *Computers and Electrical Engineering*, 99:107818, 2022.

[11] M. Kraft, M. Piechocki, B. Ptak, and K. Walas. Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle. *Remote Sensing*, 13(5), 2021. ISSN 2072-4292.

[12] D.-D. Le, T.-N. Huynh, A.-K. Tran, M.-S. Dao, and P. T. Bao. FedNolowe: A normalized loss-based weighted aggregation strategy for robust federated learning in heterogeneous environments. *PloS one*, 20 (8):e0322766, 2025.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[14] C. Liu, Y. Tao, J. Liang, K. Li, and Y. Chen. Object detection based on yolo network. In *IEEE 4th information technology and mechatronics engineering conference*, pages 799–803. IEEE, 2018.

[15] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang. FedVision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13172–13179, 2020.

[16] S. Majchrowska, A. Mikołajczyk, M. Ferlin, Z. Klawikowska, M. A. Plantykow, A. Kwasigroch, and K. Majek. Deep learning-based waste detection in natural and urban environments. *Waste Management*, 138: 274–284, 2022.

[17] S. R. Mandhati, N. L. Deshapriya, C. L. Mendis, K. Gunasekara, F. Yrle, A. Chaksan, and S. Sanjeev. pLitterStreet: Street level plastic litter detection and mapping, 2024.

[18] P. E. McKnight and J. Najab. Mann-whitney U test. *The Corsini encyclopedia of psychology*, pages 1–1, 2010.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[20] H. Pan, G. Durak, Z. Zhang, Y. Taktak, E. Keles, H. E. Aktas, A. Medetalibeyoglu, Y. Velichko, C. Spampinato, I. Schoots, et al. Adaptive aggregation weights for federated segmentation of pancreas MRI. In *2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2025.

[21] A. Patrizi, G. Gambosi, and F. M. Zanzotto. Data augmentation using background replacement for automated sorting of littered waste. *Journal of imaging*, 7(8):144, 2021.

[22] K. J. Rahman, F. Ahmed, N. Akhter, M. Hasan, R. Amin, K. E. Aziz, A. M. Islam, M. S. H. Mukta, and A. N. Islam. Challenges, applications and design aspects of federated learning: A survey. *IEEE Access*, 9: 124682–124700, 2021.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016.

[24] G. Santafe, I. Inza, and J. A. Lozano. Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review*, 44 (4):467–508, 2015.

[25] C. Shi, H. Zhao, B. Zhang, M. Zhou, D. Guo, and Y. Chang. Fedawa: Adaptive optimization of aggregation weights in federated learning using client vectors. *arXiv preprint arXiv:2503.15842*, 2025.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[27] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

[28] T. Wang, Y. Cai, L. Liang, and D. Ye. A multi-level approach to waste object segmentation. *Sensors*, 20(14):3816, July 2020. ISSN 1424-8220.

[29] J. Zhang, J. Zhou, J. Guo, and X. Sun. Visual object detection for privacy-preserving federated learning. *IEEE Access*, 11:33324–33335, 2023.

[30] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023.