ESAIM 2024 – 2nd European Symposium on Artificial Intelligence in Manufacturing

Collaborative Learning in Shared Production Environment using Federated Image Classification

Vinit Hegiste[1,*], Tatjana Legler[1,2], and Martin Ruskowski[1,2]

[1] Chair of Machine Tools and Control Systems, RPTU Kaiserslautern-Landau, Kaiserslautern, Germany
[2] Innovative Factory Systems (IFS), German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

[*] Corresponding author. Tel.: +49 631 2055059;. E-mail: vinit.hegiste@rptu.de

Abstract. The application of federated learning (FL) in industrial settings offers promising advancements in maintaining data privacy while collaboratively training machine learning models. This study focuses on the comparative analysis of federated image classification versus locally trained models within a shared production environment. Specifically, we explore the classification of windshields in truck cabins, which is a crucial task for quality inspection in manufacturing of trucks. Our research involves four clients, each producing different types of truck cabins and research based on FL process between them. Various deep learning architectures, including VGG19, ResNet50, InceptionNetv3, DenseNet-121, and EfficientNetv2-s, were evaluated under a FL framework implemented using the FLOWER framework. A custom plain averaging strategy was used for weight aggregation. The global model's performance was assessed using a combined test set from all clients and compared against models trained locally by individual clients. The results highlight the effectiveness of FL in enhancing model generalization and adaptability to new product variations in industrial applications, promoting its adoption for collaborative quality inspection tasks.

Keywords: Federated Learning · Image Classification · Quality Inspection · Deep Learning · Industrial Applications

1 Introduction

The quality of the dataset is pivotal in training machine learning models. High-quality datasets lead to the development of robust models that perform effectively across a range of applications [1]. Federated learning (FL), a distinct paradigm of machine learning, facilitates the training of a cohesive model through

the collaborative efforts of multiple clients. This approach involves the aggregation of model weights from each participant, ensuring that the training data remains on the local servers, and enhancing the model's ability to perform in unfamiliar testing environments [2]. Recent years have witnessed a surge in FL applications, driven by the growing demand for data privacy and the need for collaborative solutions across industries [3], [4]. Despite its increasing popularity, the application of FL in visual tasks within the manufacturing sector for custom datasets does not have much research compared to the FL algorithms and architectures tested on IID (independent and identically distributed) datasets such as CIFAR-10 and MNIST [4], [5]. Furthermore, there is a notable scarcity of studies comparing models trained via FL with those trained using traditional, local datasets. Such comparisons are crucial in the commercial sector, as they highlight the differences in performance on test datasets between locally trained models and those trained through a federated approach [6]. This contrast not only showcases the effectiveness of FL in enhancing model generalization across diverse datasets but also encourages more companies to engage in FL initiatives [7].

This research focuses on the comparative analysis of federated image classification within a shared production environment. Our study involves four clients, each producing different types of truck cabins, with or without windshields. We examine the efficacy of various deep learning architectures in a FL setting and evaluate two distinct strategies for weight selection in federated models. This investigation aims to shed light on the optimal configurations and strategies that enhance performance in FL applications, particularly in industrial settings.

2 Related Work

The adoption of FL in industrial applications, particularly for quality inspection and predictive maintenance, has garnered significant interest. This interest is driven by FL's ability to train models collaboratively without compromising data privacy. [8] explored failure prediction using FL on production lines, illustrating the efficacy of FL in real-world scenarios. [5] furthered this research by developing federated object detection algorithms for quality inspection tasks in manufacturing environments.

Introduced by McMahan et al. [2], Federated Averaging (FedAvg) has become a foundational algorithm in FL, enabling multiple devices to collaboratively train a model while maintaining data localization and privacy. However, there has been limited exploration into comparing different deep learning architectures within FL frameworks like FedAvg to assess their impact on model performance [9], [10]. Evaluating these architectures in an FL context is particularly important, as highlighted by [11], given their widespread use in image classification and the varying complexity they offer. Studies such as those by [10], [5], and [12] have begun addressing this gap by analyzing the performance differences between federated and centralized models in industrial settings. Additionally, [13] explored strategies for integrating new clients into FL networks, enhancing performance

in dynamic environments. Further work by [14] demonstrated FL's ensemble capabilities, showing improved object detection in previously unseen scenarios, further supporting the case for FL's application in complex industrial contexts.

3 Implementation

This section outlines the FL architecture and framework utilized, the distribution and characteristics of the dataset for experimentation, and the external test dataset employed for evaluating the globally generated models against the locally trained client models.

3.1 Federated Learning Framework

Several FL frameworks facilitate research by simplifying the integration and testing processes. Notable frameworks include TensorFlow Federated [15], PySyft [16], and FLOWER [17]. Among these, FLOWER is chosen for its ease of integration and effectiveness in research-oriented applications. FLOWER is a flexible and user-friendly framework that supports various experimental setups.

In our FL setup, all clients shared the same deep learning architecture and hyperparameters to ensure consistency across the federated learning process. The architectures varied, but the hyperparameters were kept constant: batch size=16, optimizer=SGD with momentum=0.8, learning rate=0.001, loss function=cross-entropy, and image size=300. Deep learning architectures—EfficientNetv2 (small) [18], VGG19 [19], ResNet50 [20], DenseNet-121 [21], and InceptionNetv3 [22] were selected for their proven performance in image classification tasks and their varying complexities in terms of trainable parameters. This selection allows for a comprehensive analysis of how different architectures impact the effectiveness of federated learning in handling diverse and complex data scenarios, particularly in an industrial setting. The FL strategy employed was plain averaging of model weights for the global federated model, customized within the FLOWER framework for each architecture as mentioned in Table 1.

3.2 Dataset

The primary scenario for this research involves detecting the presence of a windshield in truck cabins as a quality inspection application. The dataset comprises four clients, each identified by the color of their cabins: Blue, Green, Orange, and Red. Each client's dataset includes two labels: 'No_windshield' and 'With_windshield', as illustrated in Figure 1. The total data distribution can be referred to in Figure 2. An external test dataset was also developed to challenge the robustness of local models under FL paradigms. This dataset includes cabins of different colors (gray and purple) and features a novel type of windshield, depicted on the right side of Figure 1.

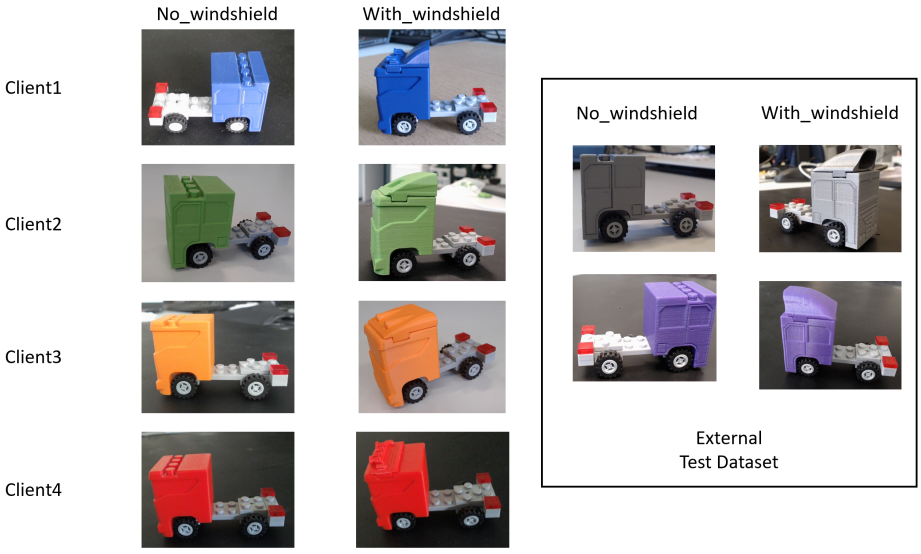

Fig. 1. Local Client dataset for each client in FL (left) and external test dataset with 2 Colored Cabins along with a novel Cabin type which none of the clients have ever seen before (right)

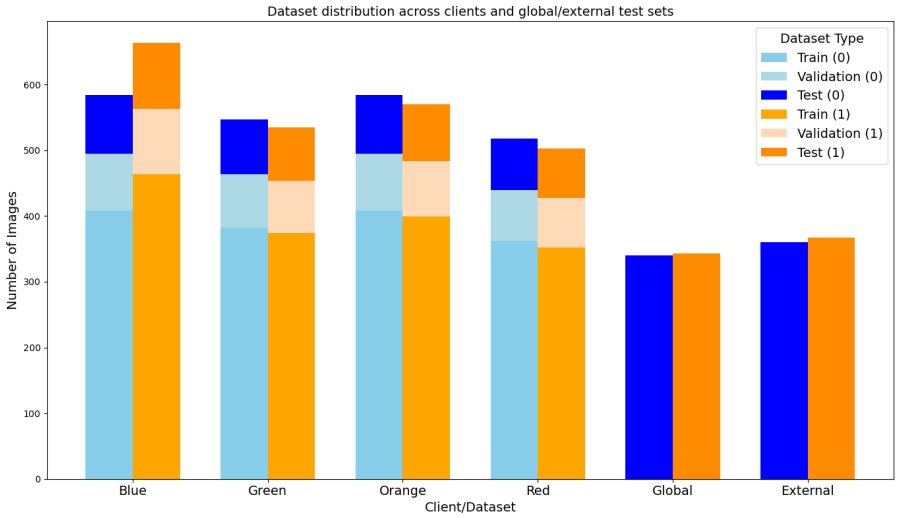

Fig. 2. Distribution across clients and global/external testsets. 'No_windshield' is represented as 0 and 'With_windshield' is represented as 1. The Global dataset is a combination of the test set of all 4 clients.

3.3 Experimental Procedure

We began with an FL architecture where all clients shared the same deep learning architecture and hyperparameters. The architectures were varied, but the hyperparameters were kept constant: batch size=16, optimizer=SGD with momentum=0.8, learning rate=0.001, loss function=cross-entropy, and image size=300, using the final weights from each local epoch. The FL strategy employed was custom plain averaging of model weights for the global federated model, customized within the FLOWER framework for different deep learning architectures as mentioned in Table 1. To expedite the testing process, a global test set was created by amalgamating the test sets of all clients. The global model was evaluated against this global test set following each communication round to achieve the best combination of global weights and hyperparameters.

4 Results and Discussion

After extensive experimentation with different communication rounds (CRs) and epochs, the optimal global federated model was achieved using 5 local epochs and 15 CRs. This section presents the performance metrics of various deep learning architectures, comparing both individual client models and the federated global model on the global test dataset. Table 1 outlines the Accuracy, Precision, Recall,

Table 1. Performance Metrics for Various Architectures on Global Test Dataset

Architecture	Metric	Client1	Client2	Client3	Client4	Global Model
DenseNet-121	Accuracy	0.7438	0.8287	0.7818	0.6823	**0.9971**
	Precision	0.8207	0.8495	0.8210	0.6837	**0.9971**
	Recall	0.7438	0.8287	0.7818	0.6823	**0.9971**
	F1 score	0.7278	0.8262	0.7752	0.6818	**0.9971**
EfficientNetv2	Accuracy	0.6633	0.7277	0.6281	0.6310	**0.9898**
	Precision	0.7694	0.7695	0.7580	0.6320	**0.9899**
	Recall	0.6633	0.7277	0.6281	0.6310	**0.9898**
	F1 score	0.6272	0.7170	0.5756	0.6302	**0.9898**
VGG19	Accuracy	0.8594	0.8389	0.8873	0.6428	**0.9912**
	Precision	0.8827	0.8399	0.8923	0.7920	**0.9913**
	Recall	0.8594	0.8389	0.8873	0.6428	**0.9913**
	F1 score	0.8574	0.8389	0.8869	0.5913	**0.9912**
ResNet50	Accuracy	0.5754	0.7130	0.5007	0.5666	**0.9941**
	Precision	0.7709	0.8148	0.7507	0.7401	**0.9942**
	Recall	0.5754	0.7130	0.5007	0.5666	**0.9942**
	F1 score	0.4834	0.6883	0.3374	0.4727	**0.9941**
InceptionNetv3	Accuracy	0.4978	0.4890	0.4890	0.4978	**0.5212**
	Precision	0.2478	0.4569	0.4704	0.2478	**0.6231**
	Recall	0.4978	0.4890	0.4890	0.4978	**0.5212**
	F1 score	0.3309	0.3636	0.3829	0.3309	**0.3998**

and F1 scores for each model. The federated global model consistently outperformed individual client models across all architectures. For instance, DenseNet-121 achieved a global model F1 score of 0.9971, significantly higher than the individual client F1 scores, which ranged from 0.6818 to 0.8262. Similar trends were observed with other architectures, where the federated model demonstrated superior performance, underlining the effectiveness of federated learning (FL) in improving model generalization.

Table 2. F1 Scores on Global Test Dataset (Merging all Client's Testset) for Centralized vs. Global FL Model

Training Type	DenseNet	EfficientNet	VGG	ResNet	InceptionNet
Centralized	0.8097	0.6145	0.8528	0.7296	**0.4286**
Federated	**0.9971**	**0.9898**	**0.9912**	**0.9941**	0.3998

Table 3. F1 Scores on External Test Dataset (Gray and Purple Cabins) for Centralized vs. Global FL Model

Training Type	DenseNet	EfficientNet	VGG	ResNet	InceptionNet
Centralized	0.8569	0.6102	0.9100	0.7654	**0.4467**
Federated	**0.9821**	**0.9876**	**0.9586**	**0.9917**	0.4394

To further evaluate the robustness of these models, we tested both centralized and federated models on an external test dataset consisting of gray and purple cabins with an unseen windshield type. This scenario simulates a real-world use case where a company integrates a new windshield type into its manufacturing process, and the goal is to assess how well pretrained models can handle such unseen data. Table 3 presents the F1 scores for both centralized and federated models on this external test dataset. The results indicate that federated models generally outperform their centralized counterparts, particularly with DenseNet, EfficientNet, VGG, and ResNet architectures, achieving F1 scores of 0.9821, 0.9876, 0.9586, and 0.9917, respectively. This demonstrates the superior robustness and generalization capability of federated models when exposed to unseen data, highlighting their potential for real-world industrial applications where new components or product variations are frequently introduced.

In summary, the results highlight that federated learning not only enhances model performance on combined datasets but also significantly improves the model's ability to generalize to new, unseen scenarios. The VGG19 model has approximately 143.67 million parameters, ResNet50 has around 25.56 million parameters, DenseNet-121 has about 7.98 million parameters, EfficientNetv2 has approximately 21.55 million parameters, and InceptionNetv3 has about 23.85 million parameters. Despite having a relatively lower number of trainable parameters, DenseNet's federated global model achieved near-perfect performance

metrics, demonstrating its efficiency and suitability for resource-constrained environments. These findings emphasize the efficacy of federated learning in industrial settings, where data privacy and the ability to adapt to new conditions are paramount.

5 Conclusion and Outlook

This study explored the application of Federated Learning (FL) for image classification within a shared production environment, focusing on classifying windshields in truck cabins. We evaluated the performance of several deep learning architectures, comparing models trained locally by individual clients with a global model obtained through FL using a custom plain averaging strategy. The experimental results indicate that FL significantly enhances model performance across all tested architectures, consistently achieving higher accuracy, precision, recall, and F1 scores compared to individual client models. This portrays FL's potential to create robust and generalized models by aggregating knowledge from multiple sources while preserving data privacy. Furthermore, testing on an external dataset with unseen windshield types demonstrated the adaptability and robustness of federated models in handling unforeseen data. Considering both performance metrics and the total number of trainable parameters, DenseNet-121 emerged as the most suitable architecture, offering near-perfect performance with fewer trainable parameters, making it both efficient and resource-friendly. In contrast, InceptionNetv3 consistently underperformed, indicating its unsuitability for this task.

The results also highlight the limitations of local models trained on isolated datasets, which perform poorly in comparison. FL addresses this by enabling a superior global model without data sharing, critical in industrial applications where data privacy is paramount. Future work will extend this approach to other quality inspection tasks and explore advanced FL strategies, such as differential privacy and secure multi-party computation, to enhance data security. Additionally, integrating FL with real-time industrial systems for continuous learning and adaptation to new production scenarios will be investigated.

Acknowledgment

This work was funded by the Carl Zeiss Stiftung, Germany under the Sustainable Embedded AI project (P2021-02-009).

References

1. Chen, H., Chen, J., Ding, J.: Data evaluation and enhancement for quality improvement of machine learning. IEEE Transactions on Reliability **70**, 831–847 (2021)
2. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS. pp. 1273–1282 (2017)

3. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST) **10**(2), 12 (2019)
4. Wen, J., Zhang, Z., Lan, Y., Jin, W.: A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics **14**, 513–535 (2023)
5. Hegiste, V., Legler, T., Ruskowski, M.: Federated object detection for quality inspection in shared production. In: 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC). pp. 151–158 (2023)
6. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 1175–1191. ACM (2017)
7. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine **37**(3), 50–60 (2020)
8. Ge, N., Li, G., Zhang, L., Liu, Y.: Failure prediction in production line based on federated learning: an empirical study. Journal of Intelligent Manufacturing **33**, 2277–2294 (2021)
9. Islam, F., Raihan, A.S., Ahmed, I.: Applications of federated learning in manufacturing: Identifying the challenges and exploring the future directions with industry 4.0 and 5.0 visions. arXiv preprint arXiv:2302.13514 (2023)
10. Hegiste, V., Legler, T., Ruskowski, M.: Application of federated machine learning in manufacturing. In: 2022 International Conference on Industry 4.0 Technology (I4Tech). pp. 1–8 (2022)
11. Alzubaidi, L., Zhang, J., Humaidi, A., et al.: Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. Journal of Big Data **8**(53), 1–74 (2021)
12. Hegiste, V., Walunj, S., Antony, J., Legler, T., Ruskowski, M.: Enhancing object detection with hybrid dataset in manufacturing environments: Comparing federated learning to conventional techniques (May 2024), presented at the International Conference on Innovative Engineering Sciences and Technological Research (ICIESTR-2024)
13. Legler, T., Hegiste, V., Ruskowski, M.: Mapping of newcomer clients in federated learning based on activation strength. Journal of Manufacturing and Federated Learning **1**(1), 100–110 (2021)
14. Hegiste, V., Legler, T., Ruskowski, M.: Federated ensemble yolov5: Enhancing object detection via federated learning. In: Proceedings of the International Conference on Machine Learning and Privacy. pp. 45–60 (2021)
15. Inc., G.: Tensorflow federated: Machine learning on decentralized data. Software available from tensorflow.org (2019), https://www.tensorflow.org/federated
16. Trask, A., Mancuso, J., Ryffel, T., Rueckert, D.: Pysyft: A library for encrypted, privacy preserving machine learning. In: 31st Conference on Neural Information Processing Systems (NIPS 2018) (2018)
17. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., Lane, N.D.: Flower: A friendly federated learning research framework (2020), https://flower.dev/
18. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML. pp. 6105–6114 (2019)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)

21. Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
22. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826 (2016)