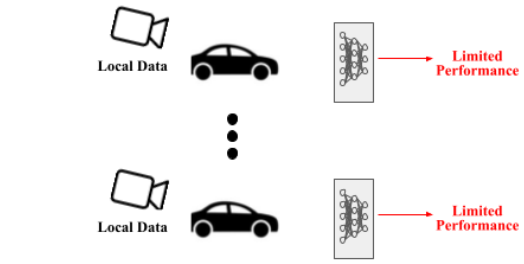
Federated Learning-based Driver Activity Recognition for Edge Devices

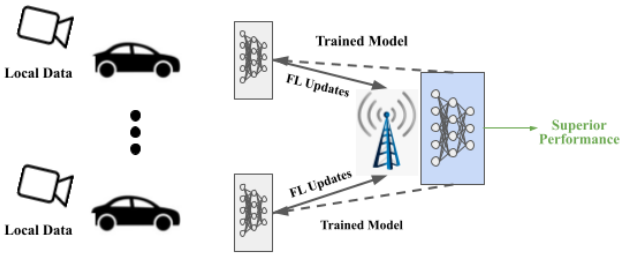
Keval Doshi and Yasin Yilmaz
University of South Florida
4202 E Fowler Ave, Tampa, FL 33620
{kevaldoshi, yasiny}@usf.edu

Abstract

Video action recognition has been an active area of research for the past several years. However, the majority of research is concentrated on recognizing a diverse range of activities in distinct environments. On the other hand, Driver Activity Recognition (DAR) is significantly more difficult since there is a much finer distinction between various actions. Moreover, training robust DAR models requires diverse training data from multiple sources, which might not be feasible for a centralized setup due to privacy and security concerns. Furthermore, it is critical to develop efficient models due to limited computational resources available on vehicular edge devices. Federated Learning (FL), which allows data parties to collaborate on machine learning models while preserving data privacy and reducing communication requirements, can be used to overcome these challenges. Despite significant progress on various computer vision tasks, FL for DAR has been largely unexplored. In this work, we propose an FL-based DAR model and extensively benchmark the model performance on two datasets under various practical setups. Our results indicate that the proposed approach performs competitively under the centralized (non-FL) and decentralized (FL) settings.



(a) Non-cooperative.



(b) Cooperative FL.

Figure 1. Non-cooperative training suffers from limited performance due to the limited representation power of local data. On the other hand, FL setup enables training on a diverse dataset while satisfying privacy and communication constraints.

1. Introduction

The identification of distracted driving is one of the most crucial, demanding, and time-critical tasks for intelligent transportation systems (ITS). There has been a steady increase in the number of traffic accidents caused by distracted driving. According to the National Safety Council, distracted driving leads to approximately 1.6 million crashes every year, with a majority being caused due to mobile phone use [7]. As a result, driver activity recognition (DAR) has become a subject of increasing interest.

Several computer vision tasks, such as image classification and action recognition, have advanced dramatically in recent years due to improved deep learning architectures and large annotated datasets. Similarly, designing robust DAR approaches require a tremendous amount of data from multiple sources and significant computational resources. However, in such a scenario, centralized training is impractical due to massive communication and storage overheads. Moreover, centralizing data may also lead to security and privacy concerns, and violate regulations such as the General Data Protection Regulation [2]. Federated Learning (FL) is offered as a distributed model training method that does not communicate raw data, therefore maintaining data privacy and saving communication bandwidth [16, 26, 36, 37]. Despite the rapid growth of FL research for computer vision, practical applications such as

DAR has received little attention to date.

Specifically, in an FL system, multiple parties train a machine learning model cooperatively without exchanging raw data [22]. The system generates a common machine learning model for the parties such that the model learnt via FL is superior to a model learned via local training with the same model architecture. Typically, a larger model trained on sufficient amount of diverse data is known to improve the overall accuracy compared to simpler models that would be normally used at the edge devices. On the other hand, naive FL techniques such as the FedAvg algorithm might overload resource-constrained edge devices with a large model, and thus would not be practical. To this end, we propose leveraging a group knowledge transfer algorithm called FedGKT [10], that minimizes edge computation while maintaining model accuracy comparable to FedAvg. Particularly, FedGKT uses knowledge distillation to transfer information from edge devices to a central model. As illustrated in Fig. 1, it is usually not easy to locally collect and train on a dataset sufficiently representative of all relevant classes/scenarios. On the other hand, gathering a diverse dataset from multiple sources and training a model in a centralized manner is also not feasible in general due to privacy and communication overhead constraints. An FL setup can enable continually updating a classification model with diverse data from multiple sources by sharing only some processed information instead of raw data.

In summary, we propose the first FL setup for distracted driver activity recognition. Our proposed approach takes into consideration the limited computational and communication resources available on edge devices, and can easily perform training on edge and real-time inference. The experimental results reveal that our approach, which has a resource-efficient FL implementation, is capable of performing competitively and is ranked fifth on the Track 3 test set of the AI City Challenge 2022 (Fig. 2), with an F1-score of 0.2921.

2. Related Works

Several recent works consider visual, auditory, and biomechanical distractions to study driver behavior. The majority of existing distracted driver studies necessitate the extraction of certain specific modalities such as head posture angle, hand and body joint position, and eye tracking [24, 28, 35]. However, such approaches require the use of specialized hardware, making it financially infeasible to deploy them on a large scale. As a result, an end-to-end driver activity detection system based on deep CNN models is proposed in this paper, which is accurate and simple to deploy.

Broadly, in the existing literature, driver activities are classified into two classes, maneuvering-based [8, 25, 27] (starting, changing lanes, etc.) and distraction-based (eating, drinking, talking, etc.) [1, 4, 9, 20, 21]. In this work, we primarily focus on the distraction-based activities. In [20], Martin et al. propose a multi-modal method to combine multiple streams involving body pose and contextual information. Similarly, Behera et al. [4] leverage LSTMs to extract spatiotemporal features for recognizing various activities. Recently, Li et al. [14] proposed an egocentric spatial-temporal interaction based approach to evaluate how drivers interact with road users.

Early FL algorithms such as FedAvg [23] and FedMA [32] employ a naive averaging approach and do not account for computational limitations in edge devices. The complexity of FL in edge computing and distributed networks stems mostly from client heterogeneity [38]. For example, heterogeneous clients have varying data quality, data quantity, calculation capability (i.e., compute resources), communication condition, and willingness to engage. More recently, several FL approaches have been proposed [3,5,18,29–31,33] that focus on minimizing the communication overhead, and yet do not account for the computational cost on edge devices. More recently, a group knowledge transfer based approach called FedGKT was proposed in [10], that leverages split learning and FedAvg to minimize computation and communication overheads without compromising the model accuracy.

[12] is the first work that applies FL to a real-world image dataset, Google Landmark [34], which has now become the standard image dataset for FL research. Recently, [6, 15, 17] applied FL on medical image segmentation tasks, where the training data may not be available at a single medical institution due to data privacy regulations. In the object detection task, [39] proposes a KL divergence method to mitigate model accuracy loss due to non-IID data. FedVision [19] is an FL framework for object detection, which supports object detection models such as FastRCNN and YOLOv3. For FL in other application domains, we refer the reader to the survey in [13].

3. Methodology

In addition to accuracy, we also target computational efficiency for edge computing in vehicles. Despite the fact that several recent approaches have demonstrated promising results on current benchmark datasets, they are also associated with a large amount of computational overhead. Moreover, it is not realistic to assume *homogeneity*, i.e., the availability of sufficient training data for all classes at each edge device. Rather, in a practical setup it is more common to observe *heterogeneity* or non-i.i.d. data distribution, where several different edge devices observe a diverse range of classes. Most existing works assume centralizing data from all such edge devices, which may lead to security and privacy concerns. In this section, we present our approach for applying FL to distracted driver activity recognition. We

Figure 2. Various views in the AICITY Track 3 dataset.

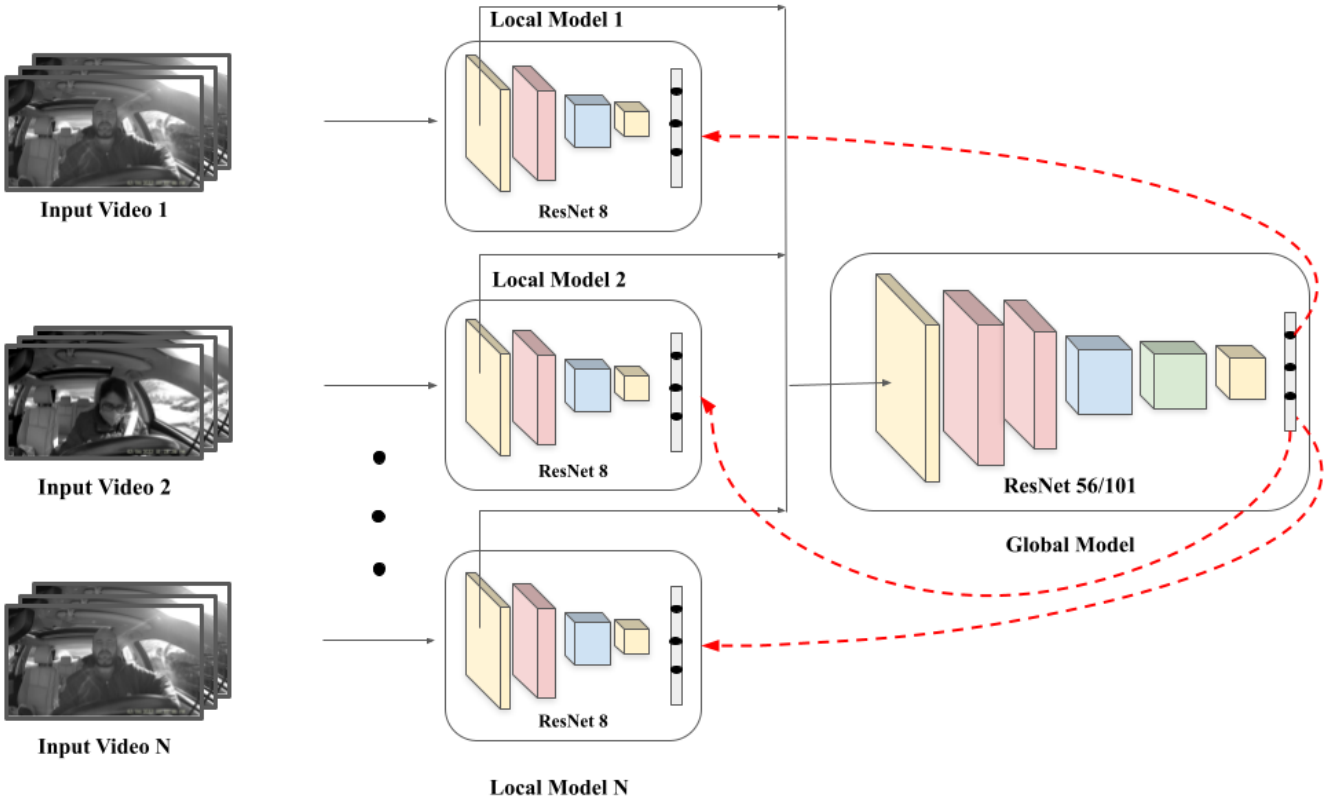


Figure 3. FL Architecture.

first present our problem formulation, and then introduce the proposed FL setup (Fig. 3).

3.1. Problem Formulation

The distracted driver activity recognition task can be defined as a multi-class classification problem where given a set of N training videos $X^{Train} = \{x_1, x_2, \ldots, x_N\}$ and labels y_n from C classes, we aim to learn a function $\mathcal{F}$ that classifies a set of test videos X^{Test} accurately. To detect an activity, we extract frames from each video and treat it as a image classification problem, primarily because of two reasons. First, video classification models typically require significant number of frames to confidently detect an activity, which might lead to a higher detection delay. Secondly, 2D CNN models require significantly less computational resources as compared to 3D CNN models and thus can easily run on resource-constrained edge devices in real-time. Hence, to train the image classification model, we learn $\mathcal{F}$ by minimizing the cross entropy loss L_x given by

$$L_x = -\sum_{n=1}^{N} \log[\mathcal{F}(x_n)]_{y_n} \tag{1}$$

where y_n is the class label index of sample n, $\mathcal{F}(x_n)$ is the predicted class probability vector, and $[\mathcal{F}(x_n)]_{y_n}$ denotes the predicted probability for class y_n.

We evaluate our proposed approach under two different settings, i.i.d. and non-i.i.d. In the i.i.d. setting, the available training data is divided homogeneously among all edge devices, whereas in the non-i.i.d. setting the data distribution among the edge devices is highly skewed.

3.2. FL Setup

In [22], FL is suggested as an alternative to centralized learning. In an FL system, a server coordinates with local nodes (clients) and sends them a global deep neural network model. The clients utilize their own data to train the model locally, then communicate it back to the server to be aggregated into an updated global model. The server repeats this approach until the global model's performance on a task converges. Thus, the data at a local node is never shared with third parties, providing privacy. To optimize the training loss across clients, FL algorithms try to obtain a global model. One of the most popular FL approach is the Federated Averaging (FedAvg) algorithm proposed in [22]. Specifically, FedAvg optimizes the local training loss using Stochastic Gradient Descent (SGD). The objective for FedAvg can be expressed as:

$$\min_{w} F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \tag{2}$$

where $F_k(w)$ is the local loss of client k, n_k is the number of training samples on client k, with a total of n training samples partitioned across all K clients. However, the primary flaw in such naive FL methods is that they require similar model architectures at the central server and edge devices. Hence it might not always be possible to train big CNNs on resource-constrained edge devices due to a lack of GPUs and adequate storage. To this end, we propose leveraging the recently proposed group knowledge transfer algorithm (FedGKT) [10]. It is important to note that FedGKT takes an alternating minimization (AM) method to FL, which fixes one random variable (the edge model) while simultaneously optimizing the server model. As a result of this change, FedGKT contributes to the development of a new group knowledge transfer paradigm, which in turn improves the server model's performance.

Specifically, FedGKT shifts the computing burden from resource constrained edge devices to the centralized server. As shown in Fig. 3, the local edge devices train on the available data and produce a feature representation of the same dimension. The feature representations are used as input to the server model which is trained by minimizing a knowledge distillation-based loss function, given by

$$L_{server} = L_x + \sum_{k=1}^{K} D_{KL}(p_k \,||\, p_s) \tag{3}$$

where L_x is the cross-entropy loss (Eq. 1), D_{KL} is the Kullback-Leibler (KL) divergence, and p_k, p_s are the probabilistic predictions of edge model k and server model, respectively, i.e., $p_k = \mathcal{F}_k(x)$. To further improve the performance of edge devices, the predicted class probabilities from the server model are used to fine tune the local edge device models using

$$L_k = L_x + D_{KL}(p_s \,||\, p_k). \tag{4}$$

During inference, the final model consists of feature extractors from the local models and the trained server model.

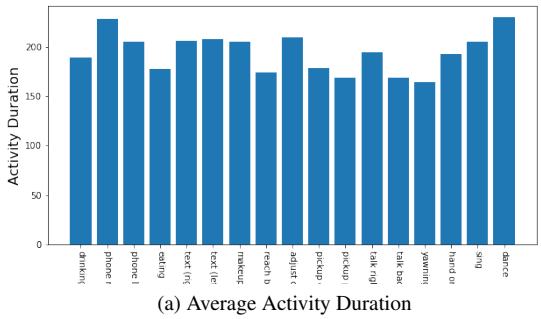
Feature Extractor: Several recent works claim that the depth of a network is a significant factor in determining the network's performance, i.e., using a deeper architecture leads to improved performance. However, training a deeper network is difficult due to the well-known gradient vanishing problem. To address this issue, He et al. [11] proposed a straightforward but effective technique known as residual neural networks (ResNet). ResNet provides a framework for training networks that are significantly deeper than those previously used by leveraging skip connections. In our implementations, we use the ResNet-56 for the server model and ResNet-8, which consists of 8 convolutional layers, as a compact edge device model. Note that, FedAvg would require all server and edge models to be the computationally expensive ResNet-56.

Implementation Details: The FedGKT framework is implemented using the FedML library and deployed in a distributed environment. For the AICITY challenge, we leverage the computationally expensive Resnet-101 model, whereas for the federated setup, we leverage the Renet-56 architecture. In the i.i.d. setup, we randomly select videos from the entire training set and assign them to each node, whereas in the non-i.i.d. setup each node is assigned a random set of classes and only observes videos belonging to those classes. The server model and client models are trained using 4 NVIDIA RTX 6000 GPUs for 20 epochs. Based on experimental results, we leverage the Adam optimizer and a learning rate of 0.0001 for i.i.d. data and SGD optimizer with a learning rate of 0.005 for non-i.i.d. data [10].

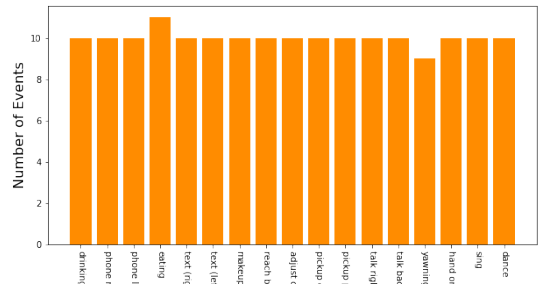
4. Experiments

4.1. Datasets

To demonstrate the efficacy of our approach, we use two benchmark datasets, namely the AI City Challenge dataset and StateFarm dataset. Since the AI City dataset has limited number of samples and the training labels are unavailable, we use the StateFarm dataset to show our FL results. In Fig. 8, we show a few examples from various classes for both of the datasets. We also analyze the the average activity duration and number of events per class occurring in the



(a) Average Activity Duration



(b) Number of Activities

Figure 4. The average activity duration and number of activities per class in the AI City Dataset.

AI City Dataset in Fig. 4. It is seen that all classes have almost equal number of samples (Fig. 4(b)) and the average activity durations are uniformly distributed (Fig. 4(a)).

AI City Challenge: The Track 3 dataset of the AI City Challenge 2022 consists of 90 videos divided into three sets, A1, A2 and B each consisting of 30 videos. Each set comprises five different drivers performing various actions, captured from three different angles, the dashboard, the rear view, and the right side view (Fig. 2). The participants were only allowed to use video data from one of the views. Each video is approximately 10-minutes long and captured in grayscale at a frame rate of 30 fps and a resolution of 1920×1080. The purpose of the challenge is to devise an algorithm that is capable of identifying all distracted driver activities with accurate start and end times. For a detected activity to be considered as true positive, both start and end times are required to be determined within one second of the ground truth. Otherwise, it is considered as false positive even if the predicted label is true and the predicted duration overlaps with the ground truth. Participants were allowed to train on the labelled A1 and unlabelled A2 sets. The evaluation metric used was the F_1 score.

StateFarm Dataset: StateFarm's distracted driver dataset is one of the first publicly available datasets and consists of approximately 102k images for 10 unique driver activities. As compared to the AI City dataset, the StateFarm dataset only consists of images captured from the right side view.

4.2. Results

AI City Challenge: We show the feature activation maps of our trained ResNet-101 model in Fig. 5. We observe that the model is successfully able to detect and activate regions where the true activity occurs, as seen in the testing part. However, there are also cases in which the model is unable to detect the activity correctly since there are no samples in the training data of a person wearing a seat belt. To offset this issue, we also train on samples from the A2 set of the AI City Challenge. In Table 1, we show the results from the leaderboard of the AI City Challenge 2022. The final F_1 score of 0.2921 placed us fifth in the challenge. We have a precision of 0.4432, which shows that the proposed approach does not suffer from several false alarms. On the other hand, our recall of 0.2179 indicates that the proposed approach misses quite a few activities. However, this can be attributed to the strict evaluation protocol, in which any activity not detected within a one second window is considered as a false negative, and any attempt failing to do that is considered a false positive. Moreover, our model is trained to detect a subset of all possible classes since there are several classes such as *texting* or *adjust control* that cannot be detected from the dashboard view, or are completely action based and cannot be detected from an image, such as *singing* or *dancing*.

Leaderboard			
Rank	**Id**	**Name**	F_1
1	72	VTCC-UTVM	0.3492
2	43	Stargazer	0.3295
3	97	CybercoreAI	0.3248
4	15	OPPPilot	0.3154
5	**78**	**SIS Lab**	**0.2921**
6	16	BUPT-MCPRL2	0.2905
7	106	Winter is Coming	0.2902
8	124	HSNB	0.2849
9	54	VCA	0.2710
10	95	Tahakom	0.2706

Table 1. Result comparison on the Track 3 test set of the AI City Challenge 2022.

StateFarm Dataset: Due to the limited number of labelled instances in the AI City challenge dataset, we leverage the StateFarm dataset to evaluate our FL setup. Specifically, we consider the performance of the proposed model under the i.i.d. and non-i.i.d. setups. In the i.i.d. setup, we uniformly divide the available training data among all edge nodes such that each node receives data from each class. In

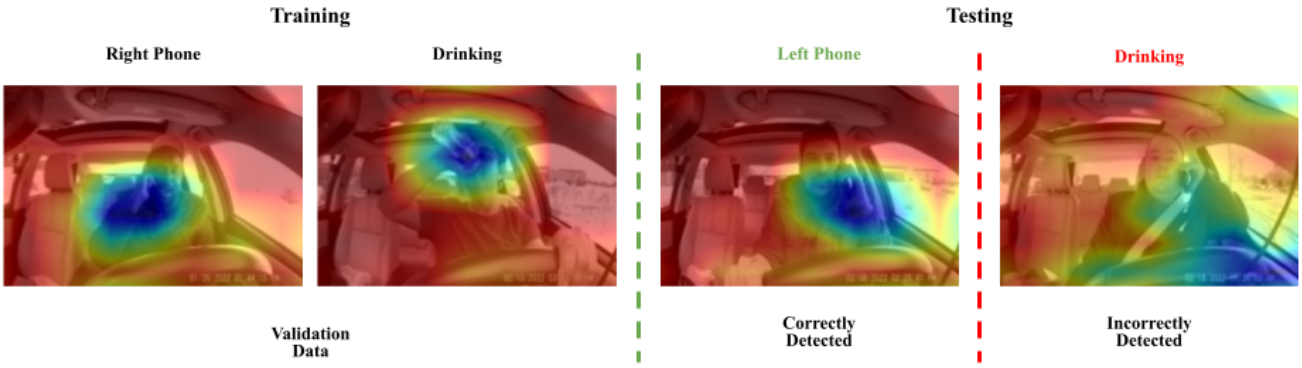
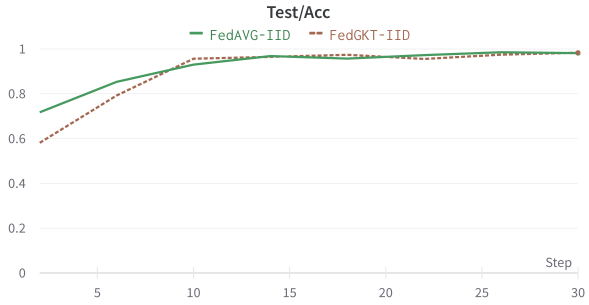
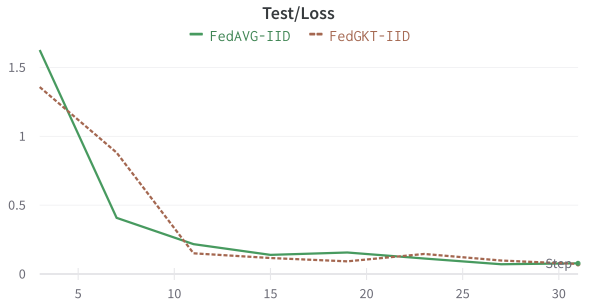


Figure 5. Feature activation map for training and testing on the AI City Dataset.

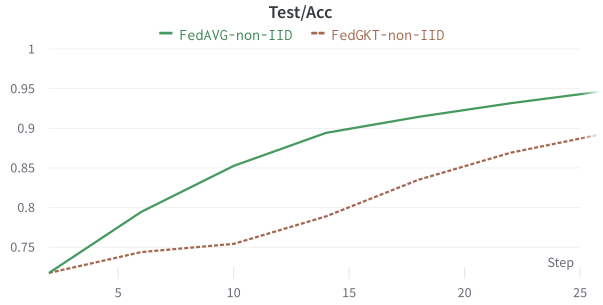


(a) Accuracy of FL approaches under the i.i.d. setup.

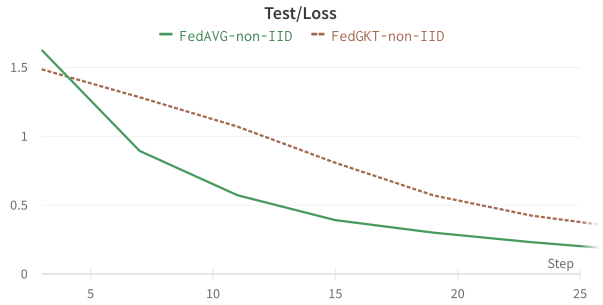


(b) Loss of FL approaches under the i.i.d. setup.

Figure 6. Comparison between the FedAvg and FedGKT algorithms under the i.i.d. setting in terms of the test accuracy and loss.



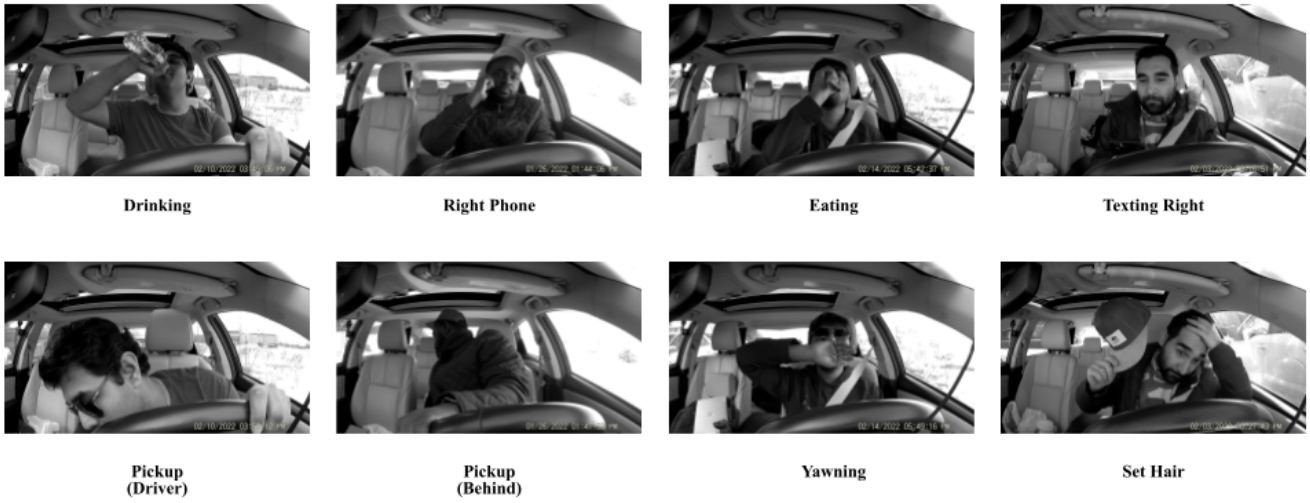
(a) Accuracy of FL approaches under the non-i.i.d. setup.



(b) Loss of FL approaches under the non-i.i.d. setup.

Figure 7. Comparison between the FedAvg and FedGKT algorithms under the non-i.i.d. setting in terms of the test accuracy and loss.

this work, we consider 4 client edge devices and 1 centralized server node. We show the performance of the proposed FL setup using FedAvg and FedGKT under the i.i.d. setting in Fig. 6. It is clearly seen that the computationally efficient FedGKT approach is able to perform competitively with respect to the more computation-intensive FedAvg algorithm, and nearly outperforms it. While FedAvg uses ResNet-56 both at the server and the edge nodes, FedGKT uses ResNet-56 as the server model and ResNet-8 as a compact edge device model. Fig. 7 shows the performance of the FL setup under the non-i.i.d. setting, where again the efficient FedGKT algorithm is able to perform competitively with respect to the FedAvg algorithm. We also train a centralized ResNet-56 model on the entire dataset which achieves an accuracy of 0.989. Overall, we observe that the performance of the various models under both i.i.d. and non-i.i.d. settings is quite close to the centralized model performance, which confirms the efficacy of the proposed decntralized FL



(a) AI City Dataset.

(b) StateFarm Dataset.

Figure 8. A few classes from the AI City dataset and the StateFarm dataset. In this work, we use the dashboard videos from the AI City dataset. The StateFarm dataset only consists of right side view images.

approach.

5. Conclusion

In this work, we propose an efficient federated learning (FL) solution for detecting distracted driver activities. The proposed solution trains the detection model in a decentralized fashion preserving privacy lowering data communications, and yet is also able to perform competitively in the 2022 AI City Challenge. Using the FedAvg and FedGKT algorithms, we demonstrated the proposed FL framework for the activity detection task. We observe that the FedGKT approach is able to achieve a close performance to the FedAvg approach, even after being computationally more efficient by several order of magnitudes. The FL results staying close to the centralized results showed that the proposed approach can be effectively trained in a privacy-preserving and communication-efficient way.

References

[1] Yehya Abouelnaga, Hesham M Eraqi, and Mohamed N Moustafa. Real-time distracted driver posture classification. *arXiv preprint arXiv:1706.09498*, 2017. 2

[2] Sanchit Alekh. Eu general data protection regulation: A gentle introduction. *arXiv preprint arXiv:1806.03253*, 2018. 1

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017. 2

[4] Ardhendu Behera, Alexander Keidel, and Bappaditya Debnath. Context-driven multi-stream lstm (m-lstm) for recognizing fine-grained activity of drivers. In *German Conference on Pattern Recognition*, pages 298–314. Springer, 2018. 2

[5] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018. 2

[6] Qi Chang, Hui Qu, Yikai Zhang, Mert Sabuncu, Chao Chen, Tong Zhang, and Dimitris Metaxas. Synthetic Learning: Learn From Distributed Asynchronized Discriminator GAN Without Sharing Medical Image Data. *arXiv e-prints*, page arXiv:2006.00080, May 2020. 2

[7] National Safety Council. Motor vehicle safety issues - distracted driving, 2022. Last accessed 13 April 2022. 1

[8] Anup Doshi and Mohan M Trivedi. Tactical driver behavior prediction and intent inference: A review. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1892–1897. IEEE, 2011. 2

[9] Hesham M Eraqi, Yehya Abouelnaga, Mohamed H Saad, and Mohamed N Moustafa. Driver distraction identification with an ensemble of convolutional neural networks. *Journal of Advanced Transportation*, 2019, 2019. 2

[10] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020. 2, 4

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[12] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated Visual Classification with Real-World Data Distribution. *arXiv e-prints*, page arXiv:2003.08082, Mar. 2020. 2

[13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning, Vol 14, Issue 1–2*, 2021. 2

[14] Chengxi Li, Yue Meng, Stanley H Chan, and Yi-Ting Chen. Learning 3d-aware egocentric spatial-temporal interaction via graph convolutional networks. In *International Conference on Robotics and Automation*, 2019. 2

[15] Daiqing Li, Amlan Kar, Nishant Ravikumar, Alejandro F Frangi, and Sanja Fidler. Fed-Sim: Federated Simulation for Medical Imaging. *arXiv e-prints*, page arXiv:2009.00668, Sept. 2020. 2

[16] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018. 1

[17] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M. Jorge Cardoso, and Andrew Feng. Privacy-preserving Federated Brain Tumour Segmentation. *arXiv e-prints*, page arXiv:1910.00962, Oct. 2019. 2

[18] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017. 2

[19] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In *AAAI*, pages 13172–13179, 2020. 2

[20] Manuel Martin, Johannes Popp, Mathias Anneken, Michael Voit, and Rainer Stiefelhagen. Body pose and context information for driver secondary task detection. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 2015–2021. IEEE, 2018. 2

[21] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2801–2810, 2019. 2

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017. 2, 4

[23] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016. 2

[24] Eshed Ohn-Bar, Sujitha Martin, Ashish Tawari, and Mohan M Trivedi. Head, eye, and hand patterns for driver activity recognition. In *2014 22nd international conference on pattern recognition*, pages 660–665. IEEE, 2014. 2

[25] Nuria Oliver and Alex P Pentland. Graphical models for driver behavior recognition in a smartcar. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*, pages 7–12. IEEE, 2000. 2

[26] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11):2204–2239, 2019. 1

[27] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7699–7707, 2018. 2

[28] Mahdi Rezaei and Reinhard Klette. Look at the driver, look at the road: No distraction! no accident! In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 129–136, 2014. 2

[29] Jinhyun So, Basak Guler, and A. Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *arXiv preprint arXiv:2002.04156*, 2020. 2

[30] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, pages 7652–7662, 2018. 2

[31] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, pages 9850–9861, 2018. 2

[32] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020. 2

[33] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*, pages 1299–1309, 2018. 2

[34] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2572–2581, 2020. 2

[35] Yang Xing, Chen Lv, Huaji Wang, Dongpu Cao, Efstathios Velenis, and Fei-Yue Wang. Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE transactions on Vehicular Technology*, 68(6):5379–5390, 2019. 2

[36] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019. 1

[37] Shengwen Yang, Bing Ren, Xuhui Zhou, and Liping Liu. Parallel distributed logistic regression for vertical federated learning without third-party coordinator. *arXiv preprint arXiv:1911.09824*, 2019. 1

[38] Dongdong Ye, Rong Yu, Miao Pan, and Zhu Han. Federated learning in vehicular edge computing: A selective model aggregation approach. *IEEE Access*, 8:23920–23935, 2020. 2

[39] Peihua Yu and Yunfeng Liu. Federated object detection: Optimizing object detection model with federated learning. In *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, pages 1–6, 2019. 2