Smart Road Danger Detection and Warning

Dissertation
zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
“Doctor rerum naturalium”
der Georg-August-Universität Göttingen

im Promotionsprogramm Computer Science (PCS)
der Georg-August University School of Science (GAUSS)

vorgelegt von

Yachao Yuan
aus China

Göttingen, November 2021

Betreuungsausschuss

Prof. Dr. Dieter Hogrefe,
Telematics Group, Institut für Informatik

Prof. Dr. Xiaoming Fu,
Computer Networks Group, Institut für Informatik

Prof. Dr. Lutz M. Kolbe,
Chair of Information Management, Wirtschaftswissenschaftliche Fakultät

Prof. Dr. Marcus Baum
Data Fusion Lab, Institut für Informatik

Mitglieder der Prüfungskommission

Referent:	Prof. Dr. Dieter Hogrefe, Telematics Group, Institut für Informatik, Georg-August-Universität Göttingen
Referent:	Prof. Dr. Shengjin Wang, Media Big-data Cognitive Computing Group, Department of Electronic Engineering, Tsinghua University
Korreferent:	Prof. Dr. Lutz M. Kolbe, Chair of Information Management, Wirtschaftswissenschaftliche Fakultät, Institut für Informatik, Georg-August-Universität Göttingen

Weitere Mitglieder der Prüfungskommission

Prof. Dr. Marcus Baum,
Data Fusion Group, Institut für Informatik, Georg-August-Universität Göttingen

Prof. Dr. Xiaoming Fu,
Computer Networks Group, Institut für Informatik, Georg-August-Universität Göttingen

Prof. Dr. Carsten Damm,
Theoretical Computer Science Group, Institut für Informatik, Georg-August-Universität Göttingen

Tag der mündlichen Prüfung

11. November 2021

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [University of Goettingen]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Göttingen, November 2021

Abstract

Road dangers have caused numerous accidents, thus detecting them and warning users are critical to improving traffic safety. However, it is challenging to recognize road dangers from numerous normal data and warn road users due to cluttered real-world backgrounds, ever-changing road danger appearances, high intra-class differences, limited data for one party, and high privacy leakage risk of sensitive information. To address these challenges, in this thesis, three novel road danger detection and warning frameworks are proposed to improve the performance of real-time road danger prediction and notification in challenging real-world environments in four main aspects, i.e., accuracy, latency, communication efficiency, and privacy.

Firstly, many existing road danger detection systems mainly process data on clouds. However, they cannot warn users timely about road dangers due to long distances. Meanwhile, supervised machine learning algorithms are usually used in these systems requiring large and precisely labeled datasets to perform well. The **E**dge-**c**loud-based **R**oad **D**amage detection and warning framework (EcRD) is proposed to improve latency and reduce labeling cost, which is an Edge-cloud-based Road Damage detection and warning framework that leverages the fast-responding advantage of edges and the large storage and computation resources advantages of the cloud. In EcRD, a simple yet efficient road segmentation algorithm is introduced for fast and accurate road area detection by filtering out noisy backgrounds. Additionally, a light-weighted road damage detector is developed based on Gray Level Co-occurrence Matrix (GLCM) features on edges for rapid hazardous road damage detection and warning. Further, a multi-types road damage detection model is proposed for long-term road management on the cloud, embedded with a novel image-label generator based on Cycle-Consistent Adversarial Networks, which automatically generates images with corresponding labels to improve road damage detection accuracy further. EcRD achieves 91.96 % accuracy with only 0.0043 s latency, which is around 579 times faster than cloud-based approaches without affecting users' experience while requiring very low storage and labeling cost.

Secondly, although EcRD relieves the problem of high latency by edge computing techniques, road users can only achieve warnings of hazardous road damages within a small area due to the limited communication range of edges. Besides, untrusted edges might misuse users' personal information. A novel Privacy-preserving edge-cloud and **Fed**erated learning-based **R**oad **D**anger detection and warning framework (FedRD) named FedRD is developed to improve the coverage range of warning information and protect data privacy. In FedRD, a new hazardous road damage detection model is proposed leveraging the advantages of feature fusion. A novel adaptive federated learning strategy is designed for high-performance model learning from different edges. A new individualized differential privacy approach with pixelization is proposed to protect users' privacy before sharing data. Simulation results show that FedRD achieves similar high detection performance (i.e., 90.32 % accuracy) but with more than 1000 times wider coverage than the state-of-the-art, and works well when some edges only have limited samples; besides, it largely preserves users' privacy.

Finally, despite the success of EcRD and FedRD in improving latency and protecting privacy, they are only based on a single modality (i.e., image/video) while nowadays, different modalities data becomes ubiquitous. Also, the communication cost of EcRD and FedRD are very high due to undifferentiated data transmission *(both normal and dangerous data) and frequent model exchanges in its federated learning setting, respectively. A novel edge-cloud-based privacy-preserving **Fed**erated **M**ultimodal learning framework for **R**oad **D**anger detection and warning named FedMRD is introduced to leverage the multi-modality data in the real-world and reduce communication costs. In FedMRD, a novel multimodal road danger detection model considering both inter-and intra-class relations is developed. A communication-efficient federated learning strategy is proposed for collaborative model learning from edges with non-iid and imbalanced data. Further, a new multimodal differential privacy technique for high dimensional multimodal data with multiple attributes is introduced to protect data privacy directly on users' devices before uploading to edges. Experimental results demonstrate that FedMRD achieves around* $96.42\,\%$ *higher accuracy with only* $0.0351\,\mathrm{s}$ *latency and up to 250 times less communication cost compared with the state-of-the-art, and enables collaborative learning from multiple edges with non-iid and imbalanced data in different modalities while preservers users' privacy.*

Acknowledgements

I would like to express my deepest appreciation and thank all the people who contributed to this thesis's successful completion.

First and foremost, I would like to extend my sincere thanks to my supervisor Prof. Dr. Dieter Hogrefe, for his valuable contributions, patient guidance, and support during my Ph.D. study. Prof. Hogrefe provides me with this great opportunity to continue my Ph.D. study. He is a knowledgeable professor and a very nice person. I deeply appreciate all his contributions of time, guidance, and funding to make my Ph.D. study complete and successful.

I also would like to extend my sincere thanks to Prof. Dr. Lutz M. Kolbe, who gave me a valuable chance to study in Germany at the University of Göttingen. I really appreciate it. His rigorous research attitude and profound professional knowledge benefit me a lot. I am very grateful to him for his support, supervision, and guidance during my Ph.D. study.

I also would like to extend my sincere thanks to Prof. Dr. Xiaoming Fu, Prof. Dr. Marcus Baum, and Prof. Dr. Shengjin Wang for their precious time, guidance, and help during my Ph.D. study. I am very grateful to Prof. Dr. Thar Baker since he never hesitates to provide me feedback, guidance, and help.

Additionally, I want to thank all my thesis committee members, Prof. Dr. Dieter Hogrefe, Prof. Dr. Shengjin Wang, Prof. Dr. Lutz M. Kolbe, Prof. Dr. Xiaoming Fu, Prof. Dr. Marcus Baum, and Prof. Dr. Carsten Damm, for investing their valuable time to my defense.

I am also very grateful to all my colleagues at Telematics Group and Smart Mobility Research Group at the University of Göttingen. They help me a lot with my study and research during my Ph.D. study. I also thank all my co-authors and my Bachelor's and Master's students for their contributions to finishing my thesis.

Last but not least, I would like to sincerely thank my parents and my family for always supporting me, believing in me, and loving me. I would not be where I am today without their support and help. I love them.

Contents

Acronyms

EcRD	**E**dge-**c**loud-based **R**oad **D**amage detection and warning framework	v
FedRD	Privacy-preserving edge-cloud and **Fed**erated learning-based **R**oad **D**anger detection and warning framework	v
ADD	Advanced Road Damage Detection model	xv
AFed	Adaptive Federated learning	xvii
MRDD	Multimodal Road Danger Detection model	xvii
FedML	Federated Multimodal Learning strategy	1
FedMRD	Edge-cloud-based privacy-preserving **Fed**erated **M**ultimodal learning **R**oad **D**anger detection and warning framework	4
FL	Federated Learning	4
DP	Differential Privacy	4
LDP	Local Differential Privacy technique	4
DFRD	Deep Feature based Road Detection model	5
HDD	Hazardous Damage Detection model	6
MRDD1	Multimodal Road Danger Detection model (solution 1)	6
MRDD2	Multimodal Road Danger Detection model (solution 2)	6
IDPP	Individualized Differential Privacy with Pixelization	6
MDP	Multimodal Differential Privacy technique	6
MDD	Multi-types Damage Detection model	8
IoT	Internet of Things	8
DCNN	Deep Convolutional Neural Network	9
DCNNs	Deep Convolutional Neural Networks	9
CycleGAN	Cycle-Consistent Adversarial Network	9
QoS	Quality of Services	10
RSUs	Road Side Units	13
GLCM	Gray Level Co-occurrence Matrix	16
RSU	Road Side Unit	21
AbRS	Abnormal Road Screening model	35
FC	Fully Connected layer	41

CNN	Convolutional Neural Network	51
FedAvg	Federated Averaging approach	60
FedML1	Federated Multimodal Learning algorithm with MRDD1	82
FedML2	Federated Multimodal Learning algorithm with MRDD2	83

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Road transportation networks are essentially social and economic components for all nations [1]. However, poor road conditions, such as road dangers, including road damages, crashed vehicles, fallen trees, and icy roads, drastically increase the risk of road accidents and may cause serious injuries or fatalities. For example, when there is a road danger (e.g., big hole) on the road but drivers are unaware of it. In this case, one vehicle may collide with another and injure the driver, passengers, or pedestrians when hitting or trying to avoid the road danger. As reported in [2], road traffic injuries claim more than 1.2 million lives and cost governments around 3% of GDP. Except for human error, road dangers are the primary causes of road accidents. Therefore, detecting road dangers and warning drivers timely can significantly lessen road accidents or fatalities and ensure road traffic safety.

However, it is still challenging for drivers to obtain accurate road danger information (e.g., dangerous types and locations) due to huge road network volumes, cluttered real-world environments, various road dangers, high intra-class differences, and difficulty of data collection. Therefore, this thesis aims to propose fast, accurate, communication-efficient, and private road danger detection and warning frameworks.

Existing road danger detection is mainly performed by road inspectors or streaming video monitors, which is labor-intensive, time-consuming, and costly. Meanwhile, expensive vehicles equipped with various sensors, High Definition (HD) cameras, and illumination devices are also employed for road damage danger inspection in countries like Germany. However, it is unaffordable for many developing countries or regions. So far, researchers have proposed many automatic road danger detection systems [3, 4], but they cannot warn drivers timely about road dangers. Few researchers have considered this issue, like [3]. However, the proposed models are mainly cloud-based, where data is processed on a central cloud, and warning messages are sent to drivers from the cloud.

Due to the long latency (i.e., waiting time of road users for receiving road danger information) between the cloud and road users, users may not receive warnings in time, resulting in serious road accidents. Further, most existing systems are sensor-based [5,6], so road dangers can only be detected when vehicles hit them, which is dangerous and not suitable for accident-preventing. To my best knowledge, there are no existing systems that can rapidly respond to road danger, detect road dangers accurately with minimum communication costs while preserving both users' privacy (e.g., users' ID and location) and sensitive information inside data (e.g., people's faces, locations, and drivers' license plate numbers).

The following research questions will be answered in this thesis to fill the research gaps and achieve our goals:

1. How to reduce latency for fast road danger detection and warning?
2. How to achieve high road danger detection accuracy (with low data collection and labeling cost)?
3. How to significantly reduce communication cost and bandwidth usage when performing road danger detection and warning?
4. How to preserve data privacy while maintaining high detection performance?

Three edge-cloud-based road danger detection and warning frameworks, i.e., EcRD [7], FedRD [8], and FedMRD, are introduced to answer these research questions. The EcRD [7] and FedRD [8] frameworks were published in the IEEE Internet of Things Journal in 2020 and the Journal of Future Generation Computer Systems in 2021, respectively. The Edge-cloud-based privacy-preserving **Fed**erated **M**ultimodal learning **R**oad **D**anger detection and warning framework (FedMRD) is under review in the IEEE Transactions on Industrial Informatics. Specifically, EcRD leverages edge-cloud computing for real-time hazardous road damage warning and long-term multi-types road damage management. FedRD improves EcRD's warning map coverage, enhances the robustness of the detection model, enables distributed learning among edges by Federated Learning (FL), and decreases privacy leakage risks by Differential Privacy (DP) which has mathematical privacy guarantees. FedMRD solves the problems of FedRD and EcRD, i.e., only based on single modality, high communication cost caused by frequent model exchanging, high dimensionality curse of Local Differential Privacy technique (LDP), by developing two multimodal deep learning-based road danger detection models, a new FL with periodic averaging and model quantization, a novel DP which largely reduces data dimension while maintaining high detection performance.

1.2 Organization of the thesis

The thesis is organized as follows: Chapter 2 introduces a novel Edge-cloud computing-based Road Damage detection and warning framework named EcRD, which utilizes the recent advances of edge and cloud computing to meet the demands of real-time hazardous road damage recognition and warning, and long-term multi-type road damage management. More specifically, the motivation and contributions of EcRD are explained in Section 2.1 and 2.2, respectively. Section 2.3 reviews the related literature. I describe the proposed EcRD framework and the proposed algorithms in detail in Section 2.4 and Section 2.5. Experimental results are illustrated in Section 2.6. Finally, I conclude this chapter with a summary in Section 2.7.

In Chapter 3, a novel edge-cloud computing and Federated learning-based framework for intelligent hazardous Road Damage detection and warning, named FedRD, is proposed. FedRD uses the recent advances of edge-cloud computing, FL, and DP to meet the requirements of privacy-preserving hazardous damage detection and warning. In detail, I introduce the motivation and contribution of the proposed FedRD framework in Section 3.1 and Section 3.2. Then, in Section 3.3, the related state-of-the-art methods are reviewed. Section 3.4 presents the developed FedRD framework. The proposed strategies and techniques are introduced in Section 3.5. Section 3.6 presents the evaluation results of the proposed FedRD framework and the proposed models. Finally, I draw a conclusion of this chapter in Section 3.7.

Chapter 4 presents a new edge-cloud computing-based Federated Multimodal learning framework, called FedMRD for intelligent Road Danger detection and warning. FedMRD leverages the recent advances of edge-cloud computing, multimodal model learning, efficient FL, LDP to meet the requirements of accurate and communication-efficient road danger detection and warning while preserving data privacy. In detail, Section 4.1 and 4.2 provides the motivation and contributions of this chapter. Section 4.3 summarizes related works. In Section 4.4, I introduce the proposed FedMRD framework. Section 4.5 explains the proposed methodologies in detail. Section 4.6 shows the evaluation results of different methods and the overall FedMRD framework by comparing with existing works. Finally, I summarize this Chapter in Section 4.7.

Chapter 5 discusses and compares the advantages and drawbacks of the proposed road danger detection models, communication costs reduction strategies, privacy protection techniques, and frameworks.

Chapter 6 describes the conclusion and future work of this thesis.

1.3 Anticipated Contributions

The anticipated contributions of this thesis are summarized as follows:

- Three novel frameworks, i.e., EcRD, FedRD, and FedMRD, for fast, accurate, communication-efficient, and private road danger detection and warning.
- A new road segmentation model named Deep Feature based Road Detection model (DFRD) to denoise the video data, and a novel Cycle-Consistent Adversarial Network-based image-label generator to automatically generate images and their corresponding labels.

- Four new road danger detection models, i.e., Hazardous Damage Detection model (HDD) from EcRD, Advanced Road Damage Detection model (ADD) from FedRD, and Multimodal Road Danger Detection model (solution 1) (MRDD1) and Multimodal Road Danger Detection model (solution 2) (MRDD2) from FedMRD.

- Two new communication cost reduction approaches are proposed, including Adaptive Federated Learning strategy (AFed) from FedRD and Federated Multimodal Learning strategy (FedML) from FedMRD strategies.

- Two novel privacy protection techniques are developed including Individualized Differential Privacy with Pixelization (IDPP) from FedRD and Multimodal Differential Privacy technique (MDP) from FedMRD.

- Extensive experiments are conducted to evaluate the performance of the proposed frameworks and algorithms. LDP-based techniques (i.e., IDPP and MDP) are also proved mathematically in this thesis.

Chapter 2

EcRD: Edge-cloud Computing Framework for Smart Road Danger Detection and Warning

2.1 Motivation

As mentioned in Chapter 1, monitoring road dangers and warning road users are critical to prevent road accidents. Among all types of road dangers, road damage type is one of the main causes of road accidents [9] since roads are crumbling in many countries due to aging, lacking periodic maintenance, or natural disasters [10]. Thus, this chapter studies road damage as an example of road dangers.

Despite the importance, road damage danger detection and warning are challenging due to huge road networks and cluttered real-world environments. Most previous works such as [3] only focus on road damage (e.g., cracking) detection, while very few researchers [11] warn drivers about serious damages like big holes in real-time. However, they process data on the cloud and send alerts to drivers, thus suffering from high latency due to long distances. Meanwhile, supervised machine learning algorithms are usually used in these systems requiring large and precisely labeled datasets to achieve good performance.

Therefore, in this chapter, EcRD: an Edge-cloud computing-based Road Damage detection framework is proposed, which is an efficient framework for detecting road damages and warning users. Unlike previous works, e.g., [3, 11], I leverage the recent advances of edge and cloud computing and further exploit the benefits of combining both edge and cloud computing in road damage detection and warning applications. More precisely, by dividing the road damage detection and warning task into two subtasks, i.e., hazardous damage detection task and multi-type damage detection task, and distributing them on edges and the cloud separately according to their urgency or delay-tolerance levels. In this way, drivers can receive critical life-threatening road damage messages from edges in real-time which prevents serious road accidents from happening, while users of the cloud (e.g., road administrations and individuals) can access more detailed road damage information (e.g., damage types and locations) of both current and historical records for long-term road management or traveling route planning. Additionally, edges' limited resources can be saved for more critical and delay-sensitive tasks. Meanwhile, clouds can be utilized for delay-insensitive tasks, data storage, and information retrieval. Further, three detection models, DFRD, HDD, and Multi-types Damage Detection model (MDD), are proposed to optimize the performance of the EcRD framework. Specifically, the DFRD model is proposed to filter out the noisy background for better road inspection, leveraging the advantages of transfer learning and superpixels. An efficient unsupervised HDD model is developed for rapid hazardous damage detection on edges. Unlike existing works [12, 13], HDD is unsupervised, which means it does not require large datasets for training to achieve good performance. Moreover, to detect multi-type road damages on the cloud with limited training images, the MDD model is introduced. The proposed approaches, in combination with the well-designed and task-oriented edge-cloud computing structure, form a fast and efficient road damage detection and warning system (i.e., EcRD).

Overall, EcRD is a general framework for road danger detection and warning, and users can adjust the subtasks according to their needs. An extensive evaluation is performed using both public and collected datasets to prove its effectiveness. Experiments show that EcRD can detect hazardous damages with 92.43% F1-score and $0.0043\,\mathrm{s}$ latency on a normal laptop which satisfies the requirement of real-time response. Besides, 94.31% AP50 is also obtained for the multi-type damage detection task on the cloud. It is further observed that damaged roads are far less than normal roads in the real world, so only damaged road data after detection is stored on the cloud to use storage resources optimally and reduce the entire storage cost. To the best of my knowledge, it is the first work to propose an edge-cloud computing-based smart road damage detection framework satisfying both real-time road damage detection and warning and long-term road management requirements.

2.2 Contributions

The contributions of this chapter, which were published in the IEEE Internet of Things Journal in 2020 [7]), include:

- A novel Edge-cloud computing-based Road Damage detection framework named EcRD is proposed. It utilizes the recent advances of edge and cloud computing to meet the requirements of real-time hazardous damages warning and long-term multi-type damages management.

- A Deep Feature-based Road Detector called DFRD is proposed to denoise the video data received from Internet of Things (IoT) devices. It reduces image/frame sizes and improves the performance of HDD.

- An unsupervised Hazardous road Damage Detector named HDD is developed for fast and accurate road damage danger detection on edges. It significantly saves data labeling efforts.
- A semi-supervised Multi-type Damage Detector called MDD is introduced for efficient multi-type road damage detection on the cloud, including an anchor-free deep learning model and a novel Cycle-Consistent Adversarial Network (CycleGAN)-based image-label generator. The image-label generator automatically produces road damage danger images and their corresponding labels for MDD.

The structure of this chapter is organized as follows: Section 2.3 gives an elaborate summary and comparison of the related state-of-the-art researches. Section 2.4 introduces the design of EcRD, including its design goals, architectural components, and working process. Details of the DFRD, HDD, and MDD algorithms are explained in Section 2.5. Experimental setup, datasets, baselines, evaluation metrics, and evaluation results are illustrated in Section 2.6. Finally, Section 2.7 concludes this chapter.

2.3 State-of-the-art

In this section, the related literature are reviewed including road segmentation techniques, road damage detection techniques, and cloud/edge computing systems. The overall summery is given in Table 2.1.

Road segmentation techniques: Traditional road segmentation methods rely on specific information such as location priors, structured information (e.g., lane markings, vanishing point, and road boundary), or visual characters (e.g., texture, edge, and color) to detect road region. Chacra et al. [14] utilized location priors to detect road parts but with the assumption that roads presented at the lower part of images, which is not practical in the real world. In [15], many low-level cues such as color intensity, entropy, and local binary pattern histograms were used for road segmentation. The work of [16] estimated vanishing points for road direction estimation to detect drivable roads. However, low-level hand-crafted features may not be suitable for complex images with cluttered backgrounds. Additionally, location priors and structured information can not hold when the specified situation does not appear in images, for example, occlusion on the road or no lane-markings. With the recent development of Deep Convolutional Neural Networks (DCNNs), many Deep Convolutional Neural Network (DCNN)-based methods, for example [17], [18], and [19], have been successfully applied in road segmentation. In [18] and [19], authors presented a DCNN-based network which simultaneously performs the road boundary detection and road segmentation utilizing the predicted results of both tasks to improve each other's performance. It improved the segmentation accuracy; however, extra information such as road contour maps and location priors are required. Despite the great success of DCNNs in road segmentation, it may not be the best solution for the road segmentation task in EcRD since DCNNs inherently require large precisely labeled datasets to learn parameters while data labeling is time-consuming and costly.

Therefore, a simple but efficient road segmentation method called DFRD was developed for the EcRD framework.

Road damage detection techniques: Most recent state-of-the-art road damage detection methods can be summarized into low-level feature-based methods and pattern-based methods, depending on the noise level of the background. Most of the low-level feature-based methods are relatively simple and require less processing time (e.g., [20]), while many pattern-based methods are relatively complex and usually need large datasets (e.g., [12]). Hence, for images with simple and clean backgrounds like road damage images, low-level feature-based methods combining low-level features like intensity [20] or structure information [21] provide relatively fast and accurate results; while pattern-based methods like DCNNs [12] required large training time and much slower on normal computers, but they are more robust in detecting damages on images with cluttered backgrounds. Therefore, combining the advantages of both methods, low-level feature-based methods along with noise reduction (e.g., road segmentation) pre-processing was used on edges for hazardous road damage detection, and a state-of-the-art DCNN model was employed on the cloud for multi-type road damage detection.

Cloud/Edge computing systems: Since real-time video analysis is a high resource-demanding task [27], computation offloading from IoT devices to servers is very important to ensure high Quality of Services (QoS). Many researchers have investigated the strategies of better resource allocation for real-time video-related services. Hossain et al. [22] studied the resource allocation for video analysis services hosted in a Virtual Machine on the cloud to reduce the storage cost and response time. Due to the shortcomings of clouds, such as high response delay, a lightweight edge computing platform was proposed to integrate more computing and networking capabilities in [23]. More specifically, the combination with edges, clouds, and private hardware for computation offloading has been widely investigated [24]. Shojafar et al. [25] proposed an adaptive resource scheduler on edge/fog centers for real-time vehicular cloud services to maximize communication and computation efficiency. The work of [28] employed the deep reinforcement learning technique to improve the performance of online resource scheduling in mobile edge computing networks with a large number of nodes. Kawano et al. [26] leveraged the advantages of edge computing and proposed a damaged lane markings detection system named Deep on edge (DoE). Edge computers are attached to garbage trucks to collect road images of a city constantly. Besides, a deep learning model was utilized to detect damaged lane markings from images on edge computers. The detection results were sent to cloud servers via cellar network for city analysis, and the notification information was sent to citizens from the cloud. However, damaged lane markings information may not be very urgent for drivers. Moreover, [28] and [26] required a large number of precisely labeled samples and have a long training time.

In summary, most of the existing works propose cloud-based solutions. Only a limited number of researchers exploited edge-based solutions [26], and no research leverages the advantages of both edge and cloud to improve the detection of both hazardous and detailed road damages. In addition,

Table 2.1: Representative Works in Road Damage Detection.

Category	Main Techniques	Advantages	Disadvantages
Road segmentation techniques	Hand-craft feature based methods [14–16]	Low computation cost	Rely on specific information like location, structure or visual characters
	DCNN-based methods [17–19]	High accuracy	Large labeled training set, high computational cost
Road damage detection techniques	low-level feature based method [20, 21]	More fast and accurate	Not robust with significant changes
	Pattern based method [12]	More robust	More complex, requiring extra training process
Cloud/Edge computing systems	Cloud-based [22]	High storage and computational resources	High response delay
	Edge-based [23]	Low latency	Low accuracy
	Edge-cloud-based [24–26]	High performance	High computational cost, no emergency consideration

there is no research studies vision-based road damage detection challenges and computational offloading schemes' service requirements. Therefore, in this chapter, the limitation and advantages of both edges and the cloud are studied, and the benefits of the edge-cloud framework in providing high QoS road damage detection and warning services with minimum cost and resources are exploited.

2.4 EcRD Framework

In this section, the design goals, architectural components, and working process of the EcRD framework are presented.

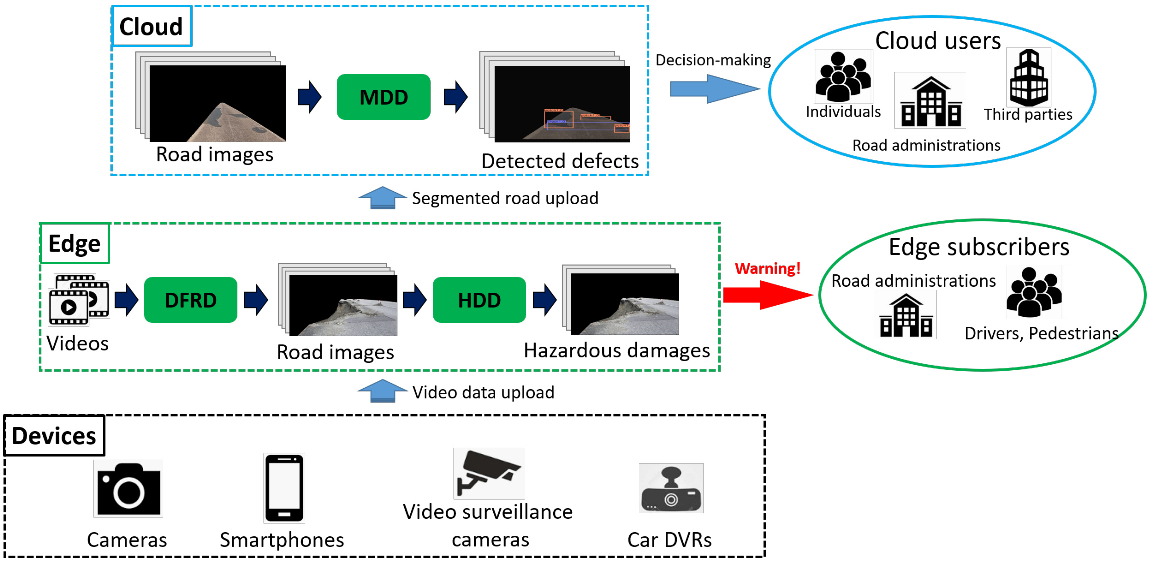


Figure 2.1: Overview of the EcRD framework.

2.4.1 Design Goals

The design goals of EcRD are listed as follows:

1. **Safety:** It is crucial for a road damage detection and warning system to provide road danger information for accident prevention and ensure road safety to users.

2. **Robustness:** A road damage detection and warning system should be able to handle unpredictable and unexpected complications arising from the surrounding environment, e.g., different weather conditions, illuminations, shadows, and obstacles like vehicles or pedestrians.

3. **Accuracy:** A road damage detection and warning system should correctly detect road damages, especially hazardous damages, with high accuracy to avoid accidents.

4. **Latency:** Fast detection and warning of hazardous road damages are critical to ensure traffic safety, especially in disaster areas such as earthquakes or floods. If drivers can not receive dangerous warning information timely, serious traffic accidents could happen when drivers are hitting or trying to avoid the damages. Therefore, hazardous road damage detection tasks should have very low latency. Clouds have rich computation and storage resources; hence, many road damage inspection systems are deployed on clouds. However, they suffer from slow responses (i.e., high latency) due to long distances. Hence, a light-weighted approach with low latency should be developed for hazardous road damage detection.

5. **Resources:** Since video analysis is a high resource-demanding task, the amount of data increases quickly over time. Besides, storage space is expensive for clouds and even worse

for edges. Therefore, designing an efficient storage strategy is critical.

6. **Cost:** Due to the high cost of video data collection and transmission, the data should be used well. Moreover, Internet communication and bandwidth costs are also unneglectable; thus, the newly developed systems need to consume low communication costs and bandwidth to apply in the real world.

2.4.2 Architectural Components

The proposed EcRD framework that satisfies the design goals mentioned above is presented in Figure 2.1. The components of this framework are described in detail as follows:

– Devices: This component gathers video data by the pervasively used IoT devices (e.g., surveillance cameras, car DVRs, or smart devices) mounted on public service vehicles (e.g., buses, taxis, or garbage trucks), and transmits the collected data to the nearest edge computers (e.g., Road Side Units (RSUs)).

– Edges: The hazardous road damage detection task is deployed on edges for rapid warning messages broadcasting and receiving, satisfying low latency requirements. Additionally, the requirements of high robustness and accuracy are achieved by the developed algorithms, and they further enhance the performance of EcRD. Once hazardous damages are detected on edges, it broadcasts a warning message including the damages' locations to nearby users and road administrations who can repair the damages promptly to improve traffic safety. Meanwhile, only the resulting segmented road images (i.e., images with only road part) are uploaded to the cloud server limiting the workload of the bandwidth and reducing the energy consumption of data transmission.

– Cloud: EcRD only collects video data once and extracts all valuable information which reduces data collection cost. After hazardous road damage detection and warning on edges, the data is further analyzed, and more detailed road damage information is extracted on clouds. Specifically, the cloud server receives road images from edges and detects multi-type road damages. It provides cloud users (e.g., road administrators, individuals, and third parties) with detailed road damage information, such as damage types and locations, for better road budget allocation and efficient road maintenance.

– DFRD: This model detects road areas from raw video frames gathered from IoT devices leveraging the advantages of transfer learning and superpixels and successfully filters out noisy backgrounds. The segmented road images are utilized for real-time hazardous road damage detection and warning on edges and then transmitted to the cloud for detailed road damage detection.

– HDD: The developed unsupervised HDD model detects hazardous road damages accurately and rapidly on edges without requiring large precisely labeled datasets for training. Then, the edge immediately informs nearby subscribers once dangerous road damages are detected. Deploying HDD model on edges further reduces latency, thus preventing road accidents and improving road safety.

– MDD: MDD consists of a state-of-the-art anchor-free deep learning model (i.e., CenterNet [29]) and a novel CycleGAN [30]-based image-label generator that provides synthetic images and their corresponding labels for the deep learning model without requiring manual data collection and annotation. Using MDD on the cloud enables further mining of useful information from the collected data. The detected road damages and corresponding GPS locations are sent to cloud users for better decision-making of road network maintenance.

2.4.3 Working Process

As illustrated in Figure 2.1, the general working process of EcRD consists of four main steps: firstly, devices from users collect videos of roads and upload them to nearby edges. Secondly, once an edge receives video data from users, it processes the videos by DFRD to segment road areas. The outputted road images are then forwarded to the HDD algorithm to detect hazardous road damages, including big holes, blowups, and fractures. The edge warns all the subscribers within its communication range once it detects any hazardous road damages. Afterward, the edge sends the segmented road images to the cloud for more detailed analysis, i.e., detect and localize all road damages, including both hazardous (e.g., big holes, blowups, and fractures) and non-hazardous damages (e.g., cracks, potholes, and patches) by MDD for long-term road maintenance and management. More details about the proposed approaches, i.e., DFRD, HDD, and MDD, are elaborated in the following sections. The meaning of the related notations is listed in Table 2.2.

2.5 Methodologies

This section presents detailed explanation of the proposed algorithms in EcRD, i.e., DFRD, HDD, and MDD.

2.5.1 Deep Feature Based Road Detector

To efficiently implement the EcRD framework, I proposed a road detection algorithm named DFRD to remove noisy backgrounds. Instead of directly using image pixels as inputs, superpixels are chosen since they preserve more compact object features (e.g., color and texture) and are less sensitive to noises than pixels [31]. Therefore, the road detection task is projected to the superpixel-level image classification problem.

The Simple Linear Iterative Clustering (SLIC) [32] method is employed as the superpixel generator since it is simple yet can generate high-quality superpixels. Given an image M with N pixels $M =$

Table 2.2: Summary of Notations.

Notation	Description
$M = \{m_1, \cdots, m_N\}$	An image with N pixels
M_i	i_{th} image in $\{M_1, \cdots, M_N\}$
m_i	i_{th} pixel of image M
N	Number of pixels
K	Number of superpixels, i.e., $K = 50$
sp_i	i_{th} superpixel
$\phi_l(x; \theta)$	Feature extractor
N_x	Number of rows of image M
N_y	Number of columns of image M
N_g	Number of gray levels and I set $N_g = 16$
(L_x, L_y)	Location of a pixel in image M
$F(i, j)$	(i, j)-th entry in a normalized GLCM matrix
μ_i, μ_j	Row and column-wise means of a GLCM matrix
σ_i^2, σ_j^2	Row and column-wise variances of a GLCM matrix
H	Weighted sum of the five features in Eq. (2.6)
$S(H)$	Sigmoid of H defined in Eq. (2.8)
y	Labels of images. $y \in \{0, 1\}$
T_h	Threshold of H and I choose $T_h = 0.135$
w_1, w_2, w_3, w_4, w_5	Weights of the features in Eq. (2.6)
A_t	Domain of road damage images
B_t	Domain of background images
t	Damage types $t \in \{crack, patch, pothole\}$
C	Domain of road scene images without damages
G_a^t	Generator A of damage type t
G_b^t	Generator B of damage type t
G_c^t	Generator C of damage type t
D_a^t	Discriminator A of damage type t
D_b^t	Discriminator B of damage type t
a_i^t	Training samples from Domain A_t
b_j^t	Training samples from Domain B_t
N_a	Number of training samples from Domain A_t
N_b	Number of training samples from Domain B_t
(x_1, y_1, x_2, y_2)	Coordinates of a image patch in a full image

$\{m_1, m_2, \cdots, m_N\}$, the image pixels can be successfully clustered into K equally sized superpixels $M = \{sp_1, sp_2, \cdots, sp_K\}$. Afterward, features of the superpixels are extracted and classified to find the superpixels that belong to road class[1]. Unlike most of the existing researches (e.g., [17, 33]), I leverage the high feature extraction ability of DCNNs and the astonishing advantages of transfer learning for feature extraction, instead of using low-level human-designed features such as Histogram of Oriented Gradients (HoG) [34], or training an end-to-end DCNN with a large precisely labeled dataset from scratch. More specifically, a DCNN model pre-trained on the

[1] An image is classified as road class when it has more than 80% pixels located in the road area.

ImageNet dataset is utilized to extract features of superpixels. Each hidden layer of the network can be viewed as a feature extractor $\phi_l(x;\theta)$, where x is the input, and θ is the parameters. Lower layers respond to general properties such as edge, corner, and color, while higher layers show higher-level properties. Since the task is simple and the dataset is small, a high-level feature extractor $\phi_8(x;\theta)$ is chosen for feature extraction. To classify the deep features extracted from superpixels, a Linear Support Vector Machines (LSVM) [35] is selected to distinguish road superpixels from non-road superpixels after testing and comparing eight commonly used classifiers, i.e., Naive Bayes, Radial Basis Function SVM, K Nearest Neighbor, Decision Tree, Artificial Neural Network, Random Forest, AdaBoost, and Quadratic Discriminant Analysis (QDA). After getting the superpixels belonging to the road class, the superpixels are merged for each unique image to form a complete and recognizable road region. The superpixels that belong to the road type are iteratively merged to obtain the segmented road. The workflow of DFRD is presented in Figure 2.2.

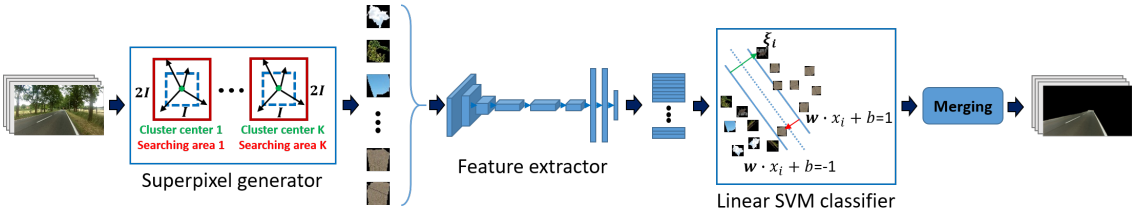


Figure 2.2: Deep feature based road detector.

2.5.2 Hazardous Damage Detector

Hazardous road damages are detected on edges, and warning information is sent to the users within their communication ranges. To achieve this goal, approaches that perform well on edges need to be developed. Although DCNNs are very popular nowadays, for example, [18], a large and precisely labeled dataset is required for training which is time-consuming, labor-intensive, and costly. Besides, it is not easy to collect massive images of hazardous road damages in the real world. Hence, the unsupervised HDD algorithm for hazardous road damage detection is developed, which has low latency, low cost, and high accuracy.

An image with N_x rows and N_y columns pixels is defined as M. Suppose that the gray level appearing at each pixel is quantified to N_g levels (I set $N_g = 16$ in the experiments). The location of a pixel in image M is defined as (L_x, L_y). An image M can be reformulated as a function that assigns some gray level values to each pixel's location at (L_x, L_y). A relative gray-level frequency matrix reveals the texture information. Each element is the relative frequency of two neighboring pixels. Such matrices are a function of the angular relationship and distance between the neighboring pixels. Gray Level Co-occurrence Matrix (GLCM) [36]-based feature representations are utilized in DFRD. I define $F(i, j)$ as the (i, j)-th entry in a relative co-occurrence matrix (i.e., GLCM). The features are defined as follows:

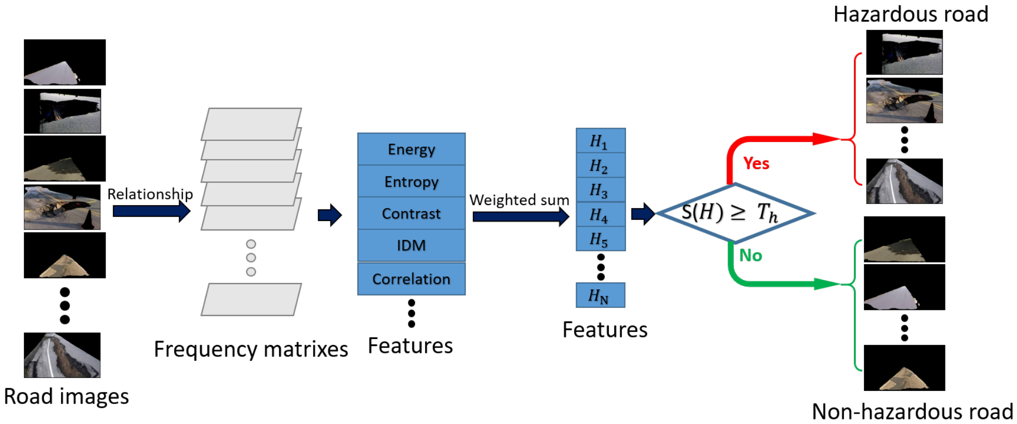


Figure 2.3: Hazardous damage detector.

1. **Energy:**

$$Energy = \sum_i \sum_j F(i,j)^2. \tag{2.1}$$

Energy measures the homogeneity of an image. In homogeneous scenes, only a few gray levels present and the values of $F(i,j)$ is relatively high thus having higher energy.

2. **Entropy:**

$$Entropy = -\sum_i \sum_j F(i,j) \log(F(i,j)). \tag{2.2}$$

Entropy measures the randomness of the amount of information contained in an image. It shows the complexity of the image. Entropy is high when all values in the relative frequency matrix are similar or pixel values show a great randomness.

3. **Contrast:**

$$\text{Contrast } = \sum_i \sum_j (i-j)^2 F(i,j). \tag{2.3}$$

Contrast reflects the clarity and regularity of the texture. The contrast value is high when the texture is clear and regular. In addition, it measures the difference of color or hue between images.

4. **Inverse Differential Moment (IDM):**

$$IDM = \sum_i \sum_j \frac{F(i,j)}{1+(i-j)^2}. \tag{2.4}$$

IDM is also known as local homogeneity. It is influenced by the homogeneity of the image. Because of the weighting factor $\left(1+(i-j)^2\right)^{-1}$, IDM gets smaller contributions from inhomogeneous areas ($i \neq j$). Therefore, IDM is relatively low for inhomogeneous images, and relatively high for homogeneous images.

5. **Correlation:**

$$Correlation = \sum_i \sum_j F(i,j) \left[\frac{(i-\mu_i)\ (j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right], \tag{2.5}$$

where (μ_i, σ_i^2) and (μ_j, σ_j^2) are respectively row and column-wise mean-variance pairs of the co-occurrence matrix. Correlation measures the linear dependency of grey levels of neighbouring pixels. It turns out that nearby pixels are highly correlated than more distant pixels.

Algorithm 1: HDD algorithm

```
Input  : Road images $\{M_1, \cdots, M_N\}$
Output : Road images with hazardous damages $\{M_1, \cdots, M_{N_h}\}$
for each road image $M_i$ do
    calculate relative co-occurrence matrix
    calculate five features of the road image
    calculate weighted sum of five features by Eq. (2.6)
    if $S(H_i) \geq T_h$ then
        $y_i = 1$
    else
        $y_i = 0$
    end
end
```

The weighted sum of these five features is defined as H, as shown in Eq. (2.6). The images are classified as hazardous (i.e., label $y = 1$) if $S(H)$ is greater than or equal to the threshold T_h, where sigmoid function S is defined by Eq. (2.8). The images are classified as non-hazardous (i.e., label $y = 0$) if H is lower than the threshold T_h, as presented in Algorithm 1. I find $T_h = 0.135$, $w_1 = -3.786$, $w_2 = 1.897$, $w_3 = 3.009$, $w_4 = -0.0316$, and $w_5 = -1.879$ are optimal from the experiments. The detailed structure of HDD is illustrated in Figure 2.3.

$$\begin{aligned} H = {} & w_1 \times \text{Energy} + w_2 \times \text{Entropy} + \\ & w_3 \times \text{Contrast} + w_4 \times \text{IDM} + \\ & w_5 \times \text{Correlation}, \end{aligned} \tag{2.6}$$

$$y = \begin{cases} 1, & S(H) \geq T_h, \\ 0, & S(H) < T_h, \end{cases} \tag{2.7}$$

$$S(H) = \frac{1}{1 + e^{-H}}. \tag{2.8}$$

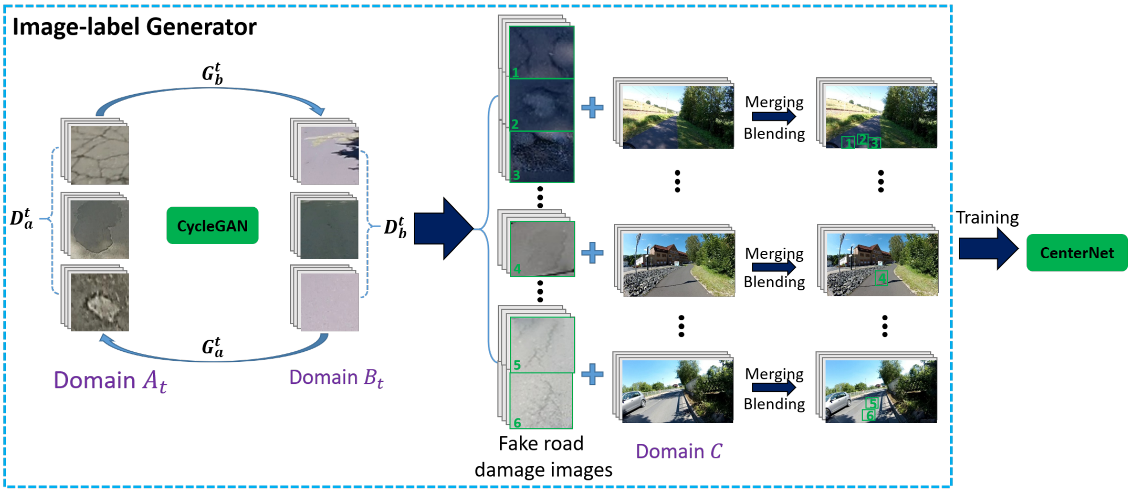


Figure 2.4: Multi-type damage detector.

The computational complexity of HDD mainly depends on the calculation of the five GLCM features, which is determined by the number of rows and columns of an image. The GLCM feature calculation's computational complexity in HDD is around $O((N_x \times N_y)^2)$, where N_x and N_y are number of pixels in the row and column of an image. Therefore, the computation complexity of HDD is about $O((N_x \times N_y)^2)$.

2.5.3 Multi-type Damage Detector

DCNNs can be utilized for the multi-type road damage detection task on the cloud server due to its advantages of large storage and high computational power. An end-to-end DCNN model (i.e., CenterNet [29]) is employed as an example to show the potential of DCNNs in combination with the proposed image-label generator on the multi-type road damage detection task. Three types of non-hazardous road damages, i.e., cracks, potholes, and patches, are selected since they are the most common road damages. Although DCNNs have achieved great success in many applications (e.g., [37]) due to their high feature extraction ability. However, it can not be directly used for the multi-type road damage detection task since they inherently require large-scale precisely labeled datasets while data collection is difficult and data labeling is time-consuming and costly. A novel image-label generator is proposed to tackle this issue, incorporating a powerful image translation technique, i.e., CycleGAN [30], to automatically generate road scene images (i.e., front view road scene images as shown in Figure 2.5) with realistic road damages and their corresponding labels. Since CycleGAN is a state-of-the-art image generation technique that does not require pairwise input images, and the preparation of pairwise images takes time. Additionally, the road damage images generated by CycleGAN are clearer and more realistic than other GAN-based models, for example, Pixel2Pixel [38]. Therefore, in the generator, CycleGAN is utilized to generate road

Algorithm 2: MDD algorithm - Image-label generator

```
Input  : Road damage images $\{M_1^t, \cdots, M_{N_a}^t\}$, road background images $\{M_1^t, \cdots, M_{N_b}^t\}$,
         and road scene images without damages $\{M_1, \cdots, M_{N_c}\}$
Output : Generated road scene images with fake damages and labels $\{M_i, (x_1^i, y_1^i, x_2^i, y_2^i, t)\}$
for number of damage types $t \in \{crack, patch, pothole\}$ do
    for number of images from domain $A_t$ and $B_t$ do
        train CycleGAN with cycle consistency loss and adversarial loss
    end
    for number of images from domain $C$ do
        for number of desired damages per image do
            Random crop a road part from an image
            Translate to a road damage image by the trained $G_a^t$
            Put the fake damage image back to the image
            Blend the fake damage image with the background of the image
            Save the coordinates $(x_1^i, y_1^i, x_2^i, y_2^i)$ and label $t$ as Ground-Truth bounding box of
              the damage in the origin image
        end
    end
end
```

damage images, i.e., images only with damages without cluttered backgrounds as illustrated in Figure 2.5. Road scene images without damages are much easier to collect; therefore, they are utilized for images and labels generation. The generated road damage images are used to synthesize new road scene images with damages.

The images and labels generation approach (see Figure 2.4 and Algorithm 2) can be formulated as a domain mapping problem, where domain A_t is a set of road damage images and domain B_t is a set of road background images as shown in Figure 2.4, where t represents damage types $t \in \{crack, patch, pothole\}$, and domain C is a set of road scene images without damages. I have a pair of images from domain A_t and domain B_t for each type of damage. I define the mapping functions between A_t and B_t as, $G_b^t : A_t \rightarrow B_t$ and $G_a^t : B_t \rightarrow A_t$, with cycle consistency loss and adversarial loss constraints, further explanations of which can be found in [30]. I define the mapping function from B_t to C_t as, $G_c^t : B_t \rightarrow C_t$. Given training samples $\{a_i^t\}_{i=1}^{N_a}$ and $\{b_j^t\}_{j=1}^{N_b}$ from two domains A_t and B_t, where $a_i^t \in A_t$ and $b_j^t \in B_t$, N_a and N_b are the number of training samples from domain A_t and domain B_t, respectively. Each damage type has two generators (G_a^t and G_b^t) and two adversarial discriminators (D_a^t and D_b^t). G_a^t is used to generate images that are similar to images from domain A_t and G_b^t is utilized to generate images that are similar to images from domain B_t. Similarly, D_a^t aims to distinguish between fake images generated by G_a^t and real images from domain A_t, while D_b^t aims to differentiate between fake images generated by G_b^t and real images from domain B_t. After training, the generators $\{G_a^{crack}, G_a^{patch}, G_a^{pothole}\}$ are applied to generate fake images with images from domain B_t. Then, road background images (as shown in Figure 2.5 (d)) cropped from domain C are mapped to domain A_t by the trained generator G_a^t.

Afterward, the translated fake road damage images are put back to the original scene images c and blend it with the background by the Opencv seamlessClone() method to make it more realistic. Meanwhile, the label t and coordinates (x_1, y_1, x_2, y_2) of the generated road damage images are saved as their annotations. Multiple fake damages can be applied to the same road scene image to have multiple damages from different types. More details are shown in Algorithm 2. After getting a large road damage dataset with annotations, an anchor-free DCNN model, CenterNet [29], is utilized for multi-type road damage detection. Unlike anchor-based detectors, CenterNet uses key points to detect objects, saving a lot of computational resources while acquiring competitive performance. The Deep Layer Aggregation (DLA) is used as the backbone to predict damages from input road images.

In MDD, the image generator's computational complexity is $O(tNv)$, where t, N, and v are the number of damage types, number of images for each damage type, and average number of desired damages for each image, respectively. The setting is, $t = 3$, $N = 760$, and $v = 3$, in the experiments. The image generator is used offline to produce training images efficiently; thus, it does not influence the latency of MDD. The computational complexity of MDD depends on the network architecture and the number of parameters. A lightweight CenterNet model with 34 layers is employed for the multi-type damage detection in EcRD, which has a low computational cost.

2.6 Experiments and Evaluation

In this section, the experimental setup and dataset are introduced first. Then, The comparison baseline algorithms are described. Following this, evaluation metrics and experiment results are introduced.

2.6.1 Experimental setup and dataset

A laptop (HP ZBook 15 G5, 64-bit Ubuntu 18.04 Operating system, 32GB RAM, Intel Core i7-8850H CPU with 2.60GHz×12) is used as the edge computer to simulate an Road Side Unit (RSU), and a high-performance server (Ubuntu 16.04 LTS system, 125.8 GB of RAM, 5.93 TB of hard disk, and 8 GTX 1080 Ti GPUs) is used as the cloud server. The client application is implemented using Python 3.6.

Road scene videos from a driver's front viewpoint are recorded by a cheap smartphone's camera (VG30+) mounted on a vehicle with a resolution of 1280×720 pixels as an example of data acquisition from IoT devices. The recorded videos are then uploaded to the edge server. After receiving road scene images from IoT devices, the edge server first segmented the road part out of the scene images by the proposed DFRD model, and then hazardous road damages are detected by the developed HDD model. Once hazardous road damages are detected, warning information will be sent to edge subscribers (e.g., drivers, pedestrians, and road administrations). After that, the segmented road images are sent to the cloud server to further analyze road conditions for

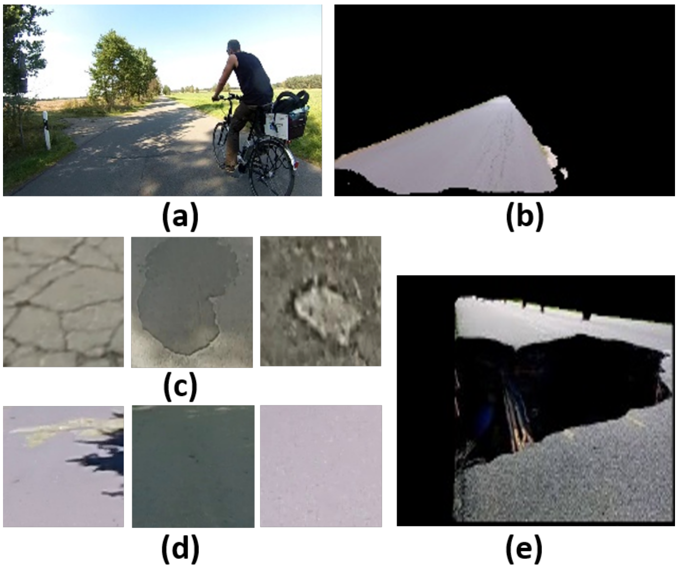


Figure 2.5: Data examples, (a) Front-view road scene image (b) Road image (c) Road damage images (d) Road background images (e) Hazardous road damage image.

long-term road network maintenance and rehabilitation. Multiple types of road damages are detected by the MDD model on the cloud in this phase. More details about the models can be found in Section 2.4.

More than 75 videos are collected covering different scene types (rural and urban), and various illumination conditions (dark, bright, and shadows), and the duration of each is more than 49 minutes which enables us to get a variety of road images. The images are automatically extracted from the videos by every 15 frames (or 0.5 sec), and the size of the extracted frames/images is 1280×720 to build the training sets. Figure 2.5 shows some examples of front-view road scene images, segmented road images, road damage images, road background images, and hazardous road damages. There are 615 frames selected from the extracted frames to train the classifiers of DFRD. Superpixels from them are automatically labeled according to the color of roads and checked manually. Further, to train the generator in MDD, 530 images with damages and 530 images without damages for each type are cut from the extracted frames. Then, square pavement areas (a random number from 1 to 6) from each of 1031 new frames without damages are randomly cropped and translated into rectangles with fake damages by the trained generator and then put back to the original frame and save the coordinates and damage types as Ground-Truth labels. To test the proposed approaches, i.e., DFRD, HDD, and MDD, and the EcRD framework, a video with 1031 frames including 200 generated non-hazardous road damages is built. It also consists of 200 frames with hazardous road damage from the Internet and 631 without damage from the collected dataset. All the damages are placed and blended in the video to make it more realistic. Figure 2.5

shows some examples of front-view road scene images, segmented road images, road damage images, road background images, and hazardous road damage images. Except for the collected dataset, benchmark datasets (KITTI and COCO) are also employed to evaluate the models.

2.6.2 Baselines

In this work, the proposed EcRD framework with the efficient DFRD, HDD, and MDD unified by a novel edge-cloud computing framework for both rapid hazardous road damage detection and warning and detailed long-term road condition monitoring.

To evaluate the performance of EcRD, I compare the performance of EcRD with the following baselines.

1) EcRD-only-edge and EcRD-only-cloud: They are two variants of EcRD, EcRD-only-edge deploys both hazardous road damage detection and multi-type road damage detection tasks on edges, while EcRD-only-cloud deploys both of them on the cloud. The performances of both EcRD-only-edge and EcRD-only-cloud are compared to EcRD to evaluate the effectiveness of the edge-cloud computing framework.

2) EcRD-no-DFRD: The EcRD-no-DFRD system is also a variant of EcRD framework with hazardous road damage detection and multi-type road damage detection subtasks; however, without image pre-processing by the DFRD algorithm. The performance of this baseline can prove the necessity of the DFRD module in EcRD.

3) EcRD-no-aug: The EcRD-no-aug system is a variant of the EcRD system without the proposed image-label generator, which automatically generates road scene images with multiple damages and corresponding labels. By comparing EcRD-no-aug with EcRD, the effectiveness of the developed generator can be evaluated.

State-of-the-art road segmentation approaches for DFRD evaluation:

1) HIM [39]: It is a graphical method that uses image decomposition to encode relational and spatial information instead of using structured predictions for road segmentation.

2) ProbBoost [40]: It is a probabilistic distribution approach for urban road detection. This method uses a Joint Boosting algorithm to detect semantic information of the road.

3) CB [15]: It is a road detector based on color, texture, and structure features and a Multiple Layer Perception (MLP) classifiers. Additionally, contextual blocks are used to provide contextual information for better road detection.

4) CN24 [33]: The CN24 is a DCNN learned to distinguish different image patches. Spatial information of image patches is incorporated into the network to enhance the road detection model.

5) StixelNet II [41]: It is a DCNN model with vertical pool layers to improve the performance of road segmentation.

6) RBNet [19]: It is a DCNN modeled by a Bayesian framework that can jointly estimate the road surface and its boundaries.

Baselines for the evaluation of the proposed HDD:

1) SMT [42]: It is a method utilizing saliency maps obtained by DRFI [42] as input instead of original images, then the saliency rate (i.e., the percentage of salient pixels) is measured to detect abnormal images.

2) GLCM-C [43]: It is a texture content-based method using GLCM and K-Means Clustering algorithms for image pixel differentiation.

2.6.3 Evaluation Metrics

The following metrics are employed to estimate the performance of EcRD and the baseline approaches.

1. Precision: It represents how correct a road damage detection model is, by measuring the proportion of the true positive predictions out of the total number of the model's positive output.

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.9}$$

   where TP is True Positives and FP is False Positives.

2. Recall: It is a measure of how many instances are identified correctly and is given as follows:

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2.10}$$

   where FN is False Negatives.

3. Accuracy: It measures how correct a road damage detection system operates by the percentage of TP and TN along with the number of false alarms and true predictions in terms of FP, FN, TP, and TN that the system produces [44] and is shown as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}, \tag{2.11}$$

   where TN is True Negatives.

4. F1-score: It is the harmonic mean of precision and recall and is presented as follows:

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}. \tag{2.12}$$

5. FNR: It is the False Negative Rate measuring the proportion of FN with respect to the sum of TP and FN as shown in the following equation:

$$\text{FNR} = \frac{FN}{TP + FN}. \tag{2.13}$$

6. FPR: It is short for False Positive Rate that represents the percentage of FP along with the number of TN and TP which is calculated as follows:

$$\text{FPR} = \frac{FP}{TN + FP}. \tag{2.14}$$

7. AP, AP40, AP50, AP75: According to COCO evaluation matrix [45], the AP40 is the average precision at IoU = 0.40, where TP is considered if IoU ≥ 0.4 else FP. Similarly, AP50 is the average precision at IoU = 0.50. and AP75 is the average precision at IoU = 0.75. The AP refers to the average precision at IoU=.50:.05:.95.

8. Runtime: It is defined by per image/frame inference time taken for road damage detection algorithms.

9. Training-time: It measures the amount of time taken by a model to train parameters. Training-time for a DCNN ranges from several days to several weeks depending on the size of the model, the resolution of the training images, and the server it runs on. To compare it to the model, I assume the training-time of all DCNN related models is approximately three weeks, i.e., 21 days, on a single NVIDIA GTX Titan GPU following [46].

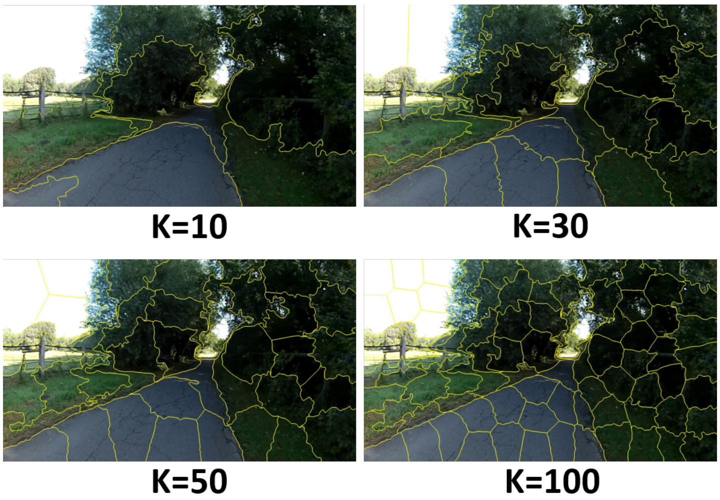
10. Latency: It is measured by the waiting time of users to receive the emergency response from servers (edge or cloud). More precisely, latency is the total data processing time (i.e., runtime) and transmission time per frame/image. Moreover, I assume the data transmission time from devices to edge is 0 because edges are close to devices, and the transmission time from devices to cloud is set to 2.452 seconds per $100\,\text{KB}$ file (per frame/image is around $100\,\text{KB}$) following [47].

2.6.4 Evaluation Results

In this subsection, the performance of the developed models and their corresponding baselines is evaluated. Following that, EcRD's evaluation results are introduced by comparing it with its baselines.

2.6.4.1 DFRD Evaluation Results

DFRD consists of three main parts, superpixel generator, feature extractor, and linear SVM classifier, as shown in Figure 2.2. The desired number of superpixels K in the superpixel generator


K=10

K=30

K=50

K=100

Figure 2.6: The effect of K of the superpixel generator in DFRD.

Table 2.3: DFRD Evaluation Results on KITTI (Um-road) Benchmark Dataset.

Model	F1-score (%)	Precision (%)	Recall (%)	FPR (%)	FNR (%)	Runtime (s)	Training-time (days)
HIM [39]	90.07	90.79	89.35	4.13	10.65	7	-
ProbBoost [40]	87.48	85.02	90.09	7.23	9.91	150	-
CB [15]	88.89	87.26	90.58	6.03	9.42	30	21
CN24 [33]	86.32	87.80	84.89	5.37	15.11	30	21
StixelNet II [41]	94.88	92.97	**96.87**	4.04	**3.13**	1.2	21
RBNet [19]	**94.97**	**94.94**	95.01	**2.79**	4.99	0.18	21
DFRD (own)	92.74	95.04	90.55	4.73	9.45	**0.00072**	**16 seconds**

significantly affects the performance of DFRD. The experiments show that the quality of superpixels is low when $K < 50$; thus, the road segmentation result is bad. While when K is very high, e.g., $K > 100$, the quality of superpixels is good; however, the computation time and storage requirement are very high. As shown in Figure 2.6, the quality of superpixels is not sufficient when $K = 10$ and $K = 30$ since some of them contain both road and non-road. The superpixels are good when $K = 100$, but it is resource-inefficient as it needs more processing time and storage space. In addition, when $K > 200$, unique features of roads are not preserved well, resulting in low road segmentation performance. Therefore, the desired number of superpixels is set as $K = 50$.

Furthermore, to prove the effectiveness of DFRD, I compare it with the state-of-the-art road segmentation models on the KITTI (um-road) benchmark dataset concerning the precision, recall, F1-score, FPR, FNR, runtime, and training-time. These evaluation matrixes are defined in Section 2.6.3. As illustrated in Table 2.3, five published monocular vision-based road segmentation algorithms are tested on KITTI (um-road) dataset. The results show that DFRD achieves competitive results in most of the evaluation metrics while with much less runtime and training time on the CPU of a normal laptop. Additionally, DFRD also achieves 93.65% F1-score on the dataset. Therefore, DFRD

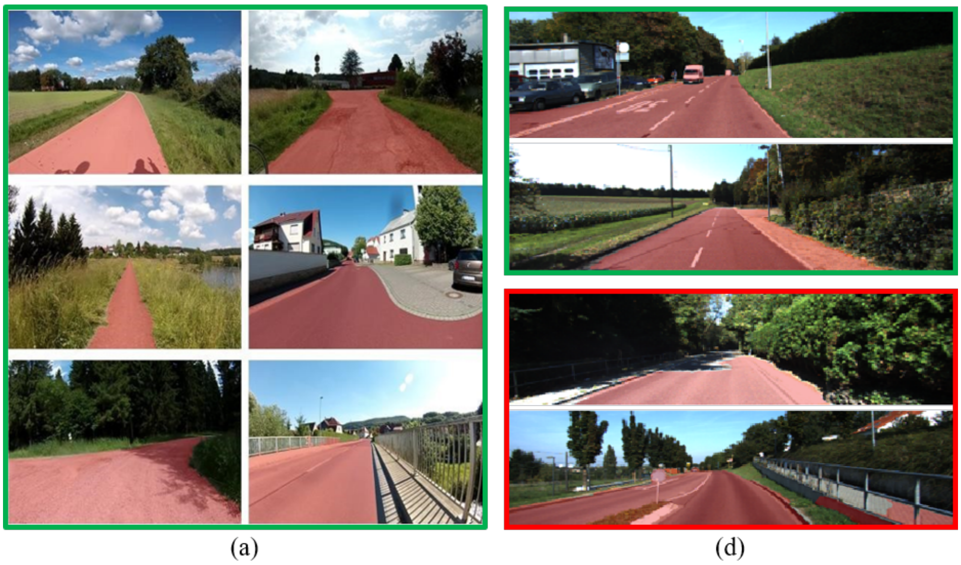


Figure 2.7: Experimental results of DFRD, (a) own dataset (b) KITTI (um-road) dataset.

Table 2.4: HDD Evaluation Results.

Model	F1-score (%)	Precision (%)	Recall (%)	Runtime (s)
SMT [42]	44.95	45.61	45.92	12.12
GLCM-C [43]	91.95	93.10	92.00	0.033
HDD (own)	**98.70**	**98.65**	**98.78**	**0.0036**

is a qualified component of EcRD framework. Image examples of DFRD's results are illustrated in Figure 2.7.

2.6.4.2 HDD Evaluation Results

I compare HDD with two approaches, i.e., SMT [42] and GLCM-C [43], with regard to F1-score, precision, recall, and runtime. The results are presented in Table 2.4. Some examples of the detected hazardous images are given in Figure 2.8. The results show that GLCM features are more useful than saliency maps for hazardous damage detection. Besides, although GLCM-C is simpler for not requiring thresholds given by humans, the F1-score is lower and takes more time than HDD. More details can be seen in Table 2.4. The results clearly show that HDD outperforms SMT and GLCM-C regarding F1-score, precision, recall, and runtime. Specifically, the F1-score obtained by HDD is about 53.75% higher compared to SMT and around 6.75% higher compared to GLCM-C, respectively. Moreover, SMT performs the worst in runtime since it needs 12.12 s to process just one image while GLCM-C and HDD only need around 0.003 s. Therefore, HDD satisfies the need for real-time accurate hazardous road damage detection and warning in EcRD. Image examples of

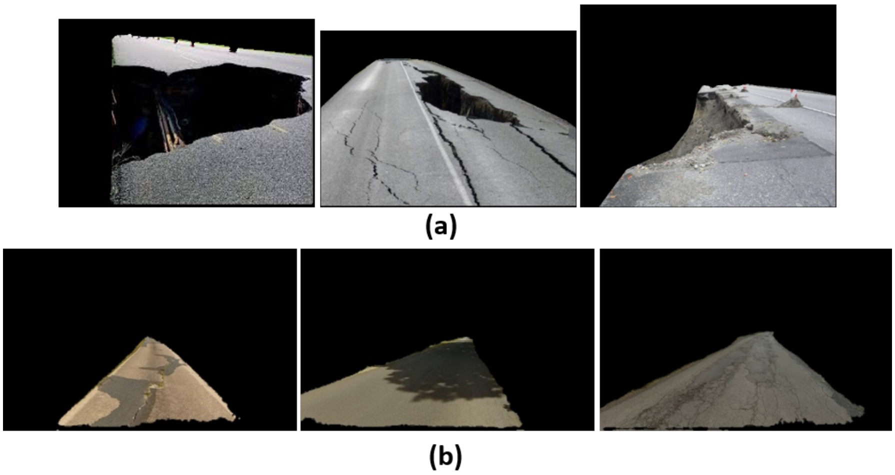


Figure 2.8: Experimental results of HDD, (a) correctly detected samples (b) wrongly detected samples.

Table 2.5: Evaluation Results for MDD Basic Model Selection on COCO test-dev Dataset.

Model	Backbone	AP(%)	AP50 (%)	AP75 (%)	Runtime(s)
MaskRCNN [48]	ResNeXt-101	39.8	**62.3**	43.4	0.09
YOLOv3 [49]	DarkNet-53	33	57.9	34.4	0.05
CenterNet [29]	DLA-34	**41.6**	60.3	**45.1**	**0.035**

HDD's results are presented in Figure 2.8.

2.6.4.3 MDD Evaluation Results

MDD is improved based on a state-of-the-art deep learning model named CenterNet [29]. It is because the CenterNet model has competitive results and the lowest runtime (0.035 seconds) compared to other models, which is shown in Table 2.5. The detection results of CenterNet from Table 2.6 prove the high road damage generation ability of our image-label generator in MDD. It can be seen that the proposed MDD model achieves about 139% improvement for AP50 compared to CenterNet. Further, Figure 2.9 also clearly shows that MDD generates clear and realistic road damages placed on real world scene images. Besides, each image has one or more fake damages marked with green rectangles. The labels of generated images are checked by three experts, and mislabeled (less than 5%) samples are corrected to measure the correctness of the labels. Image examples of MDD's results are shown in Figure 2.10.

Table 2.6: MDD Evaluation Results.

Model	Backbone	AP(%)	AP50 (%)	AP75 (%)	Runtime(s)
CenterNet [29]	DLA-34	21.6	41.2	16.5	**0.035**
MDD (Ours)	**DLA-34**	**79.2**	**98.6**	**91.1**	**0.035**

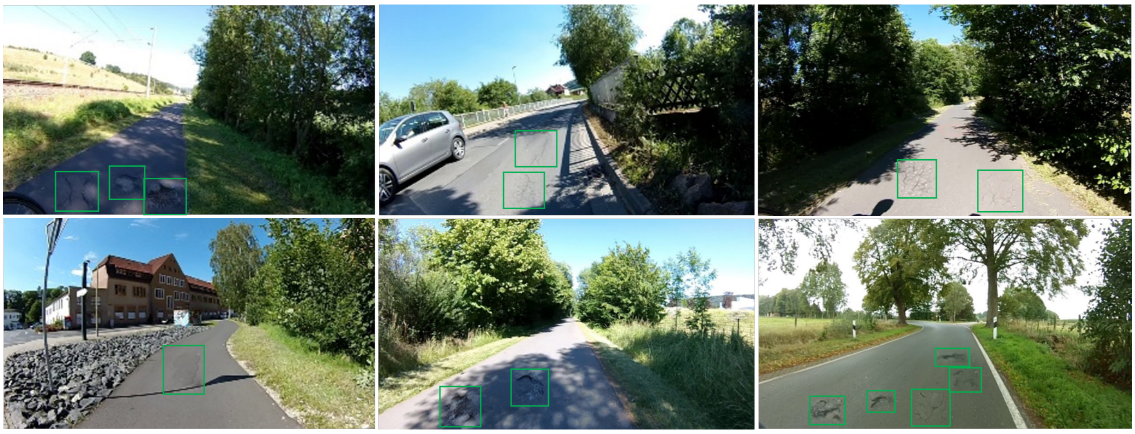


Figure 2.9: Samples of generated images with fake damages.

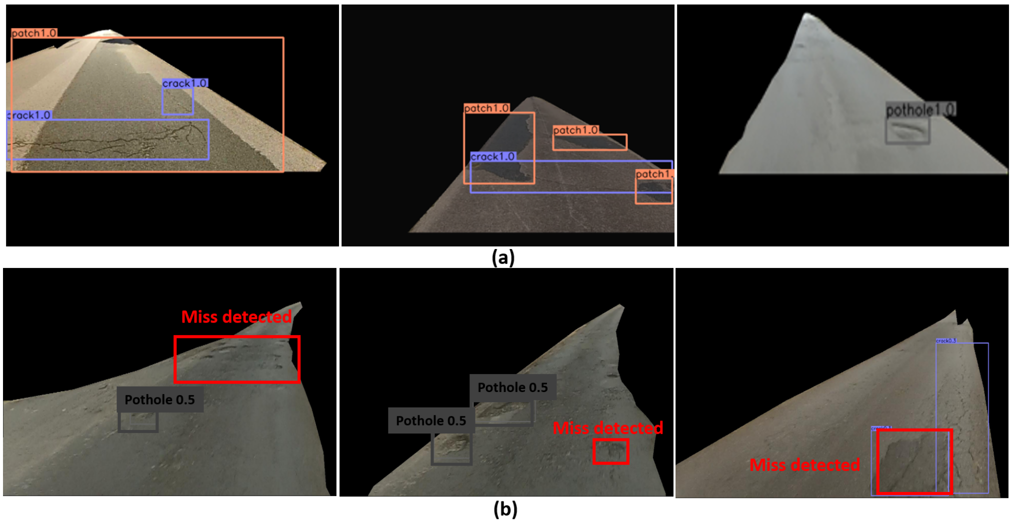


Figure 2.10: Experimental results of MDD, (a) Good samples (b) Bad samples.

Table 2.7: EcRD Framework Evaluation Results.

Model	HDD F1-score(%)	HDD Latency(s)	MDD AP50(%)	ERU
EcRD-only-cloud	**92.43**	2.4952	**94.31**	Small
EcRD-only-edge	**92.43**	0.0043	**94.31**	Large
EcRD-no-DFRD	34.34	**0.0036**	82.97	Small
EcRD-no-aug	**92.43**	0.0043	40.84	Small
EcRD	**92.43**	0.0043	**94.31**	Small

2.6.4.4 EcRD Evaluation Results

To my best knowledge, it is the first research on edge-cloud computing for smart road damage detection and warning. The performance of EcRD by comparing with the baselines (see Section 2.6.2) are presented. Since EcRD is composed of two tasks: hazardous road damage detection task on edges for the emergency response and multi-type road damage detection task on the cloud for long-term road maintenance and management, the performance of EcRD in terms of these two tasks are evaluated.

The comparison results are illustrated in Table 2.7, where ERU is short for edge Resource Utilization. Overall, the results show that EcRD outperforms its variants. Specifically, the F1-score of EcRD is about 169% times higher than EcRD-no-DFRD for the hazardous road damage detection task, while EcRD's AP50 is around 14% times higher than EcRD-no-DFRD about the multi-type road damage detection task, which proves the importance of DFRD on both tasks. In addition, the F1-score achieved by EcRD is about 131% times higher than EcRD-no-aug for multi-type road damage detection tasks, which shows that our image generator provides large-scale and high-quality images with labels for multi-type road damage detection. Concerning latency, EcRD-only-cloud has around 579 times higher latency than EcRD and EcRD-only-edge for hazardous road damage detection tasks. With this high latency, the users may be unable to receive life-threatening warning information in time. Therefore, mission-critical and delay-sensitive tasks like hazardous road damage detection tasks should be deployed on edges to ensure high QoS for the emergency response. However, although edge computing significantly reduces the latency, deploying both HDD and MDD on edges (i.e., EcRD-only-edge) does not achieve satisfactory performance. Multi-type road damage detection results are not very urgent for road users; therefore, the low latency gained by deploying MDD on edges does not improve the QoS. However, it requires massive storage space and very high computational power, which may negatively affect the performance of edge servers. Therefore, the latency of MDD is not listed in Table 2.7. Furthermore, only uploading segmented road images to the server and only saving images that are detected with damages can significantly reduce the storage cost for both edge and cloud servers. Therefore, one can conclude that with EcRD, people can not only efficiently use the available resources and produce highly accurate and fast hazardous road damage detection and warning for road users, but also can effectively detect multi-type road damage for long-term road management with very limited human-labor, time, and cost.

2.7 Conclusion

This chapter proposes EcRD framework to tackle the long latency and high data collection and labeling costs of existing road damage detection systems. EcRD leverages edge-cloud computing for real-time hazardous road damage warning and long-term multi-type road damage management. The latency of road danger warnings in EcRD is reduced by deploying road danger detection models on edges instead of the cloud. In EcRD, the DFRD model and the HDD model are developed for fast and accurate road damage danger detection and warning, which is a hand-craft feature-based algorithm that is fast for road damage danger detection. Also, MDD is introduced for multi-type road damage classification on the cloud. The communication cost of EcRD is reduced by only transmitting road images (with only road areas) instead of the original images. To protect data privacy, EcRD segments out backgrounds and only leaves road areas. Notably, EcRD is 579 times faster than cloud-based solutions.

Chapter 3

FedRD: Privacy-preserving Adaptive Federated Learning Framework for Intelligent Hazardous Road Damage Detection and Warning

3.1 Motivation

Similar to EcRD in Chapter 2, this chapter also studies road damage danger detection and warning. EcRD proposes an edge-cloud computing-based framework for low-latency road damage detection and warning by deploying the detection model on edges. Unlike sensor-based systems [5,6], EcRD detects hazardous road damages before vehicles hit them, which is much safer. However, in EcRD, drivers can only receive warnings about road damages within a small area due to the limited communication range of edges. For instance, RSUs are usually used as edges in intelligent transportation systems. However, their communication range is only around 1000 meters. Vehicles covered by an RSU can only receive road damage information within this communication range. Moreover, the detection performance of edges is strictly constrained by the amount of data collected. Some edges may fail to detect road damages if they do not have enough data for training. Even if edges or devices can directly collect pre-trained models from the cloud, it requires direct data sharing from edges or devices to the cloud, which has a high privacy leakage risk. Also, the computation power and storage space of IoT devices are limited. Despite the importance, there is *no* existing work that addresses these problems in this field. Furthermore, data collected from users' devices on edges contain massive private information, for example, people's faces, locations, and license plate numbers. Only few researches [50,51] considered the privacy problem for road condition inspection by using cryptographic techniques. However, they have a high computation cost for key generation, authentication, encryption, and decryption. Besides, *no* existing work considers the privacy problem inside image/video data in this field.

To tackle these issues, in this chapter, *FedRD*: a novel privacy-preserving edge-cloud-based federated learning framework is designed for intelligent hazardous road damage detection and warning. In FedRD, a new map construction approach is introduced. It provides drivers/users a hazardous road damage warning map, which has hundreds or even thousands of times wider coverage than EcRD [7]. Additionally, the FL strategy [52] is utilized to collaboratively learn hazardous road damage information from decentralized edges without direct data sharing between edges and the cloud. It improves the model's robustness and protects people's privacy from untrusted cloud servers. Different from [52–54], the developed AFed ensures high detection performance by only aggregating qualified models selected from top K models received from edges. In this way, high detection performance can be guaranteed within limited computation iterations. Also, it can prevent data poisoning attacks since the local models trained on poisoned data will not be used for global aggregation due to their low performance on the shared testing set on the cloud. Although FL protects privacy from untrusted clouds by keeping data locally on edges, there is still a high privacy leakage risk because FL does not protect privacy from untrusted edges. Additionally, private data can also be recovered only by shared parameter gradients of FL [55]. Hence, FL technique [56] is utilized to fill in this gap. Unlike [57–59], the proposed IDPP preserves privacy on users' devices before uploading data to untrusted edges, which is more private. Moreover, it has $3/4$ less computation cost because noise is added to pixelized images instead of original images. To the best of my knowledge, it is the *first* work to propose a privacy-preserving edge-cloud-based federated learning framework for smart hazardous road damage detection and warning addressing the problems of existing systems (e.g., long latency, small coverage, model robustness, and privacy).

3.2 Contributions

The contributions of this chapter, which were published in the Future Generation Computer Systems Journal in 2021 [8]), include:

- A novel edge-cloud computing and Federated learning-based framework for intelligent hazardous Road Damage detection and warning named FedRD is proposed. FedRD utilizes advanced edge-cloud computing, federated learning, and differential privacy techniques for fast, accurate, cheap, and private hazardous damage detection and warning.
- An ADD model for hazardous road damage detection is developed. ADD leverages high feature extraction advantages of deep learning models with hierarchical feature fusion. ADD enables fast, accurate, and robust road damage detection.
- A novel AFed is designed, which updates the learning models based on their detection performance and learning speed. It ensures more robust model learning with low communication rounds. A new map construction method is introduced to provide road users with a global warning map covering a wider area.

- A new privacy protection technique named Individualized Differential Privacy with Pixelization (IDPP) is introduced based on the advanced differential privacy technology. IDPP protects both users' sensitive information (e.g., ID and location) and the privacy inside images/videos (e.g., people's faces and drivers' plate numbers) collected at users' devices before uploading to edges.

- Extensive evaluations are performed to prove that the proposed FedRD framework can achieve a high detection performance with low latency and provides accurate warning information covering a wider area while preserving privacy, even when some edges only have limited data.

The remainder of this chapter is organized as follows: Section 3.3 gives a summary and comparison of the related state-of-the-art works. Section 3.4 explains the design of the FedRD framework, including design goals, architectural components, and working process. In Section 3.5, the proposed Abnormal Road Screening model (AbRS), ADD model, AFed strategy, map construction method, and IDPP technique are elaborated. Experimental setup, datasets, baselines, evaluation metrics, and the performance of proposed approaches and the overall FedRD framework are evaluated in Section 3.6. Finally, Section 3.7 concludes the chapter.

3.3 State-of-the-art

This section reviews the related literature about road damage classification algorithms, cloud/edge computing systems for road damage inspection, and privacy-preserving techniques.

Road damage classification algorithms: Most state-of-the-art hazardous road damage classification methods can be categorized into two groups: traditional approaches and deep-learning-based approaches.

Traditional approaches are mainly based on statistics, filters, and models. Statistical-based methods leverage statistical information of the image, e.g., the distribution of image pixel value. For example, [60] combined both GLCM and local binary pattern (LBP) feature where GLCM was used for feature extraction, and LBP was utilized for feature's robustness improvement. The KNN classifier then classifies the features. Filter-based methods describe texture information of images by several filters. This kind of approach works well when classifying road damages with strong texture features. For example, to detect surface defects, the authors of [61] used phase-only Fourier transform to detect saliency regions of images. Then, the detected saliency areas were matched with the corresponding template regions. Model-based methods construct a mixture model with some base models according to certain distributions or other attributes. For example, the authors of [62] utilized two mixture models to calculate pattern likelihoods. Defects were detected automatically by simple parametric thresholding. Despite good performance on texture-oriented defects, they may fail for data with heterogeneous textures or when there are considerable variations of defects or backgrounds. Also, most of them have high complexity, are time-inefficient, and are prone to errors. Therefore, it is not suitable for real-world applications.

Deep-learning-based approaches achieved state-of-the-art results for many applications [63]. For example, Faster R-CNN and its variants are widely used in road damage danger detection in civil infrastructure like [64, 65]. For faster computing, SSD and MobileNet are utilized. MobileNet is lightweight and specially designed for mobile applications with limited resources. To produce more robust and abundant feature representations, different deep future fusion strategies are designed. For example, in [66], a multi-scale pyramidal pooling network was proposed for defect classification, which accepts different input image sizes. The authors of [67] introduced a multilevel-feature fusion network (MFN), which fused multilevel hierarchical features from different layers of the CNN backbone into the same dimension for defect recognition. Similarly, [68–70] also fused multilevel features to build discriminative hyper features for defect monitoring.

Cloud/edge computing systems for road damage inspection: Vision-based road damage inspection is a high resource-demanding task. Suitable computing platforms must be selected to ensure high QoS, such as fast response and high accuracy. Some researchers deploy the road inspection task on clouds for their high computing power and storage capacity for data processing. For example, in [50], a cloud server was utilized to process data received from vehicles, and the results were then sent to the traffic monitoring center. Similarly, in [71], data collected by cameras was processed by machine learning or deep learning algorithms on a cloud server to automate the monitoring process. In this way, the calculation and storage burden is transferred to the cloud, thus, less burden for users or vehicles.

Nevertheless, cloud-based approaches still have many issues, for example, high latency and high bandwidth costs caused by continuously transmitting large amounts of data to the cloud. With the emergence of edge computing, some researchers explored edge-based inspection systems due to the advantages, e.g., location awareness, large scalability, and low latency. The collected data (by users/vehicles) are uploaded to the nearest edge in edge-based systems instead of the cloud in centralized cloud-based systems. For example, a new system for road condition inspection with edge computing was proposed by [51, 72]. Edge servers directly process data from users by specific algorithms and then transfer the results to a cloud. The cloud stores the results for later use. Kawano et al. [26] used edge computing for road damaged lane markings detection. Following its success, the authors' previous work [7] proposed an edge-cloud computing framework (EcRD) for intelligent road damage detection and warning. EcRD exploits the fast-responding benefit of edge and the high computational power and enormous storage space advantages of the cloud. However, there are still some limitations of EcRD: firstly, drivers are only informed about hazardous road damages within a small area covered by one edge. Secondly, since the detection model on each edge is trained with the data collected from that edge, it cannot guarantee the edges' detection performance when the edge only has limited data.

Privacy-preserving techniques: Despite the success of edge/cloud computing for road damage inspection, the privacy issue is still not addressed in many existing schemes [50,71,72]. Only a few systems considered the privacy issues, e.g., [7,50,51]. The authors of [50] preserved privacy against clouds by receiving data in ciphertext format. After validating the data source by the cloud and the authority, only data from legitimate vehicles are chosen. In [51], the privacy-preserving certificateless aggregate signcryption scheme (CLASC) was proposed for road condition monitoring by vehicular crowd-sensing using edge computing. The scheme is computing-efficient. However, it did not consider location privacy. Moreover, the authors' previous work [7] filtered out sensitive information by a road segmentation model at edges to protect users' privacy. However, there is still a high privacy leakage risk when sending data from users' devices to the edges.

In summary, following the success of deep-learning-based approaches, the deep separable convolutions as utilized in MobileNet are employed to build the AbRS model for real-time suspicious abnormal road screening on users' IoT devices. Also, the ADD model based on ResNet structure and multilevel feature fusion for hazardous road damage detection on edges are developed. Additionally, in this chapter, edges are still utilized for fast hazardous road damage detection and warning following the success of EcRD. Edges warn users/drivers immediately once any hazardous road damages are detected. It is much faster than cloud-based approaches. Also, the detected hazardous road damage information is transmitted to a central cloud. The cloud aggregates it and sends it back to users via edges. In this way, users can receive warnings from a wider range compared with [7]. Further, to improve the performance of edges with limited data, the FL strategy is leveraged to learn from multiple edges collaboratively without direct data sharing between edges and the cloud. Furthermore, although cryptographic techniques or image/video preprocessing approaches were utilized to protect users' privacy, none of the existing works considered the privacy information within images/videos. Therefore, in this chapter, privacy is protected by DP [56]. Different from [57–59], the proposed method preserves privacy at users' devices before uploading to untrusted edges, which is more private. Moreover, it has $3/4$ less computation cost because it adds noise to pixelized images instead of original images.

3.4 FedRD Framework

This section introduces the design goals, architectural components, and working process of the FedRD framework.

3.4.1 Design Goals

The following goals drive the design of FedRD:

1. **Latency:** A hazardous road damage detection and warning system must warn users timely about dangerous road damages for accident prevention, which means the latency should be low.

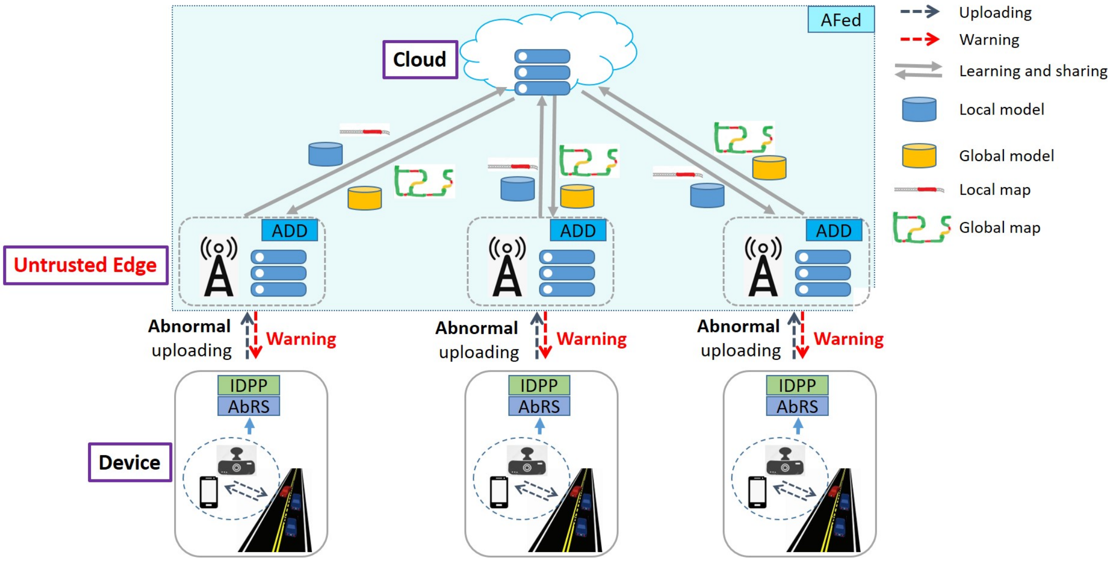


Figure 3.1: Overview of the FedRD framework.

2. **Accuracy:** The proposed system should detect hazardous road damages accurately since miss-detected dangerous road damages are fatal for road users.

3. **Robustness:** The proposed system's performance should be robust to different environments, such as different weather conditions, various illuminations, and obstacles like vehicles and pedestrians. Besides, it should achieve high performance even when some edges have limited data, common in the real-world.

4. **Coverage:** The designed framework should provide users with hazardous road damage information with wide coverage for accident prevention and route planning.

5. **Cost:** Image/video analysis is a high resource-demanding task. The size of an image/video is usually large, and the amount of accumulated data increases quickly. Also, the Internet bandwidth and data storage are expensive. Hence, the developed system requires low data transmission and data storage costs.

6. **Privacy:** There is a high privacy leakage risk from untrusted edges/clouds or during data transmission in open-access environments. The designed framework should protect the privacy of users, e.g., ID and location, and the privacy inside collected data, e.g., people's faces and license plate numbers.

3.4.2 Architectural Components

The proposed FedRD framework that satisfies the design goals is illustrated in Figure 3.1. The components of this framework are described in detail as follows:

- Devices: This component gathers video data by pervasively using IoT devices (e.g., smartphones) mounted on vehicles. AbRS deployed on devices detects suspicious road damages. Then, the data (including users' information) is processed by IDPP to protect privacy. Finally, the processed data is sent to the nearest edge (e.g., RSU).
- Edges: The ADD model is deployed on edges for fast response. ADD detects hazardous road damages. The detection performance of the ADD models is further improved by the Adaptive Federated learning strategy (AFed). Once any dangerous road damages are detected, edges broadcast warning maps to covered users for accident-preventing. The warning maps contain the detected hazardous road damage information, for example, types (e.g., big holes, fractures, blowups, pounding), levels (i.e., low, middle, and high), and locations (i.e., GPS coordinates). They are then uploaded to the cloud for aggregation.
- Cloud: The cloud serves as an aggregator which aggregates selected ADD models from edges into one model and sends it back to all edges to improve the learning process of the ADD models on edges. Meanwhile, it integrates received warning maps from edges into one warning map and sends it back to the edges for accident-preventing and route planning.
- AbRS: AbRS is deployed on devices. It is a light-weighted deep learning model and quickly detects suspicious abnormal roads from raw videos recorded by the devices. In this way, normal roads are successfully screened out, which significantly reduces data transmission cost between devices and edges. Then, only suspicious abnormal road data is transmitted to edges.
- IDPP: The IDPP technique protects privacy on users' devices before sending data to edges, including both users' privacy and privacy inside collected images/videos. It is developed based on the advanced LDP technique. Besides, a pixelization approach is utilized in IDPP to reduce computation and communication costs and improves data utility.
- ADD: The ADD model functions as a hazardous road damage detector. It is a deep learning-based model with hierarchical feature fusion. It is deployed on edges to reduce latency. ADD enables fast and accurate hazardous road damage detection and warning.
- AFed: The AFed strategy further improves the detection performance of the ADD models on edges by using the cloud as a parameter server without requiring direct data sharing between edges and the cloud.

3.4.3 Working Process

The detection models deployed on edges and clouds are defined as local models and global models, respectively. Local models learn from data on edges, while global models assist the learning process of local models by aggregating the local models on the cloud. The FedRD framework mainly consists of the following four phases that are repeated periodically for efficient hazardous road damage monitoring:

- **Phase 1:** Each vehicle collects road condition data by its carried smart IoT devices. The AbRS model on devices detects suspicious abnormal roads. The detected suspicious abnormal roads are transmitted to the nearest edge after protecting privacy by the IDPP technique.
- **Phase 2:** Each edge detects hazardous road damages by the ADD model based on data received from covered users. Once detected, it broadcasts the hazardous road damage warning information to all users within its communication range. The hazardous road damage warning information from an edge is called a local map. Then, the edge sends the trained local model and the local map to the cloud.
- **Phase 3:** The cloud selectively aggregates received local models according to their performance to generate a global model. Meanwhile, the cloud integrates all collected local maps into a global map. Afterward, the cloud sends the global model and the global map back to the covered edges. Also, the global map is sent to road administration authorities for timely repair and maintenance.
- **Phase 4:** Edges update their local models with the received global model and broadcast the acquired global map to the covered users. With the global map, users are informed about road conditions in a broader area (e.g., a city) and can select optimal routes for traveling.

3.5 Methodologies

Details of the developed algorithms, i.e., Abnormal Road Screening model (AbRS), Advanced hazardous Damage Detection Model (ADD), Adaptive Federated learning strategy (AFed) and map construction, and Individualized Differential Privacy with Pixelization technique (IDPP) are presented in this section. Table 3.1 lists a summary of all notations utilized in this chapter.

3.5.1 Abnormal Road Screening Model

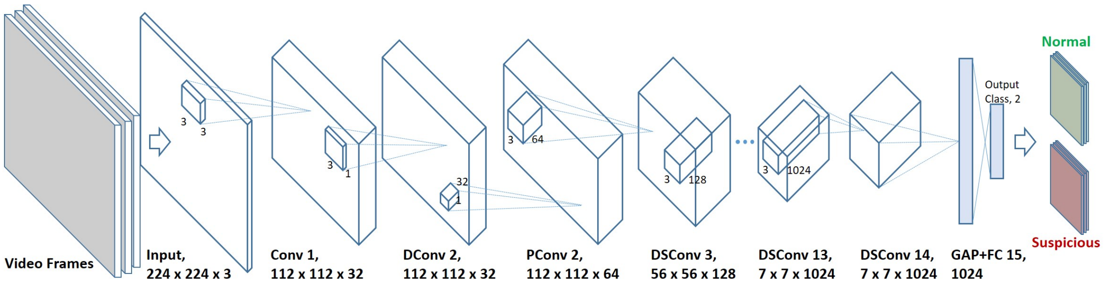
Video transmission and analysis are high resource-demanding tasks. Directly transferring video data from devices to edges would cause network congestion and seriously affect other services. Fortunately, road condition videos recorded by smart IoT devices are primarily normal roads without dangers (around 80%). Hence, it is essential to detect abnormal roads first and only upload them to edges to minimize network communication burden and data processing and storage costs. Additionally, AbRS is built based on deep learning models since they are more robust for processing real-world data with cluttered backgrounds compared to traditional machine learning methods combined with hand-craft features [7,42]. Moreover, considering that the AbRS model is deployed on devices with limited computational power, a light-weighted deep learning model structure (i.e., depthwise separable convolutions) is chosen to build the AbRS model similar to MobileNet [73]. As illustrated in Figure 3.2, the AbRS model includes one Convolutional layer

Table 3.1: Summary of Notations.

Notation	Description
V_i	The i_{th} video
N_r, N_c	No. of rows and columns in a video
M_i	The i-th image in $\{M_1, \cdots, M_N\}$
L	Prediction loss of MobileNet
$L_{threshold}$	Threshold of prediction loss L
c_i	Different classes of road damages
l_1, l_2, and l_3	Low, middle, and high level road damages
w_i^t	Parameters of local model in i-th edge at time t
w_t	Parameters of global model at time t at the cloud
t	Period of time
N	No. of edges
D_i	No. of images in the i-th edge
D	Total No. of images in all edges
b	Optimal pixels differ from neighboring images
$1/\epsilon$	Level of privacy
$F(x)$	A random function
$X, \mu, 2b_2$	Random variable, and its mean and variance
P	Pixelization function
θ_i	Value from Laplace distribution
δP	L1-sensitivity of the function P
M_i^j	A pixel of image M
l	Length of square subset
K	No. of subsets

(Conv), one Depthwise Convolutional layer (DConv), one Pointwise Convolutional layer (PConv), twelve Depthwise Separable Convolutional layers (DSConv), a Global Average Pooling layer (GAP), Fully Connected layer (FC), and a softmax classification layer. Different from standard convolution operations, a depthwise separable convolution is a form of factorized convolutions including a depthwise convolution and a 1×1 pointwise convolution [73], which makes the AbRS model more efficient. The AbRS model classifies each video frame as normal roads and suspicious abnormal roads and only upload suspicious abnormal roads to edges. In this way, the data transmission cost can be significantly reduced (by around 80%), and considerably fewer data need to be processed on edges and the cloud.

The training of the AbRS model requires a large labeled dataset, while abnormal roads are not easy to collect in some areas. Also, data samples are considerably different even within the abnormal road class, making the learning of AbRS even harder. In contrast, normal roads have a high homogeneity. In other words, data samples are similar within the normal road class. Therefore, only the AbRS model with normal roads is trained, and the abnormal roads are recognized based on the prediction loss of the model. More specifically, since the model is well-trained with normal roads, it can recognize them with considerably high confidence. If an image's prediction loss is



DConv: Depthwise Conv; Pconv: Pointwise Conv; DSConv: Depthwise Separable Conv;
GAP: Global Average Pooling; FC: Fully-Connected

Figure 3.2: Abnormal road screening model.

lower than the threshold, it is considered a normal road. On the contrary, if an image's prediction loss is higher than the threshold, it is classified as an abnormal road.

A video is defined as V. The videos collected by a user are denoted as $\{V_1, V_2, \cdots, V_N\}$, where N is the number of videos collected by the user. Additionally, a frame with R row and C column pixels in video V is denoted as M_i. The pixel value at location (r, c) of an image M is defined as $I(r, c)$. As shown in Figure 3.2, videos $\{V_1, V_2, \cdots, V_N\}$ are gathered by a road user with a smart IoT device. The AbRS model on the device preprocesses the videos to filter out normal roads. The output of the AbRS model is a prediction loss L, reflecting the probability of being a normal road. If $L > L_{threshold}$, then the input is an abnormal road; otherwise, it is a normal road.

AbRS is deployed at users' devices for video data preprocessing. Before uploading to edges, every video frame is examined and classified into normal roads and abnormal/suspicious roads. Thus, the transmission data volume can be significantly minimized, and less data needs to be processed on edges and the cloud.

3.5.2 Advanced Hazardous Damage Detection Model

The ADD model is introduced to detect hazardous road damages and measure their severity levels. ADD is deployed on edges instead of the cloud for fast response. Besides, IoT devices are not directly employed for hazardous road damage detection due to their low computational power, and each device only has limited training samples. In contrast, edge servers have much higher computational power than IoT devices and are much closer to users than the cloud. Once any hazardous road damages are detected by ADD on edges, warning messages are distributed to its covered users instantly, incorporating hazardous road damage types, levels, and locations.

According to [74], deeper models can significantly increase the classification performance but are more challenging to train and has a higher computation cost. Fortunately, the deep residual learning structure proposed by [74] is easier to optimize and can achieve high accuracy from considerably increased depth. Hence, ADD is built based on a deep residual learning structure, i.e.,

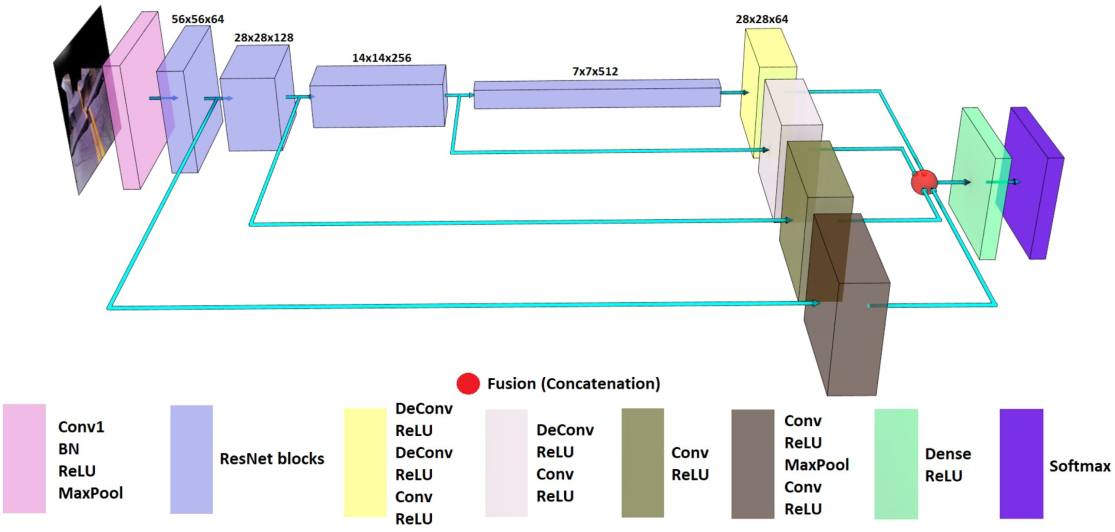


Figure 3.3: Advanced hazardous damage detection model.

residual blocks with skip connections. Although the detection accuracy increases with the growth of the number of residual blocks, the data processing time is longer. As a trade-off, ADD only uses five residual blocks, as shown in Figure 3.3. Each residual block contains a residual function performed by a shortcut connection and element-wise addition. An example of a residual block is displayed in Figure 3.4.

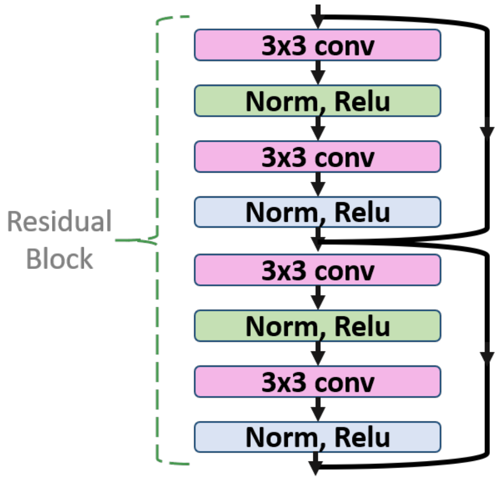


Figure 3.4: Example of a residual block.

Different residual blocks produce different levels of features. Deep layers produce high-level features, while shallow layers generate low-level features. Generally, only the last layer's feature is utilized for classification. However, low-level features also contain valuable information that can assist the final classification task. In ADD, multiple-level feature maps are extracted from different residual blocks. Extra layers are applied to the output of the four residual blocks (i.e., residual blocks 2, 3, 4, and 5) to fuse the feature maps, as illustrated in Figure 3.3. More specifically, when an image feeds into the ADD, one feature map from each residual block (except the first one) can be obtained. The feature maps are then processed by extra feature fusion layers to refine and resize them. The fusion of the four feature maps is used to classify the input image. Based on the classification result of ADD, the input image is further categorized into three dangerous levels according to visual severity. To be more specific, images without damages or only with minor damages (i.e., cracks and patches) are treated as low-level. Middle-level and high-level road damages are images with middle severity damages (i.e., potholes and fractures) and high severity damages (i.e., big holes and serious road blowups).

The input of ADD is defined as $\{M_1, M_2, \cdots, M_N\}$, where M_i is the i-th image and N is the total number of input images. The input images are classified into six classes by ADD, denoted as $\{c_1, c_2, \cdots, c_N\}$, where, c_i represents different classes of road damages, i.e., cracks, patches, potholes, fractures, big holes, and blowups. Based on the classes of the damages, the output result is further categorized into three levels, i.e., $\{l_1, l_2, l_3\}$, where l_1, l_2, and l_3 represent low, middle, and high dangerous levels.

Further, ADD is deployed on edges, and dangerous road damages are detected and rated by ADD then warning information including dangerous levels, locations are broadcasted to the users nearby.

3.5.3 Adaptive Federated Learning and Map Construction

Adaptive Federated Learning (AFed): AFed is a decentralized adaptive federated learning strategy inspired by [52]. Unlike centralized methods that train a machine learning model with data on a central cloud server, decentralized approaches (e.g., FL) train a global model by collaboratively learning from multiple edges without direct data sharing between them. Different from existing FL strategies [52–54], AFed selective aggregates Q qualified models from top K local models collected from edges, which complies with the fact that the central server has no control over local edges. Also, by only aggregating high-quality local models, high performance can be ensured in limited communication rounds, and data poisoning attacks can be alleviated to some extent.

There are mainly two phases in AFed, i.e., local update and global aggregation. The local update phase periodically updates the local models on edges, while the global aggregation phase generates a global model by aggregating the selected local models. Local models are utilized for timely hazardous road damage detection and warning. The global model is used for updating local models' parameters to improve their performance.

The AFed strategy with the setting of one cloud and N edges is considered. Let D_i denote the number of data samples in the local database held by the edge i. The local training on the i-th edge

Algorithm 3: AFed strategy

```
Input   : Local databases from edges and detection model ADD
Output: Optimal local models on edges
for each time period $t$ do
    cloud initialize $\mathbf{w}^0$;
    for every E epochs do
        for each edge $i = 1, 2, \cdots, N$ do
            $\mathbf{w}_i^t \leftarrow$ localUpdate($t, \mathbf{w}_t^i$)
        end
        cloud wait until receive $K$ local models
        $\mathbf{w}^t \leftarrow \sum_{i=1}^{Q} \frac{D_i}{D} F_{select}(\mathbf{w}_i^t)$
    end
    send $\mathbf{w}^t$ to edges if $P(\mathbf{w}^t) > P(\mathbf{w}_i^t)$
    for every E epochs do
        update $\mathbf{w}_i^t$ by $\mathbf{w}^t$ if receive
        $\mathbf{w}_i^t \leftarrow \mathbf{w}_i^t - \eta \bigtriangledown l(\mathbf{w}_i^t, b_i^t)$
    end
    send $\mathbf{w}_i^t$ to the cloud
end
```

aims to obtain a parameter set $\mathbf{w}_i^t$ of the local model at time t by minimizing the loss function. The goal of the cloud is to learn a global model $\mathbf{w}^t$ over data on the selected edges. Specifically, the cloud waits until top K local models are received. Out of them, the cloud aggregates Q best local models by:

$$\mathbf{w}^t = \sum_{i=1}^{Q} \frac{D_i}{D} \mathbf{w}_i^t, \tag{3.1}$$

where D_i is the number of samples on the i-th edge, while D is the total number of samples over the selected Q edges. Also, $D = \sum_{i=1}^{Q} D_i$.

The training process of the AFed strategy involves the following four steps. Firstly, edges locally train models on their datasets to obtain the optimal model parameters $\{\mathbf{w}_1^t, \cdots, \mathbf{w}_i^t, \cdots, \mathbf{w}_N^t\}$ at time t. Secondly, the edges send the locally trained parameters to the cloud. Thirdly, the cloud selects Q qualified local models from the top K received models. It stops receiving local models once K local models are collected. The Q qualified local models are selected by comparing the detection performance between the K collected local models. Only the local models with the top Q highest detection performance will participate in the aggregation. The detection performance of both local and global models is evaluated by the shared testing set on the cloud. Then, the cloud aggregates the parameters of the Q local models from the edges by Eq. (3.1). The aggregated global model's parameters $\mathbf{w}^t$ are sent back to edges if its performance is better than the edges' local models. Finally, edges update their local models with the global model. More details of the method are presented in Algorithm 3, where $F_{select}(*)$ means the selection of top Q local models; $P(*)$ indicates performance; η is the learning rate, $\bigtriangledown$ is the derivative, $l(*)$ is the loss function, $\mathbf{w}$

and b are the weights and biases; E is the number of epochs.

Map Construction: In FedRD, the local and global maps are generated by the local and global models. Based on the collected road condition information on edges, the local model detects dangerous roads and classifies them into three severity levels: low, middle, and high. Then, the local maps, including the types, levels, and locations, are constructed on edges. Each edge provides a fast warning by broadcasting its local map to its neighboring users. After that, each edge sends its local map to the cloud. The global map is created on the cloud by aggregating all local maps. The cloud broadcasts the global map to all users. Then, users can obtain the latest road conditions in a large area (e.g., a whole city), which helps users select optimal routes for traveling and significantly reduces the road accidents caused by hazardous road damages.

3.5.4 Individualized Differential Privacy with Pixelization

As mentioned in Section 3.1, although AFed protects privacy from untrusted clouds, there is still a high privacy leakage risk from untrusted edges. Also, researchers [75] have proven that images can be recovered from averaged gradients in FL even with a larger batch, large networks, and complex datasets. Moreover, sending data from users' devices to untrusted edges also poses great threats to users' data (e.g., location and ID) and the collected data (e.g., people's faces and license plate numbers). To protect privacy, users must sanitize all data before sending it to edges. Therefore, a new privacy-preserving technique named Individualized Differential Privacy with Pixelization (IDPP) is introduced to fulfill this requirement. IDPP is built based on the powerful LDP approach [76]. Unlike [57,76], IDPP preserves privacy at users' devices before uploading to untrusted edges. Also, images are pixelized before applying DP to reduce the computation cost of devices and the communication cost from devices to edges. Similar to [56], the Laplace mechanism is employed in IDPP.

3.5.4.1 Preliminaries

Following [57], neighboring images are defined as follows:

Definition 1. *Let two images/frames be M_1 and M_2. It can say that M_1 and M_2 are neighboring images if they have the same dimension and differ by b pixels ($b = 1$).*

According to Definition 1, images' sensitive information, such as faces and license plate numbers, can be protected by b pixels difference. The optimal b value can be selected to customize different levels of privacy according to the requirements of users and the trade-off between detection performance and privacy.

In the following content, the necessary preliminaries related to IDPP are illustrated. Then, it can be proved that IDPP achieves the ϵ-differential privacy. Following the concept of the differential privacy mechanism from Dwork et al. [56], ϵ-differential privacy is defined as follows:

Definition 2. *(ϵ-differential privacy): A random function F is said to be a ϵ-differential privacy (ϵ-dp) function, if for two different inputs $x, x' \in Dom(F)$ and one output be $z \in Range(F)$, it has:*

$$P(F(x) = z) \leq exp(\epsilon)P(F(x') = z). \tag{3.2}$$

Remark 1. *The parameter ϵ denotes privacy level. The higher the ϵ value, the more privacy leakage. Hence, ϵ is utilized to measure the trade-off between privacy leakage and detection performance.*

To accomplish the ϵ-differential privacy, the Laplace mechanism is often employed to add noises to the original data, where the noises are generated from the Laplace distribution from Definition 3.

Definition 3. *Laplace distribution: A random variable X follows the Laplace distribution if its probability density function is,*

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{|x - \mu|}{b}), \tag{3.3}$$

where the localization parameter is μ and scale parameter is b. Furthermore, the mean of the random variable X is μ and the variance of X is $2b^2$.

Remark 2. *If X follows a Laplace distribution with localization parameter μ and scale parameter b, then $X \sim Lap(\mu, b)$.*

The concept of the Laplace mechanism is given in Definition 4.

Definition 4. *Laplace mechanism: Given a function $P : R^n \rightarrow R^p$, the Laplace mechanism F can be defined as,*

$$F(x, P(.), \epsilon) = P(x) + (\theta_1, \cdots, \theta_p), \tag{3.4}$$

where $\theta_i \sim Lap(0, \frac{\Delta P}{\epsilon})$. The ΔP is the $L1$-sensitivity of the function P, which is illustrated in Definition 5.

Definition 5. *$L1$-sensitivity: The $L1$-sensitivity of a function $P : R^n \rightarrow R$ is defined as:*

$$\Delta P = sup_{x,y \in A} \|P(x) - P(y)\|_1, \tag{3.5}$$

where $\|.\|_1$ is the L1 norm.

Remark 3. *The sensitivity shows how much the function P can be changed by adding random noise while preserving privacy.*

3.5.4.2 Pixelization

An image is represented as a matrix M, and each pixel value of the image is denoted as $M_{i,j}$, where $i = 1, 2, \cdots, R$, $j = 1, 2, \cdots, C$, and $0 \leq M_{i,j} \leq 255$. The pixelization technique takes blocks/subsets of the image matrix as input. Every element $M_{i,j}$ that belongs to that block is replaced by the average value of that block. In this chapter, a square subset of length l is taken. Thus, for a matrix with dimension $R \times C$, the total number of square subsets is $S = \lceil \frac{R}{l} \rceil \times \lceil \frac{C}{l} \rceil$.

To include pixelization technique to the differential privacy concept, the pixelization global sensitivity in Lemma 1 can be defined.

Lemma 1. *The $L1$ sensitivity of the pixelization technique is $\Delta P_l = \frac{256b}{l^2}$.*

Proof. By Definition 1, given two neighboring images/frames M_1 and M_2. These images differ by at most b pixels (here we take $b = 1$). Since each pixel ranges from 0-255, then,

$$sup_{M_1,M_2}|M_1 - M_2| \leq 256b. \tag{3.6}$$

Now, the pixelization here takes a square of length l. Thus each square has l^2 pixels. Therefore, for the entire image, the global sensitivity is:

$$sup_{M_1,M_2}\|P_l(M_1) - P_l(M_2)\|_1 = \Delta P_l = \frac{256b}{l^2}. \tag{3.7}$$

□

3.5.4.3 IDPP

Algorithm 4 illustrates the procedure of IDPP. In particular, S is the number of pixel subsets. The random variable θ_i of the i-th subset is generated following the Laplace distribution. After that, this random noise is added to the average of each subset. The proof of IDPP is illustrated in Theorem 2.

Algorithm 4: IDPP

```
Input   : image $M_{(r,c)}$ and $R, C, l, b, \epsilon$
Output: image $M_{(r,c)}$ with privacy
Initialize with $S = \lceil \frac{R}{l} \rceil \times \lceil \frac{C}{l} \rceil$, $\Delta P_l = \frac{256b}{l^2}$
for $i = 1, 2, \cdots, S$ do
    $\theta_i \sim Lap(0, \frac{\Delta P_l}{\epsilon})$
    $P_l(M_{(r,c)}) = \sum_{(r,c) \in g_i} \frac{M_{(r,c)}}{l^2}$
    $\widetilde{P}_l(M_{(r,c)}) = P_l(M_{(r,c)}) + \theta_i$
    Return $\widetilde{P}_l(M_{(r,c)})$
end
```

Theorem 2. *The $\tilde{P}_l$ in Algorithm 4 satisfies ϵ-differential privacy.*

Proof. Let two neighboring images be M_1, M_2, another independent image be M, and a random variable be $\theta_i \sim Lap(0, \frac{\Delta P_l}{\epsilon})$, where $i = 1, \cdots, K$.

$$\begin{aligned}
\frac{P(\widetilde{P}_l(M_1) = M)}{P(\widetilde{P}_l(M_2) = M)} &= \prod_{i=1}^{K} \frac{exp(-\frac{\epsilon|P_l(M_{i,1})-M_i|}{\Delta P_l})}{exp(-\frac{\epsilon|P_l(M_{i,2})-M_i|}{\Delta P_l})}, \\
&= \prod_{i=1}^{K} exp(\frac{\epsilon(|P_l(M_{i,2}) - M_i| - |P_l(M_{i,1})_i - M_i|)}{\Delta P_l}).
\end{aligned} \tag{3.8}$$

Applying the triangular inequality, and from Eq. (3.8),

$$\leq \prod_{i=1}^{K} exp(\frac{\epsilon|P_l(M_{i,1}) - P_l(M_{i,2})|}{\Delta P_l}). \tag{3.9}$$

By $L1$ norm and $L1$ sensitivity in Definition 5, from Eq. (3.9), then,

$$\begin{aligned}
&= exp(\frac{\epsilon\|P_l(M_1) - P_l(M_2)\|_1}{\Delta P_l}), \\
&\leq exp(\epsilon).
\end{aligned}$$

Thus,

$$\frac{P(\widetilde{P}_l(M_1) = M)}{P(\widetilde{P}_l(M_2) = M)} \leq exp(\epsilon). \tag{3.10}$$

Finally, the proof of Theorem 2 is concluded as below,

$$P(\widetilde{P}_l(M_1) = M) \leq exp(\epsilon)P(\widetilde{P}_l(M_2) = M). \tag{3.11}$$

□

3.6 Experiments and Evaluation

In this section, the experimental setup and the dataset are introduced first. Then, the comparison baselines are described. Following this, experimental results are presented.

3.6.1 Experimental setup and Dataset

A cheap smartphone (VG30+) is used as an example of an IoT device for data acquisition. A laptop (Dell Latitude 5880, 64-bit Windows 10 Operating system, 16 G RAM, Intel Core i7 i7-7820HQ CPU with 2.9GHz) is used as the edge server to simulate an RSU. A high-performance server (Ubuntu 16.04 LTS system, 125.8 GB of RAM, 5.93 TB of hard disk, and 8 GTX 1080 Ti GPUs) is used as the

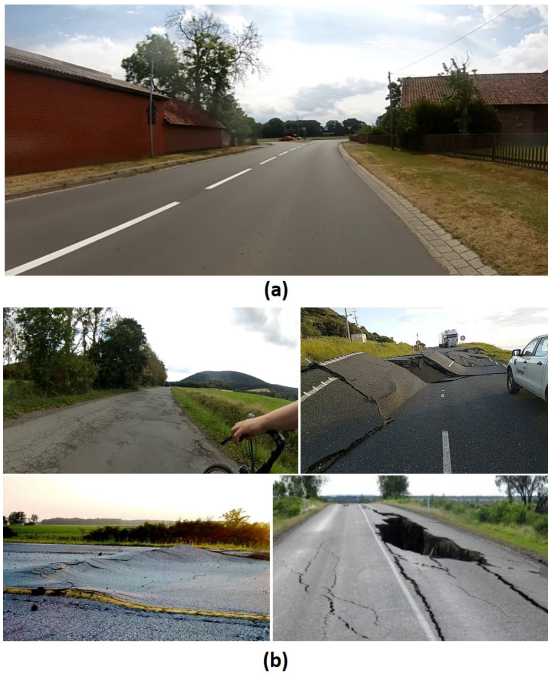


Figure 3.5: Image Examples of AbRS. (a) Normal road, (b) Abnormal/Suspicious road.

cloud server. The client application is implemented using Python 3.6. For fair comparison and good performance, the learning rate is set as 0.001, the total number of epochs as 1000, the number of epochs for local update E as 30, batch size as 64, momentum as 0.9, and weight-decay as 0.0001. The training set is split into three subsets, i.e., 60%, 30%, and 10%, for edge1, edge2, and edge3, to simulate the model training with limited and unequally-sized datasets on different edges.

To effectively evaluate the performance of FedRD, more than 75 road videos (more than 49 minutes of each) with the resolution of 1280×720 pixels from drivers' front viewpoint are collected by the IoT device. The training set is built by automatically extracting frames from the collected videos with an interval of 0.5 seconds. The training set of the AbRS model contains 1560 normal road images. Figure 3.5 presents some examples of normal roads and abnormal roads. The training set of the ADD model includes 300 abnormal road images predicted by AbRS and 600 hazardous road damage images collected from the Internet. The dataset is augmented to 2500 by rotation, skew, crop, adding noise, and padding. Some examples of different levels (i.e., low, middle, and high) of hazardous road damages are shown in Figure 3.6. Each edge has its local dataset on which it trains its local model. The cloud has a global dataset and global model. Local models and the global model share the same architecture. Further, the cloud has a small test dataset (with 60 images, 20 images per dangerous level) to decide the local models involved for global aggregation. For testing, a video including 5000 frames with 1080 abnormal roads and 3920 normal roads is built.

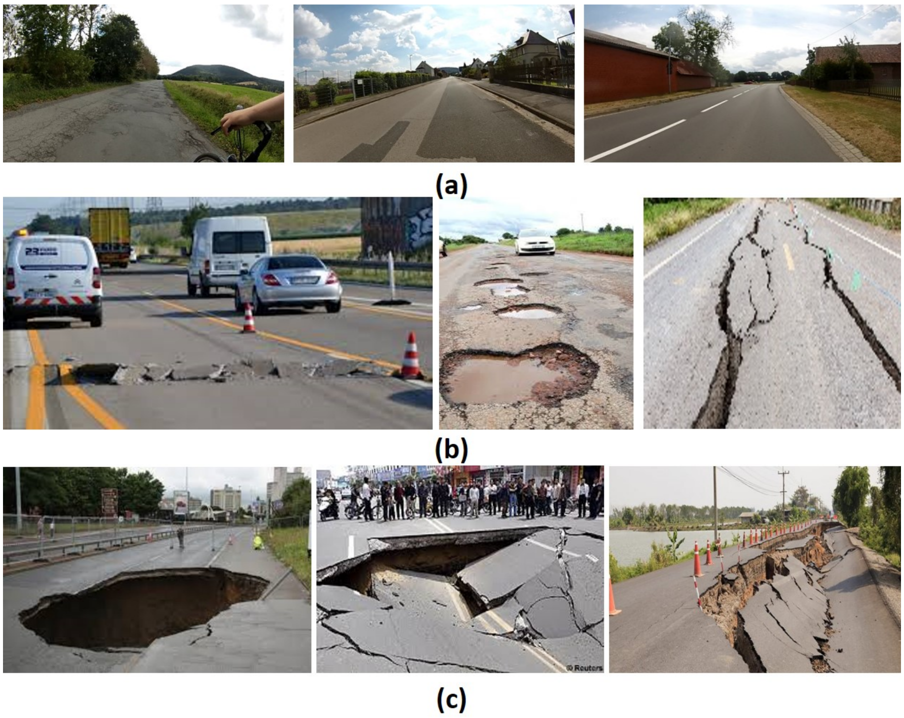


Figure 3.6: Image Examples of ADD. (a) Low level, (b) Middle level, (c) High level.

3.6.2 Baselines

In this work, the proposed FedRD, including the proposed AbRS, ADD, AFed, and IDPP, is utilized for rapid hazardous road damage detection and warning. To evaluate the performance of AbRS, ADD, and FedRD, their performances are compared with corresponding baselines.

Baselines for the evaluation of the AbRS:

1) AlexNet [77]: It is a Convolutional Neural Network (CNN) based method with five convolutional layers, three fully-connected layers, and a softmax layer.

2) GoogleNet [78]: It is a CNN-based method with 22 layers, mainly including nine inception modules. Each inception module has an extra 1x1 convolutional layer before the 3x3 and 5x5 convolutional layers. The end of each inception module is connected to a global average pooling layer.

3) VGG16 [79]: VGG16 is a CNN-based model with eight convolutional layers, four max-pooling layers, three fully connected layers, and a softmax layer.

Baselines for the evaluation of the ADD:

1) CloudRD [71]: It is a method with seven convolutional + batch normalization + ReLU layers,

six max-pooling layers, one fully-connected layer and a softmax layer. CloudRD detects road dangers on the cloud.

2) EdgeRD [26]: It is a CNN-based model with four convolutional + max-pooling layers, one global average pooling layer, two fully-connected layers, and a softmax layer. EdgeRD process data on edges.

3) ResNet [74]: ResNet is a CNN-based model using 5 residual blocks, one average pooling layer, one fully connected layer, and a softmax layer. Each residual block has an extra shortcut connection and element-wise addition.

Baselines for the evaluation of the Adaptive Federated learning strategy (AFed):

1) No-AFed represents the detection model (ADD) learning without the adaptive federated learning strategy.

2) AFed represents the detection model (ADD) learning with the adaptive federated learning strategy.

Baselines for the evaluation of the FedRD:

1) FedRD-no-IDPP: It is a variant of FedRD, just without the individualized differential privacy with pixelization approach (IDPP). By comparing with FedRD-no-IDPP, it can know the influence extended to the performance of FedRD by IDPP and what is the best trade-off.

2) CloudRD [71] and EdgeRD [26]: They are variants of FedRD, CloudRD deploys hazardous road detection task on edges while CloudRD deploys it on the cloud. Besides, both of them are not using a federated learning strategy. The performances of both CloudRD and EdgeRD are compared to FedRD to evaluate the effectiveness of the edge-cloud-based Federated learning framework.

3) EcRD [7]: The EcRD is an edge-cloud-based road damage detection framework that detects hazardous road damages only at edges, however, with a different model (i.e., HDD). The performance of EcRD can prove the necessity of a federated learning strategy as well as the performance of ADD in FedRD.

3.6.3 Evaluation Metrics

The precision, recall, accuracy, f1-score, runtime, and latency are used to evaluate the performance of FedRD as well as the baseline approaches. The definition of them is the same as in [7].

3.6.4 Evaluation Results

In this subsection, the performance of the developed models and their corresponding baselines is evaluated by the evaluation metrics defined in [7], including Precision, Recall, Accuracy, F1-score,

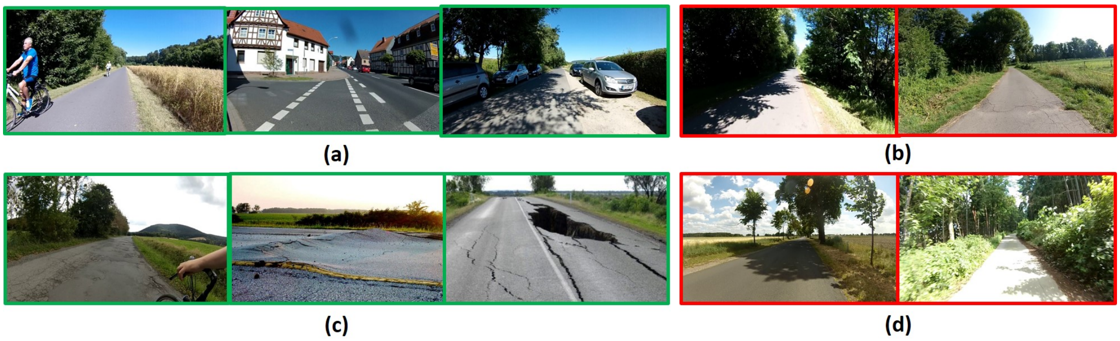


Figure 3.7: Experimental results of AbRS, (a) good results of normal roads, (b) bad results of normal roads, (c) good results of abnormal roads, (d) bad results of abnormal roads.

runtime, and latency. Following that, FedRD's evaluation results are introduced by comparing it with its baselines.

3.6.4.1 AbRS and ADD Results and Evaluation

The experimental results of the AbRS model are illustrated in Table 3.2. As shown in the table, the F1-score of AbRS is 98.79%, which is 18.11%, 2.87%, and 1%, higher than AlexNet, GoogleNet, and VGG16, respectively. Also, the accuracy of AbRS is 18.47% and 2.72% higher than AlexNet and GoogleNet, respectively. The result shows that although AbRS is more lightweight than AlexNet, GoogleNet, and VGG16, it achieves better abnormal road detection results. The reason behind this is that abnormal roads, especially for those with hazardous road damages, are vastly different from normal roads, as shown in Figure 3.5, making the separation of abnormal roads and normal roads simpler. Moreover, the runtime of AbRS is 89.21%, 92.99%, and 90.93%, lower than AlexNet, GoogleNet, and VGG16, respectively. Likewise, AbRS has the lowest runtime because it has a small model size for using deep separable convolutions instead of normal convolutions. The AbRS achieves the highest accuracy with the lowest runtime. Given the high performance of the AbRS model, the amount of data that needs to be transmitted from devices to edges is significantly reduced by using the AbRS model. Despite the high F1-score, there are still some miss-detected abnormal roads. Some results of AbRS are illustrated in Figure 3.7. The results show that AbRS may fail to detect abnormal road frames when the frames are very bright, very dark, with shadows, or very light damages.

The experimental results of the AbRS model are illustrated in Table 3.3. As shown in the table, ADD achieves a 92.52% F1-score, which is 26.34%, 15.51%, and 9.53% more accurate than cloudRD, edgeRD, and ResNet. ADD achieves worse runtime than the baselines because it contains extra feature fusion layers to capture valuable multi-scale features from different layers. However, ADD enables accurate and relatively fast hazardous road damage detection on edges. Figure 3.8 illustrates some good and bad results of different hazardous road damage levels predicted by

Table 3.2: AbRS Evaluation Result.

Model	Accuracy (%)	F1-score (%)	runtime (s)
AlexNet [77]	82.20	83.64	0.063
GoogleNet [78]	94.80	96.03	0.097
VGG16 [79]	96.50	97.81	0.075
AbRS (own)	**97.38**	**98.79**	**0.0068**

Table 3.3: ADD Evaluation Result.

Model	Accuracy (%)	F1-score (%)	runtime (s)
CloudRD [71]	75.54	73.23	0.038
EdgeRD [26]	81.21	80.10	0.054
ResNet [74]	86.35	84.47	**0.032**
ADD (Ours)	**93.48**	**92.52**	0.085

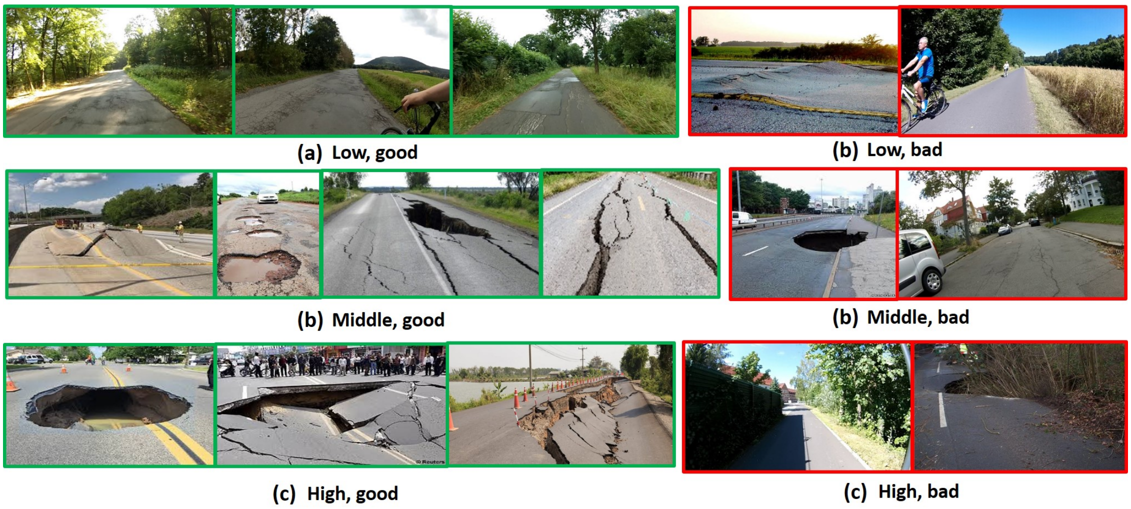


Figure 3.8: Experimental results of ADD, (a) good results of low dangerous roads, (b) bad results of low dangerous roads, (c) good results of middle dangerous roads, (d) bad results of middle dangerous roads, (c) good results of high dangerous roads, (d) bad results of high dangerous roads.

Table 3.4: AFed Evaluation Result.

Setting	Edge	Accuracy (%)	F1-score (%)
No-AFed	Edge1	89.82	84.57
	Edge2	88.35	87.26
	Edge3	89.39	88.41
AFed	Edge1	93.48	91.52
	Edge2	90.68	89.76
	Edge3	92.94	91.37

ADD. Although it achieves a relatively high F1-score, some hazardous road damages are wrongly classified due to illuminations and shadows. Also, some hazardous road damages are small for some classes, for example, minor blowups and small holes in Figure 3.8(b), Figure 3.8(d), and Figure 3.8(f), and they tend to be miss-classified as lower-level hazardous road damages, which is reasonable in practice.

3.6.4.2 AFed Results and Evaluation

Experiments with three edges and one cloud are conducted to evaluate the performance of the proposed AFed strategy. In the experiments, local models on edges update their weights every 30 iterations. The goal is to improve the detection performance of local models on edges without directly sharing data between edges and the cloud. The evaluation results of the proposed AFed strategy are presented in Table 3.4. This table shows that the F1-score of edges after applying AFed improves by maximally 6.95% than without AFed. Similarly, the accuracy of edges after using AFed increases by maximally 3.66% than that without AFed. The cloud waits until the top two local models are received, complying with the fact that the cloud has no control over edges.

3.6.4.3 IDPP Results and Evaluation

The performance of the FedRD before and after applying IDPP is given in Table 3.5. The table shows that only 1.08% F1-score and 0.7% accuracy are reduced after using IDPP, while data privacy can be well-protected. Also, the computation time of FedRD is only increased by $0.0008\,\mathrm{s}$ after applying IDPP. Figure 3.9 illustrates the detection performance of the ADD model after applying AFed with different privacy budgets ϵ, which measures the amount of noise added to the original data. Typically, the lower is the ϵ, the more noise is added, and the more private the data becomes. As shown in the figure, the detection accuracy raises with the increase of the ϵ. The detection accuracy of FedRD with $\epsilon = 0.1$ is 47.37% higher compared with $\epsilon = 0.015$. Also, the detection accuracy of FedRD with $\epsilon = 0.4$ improves by 6.00% compared to that of $\epsilon = 0.1$. However, the performance of FedRD with $\epsilon = 0.4$ is close with $\epsilon = 0.8$ and $\epsilon = 1.0$. Moreover, the F1-score of FedRD has a similar trend with the accuracy of FedRD. Figure 3.10 shows the effect of ϵ in IDPP on images. It shows that the more noise added to the image, the more private is the image content. However, if too much noise is added, e.g., when $\epsilon = 0.001$, no valuable information can be observed, including hazardous road damages without any sensitive information. Therefore, $\epsilon = 0.4$ is selected as a good trade-off for high detection performance and good privacy-preserving.

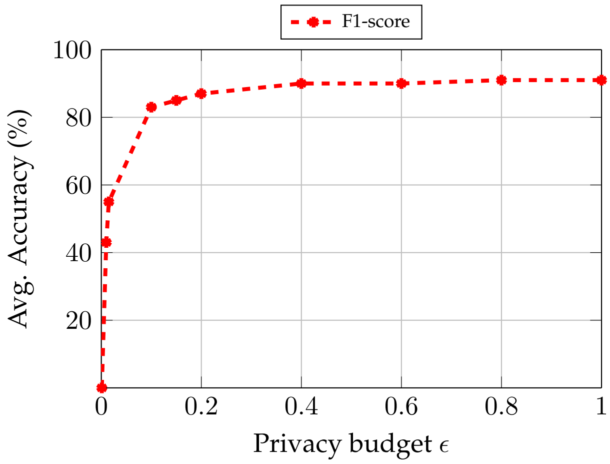


Figure 3.9: Average detection F1-score and accuracy of FedRD over different values of privacy budget ϵ.

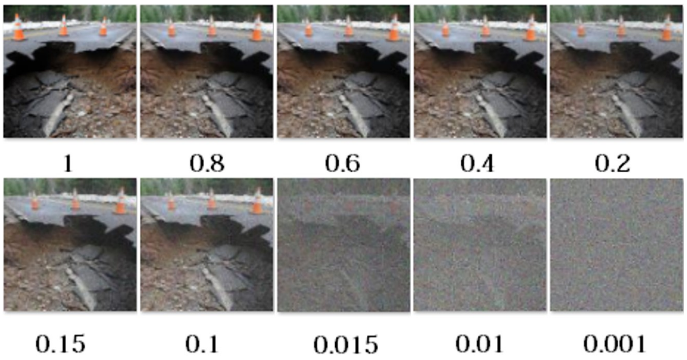


Figure 3.10: Effect of ϵ on images.

3.6.4.4 FedRD Overall Performance Evaluation.

The performance of FedRD framework is compared with cloudRD [71], edgeRD [26], and EcRD [7] to evaluate its effectiveness. The comparison results are illustrated in Table 3.6. Overall, the results clearly show that FedRD outperforms its variant and other baselines regarding the accuracy, F1-score, latency, coverage range (i.e., local or global), and privacy risk. Specifically, the detection accuracy achieved by FedRD is as high as 90.83% which is good considering the small road damage dataset as well as high inter-class divergence of the road damages as illustrated in Figure 3.6. In addition, compared to the baselines, FedRD has the lowest privacy leakage risk for the usage of AFed and IDPP. The cloudRD has a very high privacy leakage risk because clouds are usually

Table 3.5: FedRD and FedRD-no-IDPP comparison result with $\epsilon = 0.4$.

Model	Accuracy (%)	F1-score (%)	Latency (s)
FedRD-no-IDPP	91.53	91.40	0.0318
FedRD	90.83	90.32	0.0326

Table 3.6: FedRD framework Evaluation Results.

framework	Accuracy (%)	F1-score (%)	Latency (s)	Global warning	Privacy risk
CloudRD [71]	86.39	86.22	2.49	**Yes**	very high
EdgeRD [26]	81.76	81.81	0.054	No	high
EcRD [7]	**91.96**	**92.43**	**0.003**	No	high
FedRD (ours)	90.83	90.32	0.0326	**Yes**	**very low**

highly untrusted. The edgeRD and cloudRD also have a high privacy leakage risk due to untrusted edges and data transmission between devices and edges in open-access networks. Concerning latency, cloudRD has around 45 times higher latency than FedRD and edgeRD for hazardous road damage detection tasks. With such a high latency, users may not be able to receive life-threatening warning information in time. Therefore, the hazardous road damage detection and warning task should be deployed on edges to ensure high QoS. As for storage and transmission costs, storage and transmission costs for both edges and the cloud can be significantly reduced if we only upload suspicious abnormal road images to edges and save the images detected with damages. Moreover, existing edge-based frameworks, such as edgeRD [26] and EcRD [7], can only broadcast hazardous road damage warning information covered by an edge to nearby users. FedRD widens the coverage area hundreds or even thousands of times.

Discussion: To my best knowledge, it is the first research on edge-cloud federated learning-based frameworks for intelligent hazardous road damage detection and warning. The proposed FedRD efficiently utilizes the available resources from devices, edges, and the cloud for hazardous road damage detection and warning, which satisfies the design goals listed in Section 3.4 in the following ways: Firstly, the FedRD framework deploys the hazardous road damage detection model (ADD) on edges. Hazardous road damage warning messages are sent to users immediately from the edge once any hazardous road damage is detected. The latency is extremely low because edges are very close to users (satisfying design goal 1). Secondly, the AFed strategy improves local models' detection performance by updating their parameters with the latest global model when necessary (design goal 2). In this way, high performance can be guaranteed even when some edges only have limited data, which improves the robustness of hazardous road damage detection (design goal 3). Also, the robustness is further improved by training the model with diverse data, such as from different scenarios, various illuminations, and obstacles (design goal 3). Thirdly, by leveraging the edge-cloud-based global map construction strategy, drivers/users can receive road damage information on a broader area (satisfying design goal 4). For example, if FedRD (with 500 edges and one cloud) monitors a city's roads, and each edge's communication range is 1000 meters. Then, drivers can receive hazardous road damage information of the whole city, which may reach 500,000 meters. Fourthly, the communication and storage costs of FedRD are considerably reduced (design goal 5). Around 80% communication costs are reduced by filtering out normal roads using the AbRS model before transmitting to edges. The storage cost is minimized by only saving abnormal road data. Also, the 75% computation cost is reduced for applying DP by using pixelization. Finally, the FedRD framework protects data privacy (design goal 6). On the one hand, no data is directly shared from edges to the cloud, protecting privacy from untrusted clouds. On the other hand, IDPP preserves data privacy at users' devices before sending to untrusted edges and prevents privacy leakage risk during data transmission in open-access networks.

Despite the outstanding advantages, FedRD has the following limitations. Firstly, although the ADD model achieves high accuracy (93.48%), the runtime (0.085s) is not low enough for real-time hazardous road damage detection and warning. A more lightweight model should be designed to improve accuracy and reduce runtime. Secondly, the classification results in Figure 3.8 show that classifying images only based on the damage types is not always correct, especially when some damages are small. The results would be better if the road damages could be localized on the images and use the size and type for road damage rating. Depth information can also be utilized if 3D data is collected. Finally, this chapter only tests hazardous road damage as an example for general road danger detection and warning applications. More road danger types, such as traffic accidents, fallen trees, and icy roads, can also be explored in the following researches.

3.7 Conclusion

This chapter proposes the FedRD framework to improve the performance of road damage danger detection and warning concerning warning map coverage, accuracy, and privacy. In FedRD, the ADD model is designed for road damage danger detection and it improves the HDD model's (from the EcRD framework) robustness especially for data with cluttered real-world backgrounds. The communication cost of FedRD is declined by using the AFed strategy, which only uses top Q models from K edges instead of all models from all edges. FedRD proposes IDPP to protect data privacy which is based on LDP. FedRD improves EcRD's warning map coverage, enhances the robustness of the detection model, enables distributed learning among edges by AFed, and decreases privacy leakage risks by IDPP which has mathematical privacy guarantees.

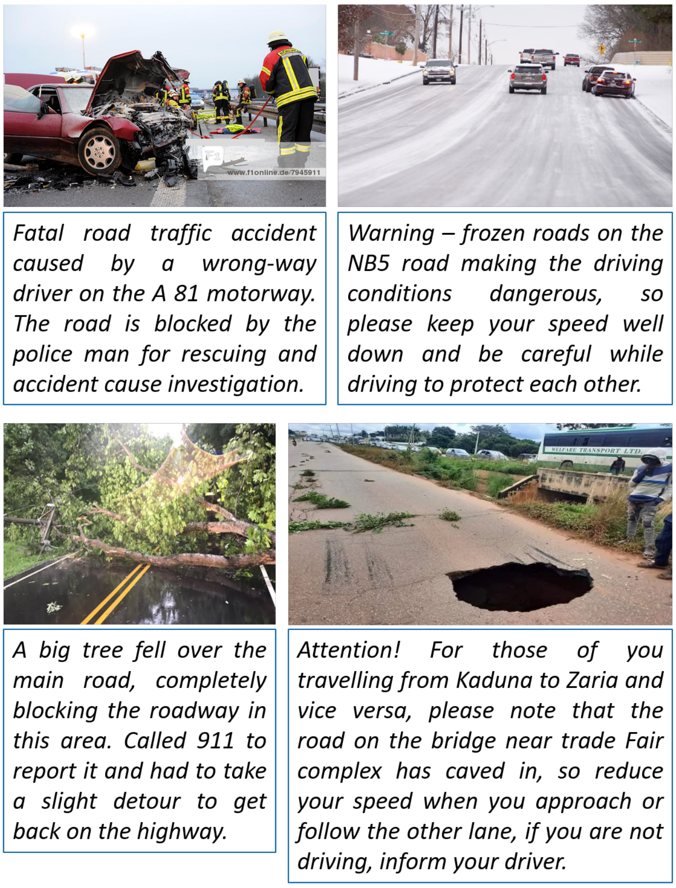
Chapter 4

FedMRD: An Edge-cloud based Privacy-preserving Federated Multimodal Learning Framework for Smart Road Danger Detection and warning

4.1 Motivation

Chapter 1 points out the importance of road danger detection and warning. Different from Chapter 2 and Chapter 3 that only study road damage danger, this chapter researches on more general road danger detection and warning, e.g., road damages, traffic accidents, fallen trees, and icy roads. Chapter 2 and Chapter 3 propose the EcRD and FedRD frameworks for smart road danger detection and warning. They are developed based on the device-edge-cloud structure and have low latency since their detection models are deployed on edges. Besides, a deep learning-based road damage danger detection model is developed in FedRD, which is more robust than EcRD 's GLCM feature-based solution. Moreover, FedRD provides road users with a much wider road danger warning map than EcRD. However, only single-modality data is utilized in these systems while ubiquitous data in other formats, e.g., text and audio, are not studied. Furthermore, many existing approaches [3,4,7,80] detect road danger dangers on users' devices, edges, or clouds by a model pre-trained on a large labeled dataset collected beforehand. However, the pre-collection of a large dataset is time-consuming, and the model cannot adaptively learn from changing data.

FL [81] enables multiple platforms to jointly learn a global model while keeping training data locally and provides privacy protection to some extend. FedRD introduces a new FL strategy named AFed for a collaborative road damage detection model that learns among different edges via the cloud in this field. The accuracy of FedRD is improved by using FL. However, the



Fatal road traffic accident caused by a wrong-way driver on the A 81 motorway. The road is blocked by the police man for rescuing and accident cause investigation.

Warning – frozen roads on the NB5 road making the driving conditions dangerous, so please keep your speed well down and be careful while driving to protect each other.

A big tree fell over the main road, completely blocking the roadway in this area. Called 911 to report it and had to take a slight detour to get back on the highway.

Attention! For those of you travelling from Kaduna to Zaria and vice versa, please note that the road on the bridge near trade Fair complex has caved in, so reduce your speed when you approach or follow the other lane, if you are not driving, inform your driver.

Figure 4.1: Examples of road danger danger with textual description.

communication cost of the AFed strategy proposed in FedRD is still high, and it is not measured precisely in EcRD. Many FL strategies [8, 54, 82, 83] are proposed to improve the performance of Federated Averaging approach (FedAvg) [81]. Despite the outstanding performance of the existing FL strategies, there are still several challenges of FL that remain to be solved. On the one hand, data on different platforms are usually heterogeneous (e.g., non-iid and skewed) in practice, and it is challenging to learn from such data. As reported by [84], the accuracy of FL reduces by up to 55% when training neural networks on highly skewed non-iid data. On the other hand, high communication cost is always the bottleneck of FL from being used in real-world applications because FL requires frequent model/parameter communication between the parties.

Privacy is one of the main concerns when users choose to use a service now. Previous researches [8, 85–89] protect users' information locally, but they did not consider the privacy preservation for multimodality data. Besides, the expected error of these approaches is too high when processing high-dimensional real-world data.

To tackle these issues, in this chapter, *FedMRD*: a novel edge-cloud based *p*rivacy-preserving *Fed*erated *M*ultimodal learning framework is designed for smart *R*oad *D*anger detection and warning. It is a federated multimodal learning framework leveraging the advantages of edge-cloud computing, federated learning, and differential privacy. FedMRD significantly reduces latency by detecting road dangers on edges similar to EcRD and FedRD. In FedMRD, both visual and textual modalities are utilized for model training achieving a much higher detection performance than EcRD and FedRD. Besides, it can collaboratively and efficiently learn from multiple edges while keeping most of the data locally on users' devices, which is more private. Further, the privacy of road users (e.g., IDs and locations) and privacy inside collected multimodal data (e.g., locations, people's faces, and license plate numbers) can be protected by the proposed multimodal differential privacy technique.

4.2 Contributions

The contributions of this chapter, which are under review in the IEEE Transactions on Industrial Informatics in 2021), are as follows:

- A novel edge-cloud computing-based Federated Multimodal learning framework, FedMRD, is proposed for privacy-preserving Road Danger detection and warning. FedMRD utilizes the recent advances of edge-cloud computing, multimodal data learning, federated learning, and differential privacy to meet the requirements of road danger detection and warning while preserving data privacy.

- An advanced Multimodal Road Danger Detection model, MRDD, is developed to improve the performance of image/video-only based solutions. MRDD leverages the advantages of the triplet loss to learn the inter-and intra-class relations, which boosts road danger detection performance, making it different from previous multimodal models.

- A novel communication-efficient and effective Federated Multimodal Learning algorithm, FedML, is proposed to collaborative learning from multiple platforms without requiring direct data sharing between edges and the cloud. FedML ensures robust and accurate road danger detection for edges with non-iid and imbalanced data. It has significantly less computation and communication costs than existing federated learning methods and converges well on such challenging data.

- A new Multimodal Differential Privacy technique, MDP, is introduced to preserve data privacy while ensuring high performance. MDP protects sensitive data on devices before uploading to untrusted edges, which preserves data privacy with minimized time and effort. Different from existing differential privacy techniques, MDP can handle both image and text modalities data and drastically alleviates the curse of high-dimensionality of local differential privacy.

The rest of the chapter is organized as follows: The related literature is reviewed in Section 4.3. Following it, Section 4.4 elaborates the design goals, architectural components, and the detailed working process of the proposed FedMRD framework. Then, the proposed MRDD algorithm, FedML strategy, and MDP technique are explained in detail in Section 4.5. Then, in Section 4.6, details of the experiments are shown, i.e., experimental setup, datasets, baselines, evaluation results, and detection result display. Finally, Section 3.7 concludes this chapter.

4.3 State-of-the-art

Road damage classification techniques: The state-of-the-art road damage classification algorithms can be classified into two categories: traditional hand-craft features-based methods and end-to-end deep learning-based methods. For the first category, robust hand-crafted feature representations combined with machine learning algorithms are widely used for road damage classification. [90] used Gabor filters and AdaBoost algorithm to classify road cracks. For the second category, Deep Convolutional Neural Networks (DCNNs) based approaches are mainly used. For example, in [91], a DCNN model is employed. To address the challenge of street-view images, the authors of [92] proposed an innovative method for large-scale road damage danger detection using a DCNN. Also, in [3,7] DCNN-based models are used to classify road damages.

Deep learning-based approaches have achieved promising results on image data for cluttered real-world environments. However, existing road damage detection systems are mainly based on visual data. To my best knowledge, there is no existing work considering multimodality data (e.g., image and text) in this field, while extensive researches in other fields have proven that data in different modalities can significantly boost the classification performance, e.g., [93–96]. To learn from such data, a direct way is to fuse features from different modalities. For example, the authors of [97] used both early fusion (i.e., fuse with features from early layers) and late fusion (i.e., fuse with each final decision) for better performance. In [98], a real-time classification method was proposed to learn from both image and text in Twitter posts by analyzing the Spatio-temporal metadata for flooding events monitoring. In [99], a multimodal deep learning system was introduced to recognize infrastructure damages from texts and images in social media messages. The authors of [96] proposed a multimodal framework with a cross-attention module for crisis events categorization.

Federated learning strategies: In 2016, FedAvg was proposed by [52] to train a global model by averaging a set of local models while keeping training data locally. The bottleneck of this form of collaborative learning that prevents federated learning (FL) from being widely used in IoT is massive communication costs caused by frequent model transmitting between clients and the parameter server during training. As studied by [100], the communication overhead for one client can soar to a petabyte to train a machine learning model on big datasets. Substantial approaches have been proposed to address this issue, which can be roughly categorized into two classes, i.e., communication frequency reduction-based and model compression-based methods. Communication frequency reduction-based approaches reduce model transmitting frequency. For example, in [52,81], each client updates its local model for several iterations before communicating instead of uploading it after every iteration. Communication frequency reduction-based methods can significantly reduce both upstream and downstream communication costs. Compression-based methods reduce communication costs by directly decreasing model size before communicating. Quantization [101] is one of the most utilized techniques to reduce model size. Quantization methods reduce communication costs by mapping an update into a smaller set, i.e., a smaller range of possible values than the update. For example, signSGD proposed by [102] is a quantization-based compression method that quantizes each gradient change to its binary sign, hence shrinking the model size by a factor of ×32 while ensuring convergence on iid data theoretically. SignSGD also considered downstream compression by integrating received binary updates by majority vote. Researchers also proposed stochastic gradient quantization methods to unbiasedly compress upstream gradients, e.g., quantized stochastic gradient descent (QSGD) [101]. QSGD can effectively quantize gradients and ensure convergence.

In the real-world, data collected by each user is usually non-iid. Unlike iid data, non-iid data are very challenging for deep learning models to learn. As stated in [84], the accuracy of federated learning drops significantly (up to 55%) for deep learning models trained on non-iid data. [84] proposed a strategy that creates a small dataset and sends it to all edges devices to learn from such data. This strategy alleviates the non-iid problem; however, it is hard to make clients' data iid since the data distributions of the clients are unknown to others. A more straightforward way is to learn directly from non-iid data [103–105]. The authors of [103] addresses the issue by proposing a modified FL which clusters clients hierarchically according to the similarity between their local models and the global model. The chapter stated that it shows a faster convergence speed on non-iid data than without the strategy. In [104], Asynchronous Online Federated Learning (ASO-Fed) for the non-iid setting was developed where devices collect data continuously while the local models were trained on the local datasets online. It learns inter-device relationships by regularization and a feature-learning model. The work of [105] tackled the catastrophic forgetting problem of FL by adding a penalty term to the loss function of the model forcing all local updates to converge to a global optimal. Moreover, [82,104] proved that data heterogeneity reduces model convergence speed. In particular, the learning rate of FedAvg must decay for the model to converge to the global optimum on non-iid data according to [82].

Privacy preserving techniques: Nowadays, privacy tends to become one of the main concerns when people use a service. This concern holds even stronger when IoT devices collect private information from individuals without explicitly informing them. Various approaches have been proposed to protect privacy, for example, encryption algorithms, anonymization, and differential privacy. Encryption algorithms were usually used to prevent malicious acquisition, misuse of data, and reverse attacks (i.e., inferring models' parameters) from the untrusted edges/clouds, e.g., [106] and [107]. However, the high computation and communication costs hinder their applications in real-world settings. [108] protects the users' privacy by the pseudonym and pseudonym certificate in fog computing. [109] utilized the trusted third party as the middle layer to accomplish users' anonymity. However, [110], and [111] proved that removing or faking the identities information of users may not protect users' privacy since de-anonymization attacks can infer this information with the existing knowledge.

Unlike anonymization and Encryption algorithms, differential privacy [56] with a strong privacy guarantee can ensure both users' privacy and model's performance. LDP is one kind of DP that can directly protect user data from portable smart devices used by users, such as mobile phones and watches. Therefore, LDP can protect users' privacy without a trusted authority (e.g., untrusted edge or cloud). The randomized response strategy was used to the encoded value in [85] and [86] to protect users' privacy locally. They are easy to carry out without additional calculation cost but have poor performance with high dimensional data. [87] and [88] employed the Expectation Maximisation (EM) based approaches that split the privacy budget for each attribute's value to protect local users' privacy for two attributes and multiple attributes data, separately. EM-based approaches may lead to a high variance because of the splitting privacy budget, which is inappropriate for high-dimensional data. The authors of [89] used the transformation technique to encode the data to a binary string. The randomized response strategy was employed to produce the string. Then, the closest center was sent differential privately. [89] can preserve the clustering information of users but is not suitable for data with high dimension and has extra cost to select the cluster center.

In summary, firstly, end-to-end deep learning-based methods have been widely explored for road damage classification and achieved promising results. However, most of them are only based on a single modality (i.e., image/video). Given the massive amount of multimodality data generated every day, single modality models' performance can be further enhanced by multimodality data. However, no existing work considers multimodal learning for road damage danger detection to the best of my knowledge. Secondly, for the collaborative learning part, communication frequency reduction-based methods are simple but effective to reduce upstream and downstream communication costs; however, the communication cost is still very high due to large model sizes. Thirdly, there are no existing LDP that handle real-world multimodal data with multiple attributes, and there is a large error when the data dimension is high. To tackle these issues, in this chapter, a communication-efficient and privacy-preserving federated multimodal learning framework is developed for road danger detection and warning.

4.4 FedMRD Framework

This section introduces the design goals, architectural components, and the working process of the proposed FedMRD framework.

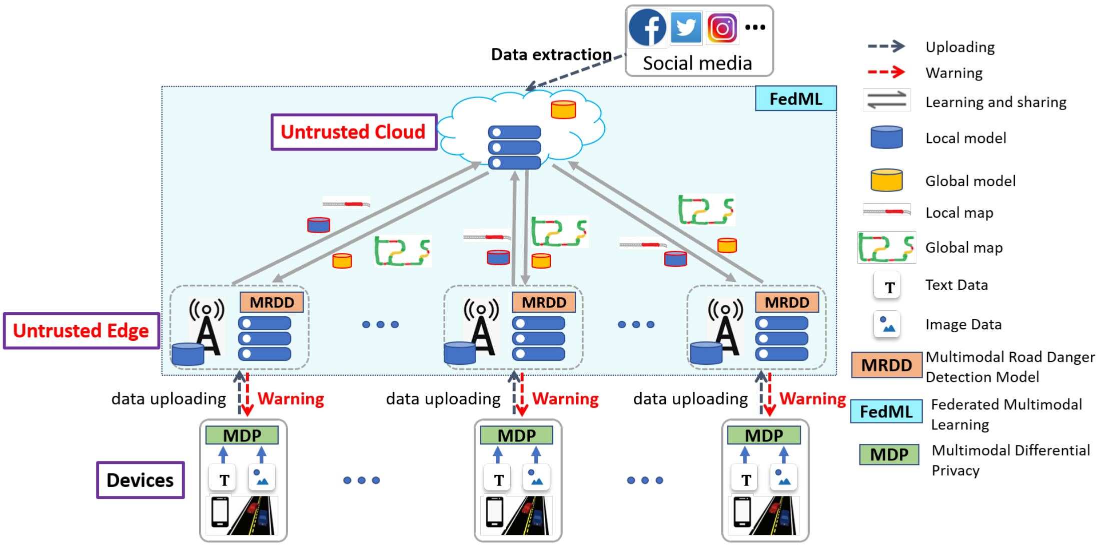


Figure 4.2: Overview of the FedMRD framework.

4.4.1 Design Goals

The design of the proposed FedMRD framework is motivated by the following goals:

1. **Safety:** It is crucial for a road danger detection and warning system to provide users with dangerous information for accident prevention, thus ensuring safe traffic.

2. **Robustness:** A road danger detection and warning system should be able to handle unpredictable and unexpected complications arising from the surrounding environment, e.g., different weather conditions, illuminations, shadows, and obstacles like vehicles and pedestrians.

3. **Accuracy:** A road danger detection and warning system should predict dangerous road dangers accurately to avoid accidents.

4. **Latency:** Fast detection and warning of hazardous road dangers are critical to ensure traffic safety. If drivers cannot receive road dangerous warning information timely, serious traffic accidents would happen when drivers are hitting or trying to avoid the dangers. Therefore, hazardous road danger detection tasks should have low latency.

5. **Resources:** Since road video analysis is a high resource-demanding task and the data size quickly increases as time goes, and cloud or edge storage is expensive. Therefore, designing an efficient storage strategy is critical.

6. **Communication Cost:** Due to the high cost of video data transmission, Internet communication, and bandwidth, the newly developed system needs to consume low communication costs and bandwidth to apply in the real-world.

7. **Privacy:** Because of the open-access transmission environment and untrusted edges and clouds, there is a high privacy leakage risk of the collected road information, including users' sensitive information (e.g., users IDs and locations) and private data inside collected data (e.g., people's faces and license plate numbers). Thus, the new framework should protect data privacy well.

4.4.2 Architectural Components

The proposed FedMRD framework is composed of the following four main components as illustrated in Figure 4.2.

- Devices: Devices collect multimodal data by users and upload it to the nearest edge (e.g., RSUs).

- Edges: Edges are responsible for receiving data from users and rapidly responding to potential road dangers in the data. In particular, the Multimodal Road Danger Detection model (MRDD) is deployed on edges for fast road danger detection. The MRDD model trained on the edge's dataset and the warning message (e.g., types and locations) of the detected road dangers is called a local model and a global map, respectively. Edges are untrusted in this setting.

- Cloud: The cloud serves as a parameter server for the federated learning process and some extra space for data and model storage. There is a global model on the cloud, which is an aggregation of the received local models. The global map on the cloud is computed by summing up the received local models and displaying them on a Google map in real-time. Moreover, road danger-related data (images and texts) are also crawled from the Internet to accelerate model learning. The cloud is also untrusted.

- MRDD: MRDD is a deep learning-based multimodal model which takes both image and text modalities as inputs and recognizes road dangers. Section 4.5.1 and Section 4.5.2 depict the MRDD1 and MRDD2 models in detail.

- FedML: The Federated Multimodal Learning strategy (FedML) improves the performance of road danger inspection by collaborative learning between edges and the cloud. The design of FedML aims to significantly reduce communication costs while maintaining high accuracy and stable model convergence. More details of FedML can be found in Section 4.5.3.

- MDP: The Multimodal Differential Privacy technique (MDP) protects both users' and collected data' privacy on users' devices before sending to nearby edges. It is an improvement of LDP, alleviating its limitation of high expectation error for high dimensional real-world data.

4.4.3 Working Process

As shown in Figure 4.2, FedMRD is built on the device-edge-cloud structure where devices are used for data collection, edges are employed to reduce latency, and the cloud is leveraged for parameter integration. The proposed MRDD is deployed on edges for fast response. Once any road dangers are detected, edges directly send warning messages to users for accident-preventing. Similar to [52], federated learning is used to collaboratively learn from multiple edges with the assistance of a cloud server. Federated learning strategies enable robust model learning without requiring direct data sharing from edges to clouds which, to some extent, protects data privacy from untrusted clouds. In FedMRD, the detection models deployed on edges are defined as local models, while the detection model on the cloud is denoted as a global model. Both the global model and local models share the same structure. The working process of FedMRD mainly consists the following four phases that repeat periodically to learn from data on decentralized edges gradually.

- **Phase 1:** Each vehicle collects image or text road danger information by carrying smart devices and then directly transmits it to the nearest edge (e.g., a RSU).

- **Phase 2:** Each edge detects road dangers within its communication range by the MRDD model (received from the cloud). Then, it broadcasts danger warning messages to all users/vehicles covered. The edge starts training its local model on its local dataset if its accumulated new data exceeds a certain threshold (it is set to 100 images in this chapter after extensive testing). Afterward, it sends the trained parameters of its local model to the cloud. The edges without enough new data are not trained to reduce their computation cost. Also, these local models are not transmitted to the cloud to reduce communication costs.

- **Phase 3:** Each cloud aggregates local models received from connected edges by Eq. (4.1) to generate a global model.

  $$\mathbf{W}^t = \sum_{i=1}^{N} \frac{D_i}{D} \mathbf{W}_i^t, \tag{4.1}$$

  where $\mathbf{W}^t$ is the global model's parameters at time t, $\mathbf{W}_i^t$ is the i-th local model's parameters at time t. D_i and D are number of training images on the i-th edge and the total number of training images over all edges participated in the training process ($D = \sum_{i=1}^{N} D_i$). N means the number of edges that transmitting their local models to the cloud. After it, the cloud sends the global model back to all the covered edges.

- **Phase 4:** Edges update their local models with the received global model.

Phases 1 to 4 are repeated in every E training epochs (it is found that the global model converges fast when $E = 10$ through extensive experiments).

Table 4.1: Summary of notations.

Notation	Description
$\mathbf{w}_t$	Global model's parameters at time t
$\mathbf{w}_i^t$	Local model parameters on i-th edge at time t
D_i	No. of training samples on the i-th edge
D	Total No. of training samples over all participated edges
N	No. of edges evolved in FL
E	Local training epochs
R	Global training rounds
f_i	Image feature extractor
f_t	Text feature extractor
M	Merging Model
L	Combined triplet loss
a_t, p_t, n_t	Anchor text, positive text, and negative text
a_i, p_i, n_i	Anchor image, positive image, and negative image
m	Margin
$d(x, y)$	Distance between x and y
η_0, η_r	Initial Learning rate and decayed learning rate
R_l, s	Previous global round and step size
B	Batch size
x_i^t	Model difference
w^{t-1}	Previous global model
$Q(\mathbf{x}_i^t)$	Low-Precision Quantizer
T	Time interval for road danger detection
K	No. of local updates received by the cloud
I, T	An image and a text vectors
d	Dimension of the input
λ	Scale of Laplace distribution
$1/\epsilon$	Privacy budget
$P_{c\times d}, Q_{d\times e}$	Random matrixes
δf	L1-sensitivity

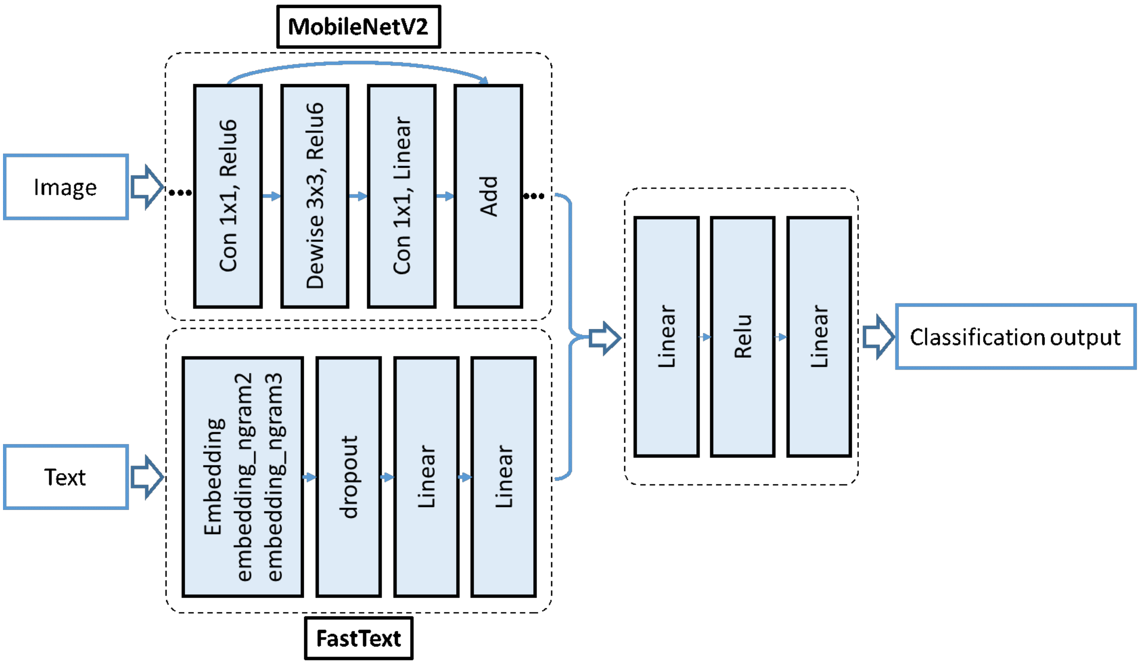


Figure 4.3: Multimodal Road Danger Detection Model - Solution 1.

4.5 Methodologies

In this section, the proposed MRDD algorithm, FedML strategy, and MDP technique are explained in detail.

4.5.1 Multimodal Road Danger Detection Model - Solution 1

The proposed Multimodal Road Danger Detection Model 1 (MRDD1) is designed to detect road dangers given image-text pairs as inputs, i.e., image-text from road users when encountering road dangers. As shown in Figure 4.3, MobileNetV2 [112] [1] and FastText [113] [2] are employed for image and text feature exaction for their high speed and effectiveness. The proposed model includes an image extractor and text extractor to exact feature maps from images and texts. Then, image and text features are fused to get the final classification output.

MobileNetV2 is used as the image extractor, which is light-weighted and particularly suitable for time-sensitive and memory-efficient tasks like road damage danger detection and warning on edges. For each image m_i, its feature vector is defined as:

$$I_i = \textbf{MobileNetV2}(m_i), \tag{4.2}$$

[1] Code available at https://github.com/tonylins/pytorch-mobilenet-v2

[2] Code available at https://github.com/facebookresearch/fastText

where I_i represents a vectorized image feature map from MobileNetV2.

FastText is utilized as the text extractor, which is also specially tailored for this task. FastText is built upon circumventing quantization to save word embeddings. It learns words based on the surrounding context with an input of raw text. Different from word2vec, it assumes a word as n-grams of characters with a changeable n from one to the length of the word, which benefits the learning process by enabling to find word representations that are not directly in the dictionary. For each text sample t_i, its feature representation is denoted as:

$$T_i = \textbf{FastText}(t_i), \tag{4.3}$$

where T_i is a text embedding vector. Both image and text features are from the last fully connected layer of the extractors to obtain abundant and high-level information. To obtain qualified feature representations of images, MobileNetV2 is pre-trained on the ImageNet dataset and then fine-tune it on the collected dataset.

After obtaining image feature representations I_i and text representations T_i, the feature maps are fused by simply element-wise adding I_i and T_i. The fused features are then fed into the classification layer to decide whether the input is dangerous or not and which danger type it is. The architecture of MRDD is shown in Figure 4.3. In many multimodal-learning tasks, different modalities contain different information. Fusing them increases the learning speed and the classification performance. The model makes the best use of knowledge from different modalities and achieves outstanding performance.

4.5.2 Multimodal Road Danger Detection Model - Solution 2

The proposed MRDD2 is designed to detect road dangers given image or text as inputs, i.e., image/text from road users when encountering road dangers. Unlike MRDD1, MRDD2 does not require image-text pairs, which is more convenient to use in practice. The structure of the proposed multimodal model consists of two phases: pre-training phase and re-training and testing phase, as shown in Figure 4.4.

A multimodal triplet loss-based model is trained during the pre-training phase to learn differentiable feature representations by distance comparisons. This phase aims to train image and text feature extractors (I_i and T_i) for distinguishing road dangers and normal roads and different types of road dangers given image/text inputs. Cosine similarity is utilized to measure embeddings distances defied by

$$d(x, y) = \frac{x \cdot y}{max(\|x\|_2 \cdot \|y\|_2, \epsilon)}, \tag{4.4}$$

where $\epsilon = 1e - 6$ is a small value to avoid division by zero. Cosine similarity is utilized because it captures the orientation of the inputs instead of the magnitude and different images and texts that belongs to the same class may have a small angle but have a large magnitude difference to each

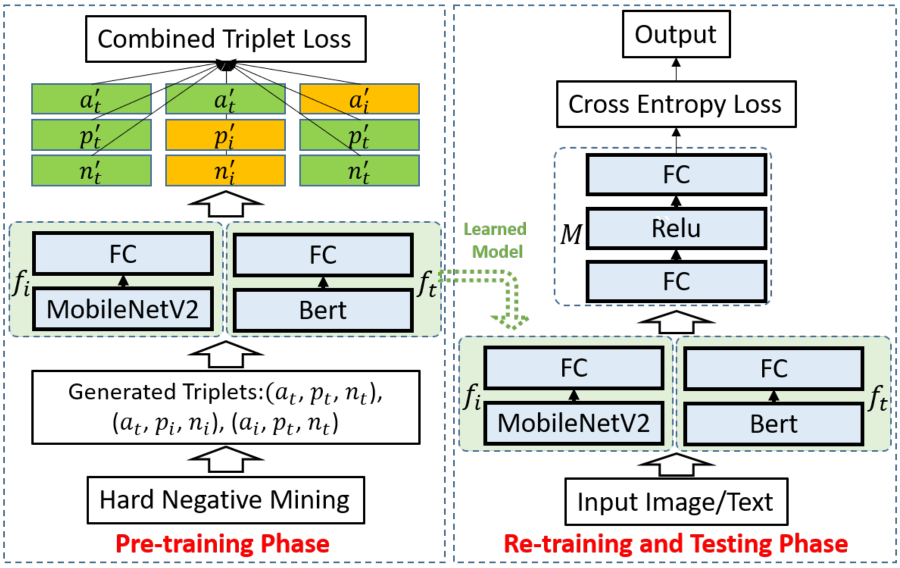


Figure 4.4: Multimodal Road Danger Detection Model - Solution 2.

other. The smaller the angle, the higher the similarity.

The multimodal triplet loss consists of image feature extractor f_i (MobileNetV2+FC) and text feature extractor f_t (Bert+FC) to exact feature embeddings from input images and texts, where FC means a Fully-Connected layer. A combined triplet loss is designed in (4.5) for the model training taking into account both inter-class and intra-class relations, where λ is a penalty factor that controls the importance of the term, and experiments show that $\lambda = 0.1$ is optimal. The reason behind it might be that text modality inputs contain high-level features while image modality inputs only include low-level features. If the text-only loss is not penalized, the combined loss would be too high for updating model weights. The combine triplet loss consists of basic triplet losses for text-only (4.6), text-against-image (4.7), and image-against-text (4.8). The basic triplet loss is defined in (4.9) following [114] where where m is a margin that could be different for different distance types. In the experiments, m is set to 0.2 for all experiments.

$$L = \lambda \cdot L(a_t, p_t, n_t) + L(a_t, p_i, n_i) + L(a_i, p_t, n_t), \tag{4.5}$$

$$L(a_t, p_t, n_t) = max\{d(a_t, p_t) - d(a_t, n_t) + m, 0\}, \tag{4.6}$$

$$L(a_t, p_i, n_i) = max\{d(a_t, p_i) - d(a_t, n_i) + m, 0\}, \tag{4.7}$$

$$L(a_i, p_t, n_t) = max\{d(a_i, p_t) - d(a_i, n_t) + m, 0\}, \tag{4.8}$$

$$L(a, p, n) = max\{d(a, p) - d(a, n) + m, 0\}, \tag{4.9}$$

Optimizing the combined triplet loss over different triplets is computationally inefficient. Hence, the image and text feature extractors (f_i, f_t) are optimized using Adam optimizer within each batch. Inspired by [115,116], the most violating negative samples are selected in each batch. Specifically, feature embeddings of three triplets $(a_t, p_t, n_t), (a_t, p_t, n_i), (a_i, p_t, n_t)$ are selected in each batch, which selects the hardest negative for training in every batch, where a, p, n represents anchor, positive, and negative samples, and i, t means image and text, respectively. In particular, if the distance (d) between an anchor and a negative sample is less than any other negative samples in a batch (the batch size is 16 due to computation constraints), then that one must be chosen. This process is denoted as a hard negative mining strategy, as shown in Figure 4.4. Theoretical guarantees of this data sampling approach are analyzed in [116]. Different from the proposed approach, the work of [117] performed both hard negative and positive mining for deep structure-preserving image-text embedding learning. In the experiments, the model converges within 40 epochs on average. However, the experiments show that the result is not improving, and it takes high computational costs.

During the re-training and testing phase, a multimodal cross-entropy loss based model consisting of the pre-trained image and text feature extractor (f_i and f_t) and merging model (M) is firstly re-trained on the training set for road danger classification, and then it can be used for real-time road danger classification. The merging model M comprises two FC layers and one Rectified Linear Unit (Relu) layer. Similarly, the Adam optimizer trained the model with the learning rate $1e-5$, batch size 16, and the model converges in about 40 epochs. In Section 4.6.3.1, the performance of the proposed model with and without pre-training by the combined triplet loss is discussed.

4.5.3 Federated Multimodal Learning Strategy

The Federated Multimodal Learning strategy (FedML) is proposed to achieve high detection accuracy on different edges with low communication costs while guaranteeing convergence on non-iid and imbalanced data. FedML mainly consists of three parts, i.e., periodic averaging, learning rate decay (LRDecay), and model quantization.

4.5.3.1 Periodic Averaging

By using FL, local models $\mathbf{w}_i^t$ on edges train locally with their collected data and update iteratively with the aggregated global model $\mathbf{w}^t$ from the cloud server. One approach is to upload the local models from edges to the cloud in every iteration of local training for global aggregation. However, the frequent model exchange between edges and the cloud would incur a very high communication cost. Instead, local models are trained on their local datasets for a certain number of local epochs E when enough new data D_i is collected before sending to the cloud. Specifically, firstly, local models only start (re-)training when the edges have sufficient newly collected data $D_{Threshold}$ which prevents frequent local model training on edges. Secondly, by only uploading the local models trained on adequate new data and periodically transmitting local models to the cloud after E local iterations, the communication rounds R between edges and the cloud reduce significantly, and it is the same for the overall communication cost. For example, if local models need to update 500 iterations to reach a high accuracy, the communication rounds are $R = 500$ when the local models upload to the cloud after every local epoch ($E = 1$); the communication rounds are $R = 500/10 = 50$ when $E = 10$, thus, the communication cost is reduced by 10 times. From the example, we can see that when the local epoch E is large, the communication cost reduces significantly. However, with the increase of E, the higher the possibility of the local models reaching their local optimums instead of the global optimal over all the data from edges. Consequently, it takes more global communication rounds R to achieve a certain accuracy than low E.

4.5.3.2 Learning Rate Decay

Despite the outstanding performance of the existing FL strategies [54, 81–83], some may fail to converge, especially on the real-world data from edges which are often non-iid and imbalanced, and only a part of edges are active at a specific time. To tackle this issue, the learning rate in FedML is decayed by (4.10) at the end of each global round inspired by [82].

$$\eta_r = \gamma^{(}\left\lfloor \frac{1}{(R_l + 1) \cdot s} \right\rfloor), \tag{4.10}$$

where η_0 is the initial learning rate which is 0.1 in the setting. γ equals to 0.5. R_l and s means the last global round and step size, respectively. The s is set to 1. At each time step t, k active edges (with abundant data) train their local models by Stochastic Gradient Decent (SGD) in parallel and upload the trained local models to the cloud after E local epochs.

Decaying learning rates is critical for the convergence of FedAvg with non-iid data [82]. Theoretically, FedAvg has full gradient when $E = 1$ and $B = len(\text{trainDataset})$. In this case, FedAvg can converge to the global optimum with a fixed and appropriate learning rate [118]. However, when $E > 1$, FedAvg may not converge to the global optimum even with $B = len(trainDataset)$. It will converge to a local optimum neighboring to the global optimum with a fixed learning rate according to [82]. A possible explanation is that FL learns a global model distributively from the connected edges while the training datasets on edges are usually non-iid and highly imbalanced in the real-world; thus, the trained local models can be biased. Only a sub-optimal global model can be learned with such local models when choosing a constant learning rate η and periodically updating local models $E > 1$ local epochs. However, the bias can be gradually erased by a decaying learning rate. With LRDecay, the global model in FedML converges well, and its learning speed is not slowed down when the data on edges are non-iid and imbalanced, and only partial edge participate for training.

4.5.3.3 Model Quantization

One bottleneck of FL is the high bandwidth cost caused by frequent model communication between edges and the cloud. In FL settings, typically, the whole gradients/models are communicated between edges and the cloud. It means that each edge must send and receive around $614\,\mathrm{MB}$ or $449\,\mathrm{MB}$ every round when using MRDD1 and MRDD2. Hence, it is a huge barrier preventing FL from being used in practical applications. In the FedML setting, each participated edge only communicates the quantized model differences between the previously received global model and the updated local model to the cloud instead of the local model to reduce communication costs. Low-Precision Quantizer (LPQ) [101] (see (4.11)) is utilized to compress the model differences since it has convergence guarantees and good practical performance.

$$\mathbf{x}_i^t = \mathbf{w}_i^t - \mathbf{w}^{t-1},$$

$$Q(\mathbf{x}_i^t) = \left\|\mathbf{x}_i^t\right\|_2 \cdot sign(x_j) \cdot \xi_t(\mathbf{x}_i^t, s), \tag{4.11}$$

$$\xi_t(\mathbf{x}_i^t, s) = \begin{cases} (l+1)/s & \text{with probability} p(\frac{|x_j|}{\|\mathbf{x}^t\|_2}, s), \\ l/s & \text{otherwise.} \end{cases}$$

Here, $\mathbf{w}_i^t$ and $\mathbf{w}^t$ are the local model of edge i and the global model $\mathbf{w}^t$ received by the edge at time t while $\mathbf{x}^t$ is the difference between them. Also, x_j is an element of $\mathbf{x}^t$, $l \in [0, s)$ is an integer, and $s \geq 1$ is quantization level.

After the training of local models and the aggregation of the local models or before transmitting the local or global models, the dynamic quantization method [3] is applied to the models to reduce model sizes further. The key idea with dynamic quantization is to dynamically determine the scale factor of the parameters during running so that as much signal is preserved for the dataset.

Overall, edges and the cloud perform FL for road danger detection in every T time slot. Within time T, only time $t \ll T$ is utilized for FL learning while the edges and the cloud are free for other services during $T - t$. Every time t, the proposed FedML performs R global rounds, and each edge updates E local epochs during one global round. In every global round, the edges with $D_i^t - D_i^{t-1} \leq D_{Threshold}$ start training their local models by SGD and send the quantized difference between the current local model and the latest global model received $Q(\mathbf{x}^t)$ to the cloud for aggregation. The cloud wakes up at the beginning of every time slot t and waits until K local updates are received. Then, the cloud aggregates the collected local updates by Eq.(4.12) and

[3] https://pytorch.org/tutorials/recipes/recipes/dynamic_quantization.html

Algorithm 5: FedML strategy

```
Input   : Local datasets on edges and detection model MRDD
Output: Optimal local models on edges
cloud initialize $\mathbf{w}^0$ and transmit it to all edges
cloud initialize learning rate $\eta_0$
for every global round $R$ do
    for every $E$ epochs do
        for each edge $i \in \{1, 2, \cdots, K\}$ do
            edge updates its local model $\mathbf{w}_i^t$ with the received global model $\mathbf{w}^{t-1}$
            edge train $\mathbf{w}_i^t$ on its newly collected local dataset by:
             $\mathbf{w}_i^t \leftarrow \mathbf{w}_i^{t-1} - \frac{\eta_0}{R+1} \bigtriangledown l(\mathbf{w}_i^{t-1}, b_i^{t-1})$
            calculate the difference/update by: $\mathbf{x}_i^t = \mathbf{w}_i^t - \mathbf{w}^{t-1}$
            edge send $Q(\mathbf{x}_i^t)$ to the cloud
            $R = R + 1$
        end
    end
    cloud wait until $K$ local updates received
    cloud aggregate the local updates by: $\mathbf{w}^t = \mathbf{w}^{t-1} + \frac{1}{K}\sum_{i=1}^{K} Q(\mathbf{x}_i^t)$
    cloud send the global model $\mathbf{w}^t$ to all edges
end
```

repeats it for R times. The proposed FedML is formally written in Algorithm 5.

$$\mathbf{w}^t = \mathbf{w}^{t-1} + \frac{1}{K}\sum_{i=1}^{K} Q(\mathbf{x}_i^t). \tag{4.12}$$

4.5.4 Multimodal Differential Privacy Technique

In this chapter, the setting is device-edge-cloud based, where edges collect data from users and detect road dangers while the cloud stores road danger data. In the setting, edges and the cloud are untrusted. The goal of the Multimodal Differential Privacy technique (MDP) is to minimize its negative influence on the accuracy of the MRDD model while largely preserving data privacy. MDP is a variant of LDP deployed on users' devices to perturb data before transmitting to untrusted edges. Notably, users' data is changing over time. It is necessary to ensure that the collection of data at each time slot satisfies LDP.

Given an image I or text T collected by a user, Laplace Mechanism [119] is applied to perturb them, which is the typical distribution for ϵ-LDP. Particularly, the perturbed image and text are defined as:

$$\forall j \in [d], I^*[j] = I[j] + Lap(\frac{2d}{\epsilon}), \tag{4.13}$$

$$\forall j \in [d], T^*[j] = T[j] + Lap(\frac{2d}{\epsilon}), \tag{4.14}$$

where $Lap(\lambda)$ means a random variable sampled from a Laplace distribution with the scale λ. The following is its probability density function:

$$PDF(x) = \frac{1}{2\lambda} exp(-\frac{|x|}{\lambda}). \tag{4.15}$$

The expected error incurred by adding Laplace noise to the data on users' devices is $O(\frac{d}{\epsilon})$, which is excessively high when data dimension d is high. To tackle this issue, the dimension of each users' image/text data is reduced before adding noise.

According to [120], important characteristics can be preserved by projecting an input vector into a random small-dimensional sub-space. However, this method can only reduce the dimension by up to $\sqrt{d}$ times, which is still very large when d is large. Half of the dimension is reduced further by projecting the data into a smaller subspace while not loss key features of the data to address this issue. Specifically, texts are firstly tokenized into vectors. Then the dimension of the text and image is reduce by multiplying with random matrixes $P_{c \times d}$ ($c < d$) and $Q_{d \times e}$ ($e < d$). The matrixes are generated by edges. Each item of P and Q, i.e., $P[i][j]$ and $Q[i][j]$, are defined as:

$$P[i][j] = Q[i][j] = sign(x) \times \frac{1}{e}, \tag{4.16}$$

where x is uniformly sampled from $U\{-1, 1\}$ and e is dimension of the output data. Therefore, given an image I and text T, the perturbed image is $I = Tanh(P \times I \times Q)$, and the perturbed text is $T = Tanh(P \times T)$ if the text dimension is high. Since text vector T only has one dimension and is small (i.e., 32) for the collected dataset, the dimension of T is not reduced. Then, this random matrix is multiplied with the data vector, and $tanh$ function is applied on the multiplication output producing the data with reduced dimension. For example, if the dimension of the image and text vectors are with 128×128 dimension. The dimension can be reduced to 1×64 by multiplying it with P and Q obtained from Eq. (4.16). The following MDP then perturbs the data after reducing dimension.

ϵ-LDP is defined following [56]:

Definition 1 ($\epsilon - LDP$). *A randomized function f satisfies $\epsilon - LDP$ where $\epsilon > 0$ and only if for any inputs x and x'*

$$Pr[f(x) = x^*] \leq exp(\epsilon) \cdot Pr[f(x') = x^*], \tag{4.17}$$

where $Pr[\cdot]$ denotes probability and ϵ means privacy budget which indicating the amount of noise added to the data. Basically, the smaller the ϵ, the more noise is added; accordingly, the more private the data while the lower the accuracy.

Based on this definition, edges that receive the perturbed data x^* cannot distinguish whether the true data is x or x' with high confidences (controlled by the privacy budget ϵ), regardless of how much the background knowledge that the untrusted edges have.

Algorithm 6: MDP

```
Input   : image $I$ or Text $T$ with dimension $d$, privacy budget $\epsilon$
Output: perturbed image and text vectors $I^*, T^*$
generate random matrixes $P_{c\times d}$ and $Q_{d\times e}$ where each entry of them has an equal
  probability to be $1/e$ or $-1/e$
reduce dimension of $I$ or $T$ by
$I'_{c\times e} = Tanh(P_{c\times d} \times I_{d\times d} \times Q_{d\times e})$
$T'_c = Tanh(P_{c\times d} \times T_{d\times 1}) \leftarrow$ only if text dimension is high
for $j = 1, 2, \cdots, d$ do
    $I^*[j] = I'[j] + Lap(\frac{2d}{\epsilon})$
    $T^*[j] = T'[j] + Lap(\frac{2d}{\epsilon})$
    Return $I^*, T^*$
end
```

To achieve LDP, Laplace noise proportional to the sensitivity of the output is added to the data. The sensitivity measures the maximum change of the output by adding noise while not revealing privacy. $L1$-sensitivity is used which is defined as:

$$\Delta f = max\{\|f(x) - f(y)\|_1\}, \tag{4.18}$$

where $\|.\|_1$ is the L1 norm.

Here, the proposed randomized function f is proved to satisfy $\epsilon - LDP$.

Proof: Let x, x' be two data points with dimension d, and f is a randomize function using noise from $Lap(0, \frac{\Delta f}{\epsilon})$.

$$\begin{aligned}\frac{Pr[(f(x) = x^*]}{Pr[(f(x') = x^*]} &= \prod_{i=1}^{d} \frac{exp(-\frac{\epsilon|f(x)_i - x_i^*|}{\Delta f})}{exp(-\frac{\epsilon|f(x')_i - x_i^*|}{\Delta f})}, \\ &= \prod_{i=1}^{d} exp(\frac{\epsilon(f(x')_i - x_i^*| - |f(x)_i - x_i^*|)}{\Delta f}), \\ &\leq \prod_{i=1}^{d} exp(\frac{\epsilon|f(x)_i - f(x')_i|}{\Delta f}). \\ &= exp(\frac{\epsilon\|f(x) - f(x')\|_1}{\Delta f}), \\ &\leq exp(\epsilon).\end{aligned} \tag{4.19}$$

Thus,

$$Pr[f(x) = x^*] \leq exp(\epsilon) \cdot Pr[f(x') = x^*]. \tag{4.20}$$

Notably, post-processing invariance is one of the essential characteristics of differential privacy.

Hence, all computations on edges based on the received data from devices are still under $\epsilon - LDP$.

4.6 Experiments and Evaluations

This section elaborates the experimental setup, dataset, and baselines first. Then, the results and evaluation of the proposed MRDD model, FedML strategy, MDP technique, and the FedMRD overall framework are presented.

4.6.1 Experimental setup and Dataset

Text and image pairs are collected from users' IoT devices. In the simulations, a cheap smartphone (Redmi Note 9S) is employed as an example. The resolution of images ranges from 100×100 to 1280×720, to simulate multiple devices. The collected data consists of annotated images and texts from six classes, i.e., road damages, traffic accidents, fallen trees, icy roads, and normal roads as shown in Figure 4.5. These types are the most common road dangers observed in our daily life. Data with different weather conditions, various illuminations, and obstacles like vehicles and pedestrians are incorporated to test the robustness of the proposed model. There are 780 images and texts in total. The 70%, 15%, and 15% of the dataset are used for training, testing, and validation, respectively. Before training, images are first resized to 256×256 for further processing, then cent-cropped to 224×224 for better image quality (e.g., no black noise on the edges). Also, texts are tokenized in word-level and padded with empty spaces if their length is shorter than 32.

FedMRD is also evaluated on the CrisisMMD public dataset [121] to prove the effectiveness of the proposed models. CrisisMMD is a multimodal crisis dataset including annotated image-tweet pairs collected during seven natural disasters in 2017. The humanitarian categorization task is considered in the experiments since it is similar to the proposed task. This task classifies the given image-tweet pairs into five classes, including Infrastructure and utility damage (1753), Vehicle damage (1187), Rescue, volunteering, or donation efforts (776), Affected individuals (injury, dead, missing, and found) (64), and Other relevant information (22). Similarly, 70%, 15%, and 15% of the dataset are utilized for training, validation, and testing, respectively.

Both MNIST [122] and collected datasets are utilized to evaluate FedML and comparing it with the state-of-the-art FL strategies. The MNIST dataset is the most popular dataset for evaluating FL strategies.

To simulate the proposed FedMRD framework, a cheap smartphone (Redmi Note 9S) is employed on the device; a computer (64-bit Windows 10 Operating system, 32 GB RAM, 1 TB of hard disk, and 1 GTX 1080 Ti GPU) is employed as the edge; a high-performance server (Ubuntu 18.04 LTS system, 64 GB of RAM, 2.5 TB of hard disk, and 2 GTX 1080 Ti GPUs) is utilized as the cloud. Python 3.8 is used in the implementation.

When training the MRDD model, the initial learning rate is set as $1e - 3$ and scheduled with the cosine annealing to rapidly decrease the learning rate to $1e-4$ or increase to $1e-3$ in 6 epochs. Also, Adam optimizer is utilized during training. The batch size and epochs are 16 and 250. Also, The early stopping strategy is leveraged.

When implementing FedML, the initial learning rate is set to 0.1. The number of edges is 30, a fraction of 0.1 participates for training in each global round. After extensive testing, the local epoch number E is set to 10, which means local models train 10 epochs before uploading. The global epoch number is 200 to monitor the convergence performance of the FL strategies. The datasets in the experiments are non-iid and highly imbalanced, i.e., among 30 edges, 1/3 edges have 1% of the training data, 1/3 edges have 30% of the training data, and the rest 1/3 edges have 69% of the training data. Also, each edge only contains three and six non-overlapping classes for the own dataset and MNIST dataset, respectively. The accuracy, precision, recall, f1-score, runtime, and latency are adopted to analyze the models' performance following the definition in [7].

4.6.2 Baselines

Several state-of-the-art unimodal models and multimodal algorithms for text and image classification are compared against the proposed MRDD1 and MRDD2 methods. MobileNetV2 [112], FastText [113], and Bert [123] are compared first. They are common light-weight unimodal image and text classification networks. MobileNetV2 is pre-trained on the ImageNet dataset, while FastText and Bert are not pre-trained. They are then fine-tuned on the collected road damage dataset.

Secondly, MRDD is compared with some recently proposed multimodal classification networks.

- Ofli et al. [93]: It utilizes a DCNN to define a multimodal architecture where a shared representation is used for feature fusion.
- Gallo et al. [94]: It encodes text onto an image directly to enrich the image and learns features with a standard DCNN.
- Choi et al. [124]: It is a robust deep multimodal fusion architecture where a terminal network is adopted for activity recognition.
- Pranesh et al. [95]: It is recently proposed to classify crisis events in social media and combine RoBERTa and ResNet50 with transformer attention.
- Abavisani et al. [96]: It employes Bert and DenseNet121 to extract features and utlizes Multimodal SSE for feature fusion.
- MRDD (own): It is the proposed model based on MobileNetV2 and Bert pre-trained on the dataset by using the triplet loss, which learns the intra- and inter-class relations.
- MRDD-noPretrain (own): It is a variant of MRDD. The only difference is that it does not utilize the weights of the MobileNetV2 and Bert that pre-trained with the triplet loss.

Thirdly, to evaluate the proposed FedML strategy, it is compared with three state-of-the-art FL strategies and two variants of FedML.

- FedAvg [81]: FedAvg is basic FL strategy. It updates local models by the global model calculated by averaging of the local models.

- FedPAQ [54]: It is a communication-efficient FL framework that uses periodic averaging and quantization.

- LRDecay [82]: This method proves that FedAvg converges slowly on heterogeneous data. Besides, it shows that decaying the learning rate at the end of each global round by $\eta_t = \frac{\eta_0}{1+t}$ improves the convergence of FedAvg, where $\eta_0 = 0.1$.

- FedML (own): It decays the learning rate to ensure that the model converges well on non-iid and imbalanced data. It is communication efficient since it only requires partial edge participation and compresses the model before transmitting for both upstream and downstream.

- FedML-Q (own): It is a variant of FedML without the model quantization method.

Finally, to measure the performance of the overall FedMRD framework, FedMRD is compared with existing high-performance frameworks, i.e., CloudRD [71], EdgeRD [26], EcRD [7], and FedRD [8].

1) CloudRD [71]: CloudRD deploys its detection model on the cloud server. This kind of system trains a detection model on the cloud, and users receive detected road danger information from the cloud.

2) EdgeRD [26]: EdgeRD deploys the detection model on edges for fast response. This type of system trains the model on edges and warns users about road dangers timely from edges.

3) EcRD [7]: It is an edge-cloud-based road damage danger detection and warning framework where the detection model is trained on edges and warning information is sent to users from edges for fast responses while the cloud server is only utilized for data storage and post-processing.

4) FedRD [8]: In FedRD, detection models are trained distributively over multiple edges via FL without direct data sharing between edges and the cloud. Besides, the privacy of the collected image data is protected by DP on users' devices.

4.6.3 Evaluation Results

This section will evaluate the proposed MRDD model, FedML strategy, MDP technique, and the overall FedMRD framework.

Table 4.2: MRDD evaluation results (own dataset).

Modality	Model	Acc (%)	P (%)	R (%)	F1 (%)
Unimodal	MobileNetV2 [112]	83.33	82.45	83.25	82.84
	FastText [113]	92.31	91.87	92.65	92.26
	Bert [123]	95.10	96.30	94.94	95.62
Multimodal	Ofli et al. [93]	94.84	95.02	95.00	94.01
	Gallo et al. [94]	95.32	71.43	95.94	81.89
	Choi et al. [124]	86.54	88.71	87.87	88.29
	Pranesh et al. [95]	85.90	85.97	85.54	85.75
	Abavisani et al. [96]	98.08	98.17	98.08	98.12
	MRDD1 (own)	97.44	97.63	97.41	97.52
	MRDD2-noPretrain (own)	97.79	97.96	97.79	97.87
	MRDD2 (own)	**99.14**	**99.15**	**99.14**	**99.14**

Acc: Accuracy; P: Precision; R: Recall; F1: F1-score

4.6.3.1 MRDD Results and Evaluation

The performance of the proposed MRDD models is evaluated by comparing it with the state-of-the-art on both the self-collected dataset and the CrisisMMD dataset. As shown in Table 4.2, it is observed that multimodal models easily outperform single modality approaches on the self-collected dataset. Specifically, MRDD1 ′s and MRDD2 ′s accuracy is up to 2% and 4% higher than unimodal methods. Compared with the state-of-the-art multimodal baselines [93–96, 124], the proposed multimodal road danger detection model (MRDD2) achieves the highest accuracy, 99.14%, which is 4%, 4%, 12%, 13%, and 1% higher than the multimodal baselines [93], [94], [124], [95], and [96]. The comparison with the state-of-the-art proves the efficiency of the proposed MRDD model for multimodal road danger classification. Moreover, without using the pre-trained triplet model's weights, the model's accuracy is decreased by around 2%. Further, MRDD2 performs better than MRDD1, proving that the model truly learns the intra- and inter-class differences after using the combined triplet loss function. We can also observe that texts contain more damage-related information while images carry more damage details. Figure 4.5 presents some example results. It shows that some images tend to be miss-classified when road dangers are small or far away from the camera. Poor illuminations and shadows are also one of the causes. Besides, sometimes crashed vehicles that ran out of road are also wrongly classified. Also, texts that are long and do not contain damage-class-related information tend to be miss-classified.

The results of the MRDD model on the CrisisMMD dataset are illustrated in Table 4.3. From the results, we can see that only one multimodal baseline [96] achieves higher accuracy than the unimodal models while other multimodal baselines are worse than them. Besides, Bert significantly outperforms FastText on the CrisisMMD dataset (by around 10% accuracy) though it is about 0.015 seconds slower than FastText, meaning that Bert is more suitable for processing more complex real-world text data than FastText. MobileNetV2 is also compared with DenseNet when choosing

Table 4.3: MRDD evaluation results (CrisisMMD public dataset).

Modality	Model	Acc (%)	P (%)	R (%)	F1 (%)
Unimodal	MobileNetV2 [112]	85.32	83.39	85.32	84.34
	FastText [113]	78.71	75.97	78.71	76.41
	Bert [123]	88.47	87.90	88.47	88.18
Multimodal	Ofli et al. [93]	87.14	86.74	87.14	86.94
	Gallo et al. [94]	84.92	83.58	84.92	84.23
	Choi et al. [124]	86.92	84.62	86.92	85.75
	Pranesh et al. [95]	85.81	84.78	85.81	85.29
	Abavisani et al. [96]	91.78	90.23	91.78	90.99
	MRDD1 (own)	89.80	89.61	89.80	89.71
	MRDD2-noPretrain (own)	86.93	87.97	86.83	87.40
	MRDD2 (own)	**92.00**	**91.05**	**92.00**	**91.52**

the image feature extractor. The results show that MobileNetV2 has about 1% higher accuracy than DenseNet, and it is 10 times faster than DenseNet, making it more applicable to devices for image processing. Moreover, the proposed multimodal models outperform all the state-of-the-art baselines by up to approximately 7% accuracy. Furthermore, the proposed MRDD2 attains the highest accuracy, i.e., 92%, which is 2% higher than MRDD1. Meanwhile, MRDD2's accuracy is 6% better than without pre-training the model by the combined triplet loss function. The results of the models on both the collected dataset and the CrisisMMD dataset prove the effectiveness of the proposed multimodal road danger detection models.

4.6.3.2 FedML Results and Evaluation

The effectiveness of FedML is evaluated by comparing it with the state-of-the-art FL strategies, i.e., FedAvg [81], FedPAQ [54], and LRDecay [82], as shown in Figure 4.6. For easy understanding, the FedML using MRDD1 and MRDD2 as the basic detection model is referred to as FedML1 and FedML2, respectively. The accuracy and communication cost are measured where the communication cost is calculated by multiplying the global rounds taken for the model to converge with the model size that needs to be transmitted between the edges and the cloud, i.e., (*quantized difference between previous global model and current local model size* $+$ *quantized global model size*) $\cdot$ *global round to converge*. The results show that the accuracy of FedML1 and FedML2 on both the self-collected dataset and MNIST dataset significantly outperforms the baselines.

FedML1 converges at 48 global rounds, which is 76% faster than FedAvg and FedPAQ on the collected dataset, as illustrated in Figure 4.6a. LRDecay converges as well as Federated Multimodal Learning algorithm with MRDD1 (FedML1) while its performance decreases around 0.1%. Similarly, FedML2 also converges fast, i.e., at 6 global rounds, and it is 88% quicker than FedAvg and FedPAQ on the self-collected dataset (see in Figure 4.6b). LRDecay here converges

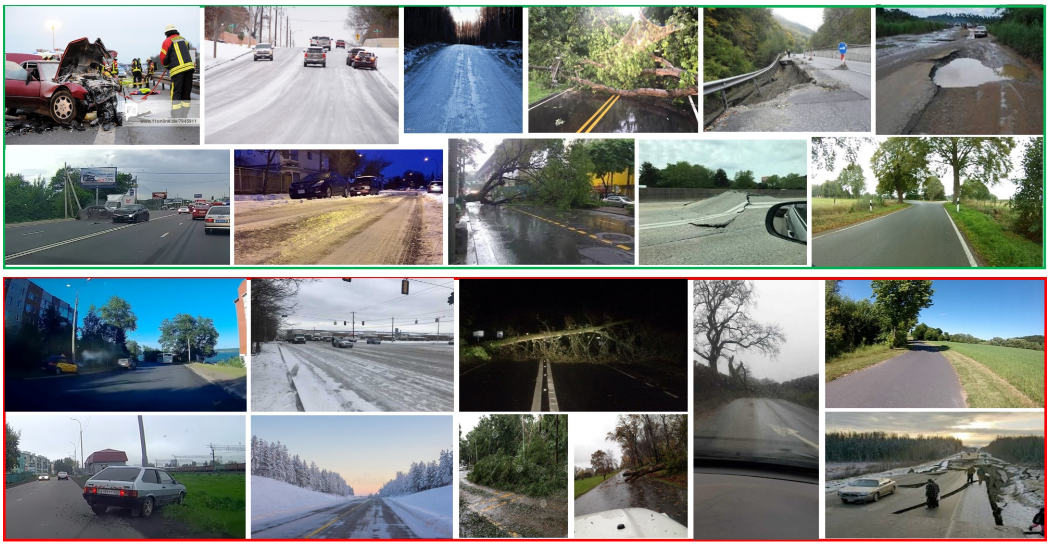
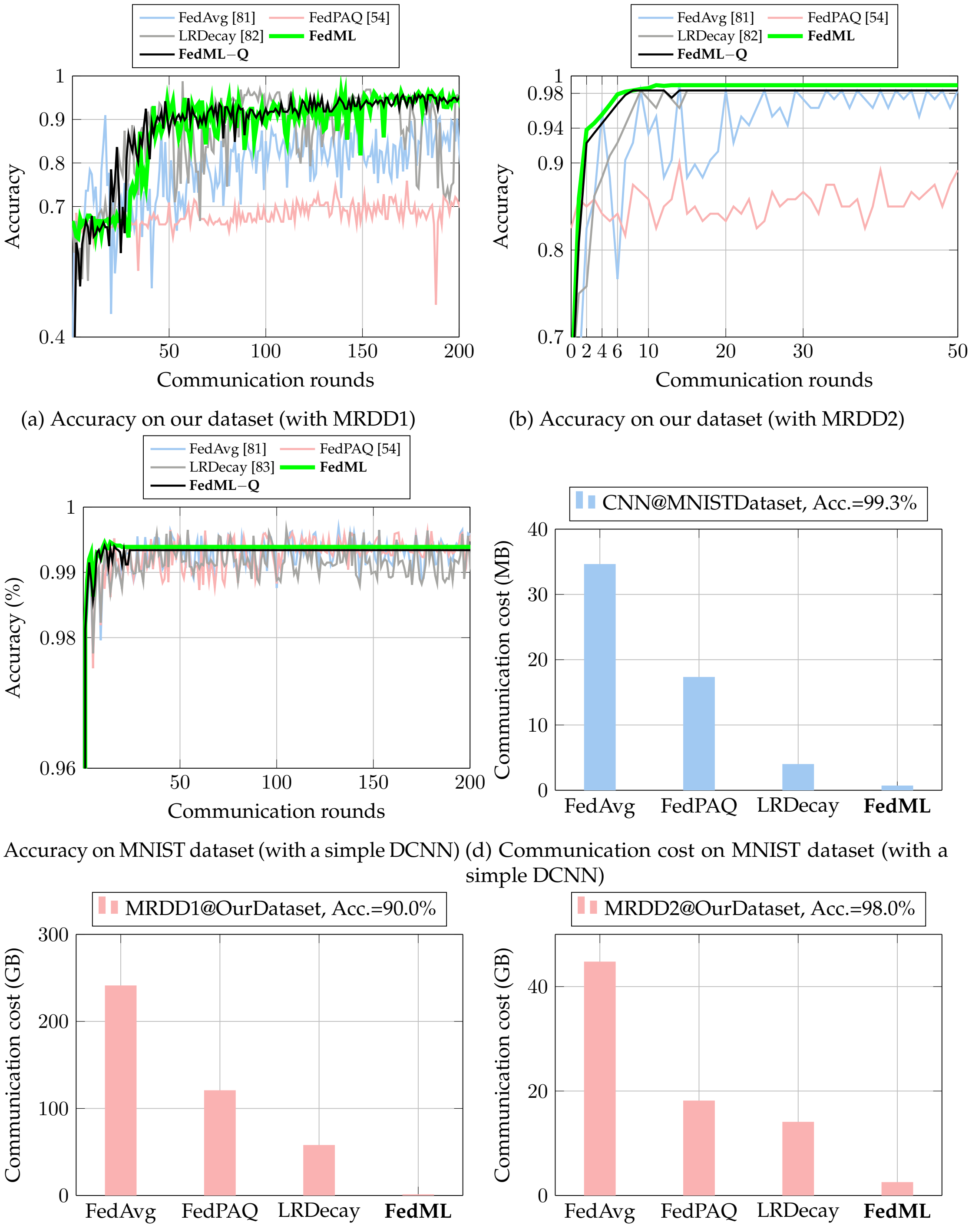


Figure 4.5: Example results where green and red represent correctly and wrongly classified images.

at about 15 global rounds, which is 150% higher than FedML but 70% lower than FedAvg and FedPAQ. Apart from the accuracy, the communication cost of FedML1 and Federated Multimodal Learning algorithm with MRDD2 (FedML2) on the collected dataset is also evaluated. From the results in 4.6e and 4.6f, we can clearly see that the communication cost of FedML1 and FedML2 is significantly lower than the state-of-the-art baselines. In particular, FedML1's and FedML2's communication cost is only 0.89 MB and 2.48 MB on self-collected dataset, which is approximately 270 and 20 times lower than their FedAvg, FedPAQ, and LRDecay baselines.

To prove the effectiveness of FedML1 and FedML2, they are tested on the MNIST dataset, and the results are presented in Figure 4.6c. Although the accuracy of FedML1 and FedML2 with model quantization (99.37% and 99.45%) is similar to without model quantization (99.34% and 99.41%), the communication cost of the former can be reduced by approximately 65 and 47 times than the latter. The accuracy of FedML1 is also slightly better than that of the FedAvg, FedPAQ, and LRDecay, while it shows a much higher convergence performance than the baselines on the MNIST dataset, as illustrated in Figure 4.6c.

Overall, FedAvg and FedPAQ have the highest communication cost since their learning processes are highly unstable, requiring much more communication rounds to converge. FedML achieves the highest accuracy with the lowest communication cost and converges well due to that it periodically averages the local models instead of in every local epoch, decay the learning rate in every global round, and quantizes the model by low precision quantizer and dynamic post quantization method. More specifically, FedML only transmits the difference of the weights instead of the whole model and quantizes them for both upstream and downstream communication. Furthermore, the results prove that the proposed FedML significantly benefits the road danger detection task in the



(a) Accuracy on our dataset (with MRDD1) (b) Accuracy on our dataset (with MRDD2)

(c) Accuracy on MNIST dataset (with a simple DCNN) (d) Communication cost on MNIST dataset (with a simple DCNN)

(e) Communication cost on our dataset (with MRDD1) (f) Communication cost on our dataset (with MRDD2)

Figure 4.6: Performance comparison of FedML on both our dataset and MNIST dataset.

proposed edge-cloud-based FedMRD framework.

4.6.3.3 MDP Results and Evaluation

The effect of the privacy budget ϵ of the proposed MDP technique on the detection performance of the FedMRD framework is evaluated, and the results are presented in Figure 4.7. In the experiments, noisy and multidimensional tuples are collected from users, and the accuracy of the multimodal model on edges is estimated. Given privacy budget ϵ, ϵ/d^2 privacy budget is allocated to each item for each image with $d \times d$ dimension (before dimension reduction), and ϵ/d privacy budget is allocated to each attribute of the text with dimension d. The expected error $O(\frac{d}{\epsilon})$ is large when dimension d is high. After dimension reduction, the expected error reduces by 99.9% (from $I_{256\times256}$ to $I_{1\times64}$). Finally, the accuracy of MRDD (with FedML) is estimated after applying MDP on the modalities data.

Overall, the road danger recognition performance of FedMRD increases with the growth of privacy budget ϵ, which complies with the theory that high ϵ indicates injecting more noise. As shown in Figure 4.7, the detection accuracy of the model is lower than 90% when ϵ is below 0.2, and it climbs quickly when ϵ increases from 0.001 to 0.8. Specifically, compared to that of $\epsilon = 0.001$, the detection accuracy of FedMRD with $\epsilon = 0.8$ improves 12.43%. Besides, the model's accuracy when ϵ equals 0.8 is around 5% and 1% higher than when $\epsilon = 0.4$ and $\epsilon = 0.6$. However, there is not much difference in the detection accuracy of FedMRD when ϵ changes from 0.8 to 1. Therefore, $\epsilon = 0.8$ is selected as the privacy level to achieve the best trade-off between detection performance of the local models and data privacy.

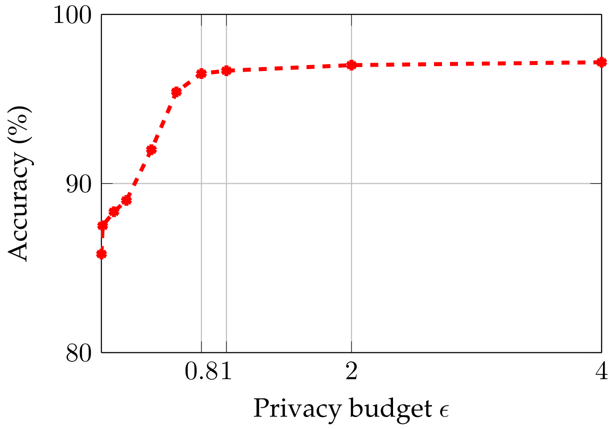


Figure 4.7: Average detection performance of FedMRD over different privacy budget ϵ.

Table 4.4: FedMRD evaluation results.

Framework	Acc (%)	F1 (%)	Latency (s)	CL	CC (MB)
CloudRD [71]	87.90	87.66	2.49	No	-
EdgeRD [26]	82.13	82.10	0.054	No	-
EcRD [7]	92.51	92.05	**0.003**	No	292.97
FedRD (own)	91.64	91.25	0.0326	**Yes**	1003.34
FedMRD (own)	**96.42**	**96.61**	0.0351	**Yes**	3.64

Acc: Accuracy; F1: F1-score; Latency: Wait time before receiving warning; CL: Collaborative Learning; CC: Communication Cost.

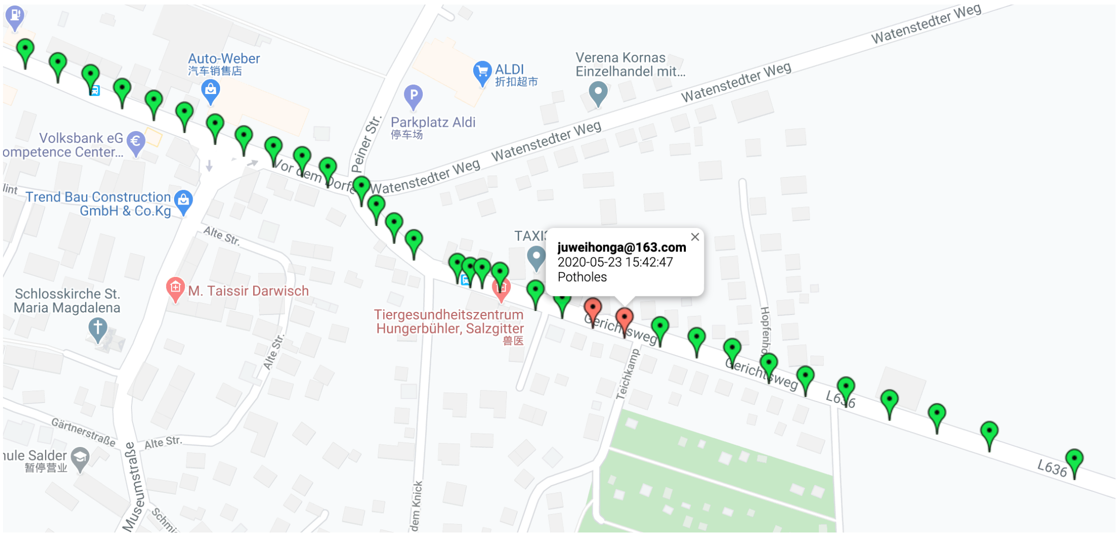
4.6.3.4 FedMRD Framework Results and Evaluation

To evaluate the performance of the FedMRD framework, it is compared with one cloud-based method, i.e., CloudRD [71], and three edge-based methods, i.e., EdgeRD [26], EcRD [7], and FedRD [8]. The cloud-based method process data on a central cloud server while edge-based approaches process it on nearby edges. Edge-based solutions have much lower latency than cloud-based methods because edges are much closer to devices than clouds. The accuracy, f1-score, latency, with/without collaborative learning, and communication cost of different frameworks are measured and compared in Table 4.4.

As shown in Figure 4.4, FedMRD achieves approximately 10%, 18%, 5%, and 6% higher f1-score than CloudRD, EdgeRD, EcRD, and FedRD, respectively. Similarly, it obtains around 10%, 17%, 4%, and 5% higher accuracy than CloudRD, EdgeRD, EcRD, and FedRD, respectively. Also, it has 98.43% and 27.78% shorter latency than CloudRD and EdgeRD. Although FedMRD has long latency than EcRD, it has a 5% higher f1-score than EcRD. Besides, FedMRD and FedRD enable collaborative learning between edges and the cloud different from other frameworks, which is more private and saves communication costs. Moreover, FedMRD's communication cost is approximately 100 and 1000 times lower than EcRD and FedRD. In summary, the results show that FedMRD responds to road dangers timely and enables collaborative learning between edges and the cloud. Besides, FedMRD preserves users' privacy from untrusted edges and clouds and significantly reduces computation cost of edges and communication cost between edges and the cloud.

4.6.3.5 Detection Result Display

Once edges detect road dangers, detailed road danger warning messages (i.e., types, GPS locations, and time) of the road dangers are transmitted to the cloud directly. The cloud analyzes the received road danger information, displays it on a road danger map. The map is updated in real-time if the cloud receives any new road dangers. With this road danger map, road users (e.g., pedestrians and drivers) or road administrations can know how the road condition is and which route is optimal for traveling.



juweihonga@163.com
2020-05-23 15:42:47
Potholes

Figure 4.8: Display of road danger detection results.

A webpage is developed to display the detection results to illustrate road danger information better. As shown in Figure 4.8, the road danger detection results are displayed on the Google Map in the designed web page in real-time. The green GPS coordinates indicate that the roads are safe (i.e., no road danger found), while red GPS coordinates mean that the roads are dangerous (i.e., one or more road dangers are found). Additionally, the GPS coordinates are clickable, and details of the road dangers such as danger types and the collection time are shown after clicking the red GPS coordinates.

4.7 Conclusion

The proposed FedMRD framework improves the road danger detection and warning task's performance in three aspects, i.e., accuracy, communication efficiency, and privacy. Specifically, the proposed multimodal road danger detection models (i.e., MRDD1 and MRDD2) creatively learn from multimodal data and significantly improve the detection accuracy. Besides, the communication cost is reduced by periodic averaging and model quantization strategies proposed in the Federated Multimodal Learning strategy (FedML). A novel Multimodal Differential Privacy technique (MDP) is introduced to protect data privacy, largely reducing data dimension while maintaining high detection performance. In summary, FedMRD solves the problems of the FedRD, EcRD and other existing frameworks (i.e., only based on single modality, high communication costs caused by frequent model exchanging, high dimensionality curse of LDP), and enables fast, accurate, communication efficient, and privacy-preserving road danger detection and warning.

Chapter 5

Discussion

To answer the four research questions listed in Chapter 1, three road danger detection and warning frameworks are proposed in this thesis, i.e., EcRD (see in Chapter 2), FedRD (see in Chapter 3), and FedMRD (see in Chapter 4). This thesis aims to provide road users with fast, accurate, communication-efficient, and private road danger detection and warning services in the real world. The proposed EcRD and FedRD frameworks only detect road damages (e.g., racks, patches, potholes, fractures, big holes, and blowups) while the FedMRD framework considers general road dangers (e.g., road damages, crashed vehicles, fallen trees, and icy roads). Also, EcRD and FedRD only process single modality data (e.g., images/videos); however, FedMRD works well on modalities data (e.g., image/video and text), and it displays road dangers on the Google map and updates in real-time. The road danger detection models from all three frameworks are deployed on edges instead of the cloud for fast responses since edges are much closer to users than the cloud. Besides, more light-weighted models are developed to reduce the latency, answering research question 1.

To improve the accuracy (research question 2) of road danger recognition, four models are proposed, i.e., HDD (see in Section 2.5.2), ADD (see in Section 3.5.2), MRDD1 (see in Section 4.5.1), MRDD2 (see in Section 4.5.2). HDD is a hand-craft feature-based approach that extracts five GLCM features of the input image and decides whether the image is dangerous or not by comparing the weighted sum of the five features with the given threshold. This method is very fast compared to deep learning-based models. However, the model's performance is highly unstable when the input images vary largely (e.g., different illuminations). To overcome this limitation, deep learning-based approaches are developed for road danger detection, i.e., ADD in FedRD, and MRDD1 and MRDD2 in FedMRD. ADD is composed of five deep residual blocks with skip connections, and four levels of features from four residual blocks are fused by simple concatenation for final road danger classification. In this way, both small and large road dangers can be classified correctly, benefiting from low-level and high-level features. Results show that around 8% of the model's accuracy is improved than without multi-level feature fusion. MRDD1 and MRDD2 are proposed to handle multimodality data. In MRDD1, MobileNetV2 and FastText are utilized for image and text feature extraction for rapid road danger inspection. The image and text features are fused together by simple adding, and then it is processed by two FC layers for final classification. To further improve the accuracy of the multimodal modal, MRDD2 is developed, consisting of MobileNetV2 and Bert for image and text feature extraction and a merging model for multimodality feature fusion and final classification. It pre-trains the multimodal model by the proposed combined triplet loss function taking into account both inter-class and intra-class relations for both image and text modalities. Then, the pre-trained MobileNetV2 and Bert and the merging model are utilized for road danger classification with the cross-entropy loss. The results show that MRDD2 achieves approximately 2% higher accuracy than MRDD1 on the collected and CrisisMMD public datasets. Different strategies are utilized in the frameworks to improve detection accuracy, apart from the road danger detection models. More specifically, DFRD (see in Section 2.5.1) is introduced in EcRD to filter out noisy backgrounds for higher accuracy. MDD (see in Section 2.5.3) is proposed for multi-type road danger classification on the cloud, complementing HDD, which can only distinguish between normal and hazardous roads. It is worth noticing that there is an image-label generator inside MDD which automatically generates images and their corresponding labels of road dangers from different classes, which significantly reduces the data collection and labeling efforts. Besides, FL is employed in FedRD and FedMRD to improve the detection model's performance.

As we all know, high communication costs (research question 3) are incurred by continuously uploading images from users' devices to edges and from edges to the cloud in the EcRD framework. Similarly, using FL in the FedRD and FedMRD frameworks also results in high communication costs due to frequent model exchanges between the edges and the cloud. To reduce the communication cost, the EcRD framework transmits road images (with only road areas) instead of the original images while the FedRD and FedMRD frameworks propose more efficient FL strategies (i.e., AFed (see in Section 3.5.3) and FedML (see in Section 4.5.3)). In AFed, the cloud waits until K local models from edges are received, which complies with the fact that the cloud has no control over edges. Also, AFed only aggregates the local models that have high detection accuracy, which improves the result. However, the communication cost of AFed is still very high. Hence, I propose the FedML strategy in the FedMRD framework. The FedML strategy achieves high detection accuracy with low communication costs while guarantees models' convergence on non-iid and imbalanced data. It includes periodic averaging, learning rate decay, and model quantization. Periodic averaging reduces communication costs by lowering model communication rounds; model quantization decreases communication costs by directly reducing the model size. Besides, the learning rate decay strategy ensures the model's convergence on non-iid data. The results prove that the communication cost is reduced significantly by using the proposed FedML strategy.

Additionally, data privacy (research question 4) in the proposed frameworks is also protected. The EcRD framework relies on the road segmentation model (i.e., DFRD) to avoid privacy leakage.

However, the privacy protection performance highly depends on the performance of the road segmentation model, while the model may fail in real-world environments where images have cluttered backgrounds and there are no mathematical guarantees. IDPP (see in Section 3.5.4) and MDP (see in Section 4.5.4) based on the differential privacy technique, are proposed to tackle this issue,and they have mathematical guarantees. They are placed on users' devices and add noise to users' data before uploading to the nearest edge, which is more private. Unlike IDPP, MDP points out the high-dimensionality curse of LDP and largely reduces the dimension by multiplying with two random matrixes. Moreover, MDP can process real-world multimodal data with multiple attributes by first converting them into fixed dimensional numerical vectors on users' devices.

Moreover, the proposed frameworks are general systems suitable for general condition monitoring tasks, e.g., traffic congestion monitoring, natural disaster inspection, infrastructure condition monitoring, and production quantity control.

Despite the outstanding performance of the frameworks, they have the following drawbacks. Firstly, although the detection models (i.e., HDD, ADD, MRDD1, and MRDD2) achieve high accuracy (up to 99%), the runtime ($0.03\,\mathrm{s}$) is not low enough for real-time road danger detection and warning (except HDD ($0.004\,\mathrm{s}$)). Although HDD's runtime ($0.004\,\mathrm{s}$) is low enough for real-time road danger inspection, its accuracy (92%) is not high enough, and its performance could be worse when the backgrounds are more cluttered. A more lightweight and robust model should be designed to improve accuracy and reduce runtime. Secondly, the proposed FL strategies (i.e., AFed, and FedML) enhance the performance of edges with non-iid and imbalanced data; however, the performance improvement for edges with no or very little data is limited. A better FL strategy that can work well on highly skewed and non-iid data distribution should be proposed. Finally, the proposed IDPP and MDP techniques preserve data privacy, and they reduce data dimensions by pixelization and matrix multiplication, respectively; however, the expected error is still not low enough since the input data' dimension is high. Important information will be lost if the dimension is reduced further. Hence, a novel LDP with low expected error for high dimension data should be developed.

Chapter 6

Conclusion and Future Work

In this thesis, the importance of road danger detection and warning and the limitations of existing road danger detection and warning systems are studied, i.e., high latency, low accuracy, high communication costs, and high privacy leakage risks. Three novel road danger detection and warning frameworks are proposed to address these issues, including EcRD, FedRD, and FedMRD.

Firstly, EcRD is proposed to ensure high detection accuracy with low labeling cost and low latency, which answers research questions 2 and 1. EcRD framework exploits the benefits of edge computing, especially the combination of edge and cloud for fast, efficient, and cheap road damage danger recognition and notification. The device-edge-cloud structure of EcRD provides high-quality services with optimum resources consumption. Also, to enhance the performance of EcRD, three models, i.e., deep feature-based road detector, hazardous road damage detector, and multi-types damage detector, are developed. More specifically, the deep feature-based road detector is utilized for road segmentation. The hazardous road damage detector is used for fast road damage danger detection on the edges. The multi-type damage detector is employed for accurate multi-type road damage detection on the cloud. It is worth mentioning that a novel image-label generator is developed in the multi-type damage detector to automatically generate realistic road dangers and corresponding labels with real-world scene backgrounds, which significantly saves data collection and labeling costs. Extensive experiments show that EcRD achieves 579 times lower latency than cloud-based approaches for road danger detection and warning and only demands limited resources.

Then, FedRD is developed to tackle the limitations of EcRD and existing cloud/edge-based systems, i.e., limited warning scope, the low performance of edges with limited data, and high data privacy leakage risks (research questions 2 and 4). In FedRD, a new map construction approach is introduced to aggregate hazardous road damage information of a wide area into a global map, which is hundreds or thousands of times wider than existing edge-based systems. The global map contains rich hazardous road damage information and helps to improve users' travel experience and reduce road accidents. Additionally, an adaptive federated learning strategy is designed to improve the performance of local models on edges, especially for the edges with limited data. The proposed adaptive federated learning strategy ensures high detection performance in limited communication rounds by selective aggregating qualified local models for global model aggregation on the cloud. Moreover, the individualized differential privacy with pixelization technique is proposed to protect data privacy. It protects privacy on users' devices before uploading to untrusted edges, and it reduces $3/4$ computation and communication costs by using pixelization. Simulation results show that the FedRD framework detects hazardous road damages accurately and warns drivers with low latency. It is robust on challenging datasets and still performs well when some edges lack data for training. It covers a broader area and has low computation, communication, and storage costs. Data privacy is also preserved. Simulation results demonstrate that FedRD achieves a high detection performance and provides fast responses with accurate warning information covering a wider area while preserving users' privacy, even when some edges have limited data.

Finally, FedMRD is introduced to address the limitations of EcRD, FedRD, and existing road danger detection and warning systems such as only based on single modality, high communication costs caused by frequent model exchanging, high dimensionality curse of LDP (research questions 2, 3, and 4). FedMRD exploits the benefits of multimodal learning, communication-efficient federated learning, and local differential privacy in device-edge-cloud computing structure for cheap, fast, accurate, communication efficient, and private road damage danger detection and warning. In FedMRD, a multimodal road danger detection model with a novel training strategy that considers inter-and intra-class differences is proposed for better road danger classification with data in different modalities (i.e., image and text). Also, an efficient federated learning strategy is developed to improve the performance of local models on edges. It significantly reduces communication and computation costs while guaranteeing the model's convergence. Moreover, a multimodal differential privacy technique is introduced to protect data privacy on users' devices before offloading to edges. The technique addresses the real-world multimodal data with multiple attributes and high-dimensionality curse of LDP by dimension reduction. Experimental results show that FedMRD provides users fast response to road danger, high accuracy, low communication cost, and well privacy protection for high-dimensional real-world data.

As part of future work, more road danger types will be incorporated in the model training to build a more general road danger warning framework. New lightweight models or learning techniques (e.g., few-shot-learning techniques) that can perform better with limited data for road danger detection will be explored. Also, more data modalities (e.g., audio) will be considered to improve the accuracy and convenience. The systems' performance can be further improved by a new FL strategy that works well on highly skewed and non-iid data from edges. A novel LDP technique with low expected error can further improve the systems' privacy. Finally, The proposed systems will be tested in real-world setups to verify the robustness further and address the challenges.

Bibliography

[1] M. Colin, F. Palhol, and A. Leuxe, "Adaptation of transport infrastructures and networks to climate change," *Transportation Research Procedia*, vol. 14, pp. 86–95, 2016.

[2] W. H. Organization, *Global status report on road safety 2015*. World Health Organization, 2015.

[3] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018.

[4] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4906–4911, 2020.

[5] A. Anaissi, N. L. D. Khoa, T. Rakotoarivelo, M. M. Alamdari, and Y. Wang, "Smart pothole detection system using vehicle-mounted sensors and machine learning," *Journal of Civil Structural Health Monitoring*, vol. 9, no. 1, pp. 91–102, 2019.

[6] A. Fox, B. V. Kumar, J. Chen, and F. Bai, "Multi-lane pothole detection from crowdsourced undersampled vehicle sensor data," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3417–3430, 2017.

[7] Y. Yuan, M. S. Islam, Y. Yuan, S. Wang, T. Baker, and L. M. Kolbe, "Ecrd: Edge-cloud computing framework for smart road damage detection and warning," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 734–12 747, ©2020 IEEE, 2020.

[8] Y. Yuan, Y. Yuan, T. Baker, L. M. Kolbe, and D. Hogrefe, "Fedrd: Privacy-preserving adaptive federated learning framework for intelligent hazardous road damage detection and warning," *Future Generation Computer Systems*, vol. 125, pp. 385–398, ©2021 Elsevier, 2021.

[9] S. D. Gleave, "Eu road surfaces: Economic and safety impact of the lack of regular road maintenance, study," *Policy Department Structural and Cohesion Policies*, 2014.

[10] K. Gopalakrishnan, “Deep learning in data-driven pavement image analysis and automated distress detection: A review,” *Data*, vol. 3, no. 3, p. 28, 2018.

[11] C. Chellaswamy, H. Famitha, T. Anusuya, and S. Amirthavarshini, “Iot based humps and pothole detection on roads and information sharing,” in *2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*. IEEE, 2018, pp. 084–090.

[12] K. Doycheva, C. Koch, and M. König, “Computer vision and deep learning for real-time pavement distress detection,” in *Advances in Informatics and Computing in Civil and Construction Engineering*. Springer, 2019, pp. 601–607.

[13] W. Song, G. Jia, H. Zhu, D. Jia, and L. Gao, “Automated pavement crack damage detection using deep multiscale convolutional features,” *Journal of Advanced Transportation*, vol. 2020, pp. 1–11, 2020.

[14] D. A. Chacra and J. Zelek, “Road segmentation in street view images using texture information,” in *2016 13th Conference on Computer and Robot Vision (CRV)*. IEEE, 2016, pp. 424–431.

[15] C. C. T. Mendes, V. Frémont, and D. F. Wolf, “Vision-based road detection using contextual blocks,” 2015, arXiv.

[16] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, “Fast vanishing-point detection in unstructured environments,” *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 425–430, 2012.

[17] Q. Wang, J. Gao, and Y. Yuan, “Embedding structured contour and location prior in siamesed fully convolutional networks for road detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 230–241, 2018.

[18] J. Zhang, Y. Xu, B. Ni, and Z. Duan, “Geometric constrained joint lane segmentation and lane boundary detection,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 486–502.

[19] Z. Chen and Z. Chen, “Rbnet: A deep neural network for unified road and road boundary detection,” in *International Conference on Neural Information Processing*. Springer, 2017, pp. 677–687.

[20] N.-D. Hoang, “Detection of surface crack in building structures using image processing technique with an improved otsu method for image thresholding,” *Advances in Civil Engineering*, vol. 2018, 2018.

[21] Y. Fujita, Y. Mitani, and Y. Hamamoto, “A method for crack detection on a concrete structure,” in *18th International Conference on Pattern Recognition (ICPR’06)*, vol. 3. IEEE, 2006, pp. 901–904.

[22] M. S. Hossain, M. M. Hassan, M. Al Qurishi, and A. Alghamdi, "Resource allocation for service composition in cloud-based video surveillance platform," in *2012 IEEE international conference on multimedia and expo workshops*. IEEE, 2012, pp. 408–412.

[23] J. Wang, Y. Hu, H. Li, and G. Shou, "A lightweight edge computing platform integration video services," in *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*. IEEE, 2018, pp. 183–187.

[24] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *computer*, vol. 50, no. 10, pp. 58–67, 2017.

[25] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Transactions on Cloud computing*, vol. 7, no. 1, pp. 196–209, 2016.

[26] M. Kawano, T. Yonezawa, and J. Nakazawa, "Deep on edge: Opportunistic road damage detection with city official vehicles," in *Proceedings of The Third International Conference on Smart Portable, Wearable, Implantable and Disability-oriented Devices and Systems (SPWID 2017)*, 2017.

[27] T. P. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranoc, and V. Eruhimov, "Computer vision workload analysis: Case study of video surveillance systems." *Intel Technology Journal*, vol. 9, no. 2, 2005.

[28] F. Jiang, K. Wang, L. Dong, C. Pan, and K. Yang, "Stacked auto encoder based deep reinforcement learning for online resource scheduling in large-scale mec networks," *IEEE Internet of Things Journal*, 2020.

[29] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[31] F. Zohourian, B. Antic, J. Siegemund, M. Meuter, and J. Pauli, "Superpixel-based road segmentation for real-time systems using cnn." in *VISIGRAPP (5: VISAPP)*, 2018, pp. 257–265.

[32] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[33] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler, "Convolutional patch networks with spatial prior for road detection and urban scene understanding," *arXiv preprint arXiv:1502.06344*, 2015.

[34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[35] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[36] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[37] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing cnn architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, 2020.

[38] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[39] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *European Conference on Computer Vision (ECCV)*, 2010.

[40] J. V. F. Giovani Bernardes Vitor, Alessandro C. Victorino, "A probabilistic distribution approach for the classification of urban roads in complex environments," *IEEE Workshop on International Conference on Robotics and Automation*, 2014.

[41] N. Garnett, S. Silberstein, S. Oron, E. Fetaya, U. Verner, A. Ayash, V. Goldner, R. Cohen, K. Horn, and D. Levi, "Real-time category-based and general obstacle detection for autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 198–205.

[42] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2083–2090.

[43] B. Ramamurthy and K. Chandran, "Content based medical image retrieval with texture content using gray level co-occurrence matrix and k-means clustering algorithms," *Journal of Computer Science*, vol. 8, no. 7, p. 1070, 2012.

[44] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 3, pp. 186–205, 2000.

[45] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[46] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[47] Y. Tang, P. P. Lee, J. C. Lui, and R. Perlman, "Fade: Secure overlay cloud storage with file assured deletion," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2010, pp. 380–397.

[48] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[49] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[50] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1779–1790, 2018.

[51] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, 2017.

[52] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[53] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," *arXiv preprint arXiv:1902.11175*, 2019.

[54] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.

[55] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?" *arXiv preprint arXiv:2003.14053*, 2020.

[56] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[57] L. Fan, "Image pixelization with differential privacy," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2018, pp. 148–162.

[58] M. Seif, R. Tandon, and M. Li, "Wireless federated learning with local differential privacy," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 2604–2609.

[59] T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie, "Edge-based differential privacy computing for sensor–cloud systems," *Journal of Parallel and Distributed computing*, vol. 136, pp. 75–85, 2020.

[60] A. A. Fauzi, F. Utaminingrum, and F. Ramdani, "Road surface classification based on lbp and glcm features using knn classifier," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1446–1453, 2020.

[61] X. Bai, Y. Fang, W. Lin, L. Wang, and B.-F. Ju, "Saliency-based defect detection in industrial images by using phase spectrum," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2135–2145, 2014.

[62] X. Xie and M. Mirmehdi, "Texems: Texture exemplars for defect detection on random textured surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1454–1464, 2007.

[63] J. Wang, Z. Wang, C. Gao, N. Sang, and R. Huang, "Deeplist: Learning deep features with adaptive listwise constraint for person reidentification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 513–524, 2016.

[64] G. Suh and Y.-J. Cha, "Deep faster r-cnn-based automated detection and localization of multiple types of damage," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*, vol. 10598. International Society for Optics and Photonics, 2018, p. 105980T.

[65] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[66] J. Masci, U. Meier, G. Fricout, and J. Schmidhuber, "Multi-scale pyramidal pooling network for generic steel defect classification," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–8.

[67] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493–1504, 2019.

[68] J. Zhong, Z. Liu, Z. Han, Y. Han, and W. Zhang, "A cnn-based defect inspection method for catenary split pins in high-speed railway," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2849–2860, 2018.

[69] J. Cao, G. Yang, and X. Yang, "A pixel-level segmentation convolutional neural network based on deep feature fusion for surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2020.

[70] J. Wang, L. Luo, W. Ye, and S. Zhu, "A defect-detection method of split pins in the catenary fastening devices of high-speed railway based on deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9517–9525, 2020.

[71] R. Fan, M. J. Bocus, Y. Zhu, J. Jiao, L. Wang, F. Ma, S. Cheng, and M. Liu, "Road crack detection using deep convolutional neural network and adaptive thresholding," *arXiv preprint arXiv:1904.08582*, 2019.

[72] K. A. Khaliq, O. Chughtai, A. Shahwani, A. Qayyum, and J. Pannek, "Road accidents detection, data collection and data analysis using v2x communication and edge/cloud computing," *Electronics*, vol. 8, no. 8, p. 896, 2019.

[73] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[75] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," *arXiv preprint arXiv:2104.07586*, 2021.

[76] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *Advances in neural information processing systems*, vol. 27, pp. 2879–2887, 2014.

[77] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[78] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[80] R. Du, G. Qiu, K. Gao, L. Hu, and L. Liu, "Abnormal road surface recognition based on smartphone acceleration sensor," *Sensors*, vol. 20, no. 2, p. 451, 2020.

[81] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[82] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[83] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, and H. Li, “Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets,” *arXiv preprint arXiv:2008.03371*, 2020.

[84] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” *arXiv preprint arXiv:1806.00582*, 2018.

[85] P. Kairouz, K. Bonawitz, and D. Ramage, “Discrete distribution estimation under local privacy,” in *International Conference on Machine Learning*. PMLR, 2016, pp. 2436–2444.

[86] T. Wang, J. Blocki, N. Li, and S. Jha, “Locally differentially private protocols for frequency estimation,” in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 729–745.

[87] G. C. Fanti, V. Pihur, and Ú. Erlingsson, “Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries.” *Proc. Priv. Enhancing Technol.*, vol. 2016, no. 3, pp. 41–61, 2016.

[88] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, “Lopub: High-dimensional crowdsourced data publication with local differential privacy,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[89] C. Xia, J. Hua, W. Tong, and S. Zhong, “Distributed k-means clustering guaranteeing local differential privacy,” *Computers & Security*, vol. 90, p. 101699, 2020.

[90] E. Zalama, J. Gómez-García-Bermejo, R. Medina, and J. Llamas, “Road Crack Detection Using Visual Features Extracted by Gabor Filters,” *Computer-Aided Civil and Infrastructure Engineering*, 2014.

[91] L. Pauly, D. Hogg, R. Fuentes, and H. Peel, “Deeper networks for pavement crack detection,” in *Proceedings of the 34th ISARC*. IAARC, 2017, pp. 479–485.

[92] K. Ma, M. Hoai, and D. Samaras, “Large-scale continual road inspection: Visual infrastructure assessment in the wild.” in *BMVC*, 2017.

[93] F. Ofli, F. Alam, and M. Imran, “Analysis of social media data using multimodal deep learning for disaster response,” *arXiv preprint arXiv:2004.11838*, 2020.

[94] I. Gallo, A. Calefati, S. Nawaz, and M. K. Janjua, “Image and encoded text fusion for multi-modal classification,” in *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018, pp. 1–7.

[95] R. R. Pranesh, A. Shekhar, and A. Kumar, “Exploring multimodal features and fusion strategies for analyzing disaster tweets.”

[96] M. Abavisani, L. Wu, S. Hu, J. Tetreault, and A. Jaimes, "Multimodal categorization of crisis events in social media," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 679–14 689.

[97] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann, "Multimedia classification and event detection using double fusion," *Multimedia tools and applications*, vol. 71, no. 1, pp. 333–347, 2014.

[98] S. Kelly, X. Zhang, and K. Ahmad, "Mining multimodal information on social media for increased situational awareness," 2017.

[99] H. Mouzannar, Y. Rizk, and M. Awad, "Damage identification in social media posts using multimodal deep learning." in *ISCRAM*, 2018.

[100] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[101] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017.

[102] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 560–569.

[103] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[104] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 15–24.

[105] N. Shoham, T. Avidor, A. Keren, N. Israel, D. Benditkis, L. Mor-Yosef, and I. Zeitak, "Overcoming forgetting in federated learning on non-iid data," *arXiv preprint arXiv:1910.07796*, 2019.

[106] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for cloudiot," *IEEE Transactions on Cloud Computing*, 2020.

[107] J. Li, X. Kuang, S. Lin, X. Ma, and Y. Tang, "Privacy preservation for machine learning training and classification based on homomorphic encryption schemes," *Information Sciences*, vol. 526, pp. 166–179, 2020.

[108] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, and J. Hu, "Appa: An anonymous and privacy preserving data aggregation scheme for fog-enhanced iot," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

[109] B. Zhao, K. Fan, K. Yang, Z. Wang, H. Li, and Y. Yang, "Anonymous and privacy-preserving federated learning with industrial big data," *IEEE Transactions on Industrial Informatics*, 2021.

[110] J. Qian, X.-Y. Li, C. Zhang, L. Chen, T. Jung, and J. Han, "Social network de-anonymization and privacy inference with knowledge graph model," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 679–692, 2017.

[111] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 2008, pp. 111–125.

[112] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[113] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[114] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92.

[115] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Machine learning*, vol. 77, no. 1, pp. 27–59, 2009.

[116] B. Shaw, B. Huang, and T. Jebara, "Learning a distance metric from a network," *Advances in Neural Information Processing Systems*, vol. 24, pp. 1899–1907, 2011.

[117] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5005–5013.

[118] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2003, vol. 87.

[119] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[120] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2001, pp. 274–281.

[121] F. Alam, F. Ofli, and M. Imran, "Crisismmd: Multimodal twitter datasets from natural disasters," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, no. 1, 2018.

[122] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[123] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[124] J.-H. Choi and J.-S. Lee, "Embracenet for activity: A deep multimodal fusion architecture for activity recognition," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 693–698.

Appendix A

Publications

A.1 Peer-Reviewed

A.1.1 **FedRD: Privacy-preserving adaptive Federated learning framework for intelligent hazardous Road Damage detection and warning**

Abstract: *Road damages have caused numerous fatalities. Therefore, the study of road damage detection, especially hazardous road damage detection and warning, is critical in improving traffic safety. Existing road damage detection systems mainly process data on clouds, however, they are not able to warn users timely due to the long latency. Recent edge-computing techniques mitigate this problem while users can only receive warnings of hazardous road damages within a small area due to the limited communication range of edges. Besides, untrusted edges might misuse users' sensitive information. In this paper, we propose FedRD: a novel privacy-preserving edge-cloud and Federated learning-based framework for intelligent hazardous Road Damage detection and warning. In FedRD, a new hazardous road damage detection model is developed leveraging the advantages of hierarchical feature fusion. A novel adaptive federated learning strategy is designed for robust model learning from different edges with limited and unequally-sized datasets. A new individualized differential privacy approach with pixelization is proposed to protect users' privacy before sharing data. Simulation results demonstrate that FedRD achieves a high detection performance and provides fast responses with accurate warning information covering a wider area while preserving users' privacy, even when some edges have limited data.*

Reference: **Yachao Yuan**, Yali Yuan, Thar Baker, Lutz Maria Kolbe, Dieter Hogrefe. "FedRD: Privacy-preserving Adaptive Federated Learning Framework for Intelligent Hazardous Road Danger Detection and Warning," in Future Generation Computer Systems Journal, 2021, 385-389.

A.1.2 **EcRD: Edge-cloud Computing Framework for Smart Road Damage Detection and Warning**

Abstract: *Road damages have caused numerous fatalities, thus the study of road damage detection, especially hazardous road damage detection and warning is critical for traffic safety. Existing road damage detection systems mainly process data at Cloud, which suffers from a high latency caused by long-distance. Meanwhile, supervised machine learning algorithms are usually used in these systems requiring large precisely labeled* *datasets to achieve a good performance. In this paper, we propose EcRD: an Edge-cloud based Road Damage detection and warning framework, that leverages the fast-responding advantage of Edge and the large storage and computation resources advantages of Cloud. There are three main contributions in this paper: we first propose a simple yet efficient road segmentation algorithm to enable fast and accurate road area detection. Then, a light-weighted road damage detector is developed based on Gray Level Co-occurrence Matrix features at Edge for rapid hazardous road damage detection and warning. Further, a multi-types road damage detection model is introduced for long-term road management at Cloud, embedded with a novel image generator based on Cycle-Consistent Adversarial Networks which automatically generates images with labels to further improve road damage detection accuracy. By comparing with the state-of-the-art, we demonstrate that the proposed EcRD can accurately detect both hazardous road damages at Edge and multi-types road damages at Cloud. Besides, it is around 579 times faster than Cloud-based approaches without affecting users' experience and requiring very low storage and labeling cost.*

Reference: Yachao Yuan, Md Saiful Islam, Yali Yuan, S. Wang, T. Baker, and L. M. Kolbe. "EcRD: Edge-cloud Computing Framework for Smart Road Damage Detection and Warning," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2020.3024885, 2020.

A.1.3 **A Light-Weight Deep-Learning Model with Multi-Scale Features for Steel Surface Defect Classification Abstract:** *Automatic inspection of surface defects is crucial in industries for real-time applications. Nowadays, computer vision-based approaches have been successfully employed. However, most of the existing works need a large number of training samples to achieve satisfactory classification results, while collecting massive training datasets is labor-intensive and financially costly. Moreover, most of them obtain high accuracy at the expense of high latency, and are thus not suitable for real-time applications. In this work, a novel Concurrent Convolutional Neural Network (ConCNN) with different image scales is proposed, which is light-weighted and easy to deploy for real-time defect classification applications. To evaluate the performance of ConCNN, the NEU-CLS dataset is used in our experiments. Simulation results demonstrate that ConCNN performs better than other state-of-the-art approaches considering accuracy and latency for steel surface defect classification. Specifically, ConCNN achieves as high as 98.89% classification accuracy with only around 5.58 ms latency over low training cost.*

Reference: Yang Liu, **Yachao Yuan**, Cristhian Balta, Jing Liu. "A Light-Weight Deep-Learning Model with Multi-Scale Features for Steel Surface Defect Classification." Materials 13.20, 4629, 2020.

A.1.4 **Unified Vision-based Methodology for Simultaneous Concrete Defect Detection and Geolocalization. Computer-Aided Civil and Infrastructure Engineering**

Abstract: *Vision-based autonomous inspection of concrete surface defects is crucial for efficient maintenance and rehabilitation of infrastructures and has become a research hot spot. However, most existing vision-based inspection methods mainly focus on detecting one kind of defect in nearly uniform testing background where defects are relatively large and easily recognizable. But in the real-world scenarios, multiple types of defects often occur simultaneously. And most of them occupy only small fractions of inspection images* *and are swamped in cluttered background, which easily leads to missed and false detections. In addition, the majority of the previous researches only focus on detecting defects but few of them pay attention to the geolocalization problem, which is indispensable for timely performing repair, protection, or reinforcement works. And most of them rely heavily on GPS for tracking the locations of the defects. However, this method is sometimes unreliable within infrastructures where the GPS signals are easily blocked, which causes a dramatic increase in searching costs. To address these limitations, we present a unified and purely vision-based method denoted as defects detection and localization network, which can detect and classify various typical types of defects under challenging conditions while simultaneously geolocating the defects without requiring external localization sensors. We design a supervised deep convolutional neural network and propose novel training methods to optimize its performance on specific tasks. Extensive experiments show that the proposed method is effective with a detection accuracy of 80.7% and a localization accuracy of 86% at 0.41 s per image (at a scale of 1,200 pixels in the field test experiment), which is ideal for integration within intelligent autonomous inspection systems to provide support for practical applications.*

Reference: Li, Ruoxing, **Yachao Yuan**, Wei Zhang, and Yali Yuan. "Unified vision-based methodology for simultaneous concrete defect detection and geolocalization." Computer-Aided Civil and Infrastructure Engineering 33.7, 527-544, 2018.

A.1.5 **Ada: Adaptive deep log anomaly detector**

Abstract: *Large private and government networks are often subjected to attacks like data extrusion and service disruption. Existing anomaly detection systems use offline supervised learning and employ experts for labeling. Hence they cannot detect anomalies in real-time. Even though unsupervised algorithms are increasingly used nowadays, they cannot readily adapt to newer threats. Moreover, many such systems also suffer from high cost of storage and require extensive computational resources. In this paper, we propose ADA: Adaptive Deep Log Anomaly Detector, an unsupervised online deep neural network framework that leverages LSTM networks and regularly adapts to newer log patterns to ensure accurate anomaly detection. In ADA, an adaptive model selection strategy is designed to choose pareto-optimal configurations and thereby utilize resources efficiently. Further, a dynamic threshold algorithm is proposed to dictate the optimal threshold based on recently detected events to improve the detection accuracy. We also use the predictions to guide storage of abnormal data and effectively reduce the overall storage cost. We compare ADA with state-of-the-art approaches through leveraging the Los Alamos National Laboratory cyber security dataset and show that ADA accurately detects anomalies with high F1-score 95% and it is 97 times faster than existing approaches and incurs very low storage cost.*

Reference: Yali Yuan and Sripriya Srikant Adhatarao, Mingkai Lin, **Yuan, Yachao**, Zheli Liu, and Xiaoming Fu. "Ada: Adaptive deep log anomaly detector." IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020.

A.1.6 **RACE: reinforced cooperative autonomous vehicle collision avoidance**

Abstract: *With the rapid development of autonomous driving, collision avoidance has attracted attention from both academia and industry. Many collision avoidance strategies have emerged in recent years, but the* *dynamic and complex nature of driving environment poses a challenge to develop robust collision avoidance algorithms. Therefore, in this paper, we propose a decentralized framework named RACE: Reinforced Cooperative Autonomous Vehicle Collision AvoidancE. Leveraging a hierarchical architecture we develop an algorithm named Co-DDPG to efficiently train autonomous vehicles. Through a security abiding channel, the autonomous vehicles distribute their driving policies. We use the relative distances obtained by the opponent sensors to build the VANET instead of locations, which ensures the vehicle's location privacy. With a leader-follower architecture and parameter distribution, RACE accelerates the learning of optimal policies and efficiently utilizes the remaining resources. We implement the RACE framework in the widely used TORCS simulator and conduct various experiments to measure the performance of RACE. Evaluations show that RACE quickly learns optimal driving policies and effectively avoids collisions. Moreover, RACE also scales smoothly with varying number of participating vehicles. We further compared RACE with existing autonomous driving systems and show that RACE outperforms them by experiencing 65% less collisions in the training process and exhibits improved performance under varying vehicle density.*

Reference: Yali Yuan and Robert Tasik and Sripriya Srikant Adhatarao and **Yachao Yuan**, Zheli Liu, and Xiaoming Fu."RACE: reinforced cooperative autonomous vehicle collision avoidance." IEEE transactions on vehicular technology 69.9: 9279-9291, 2020.

A.1.7 **Multi-Device Fusion for Enhanced Contextual Awareness of Localization in Indoor Environments**

Abstract: *Recently, with various developing sensors, mobile devices have become interesting in the research community for indoor localization. In this paper, we propose Twi-Adaboost, a collaborative indoor localization algorithm with the fusion of internal sensors, such as the accelerometer, gyroscope, and magnetometer from multiple devices. Specifically, the data sets are collected first by one person wearing two devices simultaneously: a smartphone and a smartwatch, each collecting multivariate data represented by their internal parameters in a real environment. Then, we evaluate the data sets from these two devices for their strengths and weaknesses in recognizing the indoor position. Based on that, the Twi-AdaBoost algorithm, an interactive ensemble learning method, is proposed to improve the indoor localization accuracy by fusing the co-occurrence information. The performance of the proposed algorithm is assessed on a real-world dataset. The experiment results demonstrate that Twi-AdaBoost achieves a localization error about 0.39 m on average with a low deployment cost, which outperforms the state-of-the-art indoor localization algorithms*

Reference: Yuan, Yali, Christian Melching, **Yachao Yuan**, and Dieter Hogrefe. "Multi-device fusion for enhanced contextual awareness of localization in indoor environments." IEEE Access 6: 7422-7431, 2018.