Improving Autonomous Vehicles Safety in Snow Weather Using Federated YOLO CNN Learning

Preprint *in* Lecture Notes in Computer Science · August 2021

DOI: 10.1007/978-3-030-83164-6_10

4 authors:

Gaith Rjoub
Concordia University

Omar Abdel Wahab
Polytechnique Montréal

Jamal Bentahar
Concordia University

Ahmed Batineh
Concordia University

Improving Autonomous Vehicles Safety in Snow Weather Using Federated YOLO CNN Learning

Gaith Rjoub[1], Omar Abdel Wahab[2], Jamal Bentahar[1](✉), and Ahmed Saleh Bataineh[1]

[1] Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada
{g_rjoub,ah_batai}@encs.concordia.ca, bentahar@ciise.concordia.ca

[2] Department of Computer Science and Engineering, Université du Québec en Outaouais, Quebec, Canada
omar.abdulwahab@uqo.ca

Abstract. Accurate object detection (e.g., buildings, vehicles, road signs and pedestrians) is essential to the success of the idea of autonomous and self-driving cars. Various object detection techniques have been proposed to enable Autonomous Vehicles (AVs) to achieve reliable safe driving. Most of these techniques are adequate for normal weather conditions, such as sunny or overcast days, but their effectiveness drops when they are exposed to inclement weather conditions, such as days with heavy snowfall or foggy days. In this paper, we propose an object detection system over AVs that capitalizes on the You Only Look Once (YOLO) emerging convolutional neural network (CNN) approach, together with a Federated Learning (FL) framework with the aim of improving the detection accuracy in adverse weather circumstances in real-time. We validate our system on the Canadian Adverse Driving Conditions (CADC) dataset. Experiments show that our solution achieves better performance than traditional solutions (i.e. Gossip decentralized model, and Centralized model).

Keywords: Autonomous Vehicles · Federated Learning · YOLO CNN · Edge Computing · Object Detection

1 Introduction

Increased vehicular activity has triggered considerable traffic congestion, collisions and pollution emissions. The World Health Organization (WHO) has estimated that over a million people were killed on the World's roads in 2016 according to their status report in Global Road Safety Insight of the yearbook

Supported by NSERC and Innovation for Defence Excellence and Security (IDEaS), The Department of National Defence, Canada.

J. Bentahar et al. (Eds.): MobiWIS 2021, LNCS 12814, pp. 121–134, 2021.
https://doi.org/10.1007/978-3-030-83164-6_10

2018 [10]. Thus, in average there are 3700 deaths per day on the World's roads. Over the past decade, significant investments have been put into automated vehicles performance and safety enhancements. Specifically, Information Technology (IT) has been seen as a means to revolutionize vehicle networks. Connecting vehicles via vehicular networks [2,9,28] can be used for safety purposes, such as collision alert and crash detection, as well as for supporting other applications, such as vehicle-to-vehicle and vehicle-to-cloud communications.

On the other hand, deploying services at the edge of the network has attracted increasing interest from both academia and industry to address the constraints in terms of onboard computing, connectivity, storage, and energy, while minimizing unnecessary latency in cloud computing scenarios. The next-generation wireless networks are supposed to provide ultra-reliable and ultra-low latency whenever/on-the-move. This will fully meet the current connectivity limitations for the inevitable real-time development demands for automated vehicles. In this respect, integrating machine learning capabilities into vehicles allows for more informed and real-time driving decisions such as pedestrians and cyclists detection.

While most of the current state-of-the-art object detection models are trained and benchmarked on datasets with ideal weather conditions, the aim of this paper is to improve the decision-making process of Autonomous Vehicles (AVs) in snowy conditions. One of the main challenges in object detection in AVs settings is the fact that the amount of labeled data is limited when each vehicle is learning by itself without sharing data with other vehicles. Due to limited resources on each vehicle, any image to be included in the training process requires considerable effort to collect and record. To address this challenge, we employ in this work *Federated Learning* (*FL*) [22] with the You Only Look Once (YOLO) method, a single Convolutional Neural Network (CNN) that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. The YOLO method has the advantage of being faster than the traditional CNN approaches including faster R-CNN. However, it is less accurate, so by adding the FL to the model, which adds more resource data from the vehicles and the main edge server, we increase the YOLO prediction accuracy. The main advantage of FL is that the vehicles do not share data, but rather train a single shared machine learning model locally on their own data. YOLO has recently been used to handle the object recognition models' difficulties in real-time. It can run up to 155 frames per second and processes its 24 convolutional layers at a very fast speed.

Another challenge of object detection in snowy weather environments is the high communication cost and delay needed to transfer the data (e.g., road conditions, traffic status, etc.) between vehicles, given the high number of data instances that can be collected and the long-distance communications. This challenge is also addressed by the use of the FL paradigm which allows the devices to perform mutually-distributed training of one large machine learning model without having to share the data among the endpoints. FL includes two major phases, namely, small-scaled local training and large-scale global training. At the beginning of the local training, a parameter processor (e.g., an edge server) initiates the machine learning model and provides the initial parameters to the end devices. Then, the edge server combines all the obtained updates in the local computation process to build a global machine learning model. This method is repeated until a certain degree of accuracy is achieved.

1.1 Contributions

The main contributions of the paper can be summarized as follows:

- We propose a FL-based object detection model that capitalizes on the edge computing technology to enable AVs to meet the rapid-growing demands of self-driving and object detection.
- We employ the YOLO CNN that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. This allows the vehicles to make accurate predictions in real-time.
- We study the performance of the proposed solution experimentally on the Canadian Adverse Driving Conditions Dataset (CADC)[1]. The experimental results suggest that our solution achieves a better performance compared to the traditional predicting approaches that could be used to execute FL for objects detection.

1.2 Organization

The rest of the paper is organized as follows. In Sect. 2, we conduct a literature review on the existing FL approaches in cloud and edge computing settings. We also survey the main deep and convolutional neural network learning-based image recognition approaches. In Sect. 3, we describe the details of the proposed solution. In Sect. 4, we explain the experimental environment, evaluate the performance of our solution, and present empirical analysis of our results compared to other benchmarks. Section 5 concludes the paper.

2 Related Work

In this section, we first survey the main approaches that employ FL over edge computing. Then, we study the approaches that study FL over autonomous vehicles. Finally, we give an overview of the deep learning approaches that are used for object detection over autonomous vehicles.

2.1 Federated Learning over Edge Computing

In [26], the authors propose an edge FL (EdgeFed) in which the outputs of the mobile nodes are combined at the edge server to increase the learning performance and decrease the global level of contact and communication frequency.

[1] https://cadcd.uwaterloo.ca.

In [29], the authors suggest a FL architecture, called *FedMEC*, that simultaneously combines partition techniques and differential privacy. By breaking a DNN model into two sections, *FedMEC* suggests to offload the complicated computations to the edge servers. In addition, the authors use a differential private data perturbation approach to avoid the leakage of privacy from the local model parameters where Laplace noise interferes with the updates from the local devices to the edge server.

In [23], the authors propose a Mobile Edge Computing (MEC) architecture that embeds a multi-layer, FL protocol named *HybridFL*. *HybridFL* seeks to enhance the performance and alleviate the unreliability of end-users by modulating client choice regionally, leading to an appropriate amount of local cloud updates. In [19], the authors use a Double Deep Q Learning (DDQN) to develop a trust-based and energy-aware FL scheduling method for IoT environments. The trust mechanism aims to detect the IoT devices that over-use or under-use their resources during the local training. Then, the authors develop a DDQN scheduling algorithm to take suitable scheduling decisions that take into consideration the trust values and energy levels of the IoT devices. In [8], the authors suggest a modern federated reinforcement learning architecture wherein each agent independently uses a different device to exchange knowledge, and merges individual models as well as learning parameters with the other agents to form a more mature model. The authors employ the Actor-Critic Proximal Policy Optimization algorithm for exchanging the gradients.

2.2 Federated Learning over Autonomous Vehicles

A selective model aggregation method is proposed in [25], where "fine" local DNN models are chosen and submitted by local image quality assessment and computer capabilities to their central server. The authors use two-Dimensional contract theory as a distributed paradigm for overcoming knowledge asymmetry to promote the connections between central servers and vehicle clients. In [15], the authors examine a new form of vehicular network model, i.e., a Federated Vehicle Network (FVN). This model can be regarded as a robust distributed vehicular network for supporting high-database applications such as computer distribution and FL. In order to enable the transfers and prevent the malicious behaviour, authors capitalize on auxiliary Blockchain-dependent systems.

In [7], the authors suggest an aggregation model for a FL navigation framework called *FedLoc* in the vehicular fog. They argued that their scheme effectively defends model changes trained locally and facilitates participants' fluency in a flexible way. With regards to the idea of connected automobiles that use WiFi Access Points (AP), the authors of [14] use a FL method to investigate the viability of the automobiles. Via an in-depth mathematical study, the authors were able to see how a network control FL algorithm can be better implemented with respect to the existing WiFi specifications and TCP protocol.

In order to increase cache efficiency and preserve vehicle privacy, the authors of [27] propose a modern mobility-aware proactive edge cache model for FL, known as *MPCF*. *MPCF* employs a context-conscious Auto Encoder model to approximate the popularity of content and then positions common content expected at the edge of vehicle networks to minimize latency. In addition, *MPCF* incorporates a cache replacement mobility-aware policy that allows network edges, in reaction to mobility patterns and vehicles' preferences, to add and evict contents.

2.3 Deep Learning for Object Detection over Autonomous Vehicles

In [16], the authors provide a new obstacle detection method that uses a new deep learning approach. A newly developed completely convolutional network model is presented to achieve a semantic pixel-wise marking i the following scenarios: free-space, on-road unexpected barriers, and background. The geometric cues are used with an advanced detection method that uses predictive model-based experiments to determine whether or not there are any hazards in the stereo input pictures. A real-time and lightweight traffic light detector for autonomous vehicle platforms is proposed in [11]. This system identifies all potential traffic lights in a heuristic manner and builds the training model on the GPU server and feeds it with numerous public datasets. It then classifies the outcomes using a lightweight CNN model.

In [6], the authors propose a solution that uses a multi-directional closed-loop steering controller with CNN-based feedback to increase vehicles' handling capabilities and performance in comparison to previous techniques that used pure CNN. This study shows that DAVE-2SKY, a neural network that learns how to steer vehicle lateral control using supervised pre-training and reinforcement learning using images from a camera placed on the vehicle, can perform inference steering wheel angles for self-driving vehicle. In order to achieve a better trajectory prediction, the authors introduce a sequential model in [24], that employs a neural network consisting of a CNN and a long short-term memory network. To obtain valid trajectory data, they use a box-plot to identify and exclude data from the vehicle's trajectories that seem anomalous. Moreover, the trajectories of nearby cars are predicted using CNN space expansion and LSTM time expansion.

3 Our Method

We propose to leverage FL for object detection in AVs environments. FL is an efficient technique to enable distributed training of the YOLO CNN model by involving AVs with edge cloud servers in a practical communication network. This form of collaborative learning helps the vehicles achieve fast and high accurate decisions in real-time.

3.1 FL Mechanism

Let V = $\{v_1, v_2, \ldots, v_x\}$ be a set of x AVs responsible for object detection. Let $D = \{D_1, D_2, \ldots, D_x\}$ be the set of datasets stored in the vehicles where each

D_i is the dataset stored in the vehicle v_i. As shown in Fig. 3, edge cloud adopts a FL structure comprised of the following core components:

- **Edge Server**: The edge server first trains a global YOLO CNN model on a publicly available dataset and then sends the initial parameters to the set of AVs. AVs use these parameters to perform local training on their own data. The edge server then collects the wight of the local models w_{r+1}^v for each vehicle v at the next FL communication round $r+1$ from the different AVs and aggregates them using the *FedSGD* method [1] as follows:

$$w_{r+1} = \sum_{v=1}^{V} \frac{d_v}{d} w_{r+1}^v \tag{1}$$

  where d_v is the volume of local data available at the v-th vehicle, i.e., $d_v = |D_v|.d$, and d is the size of the whole data across the selected vehicles. The edge server also notifies the AVs about the recent YOLO model updates, resulting from global model aggregation. The objective of the learning model is to minimize the global loss function, i.e.:

$$\min_{w \in \mathbb{R}} L(w) = \frac{1}{|V|} \sum_{v=1}^{V} \frac{d_v}{d} L_v(w) \tag{2}$$

  where $L_v(\cdot)$ is the loss function of each AV v on its own data. L_v can be further written as:

$$L_v(w) = \frac{1}{d_v} \sum_{i=1}^{d_v} \ell_i(w) \tag{3}$$

  where $\ell_i(w)$ is the loss function on each single data sample.
- **Autonomous Vehicles**: As illustrated in Fig. 2, AVs include embedded sensors, such as cameras, tachographs, GPS, lateral acceleration sensors, and are often prepared to host computational and connectivity tools such as CPU, memory, and data communication [27]. Data communication is the process of using computing and communication technologies to transfer data from one place to another, or between participating parties. Images captured using built-in sensors are used to train the local models to predict objects. AVs connect to the edge server to obtain the initial and aggregate model parameters, and each one trains its own YOLO CNN model using its own collected image data and hence derives an updated set of the parameters. AVs upload new local versions to the edge server (using Eq. 4) where they are compiled by the edge server into a new one.

$$w_{r+1}^v = w_r^v - \gamma \nabla L_v(w) \tag{4}$$

  where γ is the fixed learning rate, and $\nabla(\cdot)$ denotes the gradient operation.

Fig. 1. Communication process of AVs federated learning in edge cloud

3.2 YOLO CNN

The latest state-of-the-art CNNs object detections have been trained and benchmarked in ideal weather datasets, suchas the KITTI dataset [5]. However, those datasets do not show how effectively these CNNs function in real-world driving scenarios in which the weather conditions are complicated and changing over time. To address this shortcoming, we use in our work the CADC dataset, which contains annotated Light Detection and Ranging (LiDAR) and camera data in adverse weather conditions, including snow [12]. The CADC dataset can expand the CNN models to more adverse and critical weather conditions (Fig. 1).

A single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO is one of the most promising state-of- the-art, real time object detection systems [17]. YOLO utilizes features from the whole picture in order to identify and classify each bounding box as well as the classes to which they belong. Like humans, YOLO is capable of quickly recognizing both what items are inside a picture as well as where within the picture those items are located. Our model will first use an input picture, which is divided into a $N \times N$ grid. Each grid cell predicts B bounding boxes and a confidence score β for each box, where our model used $N = 13$ and $B = 5$. This confidence score estimates the probability that the item belongs to a given class, which is successfully detected. This is formally defined as per Eq. 5:

$$\beta = Pr(Object) \times IOU_{pred}^{truth} \tag{5}$$

where IOU_{pred}^{truth} is intersection over the union between the predicted box and the ground truth.

The following five predictions are included in each bounding box: cx, cy, m, h, and β. The box's centroid in reference to the grid cell's bounds is represented by the (cx, cy) coordinates, while m and h are the relative width and height. Each grid cell predicts ζ conditional class probabilities, which are expressed mathematically as follows:

$$\zeta = Pr(Class_i|Object) \tag{6}$$

Finally, we compute the bounding boxes weight by their actual probabilities of containing that object. We compute these weights by multiplying the conditional class probabilities and the individual box confidence predictions [30] as follows:

$$Pr(Class_i|Object) \times Pr(Object) \times IOU_{pred}^{truth} = Pr(Class_i) \times IOU_{pred}^{truth} \tag{7}$$

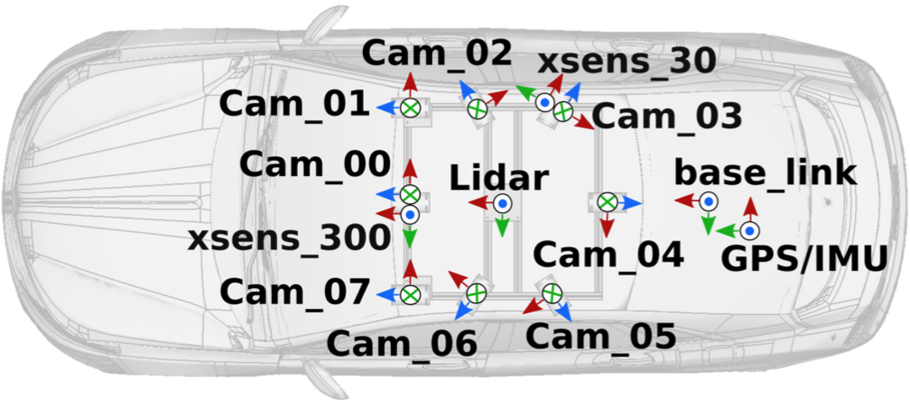

Fig. 2. Overhead view of the vehicle for sensors and cameras (from [13])

4 Implementation and Experiments

4.1 Experimental Setup

We carry out our experiments using the CADC dataset [13], which was produced using the Lincoln MKZ, modified with the Autonomoose AV platform. This autonomous driving dataset, compiled throughout the winter in the region of Waterloo, Canada, is the first dataset to examine and address several types of bad driving circumstances such as low visibility, fog, excessive precipitation snow, etc. It contains 7,000 frames of annotated data from 8 cameras (Ximea MQ013CG-E2), LiDAR (VLP-32C), and a GNSS + INS system (Novatel OEM638), collected through a variety of winter weather conditions. We train a YOLO CNN model on the dataset to determine our algorithm's efficiency and effectiveness. The YOLO model is made up of 24 convolutional layers one after the other and 2 fully connected layers at the end. For pre-training, we use the first 20 convolutional layers followed by an average-pooling layer and a fully connected layer. Keras and TensorFlow (TF) are used to implement the model. Keras is a Python-based neural network API that supports TF. TF, an open-source software framework for dataflow programming created by the Google Brain team, is widely used for a variety of purposes. The FL model is trained on AVs using the Stochastic Gradient Descent (SGD) algorithm, and the training dataset was distributed over 1000 AVs as a Non-lID setting where each AV sampled images randomly but from different subsets of the training data, standing for a more challenging but realistic setting. We varied the number of the selected AVs from 25 to 500 randomly (if its available within the coverage of the edge server) to train the local model. We evaluate the performance of the proposed solution against 1) The decentralized gossip model which does not require an aggregation server or any central component; and 2) The centralized model, the most prevalent strategy, in which a huge quantity of training data is acquired centrally and used to train models across a set of AVs.

4.2 Experimental Results

In Fig. 3, we provide experimental comparisons in terms of training and test accuracy. We run the experiments over 1000 iterations. We observe from this figure that our proposed solution achieves the highest training and test accuracy level compared to the traditional CNN approach and exhibits a better scalability and a faster convergence. This can be justified by the fact that our solution uses YOLO that is designed to be fast and includes a FL component to compensate the lack of data from which some edge servers might suffer and recover this lack by training the model on each AV's local data.

Different from Fig. 3, where we compare the training and test accuracy of our solution against the traditional CNN model, we compare in Fig. 4 the test accuracy of our solution against the centralized and gossip machine learning approaches. We provide experimental comparisons in terms of average test accuracy. The test accuracy quantifies the accuracy obtained by each AV after using the global model trained in a federated fashion to make predictions on its own data. We ran the experiments over 1000 iterations. We observe from this figure that the test accuracy obtained by our model is much higher than those obtained by the gossip and centralized approaches. In particular, the average test accuracy obtained by our model, gossip, and centralized approaches are $90.4\% - 95.2\%$, $82.4\% - 88.1\%$, and $71.4\% - 76.16\%$, respectively. This mean that our model approach enables the AVs to better learn and predict objects in bad weather conditions.

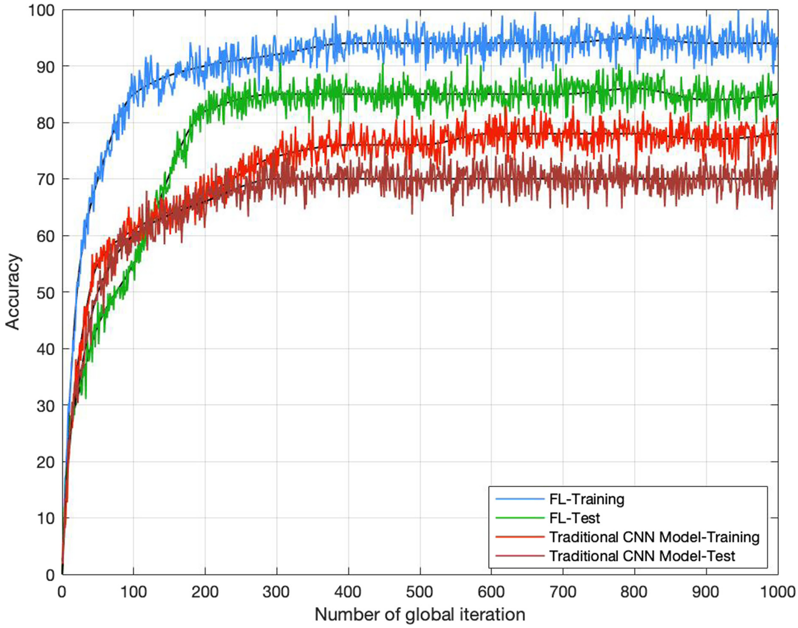

Fig. 3. Comparison of accuracy of final global model

In Fig. 5, we measure the learning time of the different studied approaches, while varying the number of AVs from 25 to 500. The main observation that can be drawn from this figure is that increasing the number of AVs leads to a modest increase in the learning time in the different studied solutions. The figure also reveals that our proposed model achieves the lowest learning time. This is justified by the fact that it distributes the training over the different AVs and aggregates the global model on the edge server, while the gossip model directly exchanges and aggregates models locally at the level of the AVs, which only have a limited resource capabilities. On the other hand, the centralized model needs to gather the training data and train the model on one of the AVs before distributing it across a set of AVs.

In Fig. 6, we measure the accuracy of the different studied approaches, while varying the number of AVs from 10 to 100 and varying also the size of the data available on each AV from 50 to 400. We notice from this figure that increasing the number of AVs leads to a modest increase in the accuracy in the different studied solutions. In fact, the less the data are on AVs, the less the features that can be capitalized on to improve the prediction accuracy would be.

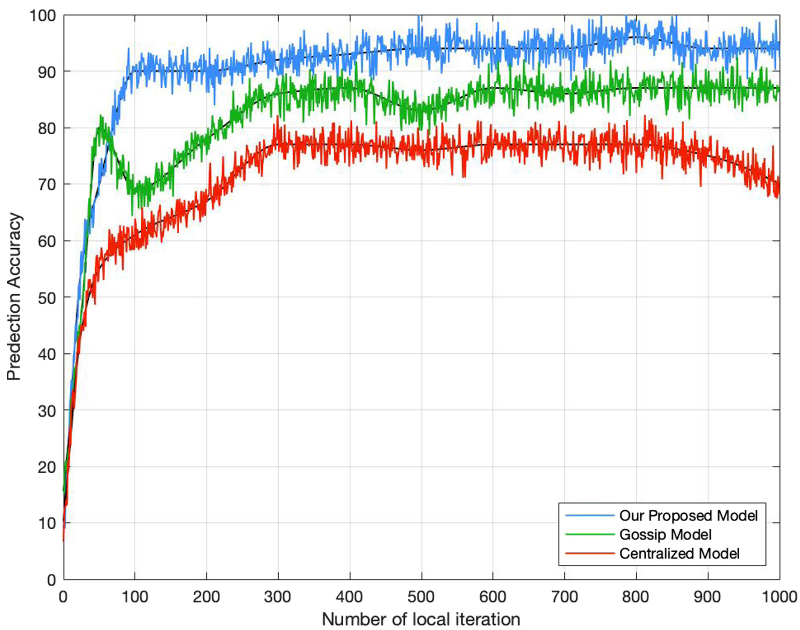

Fig. 4. Comparison of prediction accuracy of local model

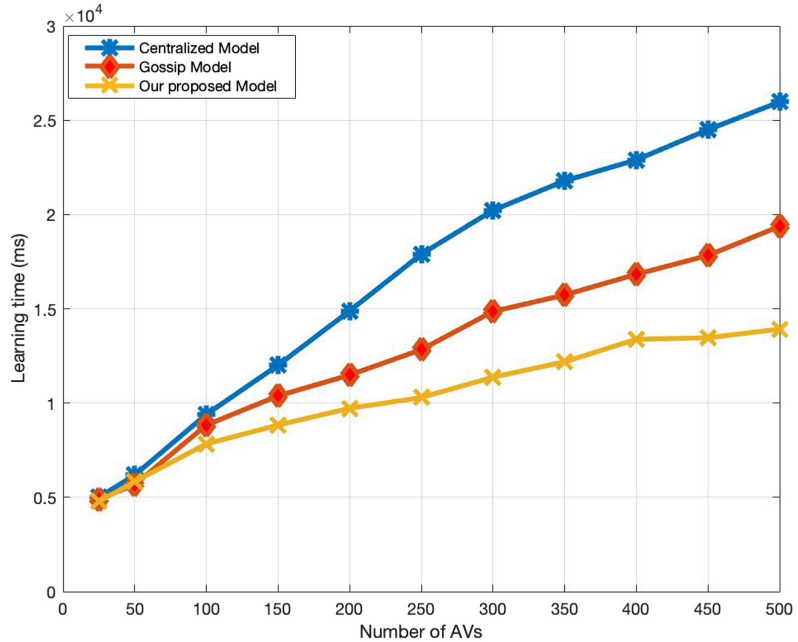

Fig. 5. Learning time versus the number of AVs

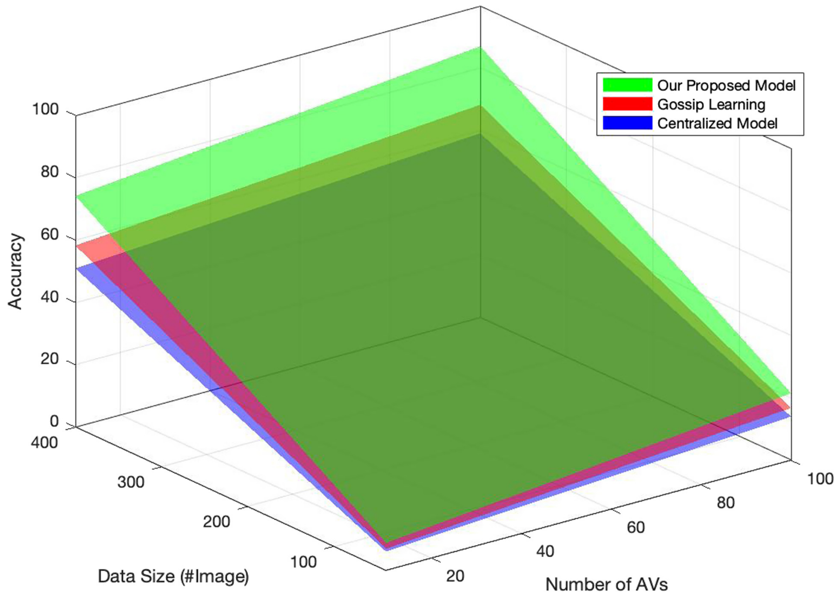

Fig. 6. Accuracy: we study in this figure the impact of varying both the number of AVs and size of data on the accuracy

5 Conclusion

In this work, we proposed a FL-powered YOLO-based approach to improve the real-time object detection predictions over AVs in bad weather conditions. Experiments conducted on the Canadian Adverse Driving Conditions (CADC) dataset reveal that our solution assures the best trade-off between speed and detection accuracy compared to three existing approaches. In the future, we plan to extend this work by investigating a scheduling approach that capitalizes on Deep reinforcement learning [20,21] and trust modeling [3,4,18] to further reduce the training time by avoiding unnecessary computations on untrusted AVs.

References

1. Amiri, M.M., Gündüz, D.: Machine learning at the wireless edge: distributed stochastic gradient descent over-the-air. IEEE Trans. Sig. Process. **68**, 2155–2169 (2020)
2. Cui, J., Wen, J., Han, S., Zhong, H.: Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network. IEEE Internet Things J. **5**(5), 3491–3498 (2018)
3. Drawel, N., Bentahar, J., Laarej, A., Rjoub, G.: Formalizing group and propagated trust in multi-agent systems. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI, pp. 60–66 (2020)

4. Drawel, N., Bentahar, J., Qu, H.: Computationally grounded quantitative trust with time. In: Seghrouchni, A.E.F., Sukthankar, G., An, B., Yorke-Smith, N. (eds.) Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS, Auckland, New Zealand, May 9–13, pp. 1837–1839. International Foundation for Autonomous Agents and Multiagent Systems (2020)
5. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
6. Jhung, J., Bae, I., Moon, J., Kim, T., Kim, J., Kim, S.: End-to-end steering controller with CNN-based closed-loop feedback for autonomous vehicles. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 617–622. IEEE (2018)
7. Kong, Q., Yin, F., Lu, R., Li, B., Wang, X., Cui, S., Zhang, P.: Privacy-preserving aggregation for federated learning-based navigation in vehicular fog. IEEE Transactions on Industrial Informatics (2021)
8. Lim, H.K., Kim, J.B., Kim, C.M., Hwang, G.Y., Choi, H.b., Han, Y.H.: Federated reinforcement learning for controlling multiple rotary inverted pendulums in edge computing environments. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 463–464. IEEE (2020)
9. Obaidat, M., Khodjaeva, M., Holst, J., Ben Zid, M.: Security and privacy challenges in vehicular ad hoc networks. In: Mahmood, Z. (ed.) Connected Vehicles in the Internet of Things, pp. 223–251. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-36167-9_9
10. World Health Organization et al.: Global status report on road safety 2018: Summary. World Health Organization. Technical Report (2018)
11. Ouyang, Z., Niu, J., Liu, Y., Guizani, M.: Deep CNN-based real-time traffic light detector for self-driving vehicles. IEEE Trans. Mob. Comput. **19**(2), 300–313 (2019)
12. Pitropov, M., Garcia, D., Rebello, J., Smart, M., Wang, C., Czarnecki, K., Waslander, S.: Canadian adverse driving conditions dataset (2020)
13. Pitropov, M., et al.: Canadian adverse driving conditions dataset. Int. J. Robot. Res. **40**(4–5), 681–690 (2021)
14. Pokhrel, S.R., Choi, J.: Improving TCP performance over wiFi for internet of vehicles: a federated learning approach. IEEE Trans. Veh. Technol. **69**(6), 6798–6802 (2020)
15. Posner, J., Tseng, L., Aloqaily, M., Jararweh, Y.: Federated learning in vehicular networks: opportunities and solutions. IEEE Network **35**(2), 152–159 (2021)
16. Ramos, S., Gehrig, S., Pinggera, P., Franke, U., Rother, C.: Detecting unexpected obstacles for self-driving cars: fusing deep learning and geometric modeling. In: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1025–1032. IEEE (2017)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
18. Rjoub, G., Bentahar, J., Wahab, O.A.: BigTrustScheduling: Trust-aware big data task scheduling approach in cloud computing environments. Future Gener. Comput. Syst. **110**, 1079–1097 (2020)
19. Rjoub, G., Abdel Wahab, O., Bentahar, J., Bataineh, A.: A trust and energy-aware double deep reinforcement learning scheduling strategy for federated learning on IoT devices. In: Kafeza, E., Benatallah, B., Martinelli, F., Hacid, H., Bouguettaya, A., Motahari, H. (eds.) ICSOC 2020. LNCS, vol. 12571, pp. 319–333. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-65310-1_23

20. Sami, H., Mourad, A., Otrok, H., Bentahar, J.: Demand-driven deep reinforcement learning for scalable fog and service placement. IEEE Transactions on Services Computing. In Press (2021). https://doi.org/10.1109/TSC.2021.3075988
21. Sami, H., Otrok, H., Bentahar, J., Mourad, A.: AI-based resource provisioning of IoE services in 6G: a deep reinforcement learning approach. IEEE Transactions on Network and Service Management. In Press (2021). https://doi.org/10.1109/TNSM.2021.3066625
22. Wahab, O.A., Mourad, A., Otrok, H., Taleb, T.: Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Commun. Surv. Tutorials **23**(2), 1342–1397 (2021)
23. Wu, W., He, L., Lin, W., Mao, R.: Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. IEEE Trans. Parallel Distrib. Syst. **32**(7), 1539–1551 (2020)
24. Xie, G., Shangguan, A., Fei, R., Ji, W., Ma, W., Hei, X.: Motion trajectory prediction based on a CNN-LSTM sequential model. Sci. China Inf. Sci. **63**(11), 1–21 (2020). https://doi.org/10.1007/s11432-019-2761-y
25. Ye, D., Yu, R., Pan, M., Han, Z.: Federated learning in vehicular edge computing: a selective model aggregation approach. IEEE Access **8**, 23920–23935 (2020)
26. Ye, Y., Li, S., Liu, F., Tang, Y., Hu, W.: EdgeFed: optimized federated learning based on edge computing. IEEE Access **8**, 209191–209198 (2020)
27. Yu, Z., Hu, J., Min, G., Zhao, Z., Miao, W., Hossain, M.S.: Mobility-aware proactive edge caching for connected vehicles using federated learning. IEEE Transactions on Intelligent Transportation Systems (2020)
28. Zhang, D., Ge, H., Zhang, T., Cui, Y.Y., Liu, X., Mao, G.: New multi-hop clustering algorithm for vehicular ad hoc networks. IEEE Trans. Intell. Trans. Syst. **20**(4), 1517–1530 (2018)
29. Zhang, J., Zhao, Y., Wang, J., Chen, B.: FedMEC: improving efficiency of differentially private federated learning via mobile edge computing. Mob. Netw. Appl. **25**(6), 2421–2433 (2020)
30. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)