Enhancing Object Detection for Autonomous Machines in Private Construction Sites Through Federated Learning

Mohammadreza Mohammadi[a*], Maghsood Salimi[b*], Mohammad Loni[b], Sima Sinaei[a]
[a] RISE Research Institutes of Sweden
Email:{mohammadreza.mohammadi, sima.sinaei}@ri.se
[b] School of Innovation, Design and Engineering, Mälardalen University, Sweden
Email:{maghsood.salimi, mohammad.loni}@mdu.se
[*] Equal Contributions.

***Abstract*—A critical enabler of autonomous construction equipment is object detection, a computer vision task integral to navigation, task execution, and safety. However, challenging conditions at construction sites, such as mud splashes, dirt, and vibrations, can degrade object detection performance by causing sensor occlusions and image blurriness. Traditional adversarial training methods, which enhance model robustness by using perturbed data, are limited in construction environments due to the scarcity of diverse real-world adversarial data and the dynamic nature of these sites. While generative models can create synthetic adversarial examples, they often struggle to generalize to the unpredictable scenarios encountered on construction sites, as they rely on rigid assumptions about data distributions. Additionally, privacy concerns and site-specific data variability hinder data sharing across different construction sites. To overcome these challenges, this paper explores Federated Learning as a solution to enhance the robustness and adaptability of object detection models while preserving data privacy. FL enables continuous online learning without direct data exchange, offering a scalable and privacy-preserving approach to training models across diverse construction environments. Experimental results demonstrate that our FL-based approach improves model performance on the ConstScene dataset by up to $\approx 4.4\%$ compared to the centralized AI model for object detection.**

***Index Terms*—Object Detection, Construction Site, Privacy, Adversarial Training, Federated Learning**

I. Introduction

The autonomous construction equipment market is expected to expand with a compound annual growth rate of 7.80% from USD 8.73 billion in 2023 to USD 15.92 billion by 2032 [1]. This growth is powered by the inherent advantages that autonomous machinery provides to construction tasks, including consistent performance, environmental sustainability, safe maneuver, and cost saving. By leveraging computer vision, the industry aims to address customer needs and improve safety and productivity of autonomous operations.

Object detection is a widely used computer vision task that plays a crucial role in navigation, task execution, and safety for autonomous construction machines such as excavators and wheel loaders [2]. However, [3], [4] environmental conditions in construction sites, such as mud splash and dirt, can occlude sensor lenses, making it difficult for the object detection models to perceive accurately. In addition, construction machines frequently operate on uneven terrains, causing significant vibrations that can result in blurry camera images [5]. Therefore, leveraging object detection models in construction sites require additional considerations to provide robust and reliable predictions.

Adversarial training methods [6], [7] are popular for defending against adversarial examples and noisy inputs. These approaches involve using intentionally perturbed data to train the model to resist such manipulations. The perturbed image data for adversarial training can be obtain either from (i) real environment, or (ii) using generative models such as generative adversarial networks (GANs) [8] and/or diffusion models [9]. However, the real perturbed data collected from a single construction site lacks diversity, as construction sites often concentrate on specific tasks. In other words, certain types of adversarial inputs, such as dirty lens, might be rarely encountered in forestry sites compared to mines and quarries. To tackle this challenge, one possible solution is to collect data from different construction sites. Fig. 1 shows an example of three different construction sites focusing on building, mining, and forestry operations. Each site presents unique adverse conditions that are ideal for collecting diverse perturbed data for adversarial training. However, since different sites are owned by different owners, data sharing between construction locations is not feasible due to privacy concerns.

On the other hand, synthetic adversarial data generation methods show promises in generating diverse data with lower cost and no privacy limits. Nevertheless, generative models rely on strong assumptions about data distribution [10], which are often ineffective in practice. Construction environments are dynamic and constantly changing—factors like shifting gravel piles, evolving machinery layouts, and the introduction of new equipment create significant outlier data that challenges generative models. Finally, adversarial training methods might overfit to adversarial examples [6]. To overcome these limitations, it is crucial to leverage continuous online learning paradigms that (i) adapt to the dynamic nature of construction sites, and (ii) guarantee privacy of clients [11].

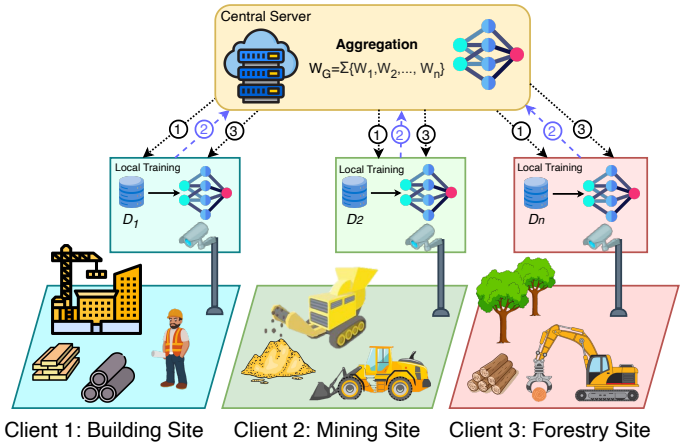

Fig. 1: Illustration of three different construction sites with different environmental conditions.

Federated Learning (FL) has recently gained attention as a method for training models without requiring clients to share data with a central entity during the training process [12], [13]. In FL, models are trained on distributed client data during each communication round. As shown in Fig. 1, client updates are aggregated iteratively to move toward the global optimum. Note that cloud servers can manage training by initiating and aggregating model updates from construction sites during each communication round [14]. Importantly, these updates are significantly smaller than the clients' entire datasets, marking a shift from traditional training methods that demand extensive data exchange.

In this paper, we leverage the idea of FL to improve adaptability and robustness of object detection models in dynamic construction environments with guaranteeing client/customer data privacy. FL has capacity to facilitate continuous online training, while minimizing data exchange makes it particularly well-suited for a wide-range of construction applications. Plus, unlike synthetic data generation methods, FL is independent of prior assumptions about the data distribution. Our results demonstrate consistent effectiveness of utilizing FL in construction environments by achieving up to $\approx 4.4\%$ higher accuracy over the centralized training paradigm on the ConstScene dataset [3].

II. Background and Related Work

A. Object Detection in Construction Industry

Vision-based sensors are prevalent in construction sites [15], generating vast amounts of image and video data for various purposes such as detecting objects (e.g. workers, material, and equipment), progress tracking, productivity measurement, and safety monitoring. While traditional computer vision methods have been employed for this purpose [16], [17], their accuracy has been limited by manual feature extraction processes and insufficient training data. To overcome these challenges, an improved Faster R-CNN approach [18] has been developed by [19], significantly enhancing real-time detection accuracy. [20] introduced a sophisticated three-stage framework for tracking multiple individuals concurrently in construction sites. Their approach initiates with a detection phase, leveraging both 2D and 3D Mask-RCNN models to locate human figures and determine their poses within images. The researchers then conducted a comparative analysis of these two approaches in terms of their detection and tracking capabilities. Following the detection stage, the second phase of their system concentrates on correlating the identified individuals across sequential frames. Construction object detection models face challenges in adverse conditions due to the limited training data and lack of robustness. To address the challenges of manually interpreting data, researchers [21] are exploring soft computing methods utilizing convolutional models emerging as a promising approach for fast construction object detection. Authors in [22] developed UIA-YOLOv5 model as a technique to enhance performance in adverse environments such as low light, fog, and rain. In the construction sites, moving obstacles often obstruct views, compromising the captured images. The research proposed by [23] adapts a U-Net-based deep learning [24] model to remove these occlusions and restore the missing background, improving image analysis in construction sites. [3] addressed the challenges of adverse weather and environmental conditions in mining construction site. [3] utilized adversarial training for improving the performance of object detection models. Although previous studies have been successful, they have largely overlooked adverse environmental issues such as image blurring caused by vibrations and dirty lenses, commonly resulting from mud splashes. Additionally, earlier research has not addressed the challenge of accessing data from other construction sites while maintaining user privacy.

B. Federated Learning (FL)

Federated learning is a distributed machine learning technique where the model is trained on a large dataset that is distributed among multiple devices or clients rather than on a centralized server [25]. FL methods play a critical role in maintaining the privacy of sensitive data where training data are distributed at the edge devices. The following section explains various terminologies related to FL:

- **Client:** refers to a device or user participating in FL. Clients maintain full control over their data, which they use to train a model.
- **Server:** The server is a central entity that coordinates the FL process. Although the server does not have direct access to the client's data, it aggregates model updates with clients during training.
- **Aggregation:** Aggregation is the process by which the server combines updates from multiple clients to form a global model. This process is designed to preserve privacy, ensuring that the global model is updated without compromising the confidentiality of individual client data.
- **Server Round:** A server round refers to a single iteration of the FL process. During each round, clients locally train

their models on their own data, send model updates to the server, and receive the updated global model in return.

- **Local Epoch:** A local epoch is the number of times a client trains its model on its local data before sending the updated model to the server for aggregation.

There are generally three steps involved in FL training (Fig. 1): (1) the central server shares an initial model. (2) Participants train their local data with the initial model and share the local model with the central server, and (3) the central server aggregates the local models and shares the global model with participants.

C. Applications and Emerging Use Cases of FL

FL has been successfully implemented in various public-facing and industrial applications, including mobile keyboard prediction [26], credit risk assessment [27], and vehicular management systems [28]. In the context of construction industry, FL plays a crucial role in enabling collaboration among various entities to build smart and safe working sites [29]–[31]. This collaboration facilitates the collective training of shared AI models while preserving user privacy. To the best of our knowledge, FL has not yet been applied to vision-base tasks in construction sites for improving robustness of predictions, making it a novel approach in this context.

III. Method

In this section, we present a detailed explanation of how semantic segmentation is integrated into the FL algorithm. The proposed method applies FedAvg for semantic segmentation across distributed construction sites (Alg. 1). The global model is initialized at the central server, which is typically a deep neural network designed for segmentation tasks. In each communication round, all the clients (e.g., construction sites) perform local training. Each client uses its local image data to update the model by minimizing a pixel-wise loss function, such as cross-entropy, over multiple local epochs. The locally updated model weights are then sent back to the central server, where they are averaged to update the global model. This iterative process continues for several communication rounds until the global model converges.

In this setup, the key parameters include the number of clients K, the number of communication rounds T, the number of local epochs E, and the learning rate η. The client-side training involves feeding image batches through the segmentation model, computing the pixel-wise segmentation loss, and updating the model using gradient descent. The advantage of this federated approach lies in preserving data privacy while enabling the global model to generalize across diverse environments by leveraging data from multiple sites.

In this work, we require a system that is accurate while simultaneously boosting privacy and communication efficiency. To justify the implementation of an FL system between construction sites, we must compare the accuracy of our FL system to that of local models. In addition, our architecture provides a higher level of privacy by preventing the server or other construction sites from accessing the data of other nodes. Furthermore, our architecture reduces communication overhead by just providing model updates, which are far less than raw data. This reduction in data transfer saves bandwidth while also speeding up the entire training process.

Algorithm 1 Federated Averaging for Semantic Segmentation

Server-Side:

Require: K: clients, T: communication rounds, E: local epochs, η learning rate, w_0: initial global model weights
  for each round $t = 1, \ldots, T$ **do**
    Randomly select a subset of clients $\mathcal{S}_t \subseteq \{1, 2, \ldots, K\}$
    for each client $k \in \mathcal{S}_t$ **in parallel do**
      $w_k^t \leftarrow \text{ClientUpdate}(k, w_{t-1})$
    end for
    Aggregate updates: $w_t \leftarrow \frac{1}{|\mathcal{S}_t|} \sum_{k \in \mathcal{S}_t} w_k^t$
  end for
  Return: Final global segmentation model weights w_T

Client-Side:

Require: k: client index, w: global model weights
  Receive model weights from server $w_k \leftarrow w$
  for each local epoch $e = 1, 2, \ldots, E$ **do**
    for each batch of image data (x, y) from client k **do**
      $y_{\text{pred}} \leftarrow \text{ModelForward}(w_k, x)$ {Perform forward pass for segmentation}
      Compute loss: $\ell \leftarrow \text{CrossEntropyLoss}(y_{\text{pred}}, y)$
      $w_k \leftarrow w_k - \eta \nabla \ell$ {Update local weights using gradient descent}
    end for
  end for
  Return: Updated local model weights w_k

IV. Experiments

A. Experimental Setup

Dataset Preparation. For this study, we utilized the ConstScene dataset [3], which is designed to address the challenges of adverse weather and environmental conditions in construction environments. This dataset includes annotated images captured under various weather conditions such as sunny, rainy, foggy, and low-light scenarios, along with environmental factors like dirt and mud on camera lenses. The ConstScene dataset is available in two versions: the original with 3470 images and the augmented version with 6240 images. To supplement the original dataset, synthetic images incorporated to mimic actual conditions found on construction sites.

These artificially generated images were added to enhance the dataset's diversity and better represent real-world scenarios. Two distinct categories of synthetic images are: those with motion blur effects and those simulating a dirty lens effect. For generating motion blur effect, the process involved creating a blur kernel - a 2D array with zeros and a single row of ones to create directional blur. The blur intensity is

controlled by the kernel size parameter, which is set to 11 in this study. For adding dirty lens effect, several synthetic dirty lens effects were used to mimic the real conditions. The amount of dirtiness is controlled by the intensity of the pixels used in each filter. In this process, for each input image, one random filter was selected and applied on the original image. Fig. 2.(c) and (Fig. 2.(d) show dirty lens and blurry samples of the dataset, respectively.

To construct a robust and diverse dataset for training and evaluation, we developed three distinct data sources (DS), which were structured as follows:

- DS1: Composed entirely of the 2,071 no-augmented images captured in normal condition.
- DS2: A combination of 37 images captured in dusty environment supplemented with 1,000 randomly selected images from the synthetic motion blur dataset.
- DS3: A combination of 662 images captured dirty lens camera supplemented with 1,000 randomly selected images from the synthetic dirty lens dataset.
- Test dataset: Composed of 348 no-augmented images captured in normal conditions, 348 synthetic dirty lens images, and 348 synthetic motion blurred images.

Then, we considered three nodes for the FL system to mimic different autonomous construction environments. In the first node, most of the data is selected from DS1 to emulate an environment with ideal conditions. The second node contains mostly data from DS2 to imitate an environment that is dusty and has a lot of lows and highs which cause blurred images. The data of the third node consists mostly of DS3 images to mimic a muddy environment that leads to a dirty lens camera. The data distribution of the three nodes is as follows:

- Node 1: 1985 images (90% DS1, 5% DS2, and 5% DS3)
- Node 2: 1087 images (90% DS2, 5% DS1, and 5% DS3)
- Node 3: 1093 images (90% DS3, 5% DS1, and 5% DS2)

Training Configuration. Our FL system has three nodes and one server. The server initiates a U-Net model [24] with ResNet18 encoder architecture [32] and sends a copy of it to the three local sites. Then, the federated training is performed with the settings specified in Table I. W employ the mean intersection over union (mIoU) metric for evaluation. mIoU is defined as: $mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{|X_i \cap Y_i|}{|X_i \cup Y_i|}$, where N, X, and Y represent the number of classes, the set of pixels predicted to belong to class i, and the set of pixels in the ground truth that belong to class I.

B. Results

We report the results in Table II.The comparison between locally trained models and the FL model highlights the advantages of using the FL approach in our specific use case. Training each node on its individual local dataset and testing on the same dataset used in the federated configuration revealed that the FL model outperformed the locally trained models significantly. The FL model provides the mIoU of **56.37%**, suggesting its superior ability to generalize across different and heterogeneous datasets. This finding highlights the effectiveness of FL in circumstances with non-identical data distribution and its potential to improve model performance in distributed environments. By utilizing FL, we not only gained superior model performance, as indicated by the higher mIoU score, but also improved the overall privacy of the system, making it a solid option for scenarios demanding severe data protection measures. The qualitative prediction results of the centralized training and FL methods against dirty lens and blurry image attacks are illustrated in Fig. 2.(e) to Fig. 2.(h). As can be seen, employing FL shows a notable accuracy improvement over centralized training (up to 4.4%).

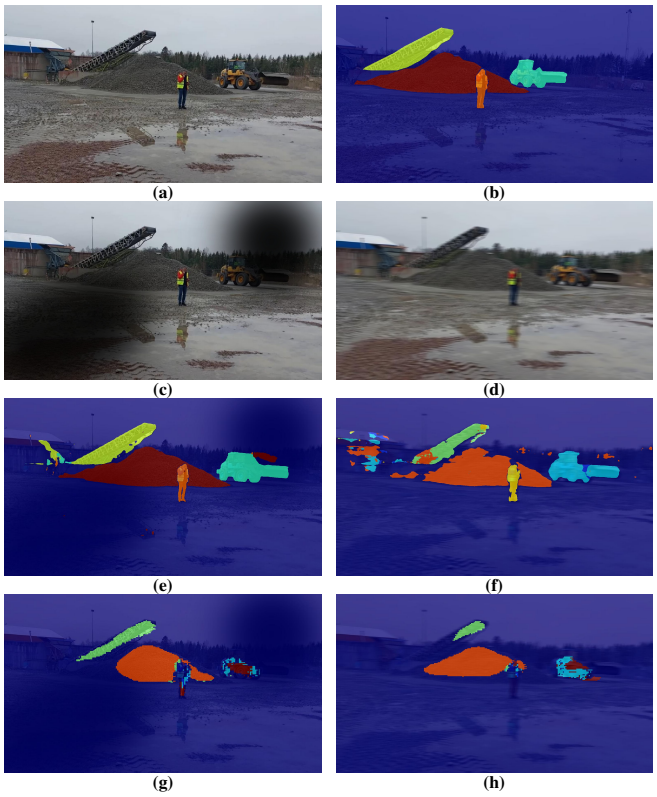

Fig. 2: Illustration of sample inputs of the prepared dataset alongside qualitative prediction results. (a) Original image. (b) Semantic label for the original image (ground truth). (c) Dirty image. (d) Blurred image. (e) Prediction result of the centrally trained model on the dirty lens image. (f) Prediction result of the centrally trained model on the blurry image. (g) Prediction result of the FL model on the dirty lens image. (h) Prediction result of the FL model on the blurry image.

TABLE I: Configurations setup of the training procedure.

Parameter	Value
Epochs for local training	1
Epochs for global training	100
Optimizer	Adam
Learning rate	3e-4
Batch size	32
Aggregation Algorithm	Fed-Avg [12]
GPU Device	Tesla M10
Global training time	3865 seconds

TABLE II: Test results of centralized and FL models on the ConstScene [3] dataset.

Model	mIoU
Node 1	53.77%
Node 2	52.08%
Node 3	51.94%
Federated learning (Ours)	56.37%

V. Conclusion

Developing robust and reliable object detection models is difficult for individual construction sites without access to diverse datasets that account for specific adverse conditions. Federated learning addresses this limitation by allowing multiple construction sites to collaborate and pool their data to train a global model, thereby enhancing accuracy across a wider range of adverse environmental conditions. Importantly, FL maintains data privacy, as there is no direct data sharing; instead, a central model is created through the aggregation of updates from individual sites. This approach also provides easy scalability, flexible training schedules, and access to larger training datasets through multi-site collaborations, which are all crucial for the successful deployment of robust object detection solutions.

References

[1] S. Singh, "Global autonomous construction equipment market overview," https://www.marketresearchfuture.com/reports/autonomous-construction-equipment-market-12648/, 2024, [Online; accessed August 2024].

[2] B. Xiao and S.-C. Kang, "Development of an image data set of construction machines for deep learning object detection," *Journal of Computing in Civil Engineering*, vol. 35, no. 2, p. 05020005, 2021.

[3] M. Salimi, M. Loni, S. Afshar, M. Sirjani, and A. Cicchetti, "Constscene: Dataset and model for advancing robust semantic segmentation in construction environments," *arXiv preprint arXiv:2312.16516*, 2023.

[4] N. Vikas, G. Pahwa, and S. Mohanty, "Camera blockage detection in autonomous driving using deep neural networks," in *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*. IEEE, 2022, pp. 1–6.

[5] C. Morikawa, M. Kobayashi, M. Satoh, Y. Kuroda, T. Inomata, H. Matsuo, T. Miura, and M. Hilaga, "Image and video processing on mobile devices: a survey," *The Visual Computer*, vol. 37, no. 12, pp. 2931–2949, 2021.

[6] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," *arXiv preprint arXiv:2102.01356*, 2021.

[7] Z. Qian, K. Huang, Q.-F. Wang, and X.-Y. Zhang, "A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies," *Pattern Recognition*, vol. 131, p. 108889, 2022.

[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[9] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10 850–10 869, 2023.

[10] H. Lee, J. Lu, and Y. Tan, "Convergence of score-based generative modeling for general data distributions," in *International Conference on Algorithmic Learning Theory*. PMLR, 2023, pp. 946–985.

[11] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469–6486, 2020.

[12] 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*. PMLR, 2017, pp. 1273–1282.

[13] P. M. Mammen, "Federated learning: Opportunities and challenges," *arXiv preprint arXiv:2101.05428*, 2021.

[14] M. F. Pervej, R. Jin, and H. Dai, "Resource constrained vehicular edge federated learning with highly mobile connected vehicles," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 6, pp. 1825–1844, 2023.

[15] S. Xu, J. Wang, W. Shou, T. Ngo, A.-M. Sadick, and X. Wang, "Computer vision techniques in construction: a critical review," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3383–3397, 2021.

[16] S. Chi and C. H. Caldas, "Automated object identification using optical video cameras on construction sites," *Computer-Aided Civil and Infrastructure Engineering*, vol. 26, no. 5, pp. 368–380, 2011.

[17] M.-W. Park and I. Brilakis, "Construction worker detection in video frames for initializing vision trackers," *Automation in Construction*, vol. 28, pp. 15–25, 2012.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[19] W. Fang, L. Ding, B. Zhong, P. E. Love, and H. Luo, "Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach," *Advanced Engineering Informatics*, vol. 37, pp. 139–149, 2018.

[20] O. Angah and A. Y. Chen, "Tracking multiple construction workers through deep learning and the gradient based method with re-matching based on multi-object tracking accuracy," *Automation in Construction*, vol. 119, p. 103308, 2020.

[21] N. D. Nath and A. H. Behzadan, "Deep convolutional networks for construction object detection under different visual conditions," *Frontiers in Built Environment*, vol. 6, p. 97, 2020.

[22] Y. Ding, M. Zhang, J. Pan, J. Hu, and X. Luo, "Robust object detection in extreme construction conditions," *Automation in Construction*, vol. 165, p. 105487, 2024.

[23] O. Angah and A. Y. Chen, "Removal of occluding construction workers in job site image data using u-net based context encoders," *Automation in Construction*, vol. 119, p. 103332, 2020.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[25] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," *Information processing & management*, vol. 59, no. 6, p. 103061, 2022.

[26] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[27] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.

[28] K. Tan, D. Bremner, J. Le Kernec, and M. Imran, "Federated machine learning in vehicular networks: A summary of recent applications," in *2020 international conference on UK-China emerging technologies (UCET)*. IEEE, 2020, pp. 1–4.

[29] S. J. S. Moe, B. W. Kim, A. N. Khan, X. Rongxu, N. A. Tuan, K. Kim, and D. H. Kim, "Collaborative worker safety prediction mechanism using federated learning assisted edge intelligence in outdoor construction environment," *IEEE Access*, 2023.

[30] X. Li, H.-l. Chi, W. Lu, F. Xue, J. Zeng, and C. Z. Li, "Federated transfer learning enabled smart work packaging for preserving personal image information of construction worker," *Automation in Construction*, vol. 128, p. 103738, 2021.

[31] H.-T. Wu, H. Li, H.-L. Chi, W.-B. Kou, Y.-C. Wu, and S. Wang, "A hierarchical federated learning framework for collaborative quality defect inspection in construction," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108218, 2024.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.