Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/49/20241067

Empirical validation of federated learning with YOLO v7-tiny for road sign detection: A simulation-based comparative study

Yizhen Bi

The Department of Electrical and Electronic Engineering, The University of Manchester, Manchester, M13 9PL, UK

yizhen.bi@student.manchester.ac.uk

Abstract. Detecting road signs is a critical component in the development of intelligent driving systems. While centralized machine learning approaches have demonstrated potential in this field, the untapped potential of Federated Learning warrants exploration. This research aims to bridge this gap by examining the feasibility of applying Federated Learning within edge Artificial Intelligence (AI) computing environments for the purpose of road sign detection. Utilizing the You Only Look Once (YOLO) v7-tiny model and a range of experimental parameters demonstrates that Federated Learning is viable and outperforms centralized approaches under specific conditions. The study's empirical analysis highlights the sensitivity of detection accuracy to varying experimental parameters. The study contributes to the existing literature by establishing the efficacy of Federated Learning in road sign detection, particularly in edge AI settings constrained by hardware limitations and privacy concerns. However, the study acknowledges limitations, including the lack of deployment on actual edge AI devices and a restricted range of experimental parameters. Future research should aim for more exhaustive experiments with broader datasets, diverse parameters, and real-world edge AI environments. These findings offer valuable insights for future implementations in intelligent automotive systems.

Keywords: Federated Learning, Road Sign Detection, YOLO v7-tiny.

1. Introduction

In an era where mobile devices i.e., edge Artificial Intelligence (AI) have become the dominant computing platform, their accessibility to large amounts of data has grown exponentially [1]. Although this data has the potential to significantly enhance the user experience, primarily through models used to improve speech recognition, text input, and image selection [2, 3], it is often fraught with privacy concerns. Recording such data to a central location for training often conflicts with the idea of protecting sensitive information inherent in the dataset.

Privacy preservation in machine learning has traditionally relied on approaches such as secure Multi-party Computation (MPC), which is effective but imposes a significant communication overhead. Despite their innovative approach of adding noise to the data, alternative approaches like differential privacy present a dichotomy between model accuracy and the danger of data exposure [4]. In this context, the concept of federated learning appears promising, offering an innovative solution to these challenges. It proposes a decentralised approach that focuses on the principle of model aggregation over data

DOI: 10.54254/2755-2721/49/20241067

aggregation [5]. As a federated learning approach, it empowers devices (called clients) to compute updates locally on their training datasets without uploading them to a central server. In this architecture, the central server maintains a global model and receives model updates only from client devices. This decentralised mechanism significantly reduces communication costs, sometimes by a factor of 10-100, compared to traditional simultaneous stochastic gradient descent methods [1].

The optimal scenarios for applying Federated Learning are diverse. The advantages of training with actual data on mobile devices outweigh the proxy data available in data centres, but due to the sensitivity or sheer size of the data, it should not be logged to a data centre solely for model training, and the data should naturally allow for label inference through user interaction [5]. In addition, object detection, a field with applications ranging from face detection to video analytics, could benefit significantly from federated learning. The requirement to centralise large amounts of annotated image data can be alleviated, and instead, models can be trained on localised data. A solid manifestation of this approach is the combination of state-of-the-art object detection algorithms like You Only Look Once (YOLO) or Faster R-CNN with federated learning [6].

The collection of data and its privacy while smart cars are in motion, has become a focal point of modern transport research. In this context, federated learning is a promising technique that allows data participants to collaborate in building machine-learning models while keeping their data secure and private. However, the practical application and training of real-edge AI devices have inherent limitations. In this study, the GTSRB German Road Signs dataset is used as the database for the experiments. First, the YOLO v7-tiny model was trained using a single Graphics Processing Unit (GPU). Subsequently, the Localised Stochastic Gradient Descent (LSGD) training mode was simulated based on the YOLO v7-tiny model. In order to deeply explore the potential and application of federated learning in the field of intelligent transport, this study specifically focuses on the impact on the Mean Accuracy Rate (mAP) of the test set and the data of each detection in different local Epoch and global Epoch scenarios. Through this series of experiments, this article aims to reveal the specific impact of training parameters on model performance in edge learning.

2. Method

2.1. Dataset preparation

The GTSRB dataset is used as the primary data source in this study. The GTSRB is a highly respected dataset in autonomous driving and traffic sign recognition, specially designed for single-image, multi-class classification problems, and containing over 50,000 images distributed across 40 classes, this dataset provides a vast and realistic database with reliable ground-truth data due to its semi-automatic annotation [7]. The sample image of this dataset is presented in Figure 1. The GTSRB images vary in size, from 15×15 to 250×250 pixels and are stored in the Portable Pixmap (PPM) format [7]. The colour scheme of the images is RGB. The details like the image filename, dimensions, and coordinates for the Region of Interest (ROI) bounding boxes [7] were also provided. These annotations were generated using the Advanced Development & Analysis Framework (ADAF) by Nisys GmbH [7].

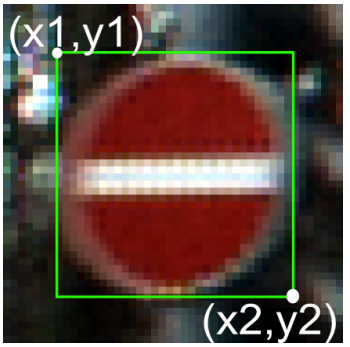

Figure 1. Example of GTSRB image with bounding box.

DOI: 10.54254/2755-2721/49/20241067

Prior to any model training, the dataset underwent a preprocessing phase. Initially, all images were converted from the PPM format to PNG for broader compatibility. Following this, annotations were adapted to fit the "You Only Look Once" (YOLO) object detection framework. The original annotations contain fields such as image filename, image dimensions (W for width and H for height), coordinates for the upper-left and lower-right corners of the bounding box (i.e., x1, y1, x2, y2), and class ID. Specifically, new text files are created with the same name as each corresponding image. These text files contain the class ID and normalized values for the centre coordinates, width, and height of the bounding box (x, y, w, h). Normalization is achieved by dividing the original coordinates by the respective image dimensions. The normalisation process used the following formulae (1):

$$x = \frac{x1+x2}{2W}, y = \frac{y1+y2}{2H}, w = \frac{x2-x1}{W}, h = \frac{y2-y1}{H} \quad (1)$$

Finally, the dataset was partitioned into training, validation, and test sets according to the 8:1:1 ratio. This complete dataset preparation process ensures that the data is appropriately prepared for subsequent modelling phases, aligning perfectly with the requirements of the YOLO object detection framework.

2.2. Object detection model

The YOLO v7 model aims to be versatile, supporting mobile and cloud-based GPU devices. Unlike conventional real-time object detectors optimised solely for architecture, YOLO v7 also focuses on optimising the training process. It introduces novel modules and optimisation techniques, collectively termed "trainable bag-of-freebies," to enhance detection accuracy without inflating inference costs [8].

The architecture of YOLO v7 diverges from mainstream real-time object detectors that commonly employ MobileNet, ShuffleNet, or GhostNet for CPU-based detection and ResNet, DarkNet, or DLA for GPU-based detection [8]. YOLO v7 incorporates strategies like model re-parameterisation and dynamic label assignment to address new challenges in network training and object detection. Specifically, it employs a planned re-parameterised model and a coarse-to-fine lead-guided label assignment method to tackle issues related to gradient propagation and dynamic target assignment for multiple output layers [8].

When considering embedded systems, particularly in the context of intelligent automotive solutions, YOLO v7 - tiny offers distinct advantages. It is designed to be computationally efficient, making it ideal for low-power, single-chip systems. The test results using the YOLO series of models for the coco dataset, shown in Table 1, reflect the substantial improvement in inference time for YOLO v7 - tiny relative to other models, further demonstrating the implementation of its models for edge AI device implementations. This feature is crucial for real-time object detection in autonomous vehicles, where computational resources are limited, yet the demand for high-speed and accurate detection is imperative.

Table 1. Comparison of state-of-the-art real-time object detectors [8].

Model	#Param.	FLOPs	Size	FPS	APtest / APval
YOLOX-S	9.0M	26.8G	640	102	40.5% / 40.5%
YOLOX-M	25.3M	73.8G	640	81	47.2% / 46.9%
YOLOX-L	54.2M	155.6G	640	69	50.1% / 49.7%
YOLOv7-tiny	6.2M	13.8G	640	286	38.7% / 38.7%
YOLOv7	36.9M	104.7G	640	161	51.4% / 51.2%
YOLOv7-X	71.3M	189.9G	640	114	53.1% / 52.9%

In using the YOLO family of models, distributed model training based on local stochastic gradient descent (LSGD) is a unique approach for YOLOv7 - tiny, based traffic sign recognition. The training consists of a two-level recurrent system involving global and local iterations. The entire training dataset is divided into several non-overlapping subsets. Each subset is trained individually through local iterations, allowing for local model fine-tuning.

Initially, the model is loaded with pre-trained YOLOv7 - tiny weights. Each subset is trained individually in a specified number of local iterations at each global iteration to generate a localised set

DOI: 10.54254/2755-2721/49/20241067

of model weights. The batch size of the architecture is pre-set to be much lower than the batch size of the complete training data. It is trained using a uniform image resolution size, which is the theoretical median of the image resolution of the training set. Each localised training run updates the model weights using the training data of the respective subset. After training on all subsets, an averaging mechanism is invoked to aggregate these localisation weights into a uniform global model weight. This is achieved through a custom function that calculates the arithmetic mean of all subset weights.

Once the global weights are updated, they will be used as a starting point for the next global iteration, ensuring that the model is continuously improved and evolved. This process is repeated for several global iterations, providing a robust mechanism for model optimisation. The LSGD strategy has the dual advantage of parallelism and improved model generalisation. The overall approach makes the training process more scalable and efficient, exploiting the distributed nature of LSGD to optimise computational resources while improving model performance. The method also can be extended to training strategies for federated learning, providing a decentralised solution. Each data subset can reside on its local edge AI device, and only the model weights are transmitted and averaged to generate the global model. That is, smart cars are allowed to collect datasets on their own and only need to synchronise the model weights to the edge AI platform after the big model is updated, and the edge AI devices can be allowed to collect reliable training datasets on their own after setting a higher accuracy threshold. The trained weights can then be uploaded to the data centre for centralised training through the above strategy. The data centre only needs to master the primary road sign data to allow the cars within the federated learning network to enjoy the road sign recognition model with gradually increasing pervasiveness and without directly collecting the video information acquired by the client.

2.3. Implementation details

The training of the YOLO v7 is initiated on a Tesla T4 GPU, selected for its comparative computational limitations, thereby emulating edge devices. A consistent learning rate of 0.001 is employed, with image resolutions and batch sizes set at 64x64 and 16, respectively. The model trained under these centralised conditions demonstrated optimal performance in terms of Mean Average Precision (mAP) after 50 epochs. The model summary reveals 200 layers, 6,119,920 parameters, and 13.4 GFLOPS.

Subsequent experiments adopt federated learning methodologies with varying configurations to emulate real-world edge device scenarios. The model summary reveals 255 layers, 6,127,312 parameters, 6127312 gradients, and 13.5 GFLOPS. For instance, one configuration employs 16 clients engaging in 17 global iterations and three local epochs, simulating the utilisation of a fractioned dataset across multiple edge AI devices. Model parameters are averaged and updated in this configuration every three local epochs, totalling 17 global communications. In another experiment, the number of simulated edge AI devices (clients) is reduced to four while retaining the global iterations and local epochs due to dataset limitations. Further experiments manipulate the local epochs and global iterations to investigate their impacts on model performance.

This multi-faceted investigation affirms the potential of YOLO v7-Tiny as a robust solution for road sign detection on edge AI devices. The experiments' hyperparameters are shown in Table 2. Furthermore, the research provides invaluable insights into federated learning configurations, elucidating the trade-offs between scalability and communication overhead in edge deployments.

DOI: 10.54254/2755-2721/49/20241067

Table 2. Comparison of experiments.

Experiment	Training Strategy	Clients	Global Iterations	Local Epochs	Observations
1	Centralized	N/A	N/A	50	Established performance baseline
2	Federated	16	17	3	Simulated 16 edge AI devices
3	Federated	4	17	3	Reduced clients count due to limited dataset
4	Federated	4	17	10	Investigated impact of increased local epochs
5	Federated	4	40	3	Explored effect of increased global iterations

3. Results and discussion

3.1. Experiment 1: centralized training strategy

The first experiment using a centralized training strategy took 4.185 hours on a Tesla T4 GPU and achieved a Mean Average Precision (mAP) of 0.642 at an IoU 0.5. This establishes a performance baseline for road sign recognition under centralized training. Limitations were highlighted by the confusion matrix, shown in Figure 2. The results confirm the model's aptitude for road sign recognition under centralized conditions. Utilizing a Tesla T4 GPU as a simulation for edge AI conditions validates the methodology and establishes its relevance in real-world scenarios. However, the confusion matrix points to lower detection capabilities for specific road signs due to dataset size, image resolution, and batch size constraints.

3.2. Experiment 2: LSGD and federated learning

The second experiment implemented LSGD and Federated Learning with approximate equivalence to centralized training (51 epochs). Each edge AI device took approximately 0.039 hours for local training. The performance deviated markedly from centralized results, registering a mAP of less than 0.1, as shown in Figure 3. The inadequate performance raises questions about the trade-off between data diversity and computational resources in a federated setting. Due to dataset partitioning, the diluted information per client is a caveat for implementations where a more extensive client base is invoked without due consideration to data distribution [9].

3.3. Experiment 3: reduced client count

The third experiment involved reducing the client count, thereby increasing the diversity of the local dataset and extending local training time to approximately 0.1 hours. An improvement was noted compared to the second experiment, as shown in Figure 4. Although there was an improvement, the aggregated model still needed more accuracy for real-world deployment. This amplifies the need for intelligent client selection and scheduling in future federated systems to avert suboptimal model performance.

3.4. Experiment 4: increasing local epochs

The fourth experiment focused on increasing local epochs and showed substantial progress. Local training time was increased to about 0.32 hours per client. The results outperformed the centralized approach, as shown in Figure 5. The findings point toward an emergent theme: localized, intensive training can compensate for less frequent global updates, thereby optimizing communication overheads. This opens avenues for further research, potentially focusing on optimizing local epochs and communication rounds to achieve a more balanced trade-off between accuracy and computational expenditure [9].

DOI: 10.54254/2755-2721/49/20241067

3.5. Experiment 5: increasing global iterations

The fifth experiment focused on increasing the number of global iterations but only yielded modest improvements. Despite longer local training times (approximately 0.1 hours), the results were still suboptimal compared to centralized training and the optimized federated strategy from the fourth experiment, as shown in Figure 6. An indiscriminate increase in global iterations does not necessarily yield proportional gains in performance. Future work may investigate optimizing these parameters to achieve a fine-grained, computationally efficient, and accurate balance [10].

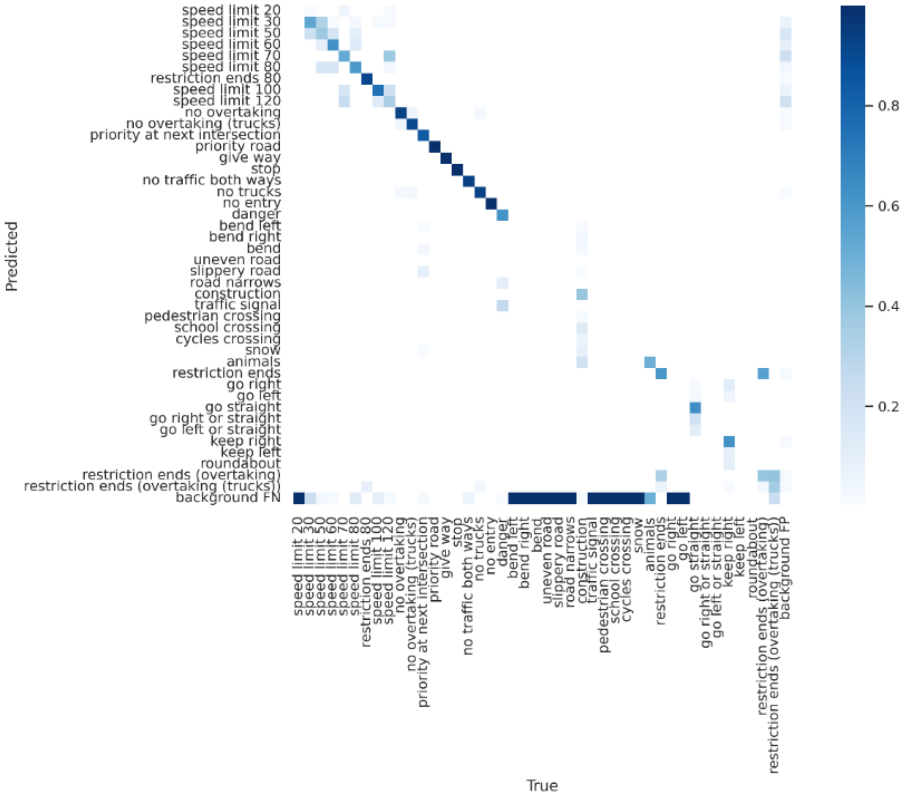

Figure 2. Confusion matrix on test dataset of experiment 1.

DOI: 10.54254/2755-2721/49/20241067

Figure 3. Confusion matrix on test dataset of experiment 2.

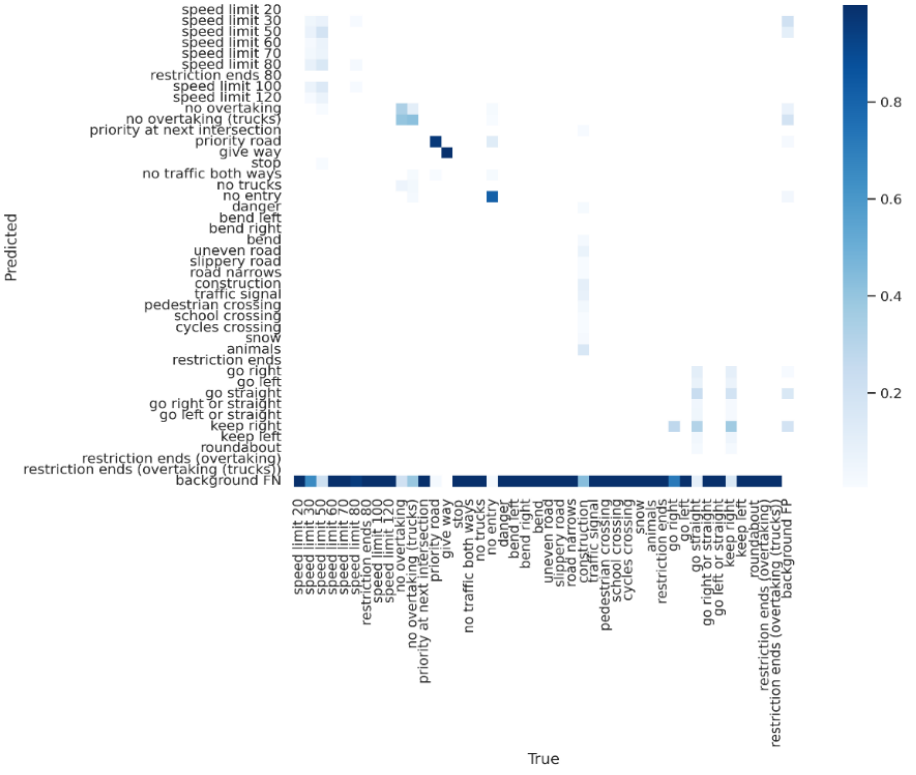

Figure 4. Confusion matrix on test dataset of experiment 3.

DOI: 10.54254/2755-2721/49/20241067

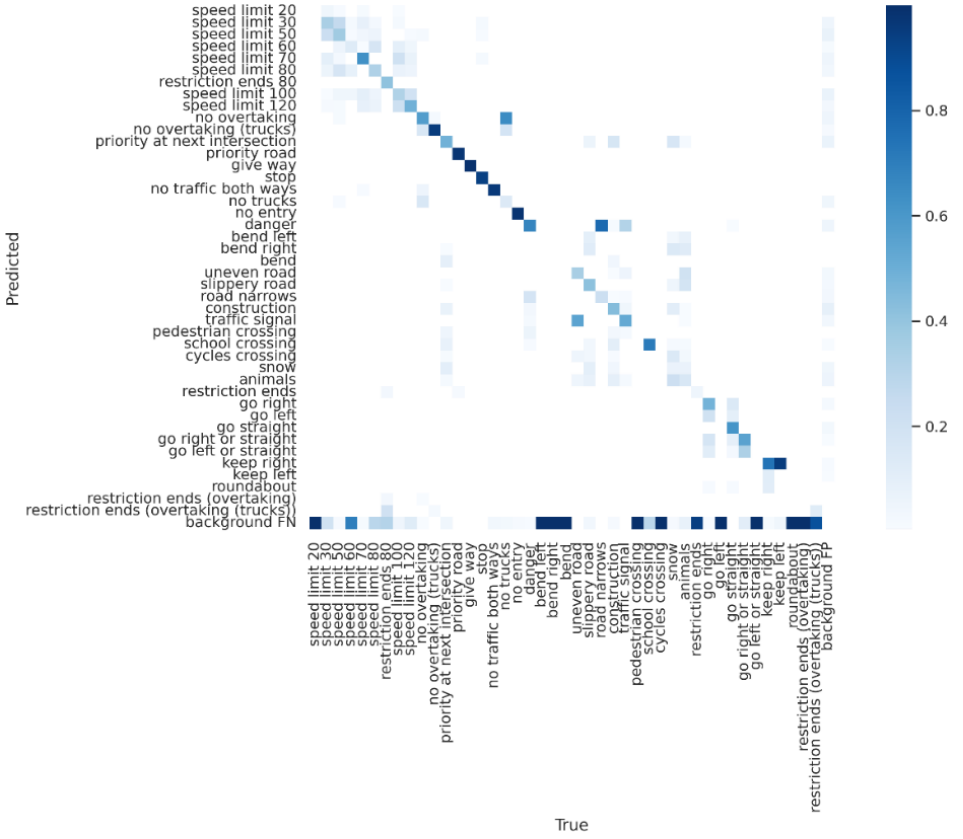

Figure 5. Confusion matrix on test dataset of experiment 4.

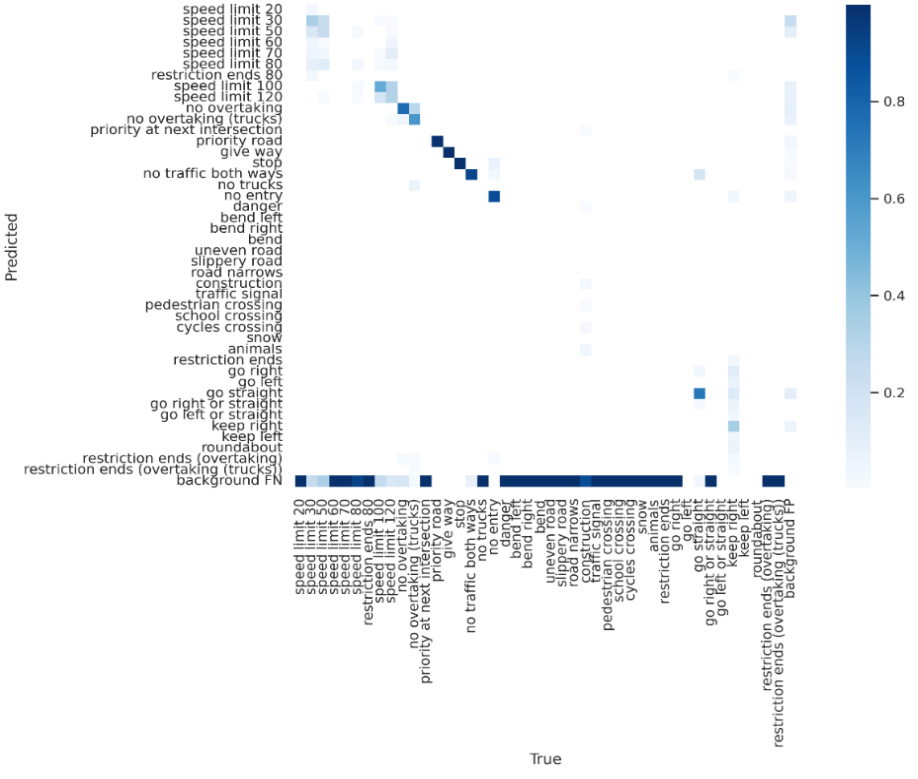

Figure 6. Confusion matrix on test dataset of experiment 5.

DOI: 10.54254/2755-2721/49/20241067

In summary, while centralized training provides a robust baseline, Federated Learning with optimized local epochs and a judicious number of global iterations offers a viable alternative for road sign detection on edge AI devices. Increased communication between the edge devices and the central model did not linearly translate to performance gains, thus warranting a balanced approach to parameter tuning in federated setups.

4. Conclusion

Considering the comprehensive experiments and discussions presented in this study affirm the feasibility of employing Federated Learning strategies in conjunction with the YOLO model for road sign detection. The research underscores the significance of leveraging Federated Learning as a viable alternative, particularly in edge AI environments constrained by hardware limitations and privacy concerns. The findings substantiate that Federated Learning enables the effective expansion and distributed improvement of a unified road sign model on local edge AI devices. This approach not only addresses the limitations of centralized models but also mitigates challenges such as the high cost of manual data annotation and privacy concerns that often plague traditional methods. Consequently, the proposed framework holds considerable promise for practical deployment in real-world road sign recognition projects.

The primary contribution of this paper lies in its empirical validation of the proposed Federated Learning and YOLO-based approach for road sign detection. By establishing the feasibility and efficacy of this framework, the study paves the way for its implementation in solving future road sign recognition challenges. While the current study provides a robust foundation, it has limitations. Specifically, the model has yet to be deployed on edge AI devices, and the experimental parameters employed needed to be sufficiently diverse. Future research should aim to conduct more exhaustive experiments using a broader road sign dataset, a more comprehensive range of experimental parameters, and real-world edge AI devices. Ultimately, the goal is to integrate this Federated Learning-based road sign detection framework into actual intelligent automotive systems. By addressing these gaps, future work can further refine the model's performance and robustness, thereby contributing to developing more efficient, cost-effective, and privacy-preserving solutions for road sign recognition in intelligent transportation systems.

References

[1] McMahan H Moore E Ramage D Hampson S Aguera y Arcas B 2017 Communication-Efficient Learning of Deep Networks from Decentralized Data null pp 1273–1282

[2] Pitaloka D A Wulandari A Basaruddin T and Liliana D Y 2017 Enhancing CNN with preprocessing stage in automatic emotion recognition. Procedia computer science 116 523-529

[3] Qiu Y Wang J Jin Z Chen H Zhang M and Guo L 2022 Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training Biomedical Signal Processing and Control 72 10332

[4] Dwork C 2008 Differential Privacy: A Survey of Results Theory and Applications of Models of Computation vol 4978 ed M Agrawal D Du Z Duan and A Li (Berlin: Springer)

[5] McMahan H B et al 2016 Federated Learning of Deep Networks using Model Averaging ArXiv abs/1602.05629

[6] Luo J Wu X Luo Y Huang A Huang Y Liu Y Yang Q 2019 Real-World Image Datasets for Federated Learning arXiv preprint arXiv:1910.11089.

[7] Stallkamp J Schlipsing M Salmen J Igel C 2012 Man vs computer: Benchmarking machine learning algorithms for traffic sign recognition Neural Networks vol 32 pp 323–332

[8] Wang C-Y Bochkovskiy A Liao H-y 2022 YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 7464-7475

DOI: 10.54254/2755-2721/49/20241067

[9] De S Huang Y Mohamed S Goswami D Corporaal H 2021 Hardware- and Situation-Aware Sensing for Robust Closed-Loop Control Systems 2021 Design Automation & Test in Europe Conference & Exhibition (Grenoble) pp 1751–1756
[10] Magalhães W Gomes H Marinho L Aguiar G Silveira P 2019 Investigating Mobile Edge-Cloud Trade-Offs of Object Detection with YOLO Anais do VII Symposium on Knowledge Discovery Mining and Learning pp 49–56 (Porto Alegre: SBC)