

Federated Ensemble YOLOv5 – A Better Generalized Object Detection Algorithm

1st Vinit Hegiste

*Chair of Machine Tools and Control Systems
RPTU Kaiserslautern-Landau
Kaiserslautern, Germany
vinit.hegiste@rptu.de*

2nd Tatjana Legler

*Chair of Machine Tools and Control Systems
RPTU Kaiserslautern-Landau
Kaiserslautern, Germany
tatjana.legler@rptu.de*

3rd Martin Ruskowski

*Innovative Factory Systems (IFS)
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
martin.ruskowski@dfki.de*

Abstract—Federated learning (FL) has gained significant traction as a privacy-preserving algorithm, but the underlying resemblances of federated learning algorithms like Federated averaging (FedAvg) or Federated SGD (Fed SGD) to ensemble learning algorithms has not been fully explored. The purpose of this paper is to examine the application of FL to object detection as a method to enhance generalizability, and to compare its performance against a centralized training approach for an object detection algorithm. Specifically, we investigate the performance of a YOLOv5 model trained using FL across multiple clients and employ a random sampling strategy without replacement, so each client holds a portion of the same dataset used for centralized training. Our experimental results showcase the superior efficiency of the FL object detector's global model in generating accurate bounding boxes for unseen objects, with the test set being a mixture of objects from two distinct clients not represented in the training dataset. These findings suggest that FL can be viewed from an ensemble algorithm perspective, akin to a synergistic blend of Bagging and Boosting techniques. As a result, FL can be seen not only as a method to enhance privacy, but also as a method to enhance the performance of a machine learning model.

Index Terms—Federated learning, FedAvg, Bagging, Boosting, Ensemble learning, Object detection.

I. INTRODUCTION

In recent years, federated learning (FL) has emerged as a promising privacy-preserving algorithm that allows multiple clients to collaboratively train a global model without sharing their raw data [1]. The concept of FL involves aggregating local model updates from distributed clients to create a more robust and accurate global model. While FL has gained significant traction for its privacy benefits, its underlying resemblance to ensemble learning algorithms such as Bagging or Boosting remains unexplored. The objective of this paper is to investigate the parallels between federated learning algorithms, such as Federated Averaging (FedAvg) and ensemble learning algorithms, and to explore the application of FL to object detection tasks. The aggregation of weights to achieve a better global model can be compared with the Bagging

algorithm, where multiple clients have data belonging to the same class/label, falling in the same feature space. And the process of sending the global model back to all clients to re-train as the starting weights is similar to the boosting algorithm, where the idea is to train weak classifiers to be able to perform better.

In this paper, we take advantage of the collaborative nature of FL to enhance the overall detection capabilities of the model, therefore creating a better generalized and more stable model: federated ensemble learning YOLOv5 (FedEnsemble YOLOv5). We examine the performance of the popular YOLOv5 (You Only Look Once version 5) algorithm [2] for object detection when trained using ensemble FL with multiple clients in comparison to the traditional centralized training approach. To accomplish this, we propose a methodology where multiple clients participate in the federated training process, with each client possessing a subset of the same dataset used in centralized training. The data distribution among clients is achieved through random sampling without replacement. This approach allows us to assess the effectiveness of FL for object detection using the YOLOv5 model in an ensemble setting. By leveraging the collective knowledge of diverse clients, we aim to demonstrate the potential of FL as an ensemble algorithm, exhibiting characteristics similar to Bagging and Boosting techniques.

II. RELATED WORK

Ensemble learning methods use different approaches to combine the outputs of many (often simpler) models into a single collective output that achieves better performance than any individual model alone [3]. Although the foundation for some of these methods can be traced back several decades [4] [5], they are gaining renewed significance due to the importance of machine learning [6]. The combination rule depends on the type of problem at hand, the model being used and may range from simple to complex: Calculating the mean value of every output (bagging) or taking the number

of votes for a prediction class (voting) are the most intuitive methods, often resulting in already good results [7]. There are different types of boosting approaches, all of which aim to reinforce a feature of the outputs between the classifiers, such as giving more weight to previously misclassified samples [8]. Stacking is usually a two-step process, that uses multiple base-level models to generate predictions, which are then used as input for a higher-level model to produce the final prediction [9].

Combining ensemble learning and federated learning can leverage the advantages of both methods, enabling efficient learning from decentralized datasets while benefiting from the predictive power of multiple learners. There exist numerous approaches to combining FL and ensemble learning, each with different objectives: FedBoost introduces boosting to FL for communication cost reduction, while assuming that pre-trained base predictors exist [10]. [11] utilizes knowledge distillation known from continual learning and further developed this into ensemble distillation of the clients, but requires labeled data to be available to the server. [12] focuses on the improvement of performance of a non-IID image classification by following a stacking approach. For object detection in our research, we adopted the YOLOv5 algorithm [2], which is an improved implementation of the YOLOv3 algorithm [13] using the PyTorch framework. The selection of YOLOv5 as our object detection algorithm aligns with the methodology employed in a previous study [14], allowing for direct comparisons between the outcomes presented in both papers. This choice ensures consistency and facilitates a comprehensive analysis of the results obtained in our current research in relation to the earlier work. Several others have also developed more advanced concepts, however, none have explored the fundamental analogies between ensemble and federated learning.

III. METHODOLOGY

In this section, we elaborate on our approach to federated ensemble learning by structuring clients that operate as weak learners. We segment the principal dataset into a number of mutually exclusive subsets, each corresponding to a specific client. This division of data, sets the stage for implementing FedAvg algorithm, which aims to empower these weak learners by enabling them to acquire distinctive features and characteristics by utilizing the global models. Interestingly, each round of communication serves to effectively boost the performance of these learners, echoing the mechanics of Boosting algorithms. This systematic approach culminates in the creation of a robust global model that surpasses its counterparts in terms of performance, displaying superior generalization capabilities in classifying and drawing bounding boxes over the target objects.

Inspired by the ensemble nature of federated learning, we designed an experiment revolving around the goal of achieving an effective object detection model. Our model leverages the YOLOv5 algorithm, utilizing a dataset identical to the one employed in centralized YOLOv5 training. In contrast to conventional FL, the centralized dataset is randomly divided

into partitions equivalent to the number of clients. Each client then engages in training a federated global model, where active participation in every communication round must be ensured. This approach allows us to make the most of the inherent ensemble behavior of federated learning, crafting an innovative solution for object detection that builds upon the strengths of existing methodologies.

The centralized dataset is divided into three subsets for training, validation, and testing purposes. Specifically, 80% of the total dataset is allocated as the training dataset, while 10% each is reserved for validation and testing. When distributing the centralized dataset to multiple clients, the same images that were part of the training subset used for training the centralized YOLOv5 model are provided. To simplify the process, we consider a fixed number of three clients and provide them with identical validation and test datasets that were originally utilized during the centralized training.

This distribution strategy remains consistent for both the dataset discussed in Section III-B. In our preliminary experiments, we conducted a test using the federated ensemble (FedEnsemble) algorithm on a dataset comprising USB-Sticks, which was previously presented in [15]. The results indicated that the FedEnsemble technique yielded superior accuracy compared to a centrally trained USB quality classification model. Encouraged by this outcome, we extended our work to include federated ensemble object detection, aiming to address practical applications beyond image classification.

A. Ensemble Algorithm

In order to leverage the ensemble behavior of federated learning, we conducted experiments with a new algorithm to develop a federated ensemble object detection model based on YOLOv5. This algorithm utilizes the same dataset that was employed for the centralized training of YOLOv5. The dataset is divided into 3 clients and the distribution strategy is the same as mentioned in Section III. The number of clients can be increased based on the size of the centralized training dataset.

The algorithm follows the steps outlined below:

- 1: Shuffle the centralized data with YOLO annotations and divide it into 'n' mutually exclusive datasets, where 'n' represents the number of clients. The validation and test datasets remain unchanged.
- 2: Utilize the same YOLOv5 model architecture (e.g., YOLOv5m or YOLOv5l) and consistent hyperparameters (e.g., optimizer, batch size, learning rate, local epochs) for all clients.
- 3: Train each client individually using their respective local dataset for a specified number of epochs and save the weights of the last epoch.
- 4: Send the weights of all clients to the server and perform federated averaging to create a global model. It is important to note that all clients actively participate in each communication round.
- 5: Evaluate the accuracy of the global model on the test dataset.

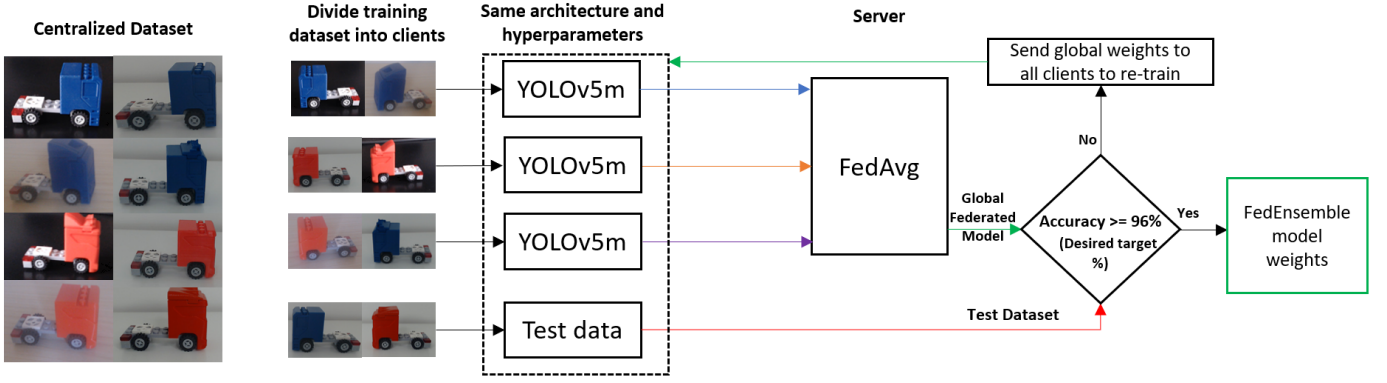


Fig. 1. FedEnsemble algorithm with example of cabin dataset

- 6: Terminate the process when the global model achieves the desired accuracy on the test dataset.

Figure 1 depicts the process flow diagram of the above-mentioned algorithm. The resulting global model is then compared to the normally trained YOLOv5 model, where identical hyperparameters are maintained. The outcomes of this comparison can be found in Section V. To expedite the process, clients can be executed on parallel GPUs while running the server code on the same system. This configuration can significantly enhance the computational efficiency of the federated ensemble algorithm. Since the federated ensemble learning approach does not tackle privacy issues as the subset used for training by every client is a part of a centralized dataset; hence this parallelization strategy can be employed without any concerns in that regard.

B. Datasets

In this paper, we experiment with two different custom datasets, which belong to our manufacturing setting, to analyze the robustness of the federated ensemble algorithm on multiple use-cases and scenarios. The first preliminary tests were done on the dataset of USB-Sticks, previously presented in [15] and [14]. The first dataset employed in our study is the truck cabin dataset, which encompasses two distinct design variations of miniature truck cabins in red and blue colors. Figure 2 illustrates this dataset, which is identical to the one utilized in a previous study [14]. The centralized cabin dataset comprises a total of 1200 images, with 600 images categorized as 'Cabin_without_windshield' and the remaining 600 images labeled as 'Cabin_with_windshield'. The dataset comprises four different windshield designs. The training data encompasses blue cabins with blue windshields of type A and B, along with red cabins featuring red windshields of type C and D. Following the distribution strategy outlined in the methodology section, this dataset is divided among three clients by shuffling the training subset originally used for training the centralized YOLOv5 model. Consequently, the clients do not possess an equal number of instances for each class, as they are randomly assigned.

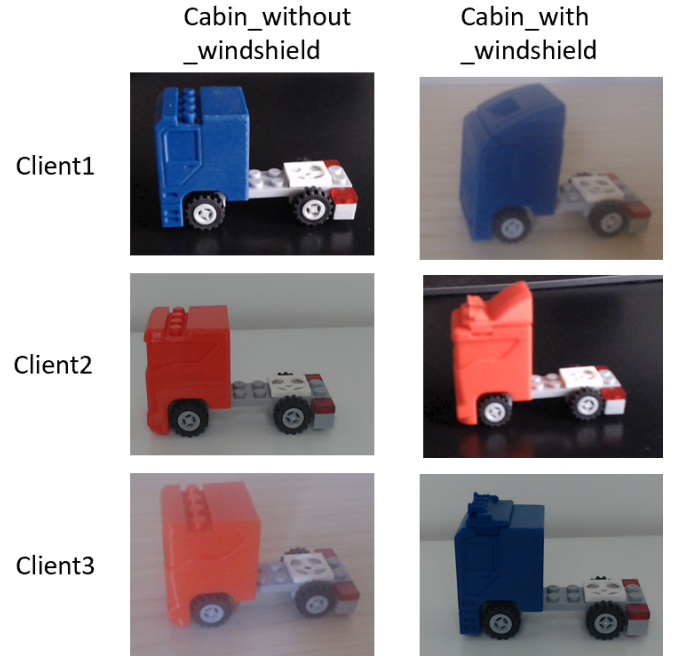


Fig. 2. A small example of distribution of centralized cabin dataset into three clients

The trailer dataset used in this study comprises a total of 900 images. Each class in the dataset consists of 300 images, and there are three different backgrounds included. Additionally, the dataset includes some deliberately introduced blurry images to enhance the robustness of the trained model. Figure 3 showcases a small subset of the dataset, providing a glimpse into its content. To facilitate the application of the federated ensemble algorithm, we partitioned the training dataset of the centralized trailer dataset into three disjoint subsets. This partitioning ensures that the validation and test datasets remain consistent across all experiments, enabling fair and accurate comparisons. It is important to note that during the capture of the trailer dataset, the images were obtained without including the chassis. Therefore, when evaluating the model's performance on trailers with chassis, interesting and

TABLE I
TRAINING DATASET DISTRIBUTION OF TRAILER DATASET (NO. OF IMAGES)

Dataset	Training images	White trailer	Blue trailer	Penholder trailer
Centralized	720	246	233	241
Client1	241	78	79	84
Client2	240	95	60	85
Client3	239	73	94	72



Fig. 3. A small example of distribution of centralized trailer dataset into three clients

informative results are observed. For the purpose of data distribution analysis, the centralized training dataset comprises 80% of the total dataset, resulting in a selection of 720 images. The distribution of these images within each class is randomized. For a comprehensive understanding of the dataset's detailed distribution, refer to Table I.

IV. IMPLEMENTATION AND EXPERIMENTS

In this section, we outline the implementation details of the FedEnsemble algorithm for training the federated YOLOv5 models. As mentioned in Section III, we distribute a centralized dataset for YOLOv5 into multiple clients (in our case, three clients). Each client utilizes the YOLOv5m architecture with the same set of hyperparameters. The training process begins by training each client's model for 10 epochs on its local dataset. After completing the local training, the client's weights are sent to a central server, where the FedAvg algorithm is applied to obtain the global weights. However, before sending the weights back to the clients, the global federated model is evaluated using the test dataset. This evaluation allows us to monitor the progress and determine if the model has achieved the desired metrics. If the global model's performance meets the desired target, the FedEnsemble learning process is stopped. However, if the metrics do not meet the target, the updated weights are sent back to each client, serving as the starting point for the subsequent round of training. It is important to note that in FedEnsemble learning, each client participates in every communication round. To illustrate this process in a more visual manner, Figure 1 presents a flowchart depicting the steps involved in the FedEnsemble algorithm. For

the cabin dataset, the centralized model was trained for 150 epochs, while for the trailer dataset, it was trained for 100 epochs (after several experimentation this epoch number was selected based on the best performing model). The FedEnsemble model for the cabin dataset yielded good results when trained with 15 local epochs and 5 communication rounds. On the other hand, the best results for the trailer dataset were achieved with 10 local epochs and 4 communication rounds. Once again, this local epochs and communication rounds number was a result of various experimentation. Also in Figure 1, the accuracy 96% was set due to the accuracy of YOLOv5 model achieved from normal training. These parameters can vary depending upon the requirement or custom use case.

In the cabin dataset, both the centralized trained YOLOv5 model and the FedEnsemble model exhibit excellent performance on the validation and test datasets. To facilitate comparison, we constructed test combinations consisting of cabins and windshields not present in the training dataset, such as blue cabins with blue windshields of type C and D, and red cabins with red windshields of type A and B (as previously utilized in [14]). Both models demonstrate high accuracy and accurately predict bounding boxes on this test dataset. However, it is important to note that the test dataset was generated using a similar background and environmental setting as the training dataset, aiming to assess the models' robustness in accurately classifying objects and drawing precise bounding boxes. It is worth mentioning two key distinctions between the truck cabin and trailer datasets. Firstly, all cabin images in the truck cabin dataset feature cabins on top of a chassis, whereas the trailer dataset includes chassis images as well. Secondly, both datasets solely contain individual objects, and neither dataset includes fully assembled trucks comprising both cabins and trailers. Based on these considerations, we designed the following experiments for this study:

- 1) Conduct a comparison between the centralized trained YOLOv5 and FedEnsemble Cabin models using a mix of cabin combinations (with and without chassis) for live classification.
- 2) Evaluate the outputs of the centralized trained YOLOv5 and FedEnsemble Cabin models on cabins assembled with various trailer types (fully assembled trucks).
- 3) Analyze the outputs of the centralized trained YOLOv5 and FedEnsemble Cabin models using images from the quality inspection module on the demonstrator.
- 4) Compare the performance of the centralized trained YOLOv5 and FedEnsemble Trailer models for trailers (with and without chassis) in live classification.

- 5) Examine the outputs of the centralized trained YOLOv5 and FedEnsemble Trailer models on cabins assembled with different trailer types (fully assembled trucks).
- 6) Analyze the outputs of the centralized trained YOLOv5 and FedEnsemble Trailer models using images from the quality inspection module on the demonstrator.

The quality inspection module is an integral part of the demonstrator located at SmartFactory-Kaiserslautern (SF-KL), which showcases the future of manufacturing with production level 4 capabilities.

V. RESULTS

In this section, we present the results of the experiments conducted in this paper. To facilitate better visualization, the model outputs are depicted with different color bounding boxes (BB). For the cabin models, the 'red color' BB represents the 'Cabin_without_windshield' class, while the 'pink color' BB represents the 'Cabin_with_windshield' class. Similarly, for the trailer model, the 'red color' BB corresponds to the 'trailer_body_blue' class, the 'pink color' BB corresponds to the 'trailer_body_white' class, and the 'orange color' BB corresponds to the 'trailer_body_white_penholder' class. For experiments number 2 and 5, the test images remain consistent for the cabin and trailer combinations to enable better comparison¹. Figures 4 and 5 present the outcomes of experiment number 1. Both figures consist of two windows, wherein the left window displays the output from the centrally trained YOLOv5 model, while the right window showcases the output from the FedEnsemble model on a live video frame. In Figure 4, the frames feature two cabins with a mixed combination (including a chassis), illustrating the effective classification and accurate bounding box generation by both models. Moving on to Figure 5, the results are noteworthy as the cabins lack a chassis, resulting in object orientations that differ from those in the training dataset. While the centrally trained YOLOv5 model correctly classifies the objects, it produces imprecise bounding boxes that lead to partial cropping of the objects. In contrast, the FedEnsemble model exhibits exceptional performance by accurately classifying the objects with high confidence scores and generating precise bounding boxes around both objects in the frame.

For experiment number 2, both cabins were assembled with different trailer types, and the results of this experiment can be seen in figure 6 and 7. Figure 6 shows output of YOLOv5 cabin model on various cabin and trailer combinations. The model produces good results apart from the images with blue trailers. The model draws a BB over the blue trailer body and classifies it as 'Cabin_with_windshield', leading to false positives. Figure 7 shows the output of FedEnsemble model on the same test images, and the model is able to predict precise BB without producing any false positives on trailer objects.

The results obtained from Experiment 3 are particularly intriguing due to the significant differences in image environment, background, and lighting conditions compared to the

¹The output images have been cropped and enlarged for improved visibility

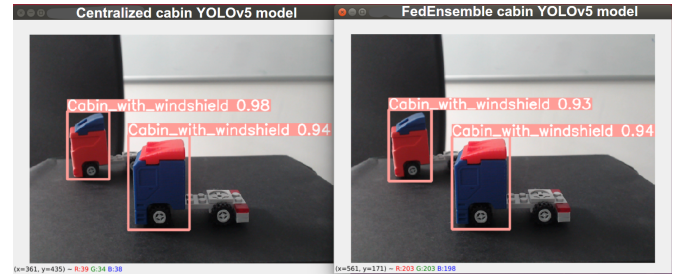


Fig. 4. Comparison of models trained using normal YOLOv5 (left) vs Federated Ensemble YOLOv5 (right) when object has a similar orientation, but a different combination not present in the dataset (Cabin with different color type windshield with chassis)



Fig. 5. Comparison of models trained using normal YOLOv5 (left) vs FedEnsemble YOLOv5 (right) when object has a different combination not present in the dataset (Cabin with different color type windshield without chassis)

images present in the training dataset. Figure 8 displays the output of the YOLOv5 cabin model on test images captured from the demonstrator. While the model performs well on cabin combinations similar to those in the training dataset, it struggles to detect the combination of blue cabin with red windshield and also produces false positives on trailer objects. In contrast, the results of the FedEnsemble cabin model demonstrate significant improvements. The bounding boxes predicted by the model exhibit high accuracy, with no false positives on trailer objects. Notably, one test image showcases a remarkable outcome: despite the flashlight of the quality inspection module being off and the object being barely visible to the human eye at first glance, the FedEnsemble cabin model is able to correctly classify the object and draw a precise bounding box around it.

Experiments 4, 5, and 6 focus on the trailer dataset models, with an emphasis on maintaining consistency in the test dataset compared to the previous results. Figure 10 and 11 illustrate the output of the YOLOv5 and FedEnsemble trailer models, respectively. Both figures exhibit different orientations of trailer bodies (with and without chassis) in the context of live classification. It is observed that both models correctly classify and predict trailer objects without a chassis (image orientation similar to the training dataset). However, as depicted in Figure 11, the YOLOv5 trailer model struggles to classify the blue trailer accurately and crops out a portion of the penholder trailer when predicting the bounding box. In contrast, the FedEnsemble model demonstrates superior performance by



Fig. 6. Output of cabin model trained with Normal YOLOv5 algorithm on cabins assembled with different trailer types



Fig. 7. Output of cabin model trained with Federated Ensemble YOLOv5 algorithm on cabins assembled with different trailer types

drawing precise bounding boxes around both objects in the frame, further highlighting its superiority over the normal centralized trained YOLOv5 model.

The test images used in Experiment 5 correspond to the same set as those employed in Experiment 2. Figure 12 showcases the output of the YOLOv5 trailer model on these images, revealing its poor performance. For instances featuring a blue trailer with a blue cabin combination, the model erroneously predicts the entire truck as 'trailer_body_blue'. Furthermore, the predicted trailer bounding box encompasses the chassis and exhibits a significant number of false positives across various images. In stark contrast, the results obtained from the FedEnsemble trailer model are truly remarkable. The FedEnsemble model avoids generating any false positives and accurately predicts and draws precise bounding boxes

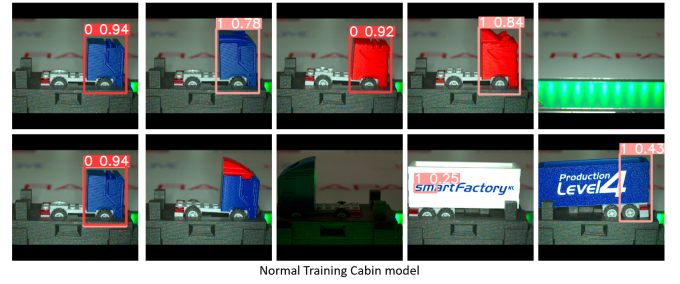


Fig. 8. Output of model trained with Normal YOLOv5 algorithm on images from the Demonstrator (0: Cabin_without_windshield, 1: Cabin_with_windshield)

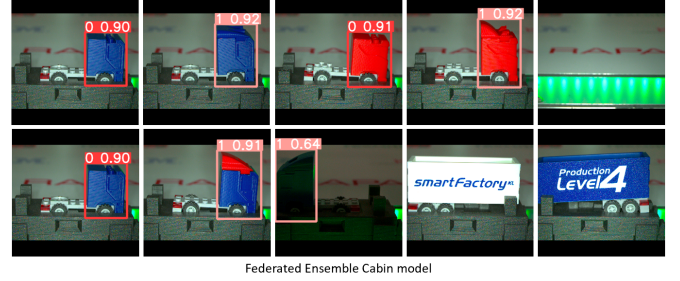


Fig. 9. Output of model trained with Federated Ensemble YOLOv5 algorithm on images from the Demonstrator (0: Cabin_without_windshield, 1: Cabin_with_windshield)

exclusively around the trailer object, effectively excluding the chassis from the prediction, as shown in Figure 13.

In the final experiment, the trailer models are subjected to testing using images from the demonstrator. Similar to Experiment 3, these test images exhibit distinct backgrounds and lighting conditions compared to the trailer's training dataset. Once again, the performance of the YOLOv5 trailer model on these images is unsatisfactory. The model fails to predict the objects accurately, whereas only identifying the white trailer, that too with an imprecise bounding box. Additionally, the model produces a false positive on an image without any objects present. In contrast, the FedEnsemble YOLOv5 trailer model excels in this challenging scenario. It not only correctly classifies the objects but also draws precise bounding boxes around them, even in an unfamiliar environment characterized by different lighting conditions that may affect the color appearance of the objects. The FedEnsemble model's ability to generalize well and maintain accurate predictions in such conditions demonstrates its superior performance compared to the standard YOLOv5 model.

VI. DISCUSSION

In this section, we discuss the implications and significance of the results obtained from the experiments conducted in this study. The experimental results clearly demonstrate the superiority of the federated ensemble approach over the centralized training approach in the context of object detection using the YOLOv5 algorithm. The federated ensemble YOLOv5 model consistently outperformed the centralized YOLOv5 model



Fig. 10. Comparison of models trained using normal YOLOv5 (left) Federated Ensemble YOLOv5 (right) when the object (trailer) orientation is similar to the training dataset



Fig. 11. Comparison of models trained using normal YOLOv5 (left) Federated Ensemble YOLOv5 (right) when the object (trailer attached to a chassis) orientation is not similar to the training images

across various scenarios for both cabin and trailer datasets. One key advantage of the federated ensemble approach is its ability to effectively leverage the collective knowledge of multiple decentralized models. By aggregating the predictions from individual models trained on local data, the federated ensemble model achieves higher accuracy, better bounding box predictions, and reduced false positives compared to the centralized trained model.

The experiments involving different combinations of cabins and windshields, as well as trailers with and without chassis, showcased the robustness of the federated ensemble model. It successfully classified and accurately predicted objects, even when presented with combinations not present in the training dataset. This indicates the model's ability to generalize well and handle variations in object appearance and orientation. Moreover, the tests conducted on images from the demonstrator, which featured different background settings and lighting conditions, further demonstrated the effectiveness of the federated ensemble approach. The model consistently achieved accurate classifications and precise bounding box predictions, even in challenging and unfamiliar environments. This highlights the Federated ensemble algorithm's adaptability and potential for real-world applications.

A. Potential for Generalization and Future Research

The positive outcomes obtained from this research indicate the potential applicability of the federated ensemble approach beyond the YOLOv5 algorithm and object detection tasks. The concept of federated learning, combining the strengths of ensemble methods and preserving data privacy, can be extended to other architectural models and various use cases,



Normal Training Trailer model

Fig. 12. Output of trailer model trained with Normal YOLOv5 algorithm on cabins assembled with different trailer types



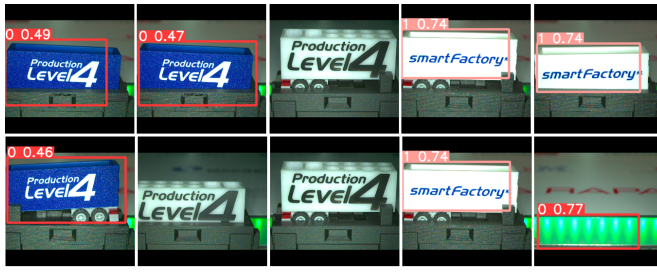
Federated Ensemble Trailer model

Fig. 13. Output of trailer model trained with Federated Ensemble YOLOv5 algorithm on cabins assembled with different trailer types (0: Trailer_blue, 1: Trailer_white, 2: Trailer_white_penholder)

including image classification, image segmentation, and small object detection. Furthermore, the potential of applying federated learning to non-vision tasks, such as spam detection and anomaly detection, warrants exploration. By decentralizing the training process and aggregating models' predictions, federated learning may provide improved performance and data privacy in these domains as well. Validation of these results through future experiments on public datasets would further support the effectiveness of the federated ensemble approach and solidify its potential for practical implementation.

VII. CONCLUSION

Following the realization, that federated learning acts as a combination of Bagging and Boosting algorithms. This was



Normal Training Trailer model

Fig. 14. Output of model trained with Normal YOLOv5 algorithm on images from the Demonstrator (0: Trailer_blue, 1: Trailer_white, 2: Trailer_white_penholder)



Federated Ensemble Trailer model

Fig. 15. Output of model trained with Federated Ensemble YOLOv5 algorithm on images from the Demonstrator (0: Trailer_blue, 1: Trailer_white, 2: Trailer_white_penholder)

empirically tested using a YOLOv5 algorithm, specifically tailored to our research context. The comparative results highlighted a superior performance of the federated ensemble YOLOv5 algorithm, in contrast to the centralized YOLOv5 model, particularly within the context of a custom dataset applied to manufacturing scenarios. This approach is not limited to YOLOv5, but can be applied to other object detection algorithms as well. Future research will explore the potential of applying this federated ensemble approach to diverse architectural models and use-cases, including but not limited to, image classification, image segmentation, and small object detection. Moreover, we also aim to extend its applicability to non-vision use cases, such as spam detection, anomaly detection, and more. There is potential for the validation of these results through future application on public datasets, which would further substantiate our findings. In conclusion, our research demonstrates that, in addition to the well-known advantages of preserving data sovereignty, FL also holds potential advantages in situations where access to a substantial portion of the data set is possible.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 2017, pp. 1273–1282. [Online]. Available: <http://proceedings.mlr.press/v54/mcmahan17a.html>
- [2] G. J. et al., "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.
- [3] Z.-H. Zhou, *Ensemble methods: Foundations and algorithms*, ser. Machine Learning & Pattern Recognition Series. Boca Raton, Fla.: CRC Press Taylor & Francis, 2012.
- [4] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik, "Boosting and other ensemble methods," *Neural computation*, vol. 6, no. 6, pp. 1289–1301, 1994.
- [5] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [6] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, no. 2, pp. 241–258, 2020.
- [7] S. Wan and H. Yang, "Comparison among methods of ensemble learning," in *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on*. IEEE, 2013, pp. 286–290.
- [8] V. Svetnik, T. Wang, C. Tong, A. Liaw, R. P. Sheridan, and Q. Song, "Boosting: an ensemble learning tool for compound classification and qsar modeling," *Journal of chemical information and modeling*, vol. 45, no. 3, pp. 786–799, 2005.
- [9] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [10] J. Hamer, M. Mohri, and A. T. Suresh, "Fedboost: A communication-efficient algorithm for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 2020, pp. 3973–3983.
- [11] S. Lin, G. Yang, and J. Zhang, "A collaborative learning framework via federated meta-learning," in *2020 IEEE 40th International Conference on Distributed Computing Systems*. Piscataway, NJ: IEEE, 2020, pp. 289–299.
- [12] H. Yu, C. Wu, H. Yu, X. Wei, S. Liu, and Y. Zhang, "A federated learning algorithm using parallel-ensemble method on non-iid datasets," *Complex & Intelligent Systems*, 2023.
- [13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [14] V. Hegiste, T. Legler, K. Fridman, and M. Ruskowski, "Federated object detection for quality inspection in shared production," 2023, unpublished, pre-print ArXiv.
- [15] V. Hegiste, T. Legler, and M. Ruskowski, "Application of federated machine learning in manufacturing," in *2022 International Conference on Industry 4.0 Technology (I4Tech)*. IEEE, 2022, pp. 1–8.