

Journal of Advanced Transportation

# Smart and Resilient Transportation Infrastructure Systems

Lead Guest Editor: Seyed Ali Ghahari

Guest Editors: Lateef Assi and Saeed Mohammadzadeh





---

# **Smart and Resilient Transportation Infrastructure Systems**

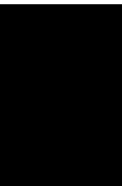
Journal of Advanced Transportation

---

## **Smart and Resilient Transportation Infrastructure Systems**

Lead Guest Editor: Seyed Ali Ghahari

Guest Editors: Lateef Assi and Saeed  
Mohammadzadeh







---

Copyright © 2022 Hindawi Limited. All rights reserved.























This is a special issue published in "Journal of Advanced Transportation." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Associate Editors

Juan C. Cano , Spain  
Steven I. Chien , USA  
Antonio Comi , Italy  
Zhi-Chun Li, China  
Jinjun Tang , China

## Academic Editors

Kun An, China  
Shriniwas Arkatkar, India  
José M. Armingol , Spain  
Socrates Basbas , Greece  
Francesco Bella , Italy  
Abdelaziz Bensrhair, France  
Hui Bi, China  
María Calderon, Spain  
Tiziana Campisi , Italy  
Giulio E. Cantarella , Italy  
Maria Castro , Spain  
Mei Chen , USA  
Maria Vittoria Corazza , Italy  
Andrea D'Ariano, Italy  
Stefano De Luca , Italy  
Rocío De Oña , Spain  
Luigi Dell'Olio , Spain  
Cédric Demonceaux , France  
Sunder Lall Dhingra, India  
Roberta Di Pace , Italy  
Dilum Dissanayake , United Kingdom  
Jing Dong , USA  
Yuchuan Du , China  
Juan-Antonio Escareno, France  
Domokos Esztergár-Kiss , Hungary  
Saber Fallah , United Kingdom  
Gianfranco Fancello , Italy  
Zhixiang Fang , China  
Francesco Galante , Italy  
Yuan Gao , China  
Laura Garach, Spain  
Indrajit Ghosh , India  
Rosa G. González-Ramírez, Chile  
Ren-Yong Guo , China

Yanyong Guo , China  
Jérôme Ha#rri, France  
Hocine Imine, France  
Umar Iqbal , Canada  
Rui Jiang , China  
Peter J. Jin, USA  
Sheng Jin , China  
Victor L. Knoop , The Netherlands  
Eduardo Lalla , The Netherlands  
Michela Le Pira , Italy  
Jaeyoung Lee , USA  
Seungjae Lee, Republic of Korea  
Ruimin Li , China  
Zhenning Li , China  
Christian Liebchen , Germany  
Tao Liu, China  
Chung-Cheng Lu , Taiwan  
Filomena Mauriello , Italy  
Luis Miranda-Moreno, Canada  
Rakesh Mishra, United Kingdom  
Tomio Miwa , Japan  
Andrea Monteriù , Italy  
Sara Moridpour , Australia  
Giuseppe Musolino , Italy  
Jose E. Naranjo , Spain  
Mehdi Nourinejad , Canada  
Eneko Osaba , Spain  
Dongjoo Park , Republic of Korea  
Luca Pugi , Italy  
Alessandro Severino , Italy  
Nirajan Shiwakoti , Australia  
Michele D. Simoni, Sweden  
Ziqi Song , USA  
Amanda Stathopoulos , USA  
Daxin Tian , China  
Alejandro Tirachini, Chile  
Long Truong , Australia  
Avinash Unnikrishnan , USA  
Pascal Vasseur , France  
Antonino Vitetta , Italy  
S. Travis Waller, Australia  
Bohui Wang, China  
Jianbin Xin , China



---

Hongtai Yang , China  
Vincent F. Yu , Taiwan  
Mustafa Zeybek, Turkey  
Jing Zhao, China  
Ming Zhong , China  
Yajie Zou , China



# Contents

## **Impact Assessment of Interlocking Systems on Single-Track Railway Lines as a Measure Leading to Resilient Railway System**

Michael Bažant  and Josef Bulíček 



Research Article (18 pages), Article ID 7025130, Volume 2022 (2022)

## **Improvement of Multiclass Classification of Pavement Objects Using Intensity and Range Images**

Elham Eslami  and Hae-Bum Yun 



Research Article (20 pages), Article ID 4684669, Volume 2022 (2022)

## **Stacking Ensemble Learning Process to Predict Rural Road Traffic Flow**

Arash Rasaizadi  and Seyedehsan Seyedabrishami 

Research Article (12 pages), Article ID 3198636, Volume 2022 (2022)

## **Resilience of Urban Road Network to Malignant Traffic Accidents**

Yiding Lu , Zhan Zhang , Xinyi Fang, Linjie Gao, and Linjun Lu

Research Article (13 pages), Article ID 3682472, Volume 2022 (2022)

## **Cloud Update of Geodetic Normal Distribution Map Based on Crowd-Sourcing Detection against Road Environment Changes**

Chansoo Kim , Sungjin Cho , Myoungho Sunwoo , Paulo Resende, Benazouz Bradai, and Kichun Jo 

Research Article (26 pages), Article ID 4486177, Volume 2022 (2022)

## **Yolov4 High-Speed Train Wheelset Tread Defect Detection System Based on Multiscale Feature Fusion**

Changfan Zhang , Xinliang Hu , Jing He , and Na Hou


Research Article (13 pages), Article ID 1172654, Volume 2022 (2022)

## **An Overview of Pavement Degradation Prediction Models**

Amir Shtayat , Sara Moridpour , Berthold Best , and Shahriar Rumi









Review Article (15 pages), Article ID 7783588, Volume 2022 (2022)

## **Evaluation of the Cost of Intelligent Upgrades of Transportation Infrastructure for Intelligent Connected Vehicles**

Zongwei Liu, Haokun Song, Hong Tan, Han Hao, and Fuquan Zhao 

Research Article (15 pages), Article ID 5841373, Volume 2022 (2022)



## **Crack Detection Method of Sleeper Based on Cascade Convolutional Neural Network**

Liming Li , Shubin Zheng , Chenxi Wang , Shuguang Zhao , Xiaodong Chai , Lele Peng , Qianqian Tong , and Ji Wang 

Research Article (14 pages), Article ID 7851562, Volume 2022 (2022)

## Research Article

# Impact Assessment of Interlocking Systems on Single-Track Railway Lines as a Measure Leading to Resilient Railway System

Michael Bažant <sup>1</sup> and Josef Bulíček <sup>2</sup>

<sup>1</sup>Department of Software Technologies, University of Pardubice, Faculty of Electrical Engineering and Informatics, Studentská 95, Pardubice 532 10, Czech Republic

<sup>2</sup>Department of Transport Technology and Control, University of Pardubice, Faculty of Transport Engineering, Studentská 95, Pardubice 532 10, Czech Republic

Correspondence should be addressed to Michael Bažant; michael.bazant@upce.cz

Received 4 January 2022; Revised 2 June 2022; Accepted 30 June 2022; Published 25 August 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Michael Bažant and Josef Bulíček. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Railway systems should be resilient to play a key role in creating sustainable development. Single-track railway lines are seen as potential bottlenecks due to limited capacity. More advanced railway interlocking systems (such as ETCS or satellite-based control systems) are being developed. On the other hand, the installation of these interlocking systems is a complex and time-consuming and costly task. For this reason, it is necessary to recognize the impact of potentially installed system with capacity, stability of timetable, quality, and other associated effects. The assessment is based on a set of simulation experiments using stochastic microscopic simulation model in the OpenTrack software tool. The focus is on railway operation with automatic block and automatic line blocking systems. If these two systems will have positive capacity effects, it is a basic presumption also for systems such as moving block (e.g., ETCS L3) to be effective. Research has shown that the significance of such measures can be best supported by linking to a matching timetable concept that will make full use of the benefits offered by these interlocking systems. The results reached in this research should be potentially applied, for example, by prioritizing of single-track railway lines for possible installation of such interlocking system. It can be achieved based on the capacity and operational effects examined.

## 1. Introduction

Railway industry plays a key role within an effort to create a sustainable environment. The issue is that there are many single-track railway lines. For illustration, 78.3% of railway network in the Czech Republic is single track (7324 km), whereas this situation corresponds with European context, where backbone lines are usually double track and single-track lines are considered for supplementary connections. Single-track lines are beneficial from the economic aspect, but they are associated with several operational problems. Occupation time is relatively long and trains often have to wait for crossing and clearing of a segment at stations, what is leading, for example, to decrease of travel speed.

Although this question has been known for years, it is becoming increasingly acute today. All railway lines are in a competitive environment with road traffic. Railway

transport must be resilient to be a successful and punctual passenger and freight transport system. Qualitative demands are increasing due to this. The lines with relative limited extent of operation are not an exception. Maximal number of operated trains on a railway line is not a crucial factor as it used to be in the past. Reduction of dwell and travel time as well as increasing speed are important features nowadays at almost all single-track lines including lines with relative low number of operated trains. Infrastructure of many single-track lines itself is considered a bottleneck in this point of view.

Another associated issue is that the relation between infrastructure extent, traffic volume, and operational quality must be balanced. Improving the interlocking system should be part of this process. The concept of “interlocking system” applied in this article integrates traffic control system, signaling system together resulting into options how the line



segment can be operated by trains. Line sections are more important than railway stations, but the two parts are interconnected.

The selection of the appropriate level of interlocking is determined by understanding of the capacity and operational effects associated with the interlocking system. In the railway sector, there is currently an extensive development of interlocking systems called as moving block that allow the presence of more than one train on a single track on a section of a railway line (e.g., European Train Control System (ETCS level) L2, L3, or satellite location-based systems). Some of these systems are based on cab signaling that do not use wayside signals, some combine cab and wayside signals. Specific way of technical solution is not crucial for this article.

The cost of these systems is significant. For this reason, it is necessary to select which railway lines and their line sections between stations are suitable for installation of such a system. Outputs of this research should be applied by prioritizing the railway lines to determine the order in which these systems should be installed. These priorities can be set after understanding how these systems will affect the capacity of railway lines and how they can contribute to the quality of operation on these lines.

It is necessary to define what are the possibilities of interlocking on single-track lines and how they will be covered in this article. In terms of the presence of multiple trains that run consecutively in a line section, line interlocking systems can be divided into four levels.

- (1) (I1) Basic level represents trains that run according to sections between neighbor stations and the presence of a single train in them (line sections are not divided into spatial sections)
- (2) (I2) The section is divided into a limited number of spatial sections by one block (the most common configuration). A larger number is possible but rather rare, whereas there can be one train in each. State-of-art blocks work automatically, so they are assumed in the article.
- (3) (I3) The equipment forming several spatial sections is adapted to the operating conditions on the line (speed and braking distances) so that the interlocking system with associated signals can maximize the capacity of the line section. This is achieved by allowing multiple trains to follow each other on the track with minimized space separation. This is referred to as an automatic line block in the Czech Republic and in some other countries.
- (4) (I4) This progressive level is so-called moving block. Trains can follow each other at different spatial intervals, the size of which corresponds to the speed of the train and the operating situation. Such a system is foreseen in ETCS L3 applications, but this principle may also be present in some other systems for regional lines or underground metro systems.
- (5) (I5) Double-tracked line segment. This variant is applied as supplementary only for possibility to have regard to next step of infrastructure extent.

The technical solution of the line interlocking system determines the technological times important for timetable design. Levels I1 to I3 are assumed in the article, especially in the configuration of automatically operating devices (shorter technological time).

Interlocking systems at level I5 can locate a train on the infrastructure considerably more accurately and precisely than ever before. This opens up new opportunities in the traffic control area. In both instances, the cost and technical possibilities of the application of such new interlocking systems are an issue—in other words, not all lines or line sections can be equipped with such systems in the short term at least. This is the reason why it is needed to conduct this research to get some information for the decision-making process how to select the lines for the installation of such a system.

The inclusion of the moving block (I4) in the article is indirect. A small volume of traffic is usually operated on single-track lines. Necessity to alternate both directions is also there. It means that usually a small number of trains run together in fleets (platoons). Although moving blocks can shorten headway times, this can be replaced with data obtained from the research focused on automatic line block as an interlocking system. This is an acceptable model simplification, but for evaluation of the relation between capacity and quality of single-track lines.

For completeness of the solution and comparability, one line section is applied as a double track in one scenario (Sc08). This infrastructure variant is systematically labeled as I5.

The authors have long been engaged in research into the capacity of the railway infrastructure based on the use of stochastic simulation models. The aim of the research is to identify technological operational indicators that can be used to facilitate the description and assessment of this capacity.

Until now, the capacity of railway infrastructure has been determined mainly by time aspects of its use. The maximum (theoretical) capacity  $n_{\max}$  can thus be determined according to the following formula:

$$\begin{aligned} n_{\max} &= \frac{T}{t_{\text{avg}}} \\ &= \frac{TN_t}{\sum_{i \in I'} t_{\text{occup}i}} [-]. \end{aligned} \quad (1)$$

This is the basic principle, the available time  $T$  [min] is divided by the average time of occupation by one train or operation  $t_{\text{avg}}$  [min]. Alternatively, this can also be expressed by the total number of trains  $N_t = |I'|$ , where  $I'$  is the set of trains under investigation and the occupation time of each train or operation  $t_{\text{occup}i}$  [min].

The second basic indicator is the occupation degree  $D_{\text{occup}}$  formula (2). It is the proportion of time used and the total available time  $T$  [min].

$$D_{\text{occup}} = \frac{\sum_{i \in I'} t_{\text{occup}i}}{T} [-]. \quad (2)$$

These indicators will continue to be used. However, they need to be complemented by qualitative perspectives and by perspectives that are more reflective of the prevailing nature of traffic and of the costs. The aim is to find a scale of infrastructure that is efficient, proportionate to operational requirements, and enables rail transport to be organized in the required quality. The main indicator that describes quality is delay, as confirmed, for example, by Börjesson and Eliasson [1].

Simulation models are inherently descriptive. The results only provide information on the replications performed, that is, on the assessed traffic variants. On the other hand, simulations are rather complex and time-consuming processes, and so, theoretical considerations must be also devoted to the simulation procedure itself. One of the additional objectives of this research is to recommend procedures performing such assessments effectively with modest demands on complexity and time.

## 2. Train Fleeting (Platooning)

Fleeting of trains means that two or more trains run in one line section in the same direction and at the same time. This is a prerequisite for an interlocking system that allows the presence of more trains on one track (by division into spatial sections) to be efficient.

For instance, in the city agglomeration of Hradec Králové-Pardubice in Eastern Bohemia, there are four cases where train fleeting can be applied because line sections are shared by two or more lines of passenger transport. Trains can be fledted at these sections due to coordination at interchanging nodes. These sections are highlighted by the arrows in Figure 1 that illustrates the frequency of such cases in practice.

## 3. Hypothesis and Aim

Specific objective is to assess the effect of a railway line interlocking system on the traffic occurring on a single-track line, using stochastic experiments in a simulation model. Changing interlocking system can be a way how the capacity as well as the quality of operation can be improved.

Specifically, various interlocking systems are looked upon as measures to increase the railway track's capacity, with emphasis on qualitative rather than quantitative benefits. The goal is to find the background for a future methodology for efficiently selecting interlocking systems for low- and medium-traffic lines, where the quantitative aspect does not play a major role and to prioritize the equipment of the line sections.

The research hypothesis can be formulated as follows. Stochastic simulation can be successfully applied for the determination of the new analytical indicators of railway capacity as well as a tool that is able to identify the contribution of individual types of interlocking systems.

Traffic stability as the qualitative aspect of capacity can be applied as the main indicator.

Traffic stability will be calculated by formula (3) as the average change of delay  $\Delta d$  [s].

$$\Delta d = \frac{\sum_{j \in J} \sum_{i \in I'} (d_{ij}^{\text{OUT}} - d_{ij}^{\text{IN}})}{N_t \cdot N_r} \quad (3)$$

It is computed as the difference between the delay at the output from simulation  $d_i^{\text{OUT}}$  [s] and the delay at input to simulation  $d_i^{\text{IN}}$  [s] by a train of  $i \in I'$  within a replication of  $j \in J$ . Trains are coming from the subset of  $I'$ . Number of assessed trains is marked as  $N_t = |I'|$  and the number of replications as  $N_r = |J|$ .

## 4. State-of-the-Art

Effect of various interlocking systems on the capacity has been discussed previously [2]. Emphasis was on the European train control system level 2 (ETCS L2), which was compared to the interlocking system NS'54/ATB. For this, the authors used the compression method as per UIC 406, accentuating only the quantitative aspect of capacity. The criterion used by us in our research is the value of the delay, whereby we attempt to accentuate the qualitative aspect, that is, to look at the problem from another side. Simulation as an adequate tool for the assessment of railway interlocking as well as for the organization of railway operation is seen also [3, 4].

Overview of studies (25 various studies) focused on railway capacity assessments is presented previously [5], where various approaches are discussed—analytical, simulation, and combined approach. There are also listed commonly used software tools for capacity assessment. The differences between the United States and European approaches are also outlined and studies are also broken into several categories and there are also highlighted key similarities and differences between the United States and European rail systems, where 14 U.S. studies and 11 European studies are considered. Our approach could be incorporated into timetable-based simulation software as is common in Europe, whereas in the United States the predominant approach is without considering timetables.

In the article by Abril et al. [6], there is pointed difference between theoretical and practical capacity with an introduction of Spanish MOM system that contains optimization module for obtaining feasible and optimized timetables. There is also comparison of various parameters of timetable (e.g., headway times, line sections length, train speed) and its impact on capacity. The article evaluates the relationship between the subsequent intermediate period and capacity on single-track and double-track lines, and attempts to tabulate this relationship. In our article, this principle is translated into schedule modifications in each scenario.

Method for designing a single-track rail line for a reliable high-speed passenger train service is presented in the article by Petersen and Taylor [7]. The operation under consideration is planned as homogeneous with one type of trains—high-speed passenger transport. Among other constraints that are relevant to our research, the primary focus is on the location and length of double-track inserts, with no fleeting considered. Different scenarios of double-track line lengths,

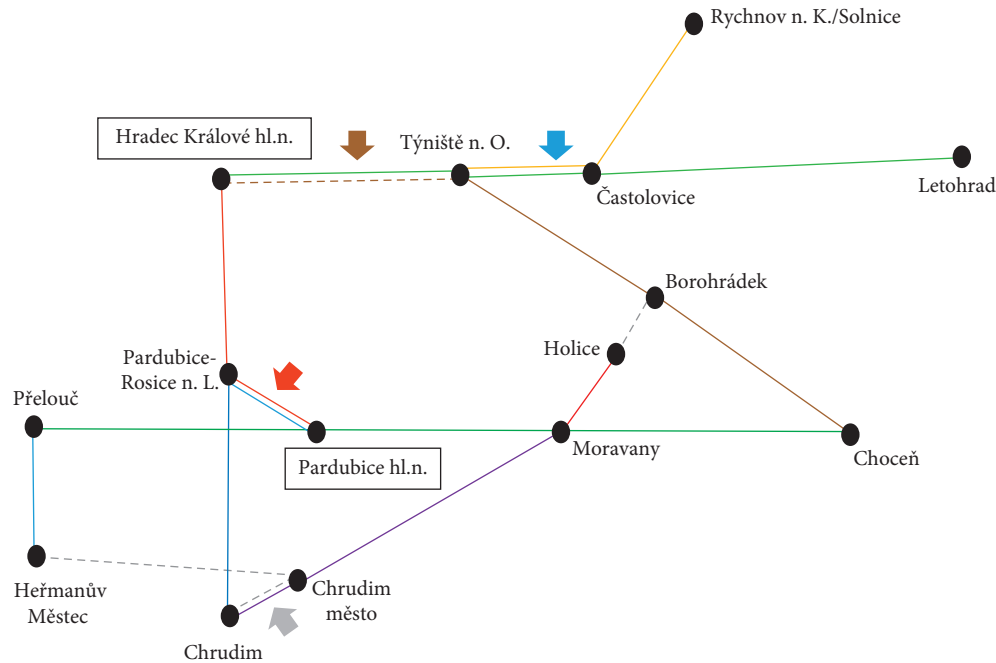


FIGURE 1: Simplified scheme of the railway network in Hradec Králové-Pardubice agglomeration with parallel regional train lines (marked with arrows).

the amount of delayed trains, as well as delay values were evaluated using computer simulation in this article.

We employ the OpenTrack tool for our simulations. This is a widely used software enabling discrete and continuous approaches to simulation to be combined [8]. It can be used in the modeling of both high-speed lines [9] and conventional lines [10] as well as suburban lines [11, 12].

Interactions of trains running at different speeds, including the capacity aspect, have been addressed previously [13], providing evidence of the importance of this topic. The author also pointed to the fact that railway traffic is a phenomenon that is affected by a number of external factors.

The same idea dominates the article by Mussone and Calvo [14]. The authors also discussed the potential of a comprehensive analytical assessment of the railway capacity. The topic is still actual, perhaps also in a new context—with respect to the question as to how to effectively select the need for and scope and method of setting up detailed microscopic simulation models.

Very often, the capacity issue in the context of an applied interlocking system is addressed for highly burdened lines in efforts to attain the maximum capacity. The article by Dicembre and Ricci [16] is an example of such a solution [12]

Unlike that solution, our research focuses on the issue of how a more advanced interlocking system can contribute to traffic quality on low- and medium-traffic intensity lines, where more importance is attached to the highest possible traffic quality than to attaining the maximum capacity.

The economic aspect is frequently stressed in the context of current railway market liberalization. Capacity assignment is a topic discussed previously [16].

Timetable-oriented point of view on the interaction between railway operation and infrastructure is presented by Široký et al. [17].

Railway capacity assessment in the international context is currently governed by the UIC 406 code. Still, despite the existence of the code, much space remains for additional research in this area. The compression method and (once again) the stochastic approach to traffic are discussed previously [18], where the application in Sweden is also described. Application in Slovakia is the topic analyzed by Šulko et al. [19].

## 5. The Simulation Model and Its Application to the Assessment of Infrastructure and Timetable Variants

In our research, a microscopic simulation model of a single-track line was set up in OpenTrack software tool. The model contains all needed data for research on microscopic level, including details about infrastructure, rolling stock, timetable, and behavior of trains in stochastic conditions.

The line is 50-km long and encompasses five interstation line sections. Input delays are stochastically generated. Train delays are generated based on the discrete probability distributions obtained through a survey made by the authors on the railway network in the Czech Republic.

We realize 200 replications for each scenario (scenarios are listed in Table 1), while replication contains traffic peak of 4 hours. To be able to collect data for delayed trains, we consider a replication time of 6 hours.

The primary objective of the research is the need to find the appropriate scale of the infrastructure and the technological equipment of the transport infrastructure to match the required volume of traffic. In addition to being cost-effective, this scale will also allow for reliable operation within the given options. The concept of the simulation

TABLE 1: Simulation scenarios.

Scenario	Infrastructure	Number of trains (full length of the line/C-D/E-F)	Train fleeting in line sections	Role in the study
Sc01	I1—trains are organized between stations	14/14/18	None	Basic (input) state
Sc02	I2 at sections C-D	13/15/18	C-D	Role of automatic block (I2)
Sc03	I2 at sections A-B, E-F	14/14/22	A-B, E-F	
Sc04		13/15/18	C-D	Role of automatic line block (I3)
Sc05	I3 at sections C-D	14/22/14	None	Role of I3 without fleeting
Sc06		14/22/14	C-D	Role of I3 with fleeting
Sc07	I3 at C-D, E-F	14/22/22	None	I3 extended to two sections
Sc08	I5 C-D double track (operated by I2)	14/22/14	C-D	Middle section C-D double track (I5)
Sc07-1	I3 at C-D, E-F	14/22/22	None	I3 at 2 sections together with increased mean value of delay

assessment is carried out as shown in Figure 2—we address all possible options of interlocking systems (I1–I4) in our research. Moving block, as it was stated earlier, is not shown in the individual scenarios, as a necessary condition for its effectiveness is the effectiveness of both the automatic block and the automatic line block, as it is essentially a higher level of both. It is not necessary to model moving block itself due to this. Moving block is replaced by automatic block and automatic line block.

The set of applied simulation scenarios is based on the logic shown in Figure 2. Variants of the types of line interlocking systems and various timetable variants were considered. The need of different timetables is caused by the fact that certain types of interlocking systems will only have a positive capacity effect in combination with a timetable that takes this into account.

As can be seen in Figure 3, the 50 km long line consists of five line sections connecting six stations marked A–F. Signaling devices enabling the presence of more trains in the spatial section are primarily inserted in the middle section C–D, in some cases in the outermost sections A–B and E–F. The timetable is based on 14 trains running throughout the line. The traffic is then reinforced in the last (suburban) section E–F by four or eight trains depending on the scenario. Further reinforcement is then added in scenarios Sc01–Sc05 and Sc07–1 in sections C–D, where eight section trains are added. This range of services covers a time window of 4 hours and consists of two types of trains—regional (slow) trains and long-distance (fast) trains. An overview of the simulation scenarios comprising this study is given in Table 1. The symbols I1–I5 stand for individual variants of infrastructure. The scope and infrastructure configuration are shown in Figures 3–8.

Scenario Sc07-1 is complementary and works with a higher mean value of stochastically generated delay of 12 min. Thus, a situation of relatively unstable traffic is created by this scenario. All other scenarios work with a mean delay value of 4 min.

## 6. Features of the Stochastic Modeling

Stochastic modeling provides the possibility to assess and evaluate not only the quantitative view, but also the

qualitative aspects. These are very important for resilient railway systems.

Different train delay characteristics were applied in the scenarios:

- Typical delay: 63% trains at the input meet the exponential distribution patterns with a mean delay time of 255 s, the longest delay generated was 1200 s.
- Larger delays due to rebuilding or other building works on the adjacent sections: 80% trains at the input obey the exponential distribution with a mean time of 720 s, the longest delay generated was 1200 s.

The output parameters obtained from the above scenarios are described in the sections that follow.

There were calculated 200 replications for all considered scenarios (values about 5 seconds for half widths of the arithmetic means with 95% confidence interval as presented).

Unless otherwise stated, delay values represent average value over all 200 replications.

## 7. Simulation Scenarios and Results Obtained

Basic timetables can be modified in individual scenarios accordingly to be suitable for the applied interlocking systems as well as to be suitable for the assessed effects.

*7.1. Scenario Sc01: Single-Track Layout along the Entire Length of the Line without Further Organization of Train Movements in Spatial Sections.* The initial situation is a single-track line on which the traffic is organized according to line sections. Related train diagram (timetable) is attached in Appendix Figure 14.

There are four extra suburban trains of Direction 1 inserted in the final spatial section E–F. However, the timetable design has shown the limited capacity imposed by this traffic organization, which has necessitated the need to connect these trains in Direction 2 with the basic trains. This effect does not occur in some other scenarios when using the automatic block.

Figure 4 shows that the line tends to reduce the input delay in both directions during the run. In Direction 2 at

Interlocking system on railway line	No block I1	Block in the middle I2	Automatic block I3	Moving block I4-I5
Effect on capacity	min ←  → max			
Simulation scenarios	Sc01	Sc02, Sc03, Sc08	Sc04-Sc07	---

FIGURE 2: Concept of the simulation assessment within performed scenarios.

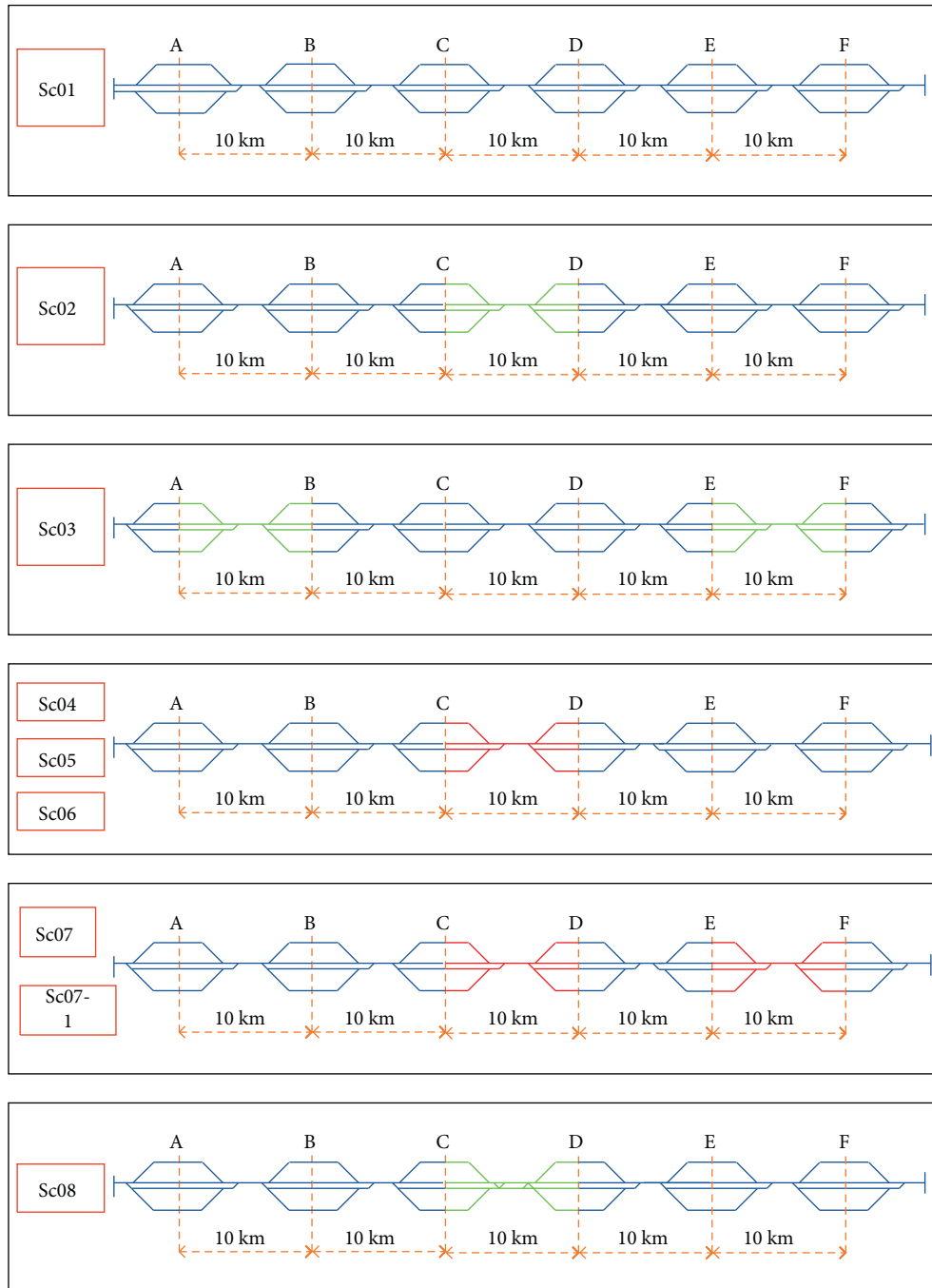


FIGURE 3: Infrastructure layout for scenarios with highlighted changes in line sections.

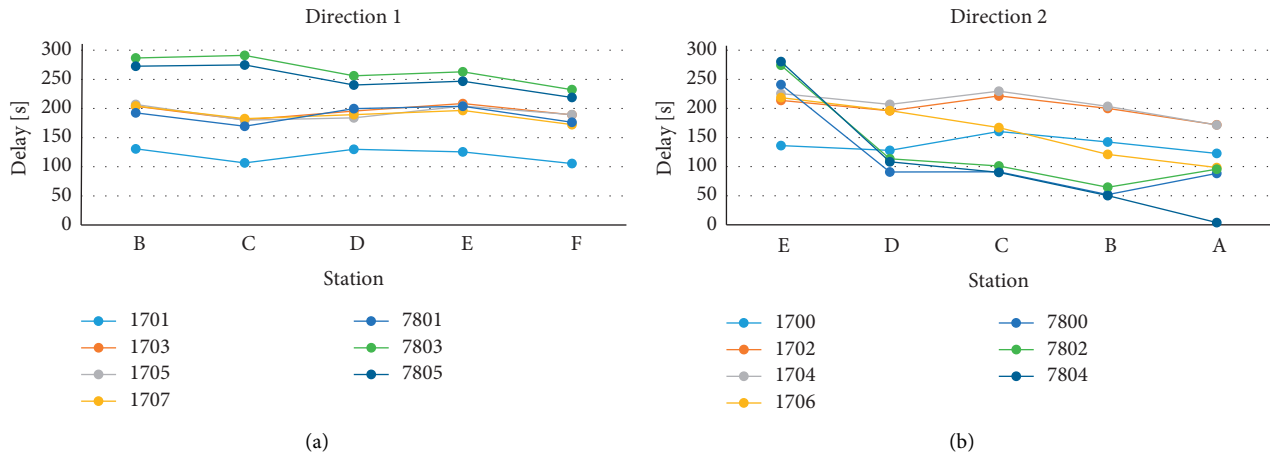


FIGURE 4: Evolution of average delay on arrival at individual stations in scenario Sc01.

station D, there is a significant reduction in delay, however, this is due to the crossing of trains at station E where passenger trains are scheduled to stay for 8.5 min. Trains 7800 and 7802 increase delay in the last section B-A due to the fact that they cross with oncoming trains at station B and depart (for the crossing operating interval) only 1.5 min after the oncoming train arrives.

The conclusion on this scenario is that a lower capacity can be expected in the interstation sections of the single-track line with the given mode of traffic organization.

**7.2. Scenario Sc02: Single-Track Layout along the Entire Length of the Line with an Automatic Block in Sections .** This scenario is principle based on the situation in scenario Sc01, which is extended by the introduction of an automatic block in the middle of section C-D, that is, in the middle of the entire modeled line. The placement of this section to the middle section of the line is based on a general judgment rather than an analysis of the timetable and traffic volume. The concept and the traffic volume are the same as in scenario Sc01, however, the timetable is modified to consider the option to run planned trains as fleeted on the section with the block. Trains are fleeted in a fast-slow sequence. Train diagram is in Appendix Figure 15.

The results for this scenario are shown in Figure 5. In Direction 1, train 1701 runs alone in sections A-D, so there is a noticeable attempt to reduce the stochastically generated input delays. At stations D and E, it crosses closely with oncoming trains, the average delay values start to increase. Another interesting effect is that for trains 1703, 1705, and 1707, the average delay values increase in the C-D section, while they decrease for trains 7803, 7805, and 7807. This is because the regional trains 78xx are overtaken at station C by the express trains 17xx. Therefore, the 78xx regional trains stay there for 16.5 min, which causes the delay to decrease, and the situation is similar in the opposite direction. These stays cause an almost absolute reduction of generated entry delays to zero for the price of unattractive stays for passengers and taking advantage of the fleeting opportunity.

**7.3. Scenario Sc03: Single-Track Layout along the Entire Length of the Line with Automatic Block in Sections A-B, E-F.** In contrast to the previous Sc02 scenario, there is an attempt to adapt the location of the automatic block to the extent of traffic, so one is placed in the last section of E-F, where the extent of suburban traffic is increased. The second automatic block is then inserted in section A-B (allowing a shorter interval between trains departing from both terminal stations).

The timetable was also adapted to the introduction of automatic blocks (see Appendix Figure 16). Train fleeting is used on both sections A-B and E-F. A significant change is the fact that the inserted trains 178xx in the (suburban) section E-F are introduced in both directions on the hourly interval and are thus no longer coupled in Direction 2 with the basic trains running on the entire line.

The evolution of delay of individual basic trains (operating on the entire line) is shown in Figure 6.

As in the previous cases, two basic elements affecting the stability of the timetable are evident in this operational scenario. The first one is the influence of the length of stay in the station (for traffic reasons), where its extension has a positive effect (see trains 78xx in Direction 1 with a stay in station C of 8 min). The second is that crossing with an oncoming train (almost) at the crossing interval can have a negative effect on stability (trains 17xx and crossing at stations B, D, and E in Direction 1). Train 1707 lacks a crossing at E at the end of the analysis period, which is reflected in the figure.

Subconclusion from the Sc01 to Sc03 scenarios: the introduction of an automatic block may not have a clear impact on the increase in capacity (analytically determined), whereas it depends on the constructed timetable. It also has been shown that the automatic block has benefits in terms of introducing some operational concepts (e.g., the possibility of running embedded commuter trains in the E-F section separately in both directions). On the other hand, a certain paradox has emerged, namely, that unattractive long stays in stations for traffic reasons can lead to increased timetable stability from the passengers' point of view.

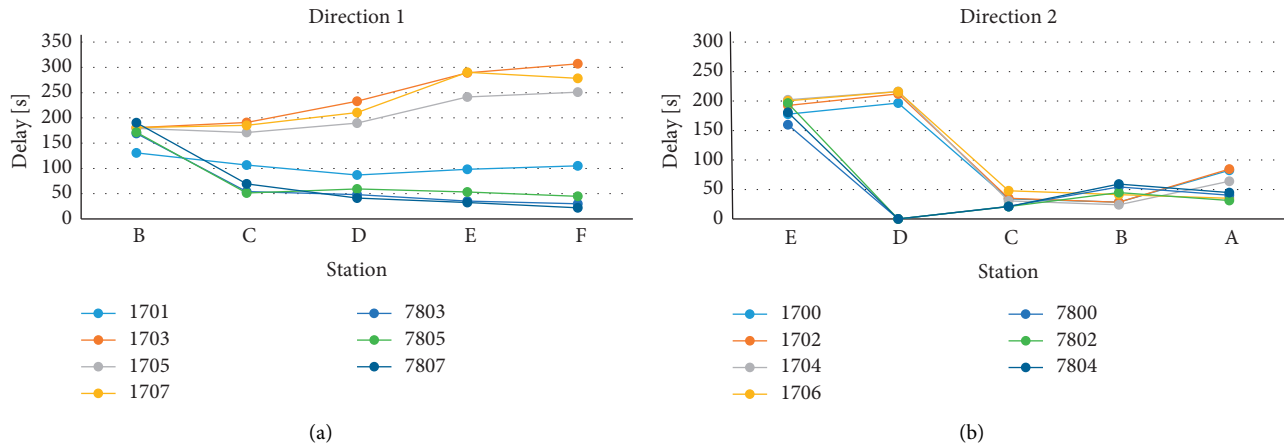


FIGURE 5: Evolution of average delay on arrival at individual stations in scenario Sc02.

For complexity, the segment E-F was assessed in an analytical way using Sc0–Sc03, as given in Table 2.

**7.4. Scenario Sc04 Is Focused on the Comparison between Automatic Line Block and Automatic Block.** The first important thing is to decide whether to use an automatic line block or an automatic block (or a more advanced interlocking system). Two (side) variants are compared. Where an automatic block is used, section C-D, that is in the middle of the line, is divided into two spatial sections. The automatic line block divides the same section into partial spatial sections 1 km each.

Four couples of long-distance trains and three couples of regional trains with longer running times are used. The timetable has been set up so that fleeted traffic is practiced in the middle section with an automatic line block, which means a long-distance train and a regional train running one after the other so that the division of the section into two spatial sections is deliberately used. The same timetable (Appendix A2) is used in the variant with an automatic block, though the latter would enable the ensuing interval to be shortened.

The simulation revealed that the mean delay of the trains arriving at the destination station (F or A), except for the regional trains in Direction 1 was shortened by the introduction of the automatic block (Table 3).

From Table 3, it is clear that the introduction of the automatic block is most beneficial to the long-distance trains in Direction 1, where the mean delay at the arrival at the destination station at the end of the 50 km line was 80 seconds. The largest increase in the delay, on the other hand, was found for the regional trains in the same direction where the mean delay is 73 seconds. This is kind of paradoxical, because the regional trains run tightly following the long-distance trains, and so one would expect that the change of the interlocking system enables more trains to move within the interstation section.

At this point, it can be concluded that the replacement of an automatic line block with an automatic block in one

“isolated” section is qualitatively beneficial to some (small) extent.

**7.5. Scenarios Sc05 and Sc06: Train Fleeting.** Fleeting was assessed on the middle section C-D of the single-track line model—the only section that is equipped with an automatic block with 1-km long spatial sections. For emphasizing, the traffic in this section was made denser by inserting additional trains running only in this section of the model. The rate is one couple of added trains per hour. The traffic concerns six trains per hour (three in either direction: long-distance, regional, and added train sets).

Two timetable variants (named as scenario Sc05 and scenario Sc06) were set up and compared: the number of trains was identical, but the added train running patterns were different. Scenario Sc05 included alternating train runs, the automatic block being thus virtually unused, as shown in Figure 7. Train diagram for scenario Sc05 is attached as Appendix Figure 17 and train diagram for Sc06 as Appendix 18.

The scenario Sc06 includes running the trains as fleeted to use the section division into spatial sections. To preserve the real aspects of traffic on medium burdened lines, only two trains in the same direction are considered in a fleet, but the interlocking systems allow even more. Moreover, the trains are fleeted in the odd direction only, whereas this approach provides the opportunity to compare group traffic (Direction 2, trains with even numbers) and fleeted traffic (Direction 1, trains with odd numbers).

First, the development of the mean delay times at the arrival to the stations (A–E) is compared for the entire line applying alternating traffic of the added trains as shown in Figure 7. This development is shown in Figure 9.

This can also be compared to the development of the delay of the same trains with additional trains being run within sections C-D in the fleeted mode in one direction and in the group mode in the other direction (Figure 8). The results are shown in Figure 10.

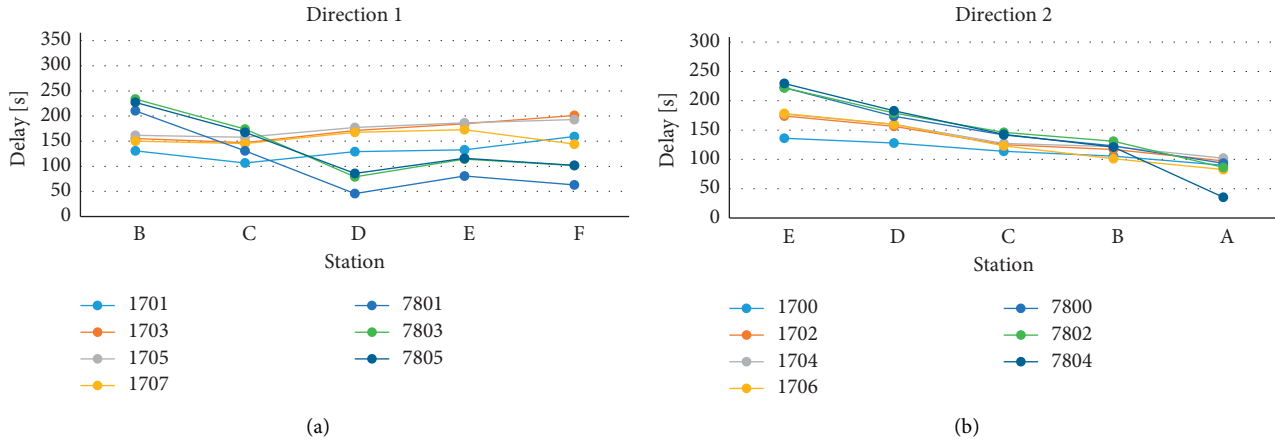


FIGURE 6: Evolution of average delay on arrival at individual stations in scenario Sc03.

TABLE 2: Analytical assessment in segments E-F.

Scenario	Sc01	Sc02	Sc03
$n_{\max}$ (trains/4h)	27	27	31
$D_{\text{occup}}$ [-]	0.648	0.648	0.708

TABLE 3: Comparison of an automatic line block and an automatic block in sections C-D.

Average delay at arrival (s)	Direction 1		Direction 2	
	Line block	Automatic block	Line block	Automatic block
Type of trains				
Regional	32	105	29	26
Long distance	235	155	67	51

For the trains with odd number (in Direction 1), the delay decreased slightly, by 10.0 s in average, for the long-distance trains (1701, 1703) and increased by 1.6 s for the regional trains (7801, 7803, 7805) that run along the entire line.

The patterns for the long-distance trains and regional trains are also different from the even trains (in Direction 2). The delay of the regional trains (7800, 7802, 7804) decreases continuously along the entire line. Compared to the alternating traffic mode, the mean delay is 2.6 s lower. This traffic model is inconvenient for long-distance trains, the delay is 5.3 s longer. This is due to the structure of the timetable (timetable composition), where the regional traffic trains are run within section F-E 15 minutes after the preceding trains, whereas the long-distance trains leave station F after a tight crossing (to the interval) the passenger train in the opposite direction. The regional trains pass the long-distance trains in E, the latter reducing their delay.

The situation of the added trains (22xxx series) running within the middle section C-D only is as follows. The stochastically generated delay is increased in both traffic variants—by 18.1 s (in average) in the odd Direction 1 and 9.2 s

in the even direction for the alternating traffic mode. If the traffic is organized in the fledted/group mode, the delay also increases, but only by 4.8 s in the odd Direction 1 and 1.3 s in the even Direction 2.

The simulation indicates that the installation of an automatic block in the middle section is beneficial to some extent but not very much from the global aspect. What is found to be significant is the link to the timetable structure in cases when timetable is designed, so that it respects infrastructure specifics.

Scenario Sc07: assessment of the extended application of automatic block on one of the suburban line sections with increased volume of suburban traffic: impact of different delay values.

Automatic block in two sections within the railway line is considered. The first one is sections C-D in the middle of the line. The second one is the suburban section E-F at the end of the line in Direction 1. There are eight more suburban trains inserted to this section. Average train delay values on arrivals at individual stations for average delay of 4 min are in Figure 11. Train diagram is in Appendix Figure 19.

7.6. Scenario Sc08: Assessment of Double-Track Line Sections C-D in the Middle of the Line. Double-track section C-D is used for crossing trains by moving both trains. The aim is to assess whether this approach can be effective also for the improvement of operational stability (reliability). For that reason, regular crossing of trains moving in the section is not planned in timetable. Train diagram attached in Appendix A5 is applied also for this scenario.

When the results in Figures 10 and 12 are compared, it is obvious that more significant change occurred in the Direction 2 only. Average delay of long-distance trains decreased from the span 120–160 s to values slightly less than 100 s. This positive effect is related to the fact that these long-distance trains cross with a pair of added and long-distance trains that run in the opposite directions. If these trains are delayed, they cross at double-track line sections.

Partial conclusion: considering double-track line only in one of the line sections is beneficial for operational stability



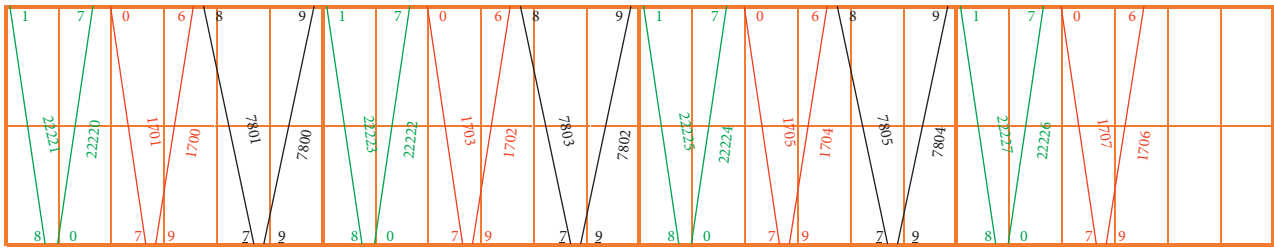


FIGURE 7: Scenario Sc05—trains running within the middle section of the line, the automatic block was unused (red, long-distance trains; black, regional trains; green, trains added to the middle section only).

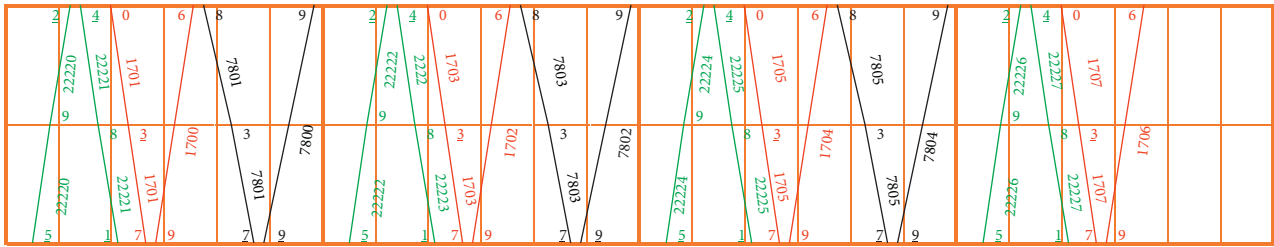


FIGURE 8: Scenario Sc06—trains running in the middle line section, as fledged.

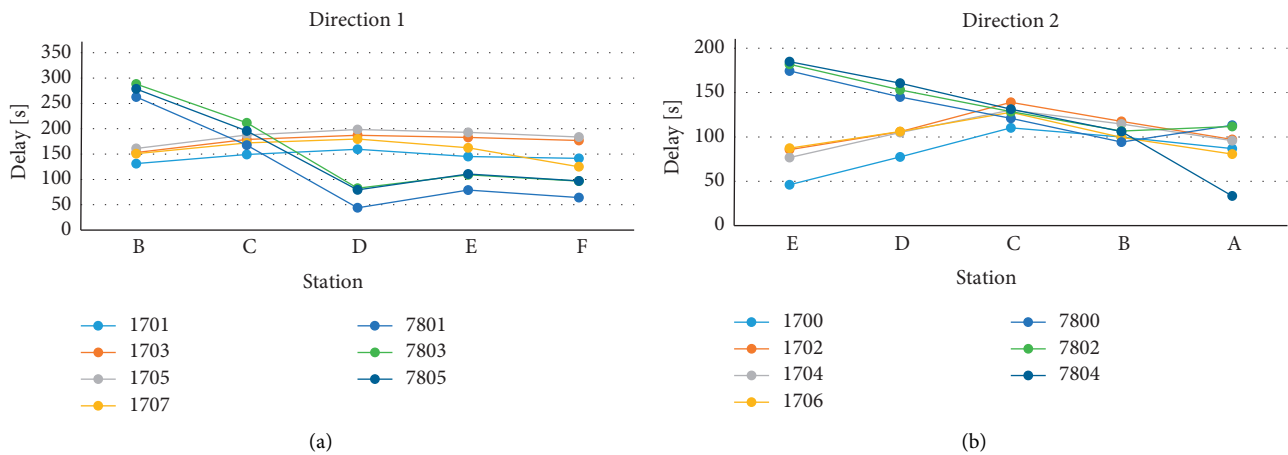


FIGURE 9: Evolution of average delay on arrival at stations in scenario Sc05.

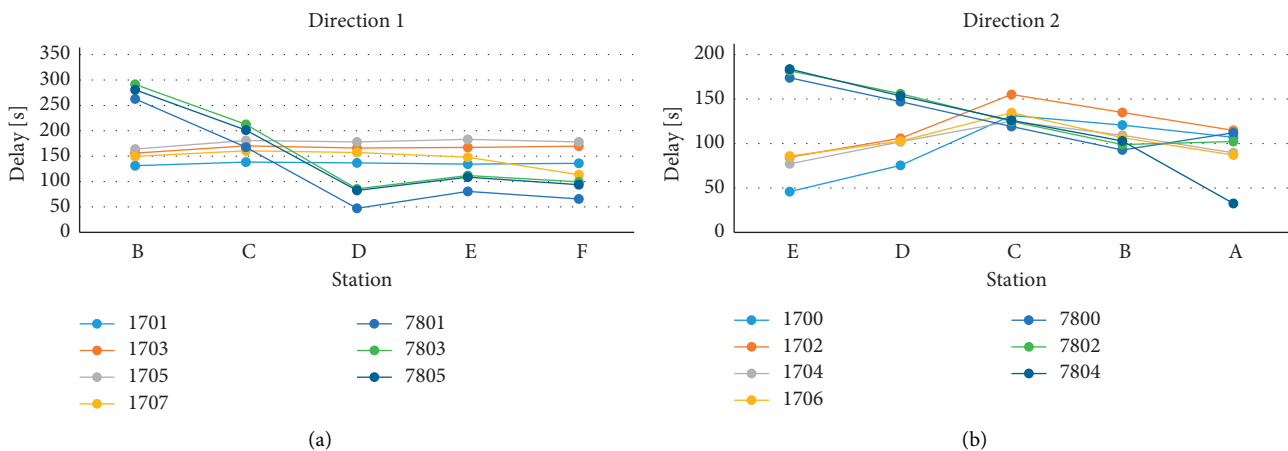


FIGURE 10: Evolution of average delay on arrival at stations in scenario Sc06.

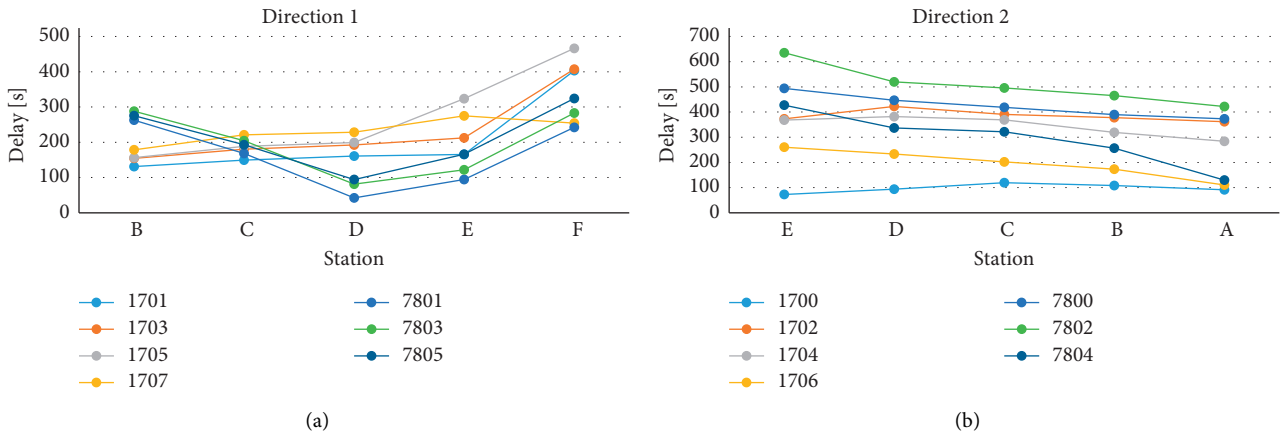


FIGURE 11: Evolution of average delay on arrivals at stations in scenario Sc07.

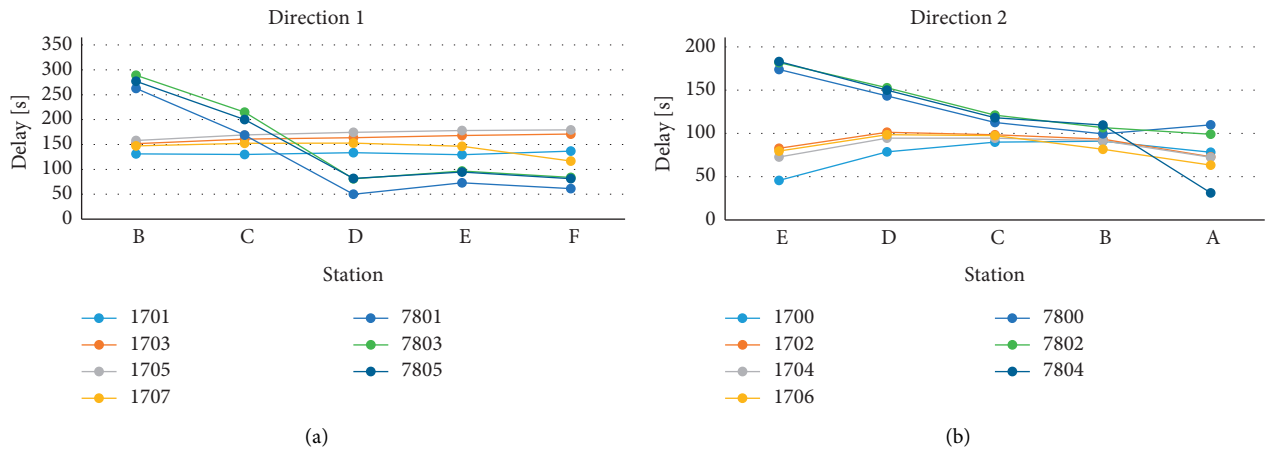


FIGURE 12: Evolution of average delay on arrivals at stations in scenario Sc08.

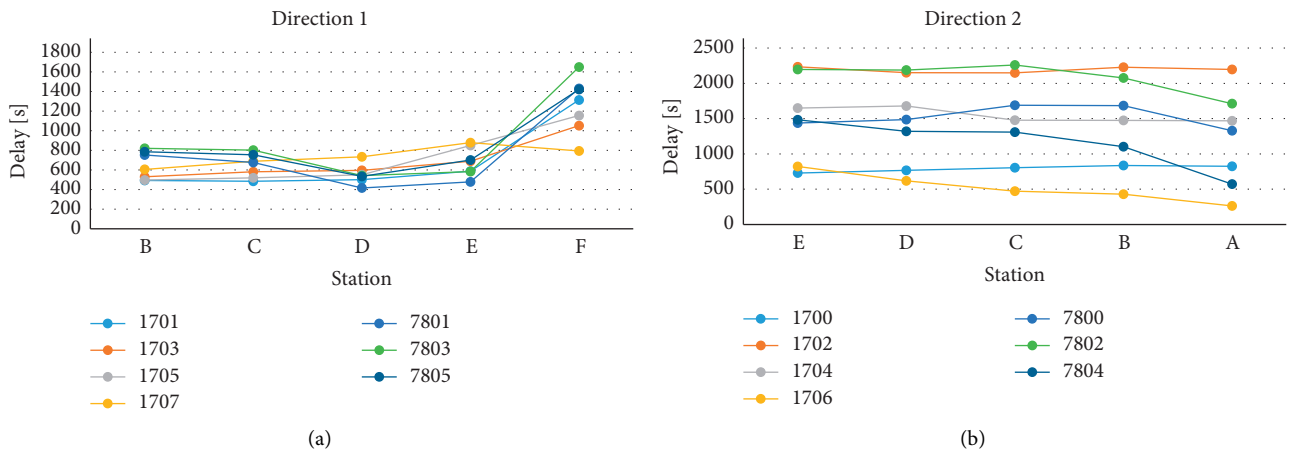


FIGURE 13: Evolution of average delay on arrival at individual stations in scenario Sc07-1.

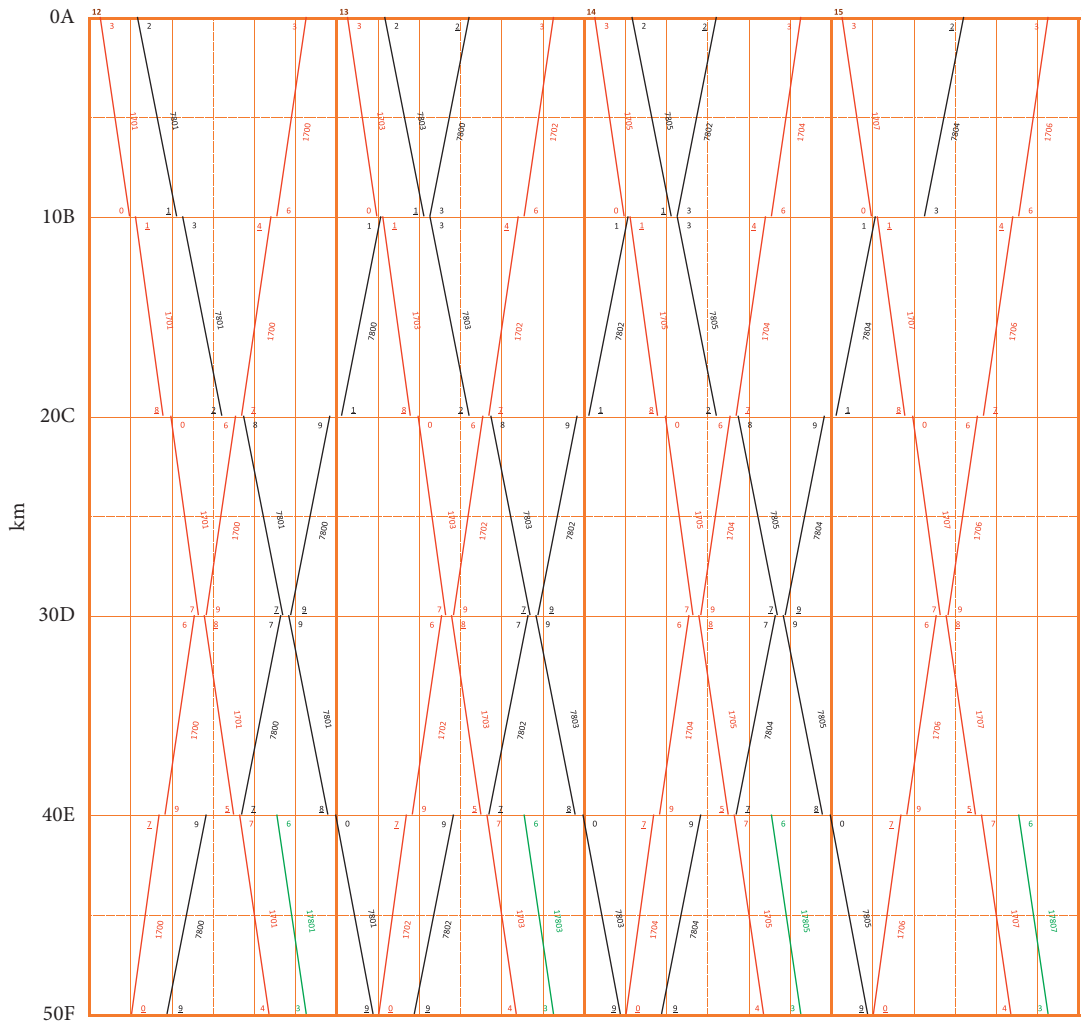


FIGURE 14: Sc01.

when crossing by moving is possible due to the structure of the timetable and due to a delay.

*7.7. Scenario Sc07-1: Assessment of the Influence of Increased Delay.* The assumptions are the same as for the scenario Sc07, including the use of the train diagram in Appendix A6. The goal of the scenario Sc07-1 is to simulate an operation with high variability in the timetable (with almost random operation)—mean value of train delay is 12 min. Average delay reached on arrival to stations is shown in Figure 13.

Automatic block in the line sections E-F supports the operation of suburban traffic, but the resulting values of delay increases due to the high volume of traffic. There is a registered decrease of delay values for regional trains in Direction 1 in sections C-D. On the other hand, delay increases for long-distance trains. Therefore, a close relation to the time positions of individual train routes is

evident. No substantial impact of automatic block application can be found in Direction 2 in the line sections C-D.

Partial conclusion: automatic block can be helpful, but significant contributions are more related to the opportunity to realize the defined operational concepts (e.g., train fleeting) than to operational reliability in general. Impact of timetable composition is more important.

### 8. Discussion

*8.1. Comparison of Scenarios.* From the mutual comparison of the scenarios Sc01, Sc02, Sc03, we can perceive certain connections with the application of the automatic block and automatic line block as basic measures enabling the presence of multiple trains running in the same direction. Not only the link to the timetable has been demonstrated, but also the fact that sometimes unattractive train sojourn times due to

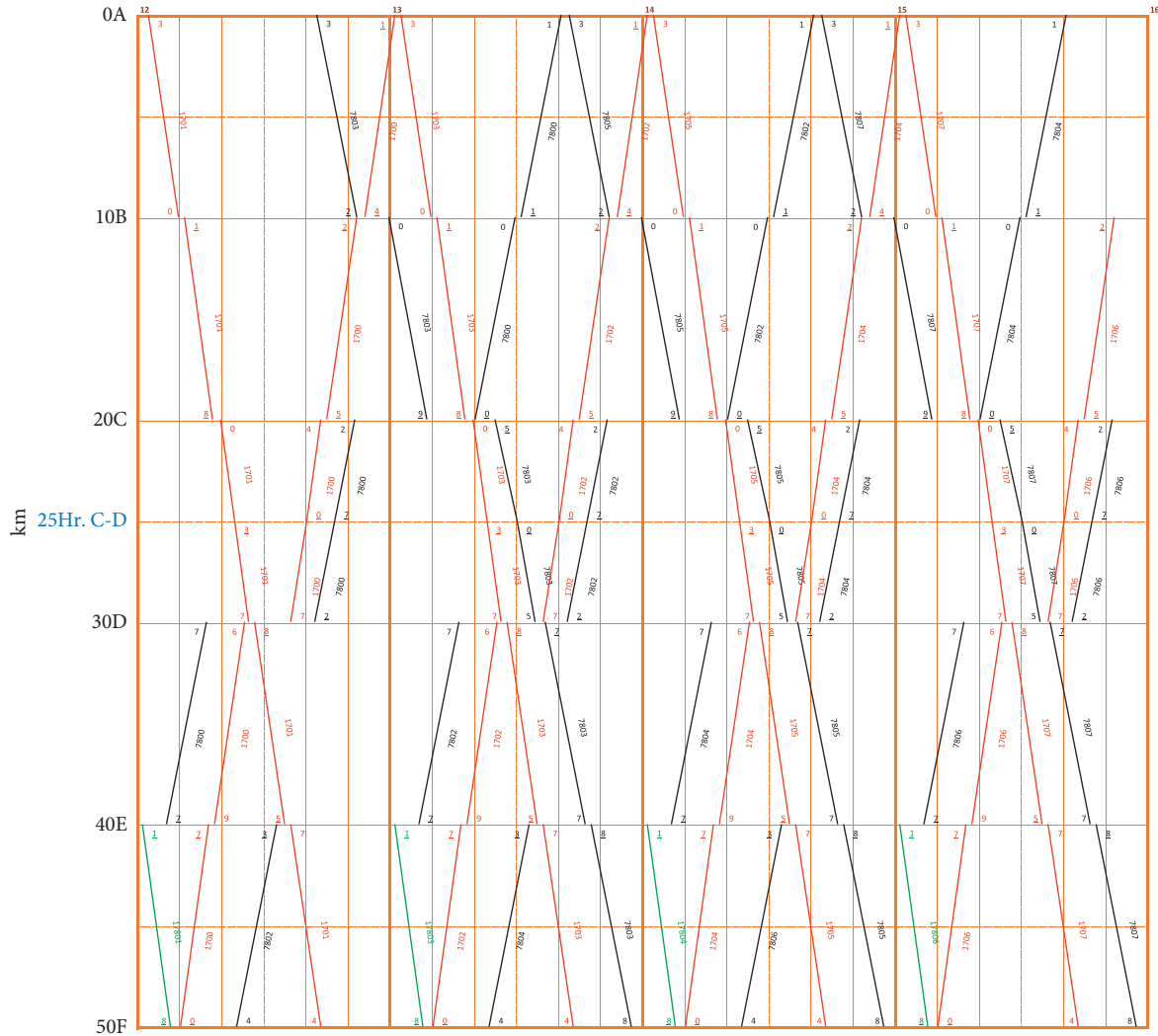


FIGURE 15: Sc02 and Sc04.

crossing can paradoxically lead to increased timetable stability.

The Sc04 scenario was focused on comparison of the automatic block and automatic line block as a measure against each other. The automatic line block leads to a reduction in the magnitude of the delay (Table 2). The delay was reduced by 10.3–34.0%, but in one case even increased more than threefold. This is because the automatic line block allows to shorten the subsequent headway. On the contrary, however, the case of increased delay again shows the particularistic nature of the solution and the link to the timetable. If a decision must be made on which type of

interlocking system allows the presence of more trains in a section should be chosen on a particular line (e.g., due to the difference in investment costs), a more in-depth assessment with a simulation model can only be recommended.

The comparison of the Sc05 and Sc06 scenarios is interesting in terms of the influence of planned fleetings, which is a prerequisite for the effectiveness of such interlocking systems, in this case an automatic line block, where alternating and fleetings modes of train passing are compared. The results show similar delay values for individual trains in Direction 1, while in Direction 2 the delay values are slightly higher in the case of fleetings. On

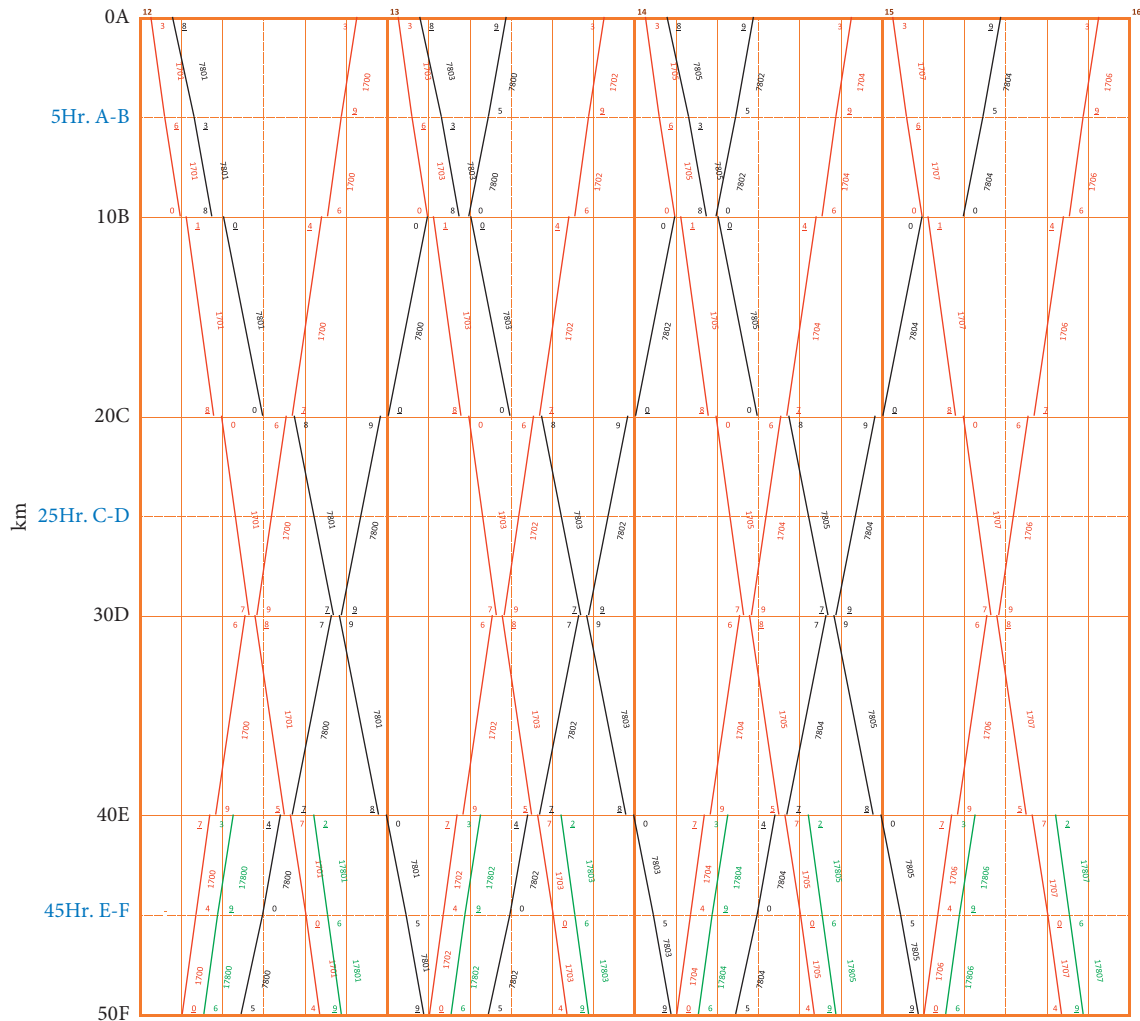


FIGURE 16: Sc03.

one hand, the increments are about 20 s per train, which does not indicate a very significant problem, but it is fully consistent with the technological interpretation of the issue that fleeted traffic can be expected to be more susceptible to delay increments. It should be noted that in both scenarios the extent of traffic in the C-D section under consideration is increased to almost full occupancy by running the newly added trains only in this section.

When comparing the Sc05 and Sc07 scenarios, where the Sc07 scenario assumes automatic line block in two sections (C-D, E-F), it was found that greater delays are experienced in the Sc07 scenario. However, this is due to the change in train sequence on departure from terminus F. Again, this points to the context of the chosen timetable and practical conditions.

The impact of unstable, but in a way irregular, traffic was monitored by comparing the Sc07 and Sc07-1 scenarios, where mean input delay values of 4 and 12 min, respectively, are applied. The achieved (output) delay values were higher, but the trend of stability was the same—in Direction 1, with a tendency of delay reduction for selected trains in the part of

line A–D and instability at station F. While the opposite direction 2 was slightly asymptotically stable in the Sc07 scenario (delay decreased throughout the line), in the Sc07-1 scenario there is also a slight increase in delay for some trains.

From a comparison of the Sc05 and Sc08 scenarios, where the Sc05 scenario assumes an automatic line block in the C-D section and the Sc08 scenario assumes the C-D section as double track and equipped with automatic block, there is no significant difference in terms of the achieved delay values.

The comparison has shown that the benefit of interlocking systems enabling the presence of multiple trains running consecutively on one section of a single-track line must be seen primarily in the context of the applied timetable. Such interlocking system is beneficial in sections where it is necessary to reduce the subsequent interval between trains. This can be recommended, for example, in the first sections after stations that are the hub of an integrated timetable so that a slower regional train can leave earlier to follow a fast train or in the case that there are running trains of two lines of passenger transport (which are



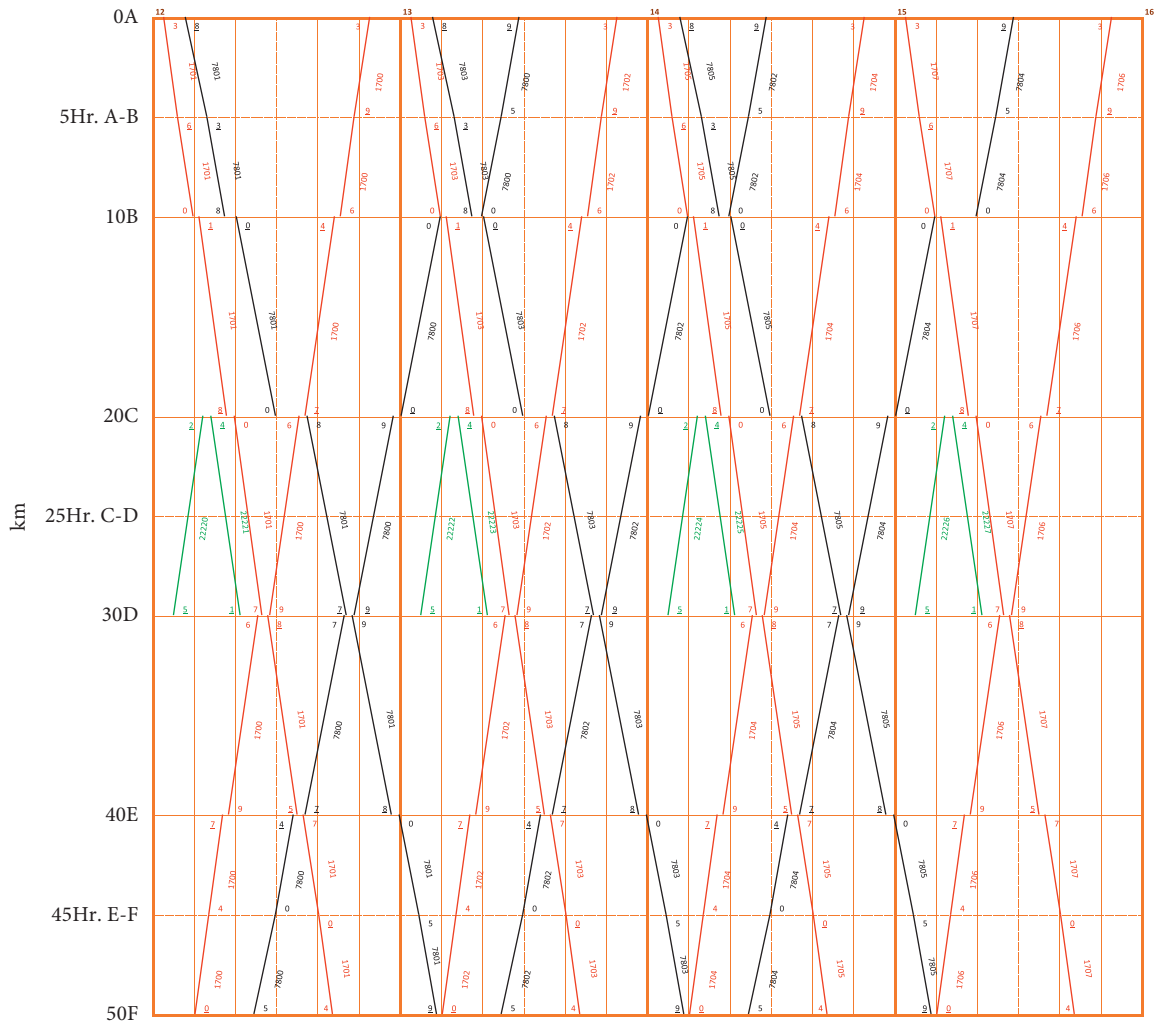


FIGURE 18: Sc06 and Sc08.

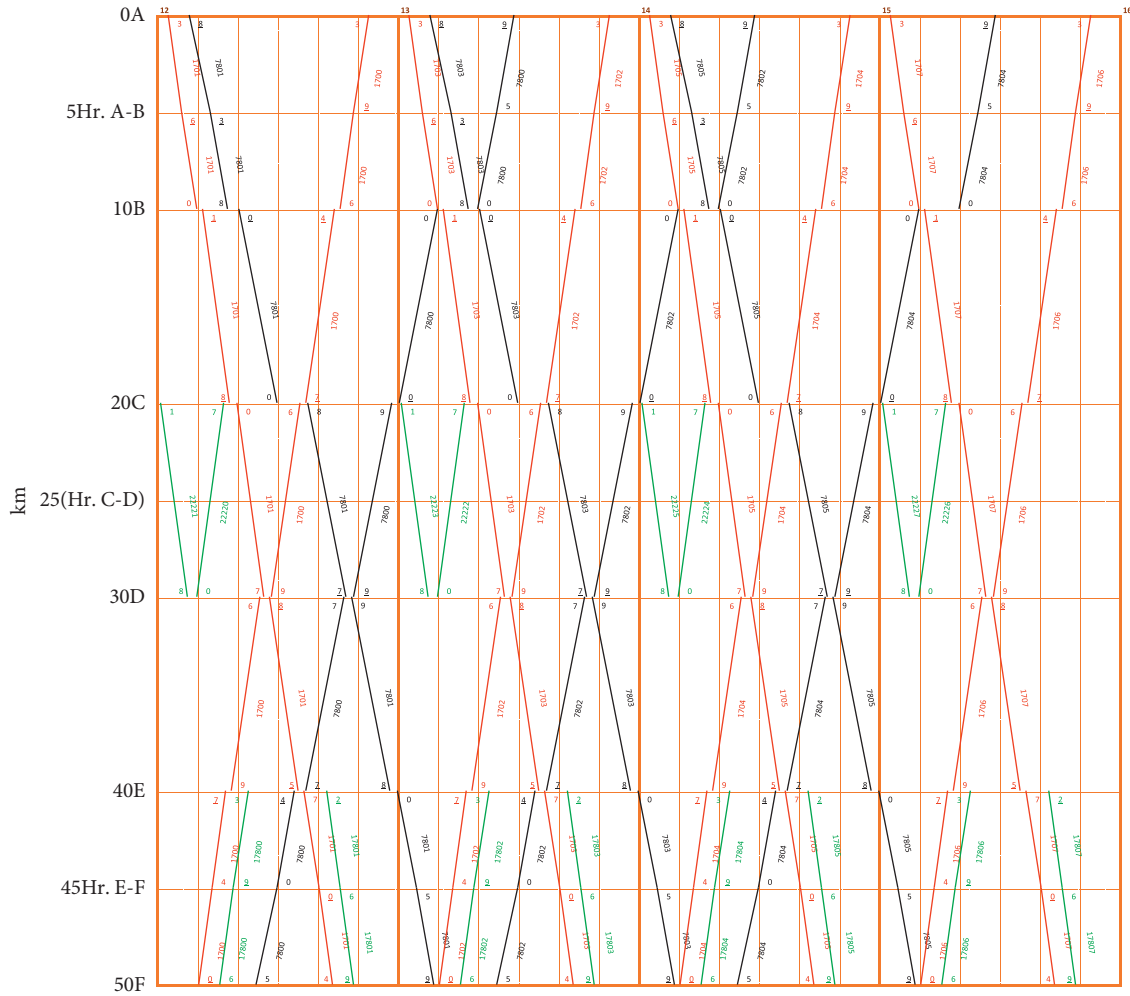


FIGURE 19: Sc07 and Sc07-1.

system) contributes to stability (quality) only for a small part. Benefit cannot be expected automatically in comparison with stochastic aspects of operation assessed with the simulation model. However, almost all partial assessments have shown that if the measure is linked to a timetable concept that would support the positive effects of the facility, its effectiveness can be increased, even if it is installed only in selected sections. The results and recommendations are that the scope of the infrastructure must be planned together with the operational concept.

Research hypothesis has not been rejected. It was recognized that stochastic simulation can also assess the possible benefits of the selected types of line interlocking system on operational quality and stability on moderately loaded lines. The performed simulation assessments resulted in some recommendations that can be technologically justified and possibly generalized. Naturally, especially in extreme cases, if the expected benefit of the device is ambiguous, it is only possible to recommend the application of a microscopic simulation model focused on the assessed line in specific (and thus more precise) conditions.

## 9. Conclusions

The research shows that there are several other possibilities and conditions in this area which could be the subject of similar assessments using stochastic simulation to create a comprehensive view of the issue. This provides the possibility of further research in this area and clarification or extension of conclusions.

The research presented in the article confirmed that the issue of capacity of railway lines in the context of quality (stability) of traffic is an interesting topic even in the case of railway lines with a medium level of traffic. The application of microscopic simulation in the OpenTrack tool can be beneficial not only for the assessment of specific railway lines, but also at the theoretical level.

Specific results of the research are mentioned above in the discussion part of the article. In general, the relation between train fleeing and the presence of interlocking system that allows the presence of multiple trains in a single line section was assessed as key aspect. As a result, a recommendation is made to consider the installation of such interlocking systems on single-track line sections where multiple passenger services are operated as a priority. It was



confirmed that train delay in simulation should be a qualitative indicator of capacity also in this specific issue, which is the subject of this article. For a comprehensive assessment, it is still recommended to build a specific simulation model containing all local specifics. On the other hand, this research forms the basis for finding ways to use single-track lines in a resilient railway transport system.

## Appendix

### Timetables for Individual Scenarios

A1: Timetable for the Scenario Sc01—initial state: trains are organized only between stations. Automatic block (I2) and automatic line block (I3) are not applied at any section (Figures 14–19).

Figure 15: Timetable for the Scenarios Sc02 and Sc04.

Figure 16: Timetable for the Scenario Sc03—automatic block (I2) applied in the border sections A-B and E-F.

Figure 17: Timetable for the Scenario Sc05.

Figure 18: Timetable for the Scenarios Sc06 and Sc08.

Figure 19: Timetable for the Scenarios Sc07 and Sc07-1—automatic block (I2) applied in the border sections A-B and E-F, even-spacing operation in section C-D (also with automatic block).

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This study was supported by the PosiTrans project: ERDF/ESF Cooperation in Applied Research between the University of Pardubice and companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans) (CZ.02.1.01/0.0/0.0/17\_049/0008394).

### References

- [1] M. Börjesson and J. Eliasson, "On the use of 'average delay' as a measure of train reliability," *Transportation Research Part A: Policy and Practice*, vol. 45, no. 3, pp. 171–184, 2011.
- [2] R. M. P. Goverde, F. Corman, and A. D'Ariano, "Railway line capacity consumption of different railway signalling systems under scheduled and disturbed conditions," *Journal of Rail Transport Planning & Management*, vol. 3, no. 3, pp. 78–94, 2013.
- [3] E. Quaglietta, "A simulation-based approach for the optimal design of signalling block layout in railway networks," *Simulation Modelling Practice and Theory*, vol. 46, pp. 4–24, 2013.
- [4] M. Rondón and F. Gomide, "Line block analysis in railway dispatch and simulation systems," *IFAC Proceedings Volumes*, vol. 33, no. 9, pp. 389–393, 2000.
- [5] H. Pouryousef, P. T. Lautala, and T. White, "Railroad capacity tools and methodologies in the U.S. and Europe," *Journal of Modern Transportation*, vol. 23, no. 1, pp. 30–42, 2015.
- [6] M. Abril, F. Barber, L. Ingolotti, M. A. Salido, P. Tormos, and A. Lova, "An assessment of railway capacity," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 5, pp. 774–806, 2008.
- [7] E. R. Petersen and A. J. Taylor, "Design of single-track rail line for high-speed trains," *Transportation Research Part A: General*, vol. 21, no. 1, pp. 47–57, 1987.
- [8] A. Nash and D. Huerlimann, "Railroad simulation using OpenTrack," *Comprail*, 2020.
- [9] Z. Chen and B. M. Han, "Simulation study based on OpenTrack on carrying capacity in district of Beijing-Shanghai high-speed railway," *Applied Mechanics and Materials*, vol. 505-506, pp. 567–570, 2014.
- [10] H. Haramina, A. Schöbel, and M. Bojić, "RačunalnomodeliranjeisimulacijazeležničkogprometanarelacijiSavskiMarof - DugoSeloprimjenomprogramskogalataOpenTrack," *Hrvatskihželjeznica*, vol. 13, no. 2, pp. 17–20, 2014.
- [11] S. Harrod, F. Cerreto, and O. A. Nielsen, "OpenTrack simulation model files and output dataset for a Copenhagen suburban railway," *Data in Brief*, vol. 25, Article ID 103952, 2019.
- [12] M. Milosavljević, S. Milinković, S. Vesković, I. Branović, and S. Aćimović, "AnalizisistemaBgvozaprimenomsimulacionogpaketaOpenTrack," *YU INFO*, vol. 20, pp. 473–478, 2014.
- [13] S. Harrod, "Capacity factors of a mixed speed railway network," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 5, pp. 830–841, 2009.
- [14] L. Mussone and R. W. Calvo, "An analytical approach to calculate the capacity of a railway system," *European Journal of Operational Research*, vol. 228, no. 1, pp. 11–23, 2013.
- [15] A. Dicembre and S. Ricci, "Railway traffic on high density urban corridors: capacity, signalling and timetable," *Journal of Rail Transport Planning & Management*, vol. 1, no. 2, pp. 59–68, 2011.
- [16] N. Stojadinović, B. Bošković, D. Trifunović, and S. Janković, "Train path congestion management: using hybrid auctions for decentralized railway capacity allocation," *Transportation Research Part A: Policy and Practice*, vol. 129, pp. 123–139, 2019.
- [17] J. Široký, P. Šrámek, K. Magdechová, E. Tischer, and P. Hlavsová, "Timetable Performance Evaluation," in *Proceedings of the 23rd International Scientific Conference. Transport Means 2019*, pp. 1427–1432, Lithuania, October 2019.
- [18] N. Weik, J. Warg, I. Johansson, M. Bohlin, and N. Nießen, "Extending UIC 406-based capacity analysis - new approaches for railway nodes and network effects," *Journal of Rail Transport Planning & Management*, vol. 15, Article ID 100199, 2020.
- [19] P. Šulko, J. Gašparik, and M. Dedik, "Line track capacity - analysis if the implementation of the UIC 406 methodology in ŽSR conditions," *Trans Motauto World*, vol. 3, no. 4, pp. 181–184, 2018.

## Research Article

# Improvement of Multiclass Classification of Pavement Objects Using Intensity and Range Images

Elham Eslami  and Hae-Bum Yun 

*Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, FL, USA*

Correspondence should be addressed to Hae-Bum Yun; [haebum@mac.com](mailto:haebum@mac.com)

Received 25 February 2022; Accepted 23 June 2022; Published 9 August 2022

Academic Editor: SeyedAli Ghahari

Copyright © 2022 Elham Eslami and Hae-Bum Yun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automated recognition of road surface objects is vital for efficient and reliable road condition assessment. Despite recent advances in developing computer vision algorithms, it is still challenging to analyze road images due to the low contrast, background noises, object diversity, and variety of lighting conditions. Motivated by the need for an improved pavement objects classification, we present Dual Attention Convolutional Neural Network (DACNN) to improve the performance of multiclass classification using intensity and range images collected with 3D laser imaging devices. DACNN fuses heterogeneous information in intensity and range images to enhance distinguishing foreground from background, as well as to improve object classification in noisy images under various illumination conditions. DACNN also leverages multiscale input images by capturing contextual information for object classification with different sizes and shapes. DACNN contains an attention mechanism that (i) considers semantic interdependencies in spatial and channel dimensions and (ii) adaptively fuses scale-specific and mode-specific features so that each feature has its own level of contribution to the final decision. As a practical engineering project, dataset are collected from road surfaces using 3D laser imaging. DACNN is compared with four deep classifiers that are widely used in transportation applications. Experiments show that DACNN consistently outperforms the baselines by 22–35% on average in terms of the F-score. A comprehensive discussion is also presented regarding computational costs and how robustly the investigated classifiers perform on each road object.

## 1. Introduction

Automation in road condition assessment is a crucial yet challenging task in smart transportation management. The goal is to label various road objects in pavement images and to establish appropriate maintenance and repair strategies to ensure road serviceability and safety. Manual road assessment, however, is labor intensive, time-consuming, and inconsistent. Automated road object detection is an alternative way for objective and scalable assessment of road networks. Fast and accurate automated road assessment can be used as quantitative data for optimal maintenance and rehabilitation practices to improve road performance and decrease the overall life-cycle cost.

To automate the road condition assessment, data are usually collected by surveying vehicles equipped with digital cameras that acquire images from pavement surfaces at high

speed. There are two main high-resolution imaging techniques frequently used in road survey projects: (i) two-dimensional (2D) imaging technology in which line-scanning cameras are used to generate 2D intensity images; (ii) three-dimensional (3D) imaging technology that provides additional range (depth) images in addition to the intensity images. Recently, the 3D imaging technology has been increasingly adopted by state and local transportation agencies for data collection of road networks [1, 2]. The 3D imaging equipment employs high-resolution laser imaging devices associated with a high-precision inertial measurement unit (IMU) to capture 3D pavement surface profile data at highway speed. One of the main advantages of the 3D technology is that it is less sensitive to light effects and less prone to noises coming from oil or water stains, dirt or sand, skid marks, etc. Furthermore, the combination of intensity and range images provides additional information to model

object boundaries and global layouts and to better recognize pavement defects.

Despite those advantages of new 3D imaging technology, existing kinds of literature [3–6] lack investigations to quantify improved performance in road object detection due to 3D technology using additional range images, compared to traditional 2D technology relying on intensity images only. Existing studies address the recognition of pavement defects, mostly cracks, using intensity images by employing deep convolutional neural networks (CNNs) [7–9]. CNNs have been successfully employed for various visual recognition tasks including image classification [10, 11], object detection [12], and semantic segmentation [13]. Although CNNs have demonstrated good performance on pavement defects recognition using intensity images, the performance tends to be degraded when detecting defects in complex scenes. The complexity comes from intensity inhomogeneity, low contrast, background noises, objects diversity in terms of shape and size, variety of lighting conditions, etc., when using intensity images only. For example, when there exists low contrast between cracks (as the foreground) and asphalt (as the background) or when dealing with thin cracks, it is difficult to distinguish between background and foreground based on only intensity data. In the case of objects with similar color and texture (such as crack seals and patches), it is easy to misclassify those objects into the same categories. Moreover, intensity-based features extracted from pavement 2D images are sensitive to illumination differences among images. The abovementioned limitations motivate the joint use of range and intensity images to enhance the classification of pavement objects. Figure 1 shows a surveying vehicle installed with a 3D laser imaging device developed by Korea Institute of Civil Engineering and Building Technology (KICT) used in this study, and a sample of intensity and range images collected by the system.

We present the novel Dual Attention Convolutional Neural Network (DACNN) to utilize additional range of input images along with intensity images to improve pavement objects classification. In this paper, DACNN classifies pavement tiles into 8 classes, including crack, crack seal, patch, pothole, marker, manhole, curbing, and asphalt. DACNN leverages multiscale input tiles that capture scale-sensitive information for multiclass classification of various road objects with different sizes and shapes. Furthermore, DACNN adopts two attention modules to effectively fuse heterogeneous features in terms of (i) scales (multiscale input tiles) and (ii) modes (range and intensity tiles). The scale and mode attention modules focus on spatial and channel-related informative features and suppress the noninformative ones for performance improvement. The dual attention mechanism is designed to identify semantic image regions relevant to specific pavement objects. Pruning feature maps in both spatial and channel dimensions enhance the quality of feature representation, contributing to more accurate and efficient object classification.

The contribution of this study is not only limited to the architectural design of DACNN. We also evaluate the effectiveness of the additional range of data in 3D technology

over 2D technology through quantitative comparison using different CNN models, including VGG16, VGG19, ResNet50, DenseNet121, as well as the DACNN. The goal of the above comparisons is (i) to understand the effects of the additional range data to improve object classification, (ii) to understand how the scale and mode attention modules can effectively fuse heterogeneous information to improve objects classification, and (iii) to understand the effects of CNN model selection to the number of trainable variables, training time, inference time, and classification accuracy. Our main contributions in this paper are summarized as follows:

We present the new DACNN framework to systematically utilize both intensity and range images collected with 3D imaging devices for multiclass classification of pavement images. Considering the variety of pavement objects and surveying field conditions, DACNN extracts scale-specific and mode-specific features from images robustly. The dual attention mechanism used in DACNN is designed to adaptively fuse multiscale multimodal features, helping the network to capture discriminative object-specific features related to their spatial and channel information.

The classification performance comparison is conducted for 8 different pavement objects using CNN models. The results show that our DACNN outperforms other models for all road object classes. We also present quantitative comparisons to understand how the additional range of images in 3D technology can improve object classification performance for compared CNN models.

## 2. Related Works

*2.1. Deep Learning in Pavement Assessment.* Conventional image processing and more recent deep learning methods are two main approaches for automated pavement image analysis. The image processing methods can be considered as feature engineering techniques in which images are represented with human-specified feature vectors. They can be sorted into intensity-thresholding [14], edge detection [15], wavelet transforms [16, 17], and texture-analysis [18, 19]. A major problem with the conventional methods is that the prediction performance mainly relies on the validity of human-specified features. Extracting those features can be subjective, domain-specific, and inefficient, which makes the detection process ungeneralizable and tedious. Especially in pavement applications, hand-crafted features are not robust enough to detect distresses in the complex background with high variations. For instance, thresholding approaches for crack detection only achieve acceptable results under certain scenarios. If there exists a complex background or the illumination changes, either the parameters should be adjusted or the method is not applicable to the new scene.

Deep learning methods overcome the drawbacks of conventional image processing methods by automatically capturing complex structures of data with multiple



FIGURE 1: 3D laser imaging system developed by Korea Institute of Civil Engineering and Building Technology (KICT); sample of high-resolution intensity and range road surface images.

processing layers. CNNs are the most studied deep learning models using vision-based input data in which automated feature learning is done at many different levels of abstraction to catch the topology of input images. Partial connections, sharing weights, and pooling layers in CNNs not only decrease the computations but also demonstrate state-of-the-art results in computer vision tasks [20, 21]. Detection, classification, and segmentation of pavement distress, especially cracks, are the main three branches of deep learning research in automated pavement assessment. Alfarrarjeh et al. [22] employed YOLO [23] as the object detection method to detect distresses, including cracks, potholes, and rutting, in pavement images. Maeda et al. [24] adopted SSD [25] as the training algorithm to detect the same defects on pavement surfaces. Song et al. [26] utilized Faster R-CNN [27] algorithm to detect pavement distresses, including cracks, potholes, and bleeding. Li et al. [28] presented a CNN model to classify pavement tiles into different types of cracks including longitudinal, transverse, alligator, and block cracks. Gopalakrishnan et al. [29] utilized a pre-trained VGG16 [30] on ImageNet and then fine-tuned it on a pavement dataset for a binary crack classification. Lau et al. [31] proposed a U-Net [32] based model in which the encoder is a pretrained ResNet34 [33] to segment pavement crack images. Inspired by SegNet [34], Chen et al. [35] proposed a fully convolutional neural network (FCNN) to detect pavement cracks at pixel level.

**2.2. Attention in Deep Learning.** The performance of deep learning-based approaches has been constantly improving by developing new architectural designs, and the attention mechanism is one of them. The main idea behind an attention mechanism is to give higher weights to relevant features while minimizing the irrelevant ones by giving lower weights. Focusing on the distinctive parts when processing large amounts of information, the attention

mechanism enhances the quality of feature representation, contributing to a more accurate and efficient performance of the designed network. Attention was initially proposed by [36] for machine translation. Then, it was employed for various tasks, such as action recognition [37–39], speech recognition [40, 41], image captioning [42, 43], and recommendation [44, 45]. More specifically, the attention mechanism is investigated in computer vision community in three aspects: (i) spatial attention in which the network learns the locations that should be focused on [46, 47]; (ii) channel attention in which the network adaptively recalibrates channel-wise features by modeling interdependencies between channels [48, 49]; and (iii) Self-attention in which long-range dependencies are captured by the network [50, 51]. In pavement applications, attention modules have been also applied for defect detection. Song et al. [52] presented a channel of attention to detect and classify different types of cracks in pavement images. Wan et al. [53] proposed an encoder-decoder network, called CrackResAttentionNet, containing spatial and channel attention modules after each block in the encoder to segment pavement cracks. Similarly, Qiao et al. [54] proposed CrackDFANet in which a channel-spatial attention module is designed to increase the generalization ability of the model in predicting cracks under different conditions of roads. Wang et al. [55] proposed using DenseNet121 as an encoder and a spatial attention module to combine multiscale features. Eslami et al. [56] designed a channel-spatial attention module to adaptively fuse multiscale features for pavement image classification. Zhou et al. [57] presented a VGG16-based network to predict crack maps, and employed spatial and channel attention modules to further refine the model. Qu et al. [58] employed Res2Net [59] along with an attention module to capture global context and long-range dependency for a better pavement segmentation. Pan et al. [60] proposed SCHNet with VGG19 as the base net in which a self-attention module is designed to global as well

as semantic interdependencies in the channel and spatial dimensions. Finally, Li et al. [61] proposed a self-attention module along with a scale-attention module to enhance feature representation for pavement crack segmentation.

In this study, we propose a dual attention approach to capture semantic interdependencies in both spatial and channel dimensions for scale and type of input images. The dual attention mechanism achieves a fast focus on more important features and enhances the representativity of more relevant features for better classification performance. The dual attention approach enables modeling global context as well as multimodal features to improve classification performance for both small objects (e.g., cracks) and large objects (e.g., patches), which are in trade-off using other CNN models.

**2.3. 3D Image Data in Pavement Assessment.** Most of the existing deep learning studies were based on only intensity images using 2D imaging devices in transportation applications. With 2D intensity input images, CNNs suffer from some important limitations. The complexity of scenes, diversity of objects, background noises (stains, oil spills, and tire marks), and surrounding changes (light and shadow) make it difficult to distinguish foreground objects (defects) from the background (asphalt) in 2D images. With the advances in sensor technology, 3D imaging systems are available and increasingly employed by state and local transportation agencies for automated road condition assessment. A survey showed that 18 states in the U.S. adopted a 3D data collection system by 2017, and 17 states intended to utilize this technology by 2019 [1]. Different approaches have been studied for transportation applications such as GPR, LiDAR, Microsoft Kinect, and laser profilers [3]. In pavement applications, laser profilers are commonly used in surveying road roughness and megatexture (ASTM E950, ASTM E1926, and ISO 13473–5) [62–64]. Other techniques offer limitations such as relatively low resolution (in case of LiDAR) or low frequency (in case of Microsoft Kinect) to collect road surface profiles. The 3D laser imaging technique, such as Laser Crack Measurement System (LCMS) [65], is commercially available to collect high-resolution road surface profiles. This system utilizes surveying vehicles equipped with two laser imaging devices (left and right) and IMU. Using the 3D imaging system, intensity and range images can be acquired at speeds up to 100 km/h on on-road lanes with 4 m width under various lighting conditions. The 3D laser imaging technology has been used to evaluate crack [66, 67], pothole [68], raveling [69], rutting [70], joint [71], and texture [72]. Ghosh et al. [73] employed YOLO and Faster R-CNN to detect cracks in range images collected by the 3D imaging system. Yang et al. [74] utilized 3D laser technology to measure the growth of crack lengths when they are sealed and non-sealed to quantify the crack sealing benefit. Li et al. [28] proposed a CNN framework to classify range images into transverse cracks, longitudinal cracks, block cracks, and alligator cracks. Lang et al. [67] proposed a clustering-based algorithm to classify range images into the same categories of cracks as Li et al. [28]. Fei et al. [75]

presented a deep CNN, called CrackNet-V, to segment cracks on asphalt range images. Li et al. [76] applied a filter-based method to segment cracks using 3D pavement images. Zhang et al. [77] proposed a recurrent neural network (RNN), called CrackNet-R, to detect pavement cracks at pixel-level in range images. Gui et al. [78] utilized laser-scanning 3D to detect pavement cracks by extracting hand-crafted features. Tsai and Chatterjee [68] proposed a threshold-based method to detect pavement potholes in range images collected by 3D laser technology. Zhang et al. [79] proposed a CNN-based architecture, called CrackNet to segment cracks in 3D pavement images. Zhang et al. [80] improved the crack segmentation results on 3D pavement images by proposing a deeper network, CrackNetII, in which the need for hand-crafted features is eliminated. Li et al. [81] presented a frequency analysis to detect pavement cracks from background texture in range images.

While there are existing studies using 3D laser imaging technology, they are limited to the use of either range or intensity images. In this study, we show that extracting features from both intensity and range (depth) images can significantly improve the CNN performance. We also show that by fusing intensity-specific and depth-specific features systematically, one can robustly and accurately classify not only cracks but also other pavement objects, including crack seals, patches, potholes, markers, manholes, and curbing in multiclass classification.

### 3. Data Preparation

**3.1. Ground-Truth Labeling.** The dataset used in this study contains 296 intensity images and the same number of range images with the size of  $3700 \times 10000$  pixels spatial resolution of 1 mm/pixel. The gray-scale intensity and range images are collected by the 3D laser imaging device developed by Korea Institute of Civil Engineering and Building Technology (KICT) shown in Figure 1. The technical specifications of this device are provided in Table 1.

We provide pixel-level annotations of road objects for 8 categories, including 4 distress classes (crack, crack seal, patch, and pothole), 3 non distress classes (marker, manhole, and curbing), and 1 pavement class (asphalt) as the background. We annotate the intensity images using an in-house developed semiautomated software that makes the annotation process fast yet accurate. The annotation procedure is performed in two steps: (i) labeling area objects (all classes except for cracks) and (ii) labeling linear objects (i.e., cracks). To label area objects, the original image, shown in Figure 2(a), is grouped into homogeneous regions, called superpixels [82, 83]. As shown in Figure 2(b), superpixel segmentation preserves the edges and boundaries of objects. Therefore, superpixel-level labeling, rather than pixel-level labeling, can be performed, which reduces the labeling work significantly. To further facilitate the annotation process, an unsupervised mean shift clustering is applied, which groups the neighboring superpixels into a bigger cluster. The result of the superpixel clustering procedure is shown in Figure 2(c). Then, the human annotator can easily select the clusters that belong to the same object and label them. Also,

TABLE 1: Technical specifications of KICT 3D laser imaging device.

Scanning frequency	Transverse range	Lateral resolution (mm)	Vertical resolution (mm)	Data rate
5600 profiles per second	4 m (4096 points per profile)	1	0.5	10.4 Gb/km (720 Mb/km compressed)

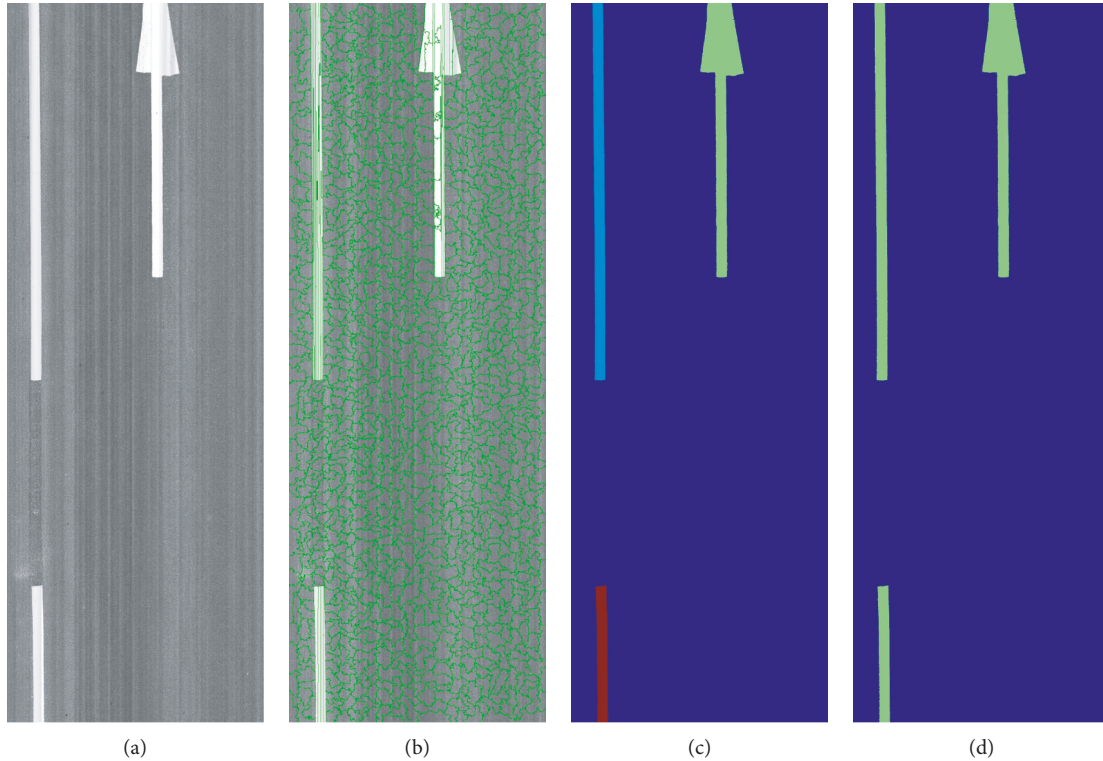


FIGURE 2: Annotation procedures for areal objects. (a) Original image; (b) superpixel segmentation; (c) unsupervised mean shift clustering; (d) human correction of false clustering and classification.

the annotator is able to define new segments, which are missed by the clustering algorithm. Figure 2(d) demonstrates the final pixel-level labeling mask. Although the superpixel segmentation technique is beneficial for labeling area objects in the dataset, it is not effective for linear object labeling such as cracks. To label cracks, a morphological technique, called MorphLink-C, is employed to extract crack pixels in original images. MorphLink-C consists of a series of morphological operations, which is proposed by Wu et al. [84]. The original image in Figure 3(a) is a zoomed-in pavement image for better visualization of the existing crack. The cracks detected by MorphLink are shown in Figure 3(b) with the bounding boxes. Having the detected cracks, the human annotator can select the truly detected cracks within the image, as shown in Figure 3(c).

Figure 4 demonstrates the contents of different objects in the dataset. We observe that the population of road object pixels are highly imbalanced, for example, there are more than three million of asphalt pixels but only more than 4000 crack seal pixels in the dataset. Detecting objects with high variations in shape and size within a highly imbalanced dataset is a major challenge in pavement applications.

**3.2. Data Preprocessing.** In road surveying projects, the depth information in range images is often used to measure the macrostructure of pavement surface (ISO 13473-1) [64]. Although the depth resolution of the laser device on an absolute millimeter scale is important to determine the mean profile depth (MPD) in macrotexture surveying, a small variation in surface profile (e.g., crack depth) and low contrast in range images could be a disadvantage in road objects detection. To enhance the contrast, a histogram equalization (HE) can be applied to range images. HE enhances the contrast by effectively spreading out the most frequent intensity values (stretching out the intensity range of the image). It allows for areas with lower local contrast to obtain a higher contrast. In this study, Contrast Limited Adaptive Histogram Equalization (CLAHE) [85] is applied to a range of images. CLAHE differs from ordinary HE algorithms in two ways: (i) An adaptive HE computes several histograms, each corresponding to a small region of the image rather than computing the histogram for the entire image. Therefore, it improves the local contrast and edges in each region of the image. (ii) CLAHE sets a threshold to limit the contrast in each small region. The contrast limiting

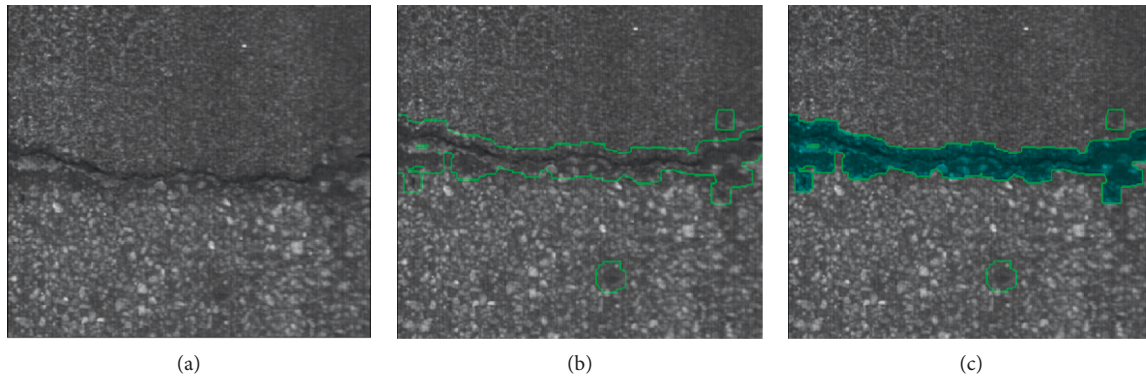


FIGURE 3: Annotation procedures for linear objects. (a) Original image; (b) automatic crack detection by MorphLink technique; (c) human selection of truly detected cracks.

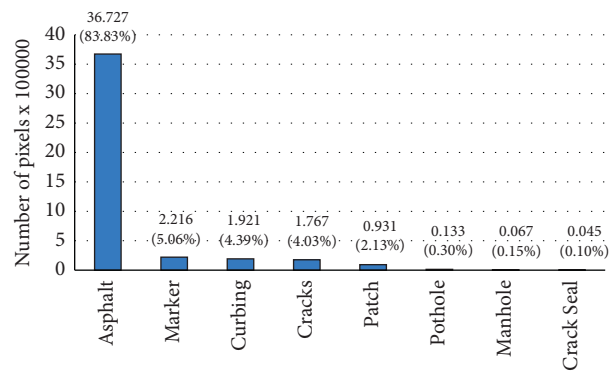


FIGURE 4: Number of pixels in our pavement classes.

procedure prevents the over-enhancement and amplification of noise in the image. Figure 5(a) shows a range image with cracks spreading all over the image. Also, the intensity distribution of the image and the cumulative distribution are presented for the range image as histogram and cdf, respectively. Figure 5(b) demonstrates the range image after using CLAHE enhancement and its corresponding histogram and cdf. We can see that the visibility of cracks is improved by redistributing the lightness values of the image without introducing noises to the image. Comparing the histograms before and after applying CLAHE to the image, the intensity range of the road image is expanded within the lower range (dark pixels 0–50) by redistribution of the values, as shown in Figure 5(b).

After the contrast enhancement of range images, we divide the original images into nonoverlapping  $50 \times 50$  tiles to conduct multiclass classification experiments on pavement images. Then, each image tile is assigned to one of 8 categories of road objects. When a  $50 \times 50$  tile has more than one class of pixels, the tile class is determined by a majority vote between the pixel number of nonbackground classes if exists, otherwise, the tile is classified as the background (asphalt). By aggregating the assigned classes for all tiles generated from an original image, a segmentation mask with a resolution of  $50 \times 50 \text{ mm}^2$  can be produced. The reason for  $50 \times 50$  tile generation comes from two sources: (i) Due to the large size of the original images ( $3700 \times 10000$ ), the

segmentation task on the whole image is memory intensive and not practical; (ii)  $50 \times 50$ -pixel tiles, equivalent to  $50 \times 50 \text{ mm}^2$ , is small enough to contain only one pavement object for the classification task. Therefore, assembling the classification results into the whole image produces a segmentation mask with a high-resolution, which is satisfactory in pavement applications. Although having small input tiles results in high-resolution segmentation masks, it sacrifices the contextual information required from the deep networks to perform well. Due to the importance of contextual information for the classification task, we generate  $250 \times 250$ , and  $500 \times 500$  tiles surrounding each  $50 \times 50$  tile with the same center. Feeding multiscale tiles into the deep networks improves the classification performance of the smallest tile, which will be explained in Section 4.1.

## 4. Method

*4.1. Dual Attention Convolutional Neural Network Architecture.* The Dual Attention Convolutional Neural Network (DACNN), illustrated in Figure 6, is presented to classify pavement image tiles into one of the 8 existing classes in the dataset. The DACNN provides a systematic way of data fusion for heterogeneous input images including (i) intensity and range images (i.e., mode), and (ii)  $50 \times 50$ ,  $250 \times 250$ , and  $500 \times 500$  (i.e., scale), which is more effective than a simple feature concatenation. For this, the DACNN

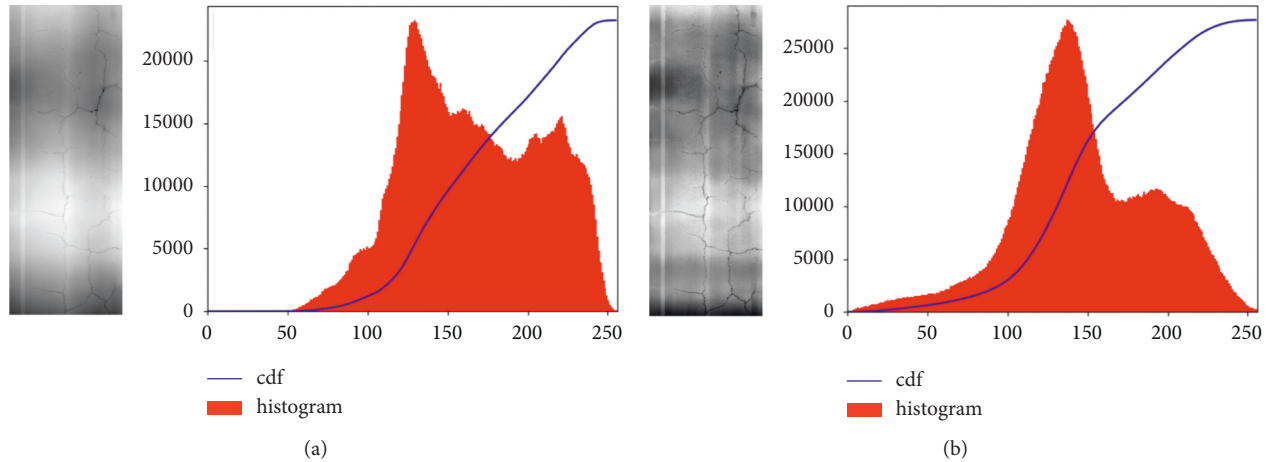


FIGURE 5: (a) Original range image and (b) CLAHE enhanced range image with corresponding histograms and cumulative histograms.

consists of two main streams of intensity and range modes, which are merged later by a mid-fusion strategy (i.e., mode-level attention module). Each mode stream consists of three scale streams to extract multiscale features, which are combined later using a mid-fusion strategy (i.e., scale-level attention module). The high-level architecture of the DACNN is shown in Figure 6.

*Multiscale Input Tiles.* Input tiles are extracted from the original intensity and range images at three scales,  $50 \times 50$ ,  $250 \times 250$ , and  $500 \times 500$ . All the input tiles are resized to  $50 \times 50$  before they are fed to the DACNN.

*Feature Extraction (Scale).* A conventional to combine multiscale multimodal input data is directly concatenating them at the input level. This approach has a disadvantage in that only similar patterns will be captured across the scales and modes. Instead of concatenating heterogeneous input data in an early fusion, we propose to feed input tiles to 6 separate CNNs to extract scale-specific and mode-specific features. Each CNN consists of three convolution layers with the filter numbers 32, 32, and 64, respectively. The filter size is  $3 \times 3$  pixels for all convolution layers. Each convolution layer is then followed by a Batch Normalization layer and a rectified linear unit activation (ReLU), which are not shown in Figure 6 because of space limitation. It should be noted that up to this point the extracted feature maps are processed independently at each scale and mode level.

*Mid-Fusion with Scale-Level Attention Module.* The main idea of using multiscale input tiles is to allow features extracted from different levels of spatial context around the smallest tile ( $50 \times 50$ ) to contribute to the classifying decision. The level of contribution at each scale for different objects varies for different objects. For example, scale 1 is more informative for small objects (e.g., cracks), while scale 3 is more informative for classifying large objects (e.g., patches). Therefore, we use a scale-level attention module that decides how much attention to pay to scale-sensitive features. Unlike simple concatenation of multiscale features,

the scale-level attention module weights the features from different input scales at each mode. The scale-level attention module consists of three convolution layers of  $1 \times 1 \times 64$ , and one sigmoid layer to generate the weight scores for each scale. The generated score maps reflect the importance of scale-specific features at a specific position and scale for classifying the object in the tile.

*Feature Extraction (Mode).* After the mid-fusion with the scale-level attention module, the weighted feature maps get concatenated in intensity and range modes, separately. Then, they are passed through three convolution layers with the filter number of 128 and max-pooling layers. At this stage, the network is expected to extract more complex multiscale features in each mode. Depth-specific patterns can complement intensity patterns and help the overall model with this useful information.

*Mid-Fusion with Mode-Level Attention Module.* For the effective mid-fusion of complementary information of intensity and range data, we use a mode-level attention module that weights the mode-sensitive features extracted from intensity and range images, determining the contribution level of mode-sensitive features to the final classification output. In this way, the feature maps can be fused with different weights based on the contribution levels of road object classes, instead of being treated uniformly.

*Feature Extraction (Classification).* For each mode, the mode-level attention module outputs weight maps that are multiplied by the feature maps. The weighted feature maps get concatenated and passed through shared layers. Four convolution layers with the filter size of 256, 512, 512, and 1024 with two max-pooling layers are applied to extract higher-level multimodal features. Then the feature maps are flattened and passed to six fully-connected layers with the sizes 2048, 1024, 512, 256, 128, and 8.

*Classifier's Output.* The last fully-connected layer generates 8 numbers showing the probability of the  $50 \times 50$  tile belonging to the 8 existing classes in the dataset. The higher the number is, the more probable the tile belongs to that specific pavement class. By assembling the predicted labels for the smallest tiles into the whole image,



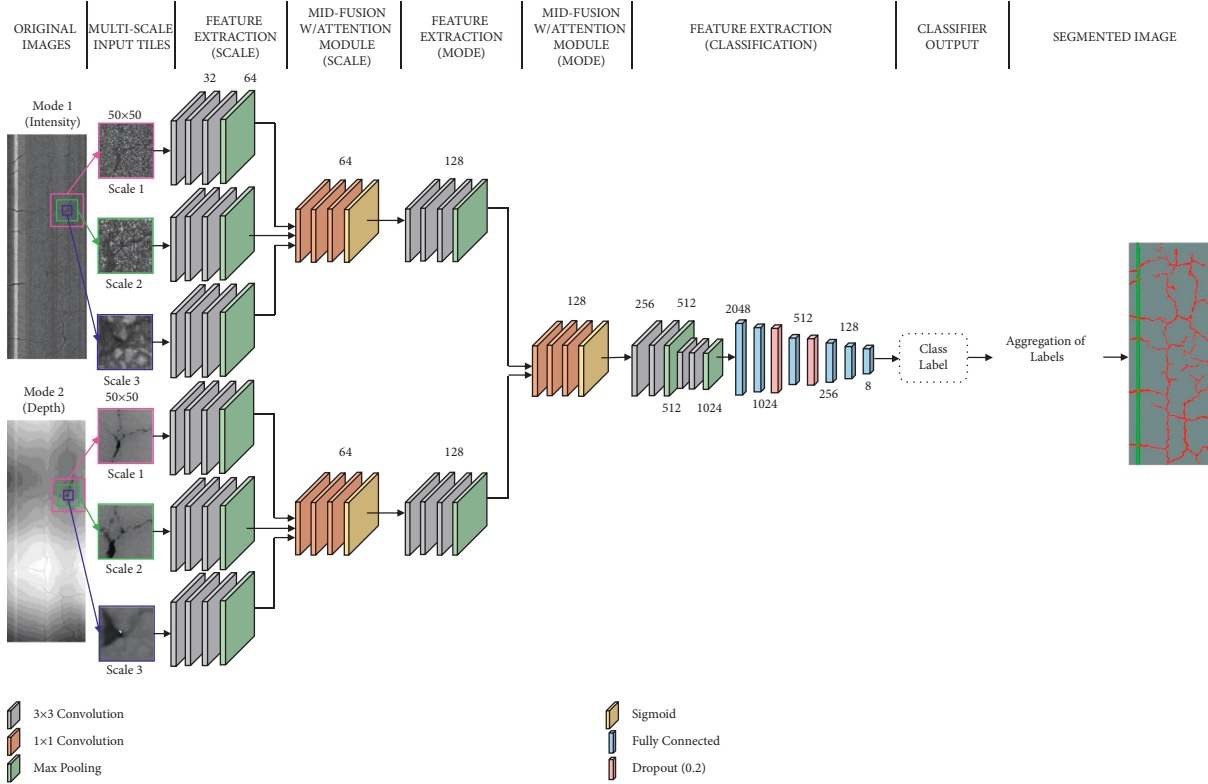


FIGURE 6: An overview of the DACNN. Range and intensity image tiles are generated at three scales to capture local and global information in each mode. The adaptive fusion of multiscale multimodal features is performed through scale-level and mode-level attention modules. The final class prediction for input tiles is assembled into the original image to create a mask.

the segmentation mask with the spatial resolution of  $50 \times 50 \text{ mm}^2$  is created.

*Effects of Range and Intensity Input Image Tiles.* Range and intensity input images provide complementary information about road objects, which can improve object classification performance compared to intensity-only input images. Depth is a key feature for road object classification, such as cracks and potholes. These objects can be small or have a similar color and texture to the clean asphalt, and it makes them difficult to detect in gray-scale intensity images. However, they appear more clearly in range images due to their depth differences. Other pavement objects, such as markers, that have a distinct color or texture or do not have a significant depth can be easier to detect from intensity images. Figure 7 demonstrates the advantage of using intensity and range images over intensity images only containing markers, patches, and cracks.

*4.2. Attention Modules.* We design two types of attention modules as a mid-fusion strategy to adaptively aggregate multiscale multimodal features extracted from intensity and range image tiles. The mechanism of an attention module is to attend to relevant parts of input features, which is important for having a robust classification. The scale-level and mode-level attention modules enable the deep network to focus on visual representations that are more informative for the classification of the object in the input tile. Scale-level

and mode-level modules incorporate both spatial and channel-wise attention into the network.

As illustrated in Figure 8(a), the scale-level attention module generates the score maps ( $\mathbf{S}^m$ ) with the dimension of  $C \times H \times W$  for each scale, where  $m \in \{1, 2, 3\}$  is the scale number,  $C$  is the number of channels,  $W$  is the width, and  $H$  is the height of the input features ( $\mathbf{F}^m$ ). The weighted feature maps,  $\mathbf{F}^m$ , are generated by the inner product of:

$$\mathbf{F}^m = \mathbf{F}^m \cdot \mathbf{S}^m, \quad (1)$$

or

$$f_{w,h,c}^m = s_{w,h,c}^m \cdot f_{w,h,c}^m, \quad (2)$$

where  $\tilde{f}_{w,h,c}^m$  is the weighted feature at the spatial position  $(w, h)$  for the channel number  $c$  at the scale  $m$ ; and  $s_{w,h,c}$  is the score corresponding to the input feature  $f_{w,h,c}^m$  at the spatial position  $(w, h)$  for the channel number  $c$  at the scale  $m$ . The attention module assigns a score between 0 and 1 to the feature maps of each scale in each channel and spatial position. Therefore, each element in the feature map  $x_{w,h,c}$  is revised to  $x_{w,h,c}$ , in which scale, channel, and spatial information is considered. This module not only localizes the object spatially but also selects the most discriminative channel.

The mechanism of the mode-level attention module, shown in Figure 8(b), is similar to the scale-level one. In this module, the shared module among the modes generates the score maps for each mode to focus on the most

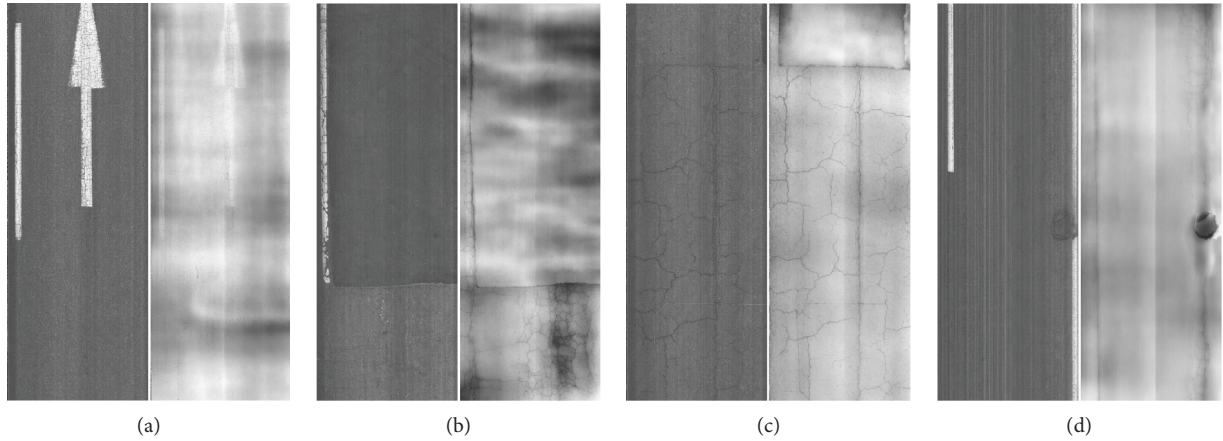


FIGURE 7: Illustration of pavement objects in intensity and range images: (a) markers, (b) marker, patch, and cracks, (c) patch and cracks, and (d) marker and pothole.

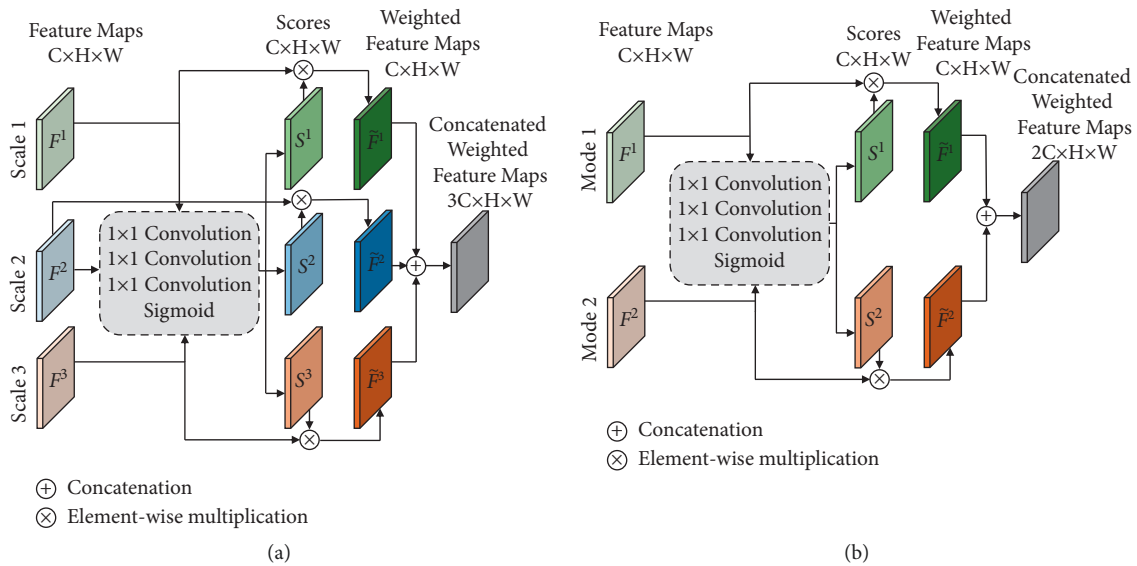


FIGURE 8: The details of (a) scale-level attention module and (b) mode-level attention module.

discriminative part of visual representations. The attention module assigns higher weights to the channel and regions of the mode features that are more relevant and informative for the classification step of that particular object.

**4.3. Implementation Details.** We train the classifiers in a fully supervised manner. The Adam optimizer with a learning rate of  $\alpha = 0.0001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1e-8$  is used, where  $\beta_1$  and  $\beta_2$  are exponential decay rates, and  $\epsilon$  is a constant for numerical stability. The Adam optimizer inherits the advantages of other optimization algorithms, including the momentum feature of SGD and the adaptive learning feature of AdaDelta. The Adam optimizer also provides faster computation time and requires fewer parameters for tuning. The networks are trained for 800 epochs with a mini-batch size of 200. In each epoch, the network uses 60,000 random tiles out of more than 6 million tiles in the training dataset.

The model with the best performance on loss for the validation dataset is selected as the model used in the testing mode. The training is conducted on an NVIDIA TitanX GPU with a memory configuration of 12 GB. The codes are implemented in Python 3.7.3 and TensorFlow 1.14.0.

## 5. Experiments

**5.1. Baseline Models with Single-Scale Input Images and Results.** Four different baseline classifiers, widely used in pavement applications, are trained to classify pavement image tiles into one of the existing 8 classes in the dataset. The deep CNNs compared in this study can be divided into three categories. (i) VGGNet was proposed by Simonyan and Zisserman [30] for ImageNet challenge 2014. The main idea behind VGGNet is to use filters with a small size ( $3 \times 3$ ), decreasing the number of parameters, and stack more of them to achieve the same receptive field as if a larger filter

were used. VGG16 and VGG19 have a total number of 16 and 19 convolutional and fully-connected layers, respectively. The deep architecture of VGGs is proved beneficial for image classification tasks. However, the gradient vanishing problem has appeared with the deeper architectures. (ii) ResNet proposed by He et al. [33] for ImageNet challenge 2015, alleviates the gradient vanishing problem by introducing skip-connections so that the input in each layer is passed to the next layer. Using identity skip-connections as well as batch normalization allows for training deep networks. ResNet50 has a total number of 50 convolutional and fully-connected layers. (iii) DenseNet proposed by Huang et al. [86] in 2017, extends ResNet's idea by including skip-connections from all previous layers. The dense concatenation to all subsequent layers preserves the features in preceding layers and allows for the classification of images in a wide range of scales. DenseNet121 has a total of 121 convolutional and is fully connected.

Figure 9 shows an overview of the deep networks used for pavement object classification in this study. The classifiers are trained with only intensity input tiles as well as intensity and range input tiles to evaluate the effect of exploiting depth information along with intensity information. As shown in Figure 9(a),  $50 \times 50$  image tiles are generated and are concatenated as a 3-channel image to train the deep networks with only intensity images. When training the networks with both intensity and range images, as shown in Figure 9(b),  $50 \times 50$  image tiles of each mode are concatenated at the input level as a 2-channel image (early fusion) and fed to the network.

Table 2 summarizes the results for all classifiers using (i) only intensity and (ii) intensity and range input pavement tiles. The performance of each classifier is evaluated on each pavement object and on average in terms of precision, recall, and F-score.

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP}, \\ \text{Recall} &= \frac{TP}{TP + FN}, \\ F * \text{score} &= \frac{2TP}{2TP + FP + FN}, \end{aligned} \quad (3)$$

where TP, FP, and FN are true positives, false positives, and false negatives, respectively. The precision determines how many of positive predictions are really positive, while the recall shows the ability of the network in predicting all the relevant instances. The F-score is a harmonic mean of precision and recall that is a useful measure to find the balance between these two metrics. The results show that using both range and intensity images improves the performance of all classifiers in terms of overall precision, recall, and F-score.

In more detail, we compare the baseline models' performances for different classes when they are trained with intensity-only images and intensity-range images. To interpret the results, we divide the classes into two categories: (i) the pavement objects having a height difference with adjacent pixels including crack, crack seal,

pothole, manhole, and patch; (ii) pavement objects having no significant height difference with adjacent pixels including marker, curbing, and asphalt. Using range-intensity input images improved the performance of VGG16, VGG19, ResNet50, and DenseNet121 on the first category of objects, including crack, crack seal, pothole, manhole, and patch, on average by 18.8%, 20.6%, 11.9%, and 14.5% in terms of F-score. The average improvement of the baseline models on crack, crack seal, patch, pothole, and manhole are 12.6%, 22.5%, 21.6%, 22.1%, and 3.6% in terms of F-score. The lower improvement of manhole classification compared to the other four objects comes from the fact that manholes have distinct shapes and textures in intensity images. Therefore, providing range data as complementary information to the network has a milder effect. Incorporating range images into the network barely changes the performance of baseline models on the classification of pavement objects in the second category. In fact, the range image of marking, curbing, and asphalt provide no extra information to the networks for the classification task.

Providing depth information to the DACNN improves the classification results on the first category of objects by 3.2% in terms of F-score. In more detail, utilizing range-intensity images increases the performance of the DACNN on the classification of crack, crack seal, patch, pothole, and manhole by 2.4%, 7.8%, 1.2%, 2.6%, and 2.3% in terms of F-score, respectively. The improvement of DACNN performance by adding depth information is less than such improvement in baseline models. This is because of the high performance of the trained DACNN with intensity-only images which creates less capacity for improvements. As shown in Table 2, the average F-score for DACNN with intensity-only images is 92.9% while the number for VGG16, VGG19, ResNet50, and DenseNet121 is 59.9%, 59.9%, 62%, and 63.4%, respectively. The DACNN also outperforms VGG16, VGG19, ResNet50, and DenseNet121 on average by 23.3%, 22%, 25.4%, and 22.4%, respectively, in terms of F-score when the networks are trained with range-intensity input data. The significant improvement of DACNN classification performance over the baseline models comes from encoding contextual information to the network and adaptively fusing the features through the attention modules. In section (5.2), we show that the performance of baseline models improves by providing multiscale input tiles to the networks. However, DACNN still outperforms those models by having an effective fusion strategy for combining multiscale multimodal features.

Figure 10 demonstrates sample segmentation at a spatial resolution of  $50 \times 50 \text{ mm}^2$  for different algorithms when trained with intensity-only and intensity-range pavement tiles. It can be seen DACNN achieves the best results by extracting a robust representation of range and intensity images. In more detail, we can see that cracks at the top left corner of the image are identified better when the depth information is encoded into all the networks. Range data provide more distinctive features helping the networks to distinguish between foreground and background when intensity values are not distinctive.

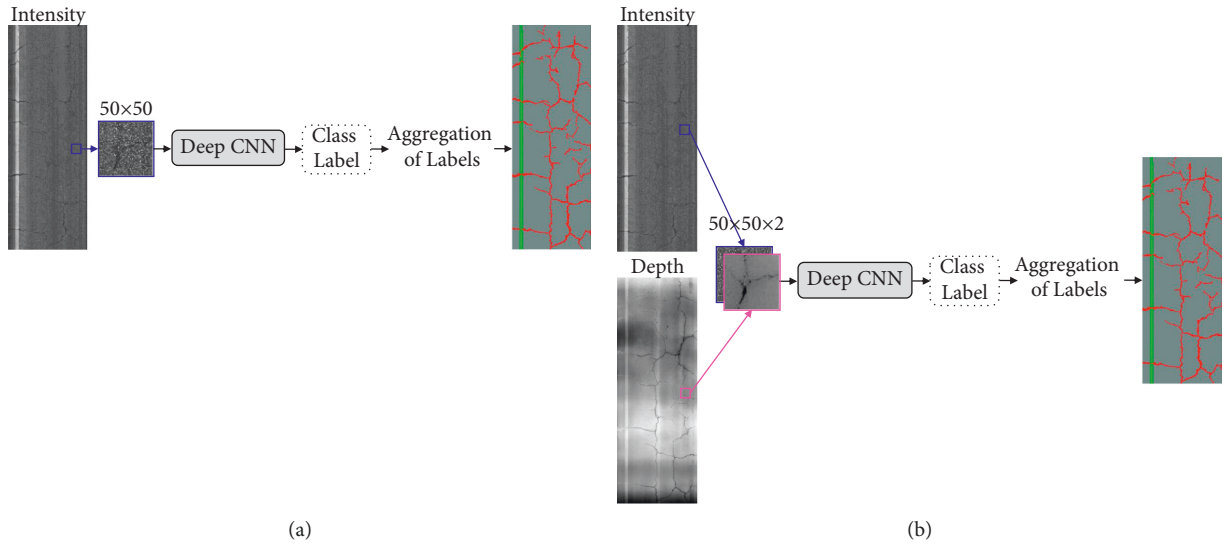


FIGURE 9: An overview of baseline classifiers trained with single-scale (a) intensity images, and (b) intensity and range images.

TABLE 2: Comparison of deep CNNs for classification of pavement objects using single-scale intensity and range input tiles.

Metric	Method	Input image	Crack	Crack seal	Patch	Pothole	Marker	Manhole	Curbing	Asphalt	Avg
Precision	VGG16	Intensity	0.660	0.533	0.676	0.529	0.923	0.850	0.947	0.945	0.758
		Intensity + Range	0.732	0.684	0.849	0.637	0.944	0.875	0.956	0.959	0.830
	VGG19	Intensity	0.593	0.577	0.653	0.636	0.932	0.840	0.944	0.95	0.766
		Intensity + Range	0.670	0.707	0.823	0.570	0.940	0.887	0.955	0.963	0.814
	ResNet50	Intensity	0.637	0.702	0.661	0.554	0.928	0.831	0.945	0.949	0.776
		Intensity + Range	0.679	0.777	0.793	0.593	0.931	0.879	0.954	0.959	0.821
	DenseNet121	Intensity	0.647	0.529	0.688	0.543	0.928	0.850	0.946	0.949	0.760
		Intensity + Range	0.737	0.875	0.789	0.608	0.935	0.895	0.952	0.961	0.844
	DACNN (ours)	Intensity	0.864	0.897	0.965	0.947	0.966	0.919	0.983	0.986	0.941
		Intensity + Range	<b>0.887</b>	<b>0.942</b>	<b>0.972</b>	<b>0.953</b>	<b>0.971</b>	<b>0.965</b>	<b>0.993</b>	<b>0.987</b>	<b>0.959</b>
Recall	VGG16	Intensity	0.256	0.018	0.345	0.253	0.917	0.646	0.985	0.988	0.551
		Intensity + Range	0.438	0.241	0.563	0.463	0.908	0.669	0.988	0.985	0.657
	VGG19	Intensity	0.373	0.033	0.348	0.139	0.904	0.669	0.991	0.981	0.555
		Intensity + Range	0.480	0.209	0.596	0.571	0.920	0.727	0.986	0.983	0.684
	ResNet50	Intensity	0.33	0.073	0.364	0.271	0.911	0.618	0.990	0.984	0.568
		Intensity + Range	0.453	0.163	0.533	0.450	0.925	0.645	0.988	0.984	0.643
	DenseNet121	Intensity	0.324	0.100	0.363	0.319	0.912	0.655	0.990	0.984	0.581
		Intensity + Range	0.431	0.218	0.648	0.528	0.926	0.676	0.992	0.985	0.676
	DACNN (ours)	Intensity	0.780	0.837	0.942	0.909	0.957	<b>0.937</b>	<b>0.990</b>	0.991	0.918
		Intensity + Range	<b>0.805</b>	<b>0.947</b>	<b>0.958</b>	<b>0.956</b>	<b>0.966</b>	<b>0.937</b>	<b>0.990</b>	<b>0.993</b>	<b>0.944</b>
F-score	VGG16	Intensity	0.369	0.034	0.457	0.343	0.920	0.734	0.966	0.966	0.599
		Intensity + Range	0.548	0.356	0.677	0.536	0.926	0.758	0.972	0.972	0.718
	VGG19	Intensity	0.458	0.063	0.454	0.223	0.918	0.745	0.967	0.965	0.599
		Intensity + Range	0.560	0.323	0.691	0.602	0.930	0.799	0.970	0.973	0.731
	ResNet50	Intensity	0.434	0.133	0.469	0.364	0.919	0.709	0.967	0.966	0.620
		Intensity + Range	0.543	0.269	0.638	0.512	0.928	0.744	0.971	0.971	0.697
	DenseNet121	Intensity	0.432	0.169	0.475	0.402	0.920	0.740	0.967	0.966	0.634
		Intensity + Range	0.544	0.349	0.712	0.566	0.930	0.770	0.971	0.973	0.727
	DACNN	Intensity	0.820	0.866	0.953	0.928	0.961	0.928	0.986	0.988	0.929
		Intensity + Range	<b>0.844</b>	<b>0.944</b>	<b>0.965</b>	<b>0.954</b>	<b>0.969</b>	<b>0.951</b>	<b>0.992</b>	<b>0.990</b>	<b>0.951</b>

5.2. Baseline Models with Multiscale Input Images and Results. Figure 11 shows an overview of the deep networks trained with multiscale input tiles to classify pavement objects. The multiscale image tiles are generated at three scales,  $50 \times 50$ ,

$250 \times 250$ , and  $500 \times 500$ , for each mode of intensity and depth. As shown in Figure 11(a), multiscale tiles are concatenated as a 3-channel image to train the deep networks with only intensity images. When training the networks with

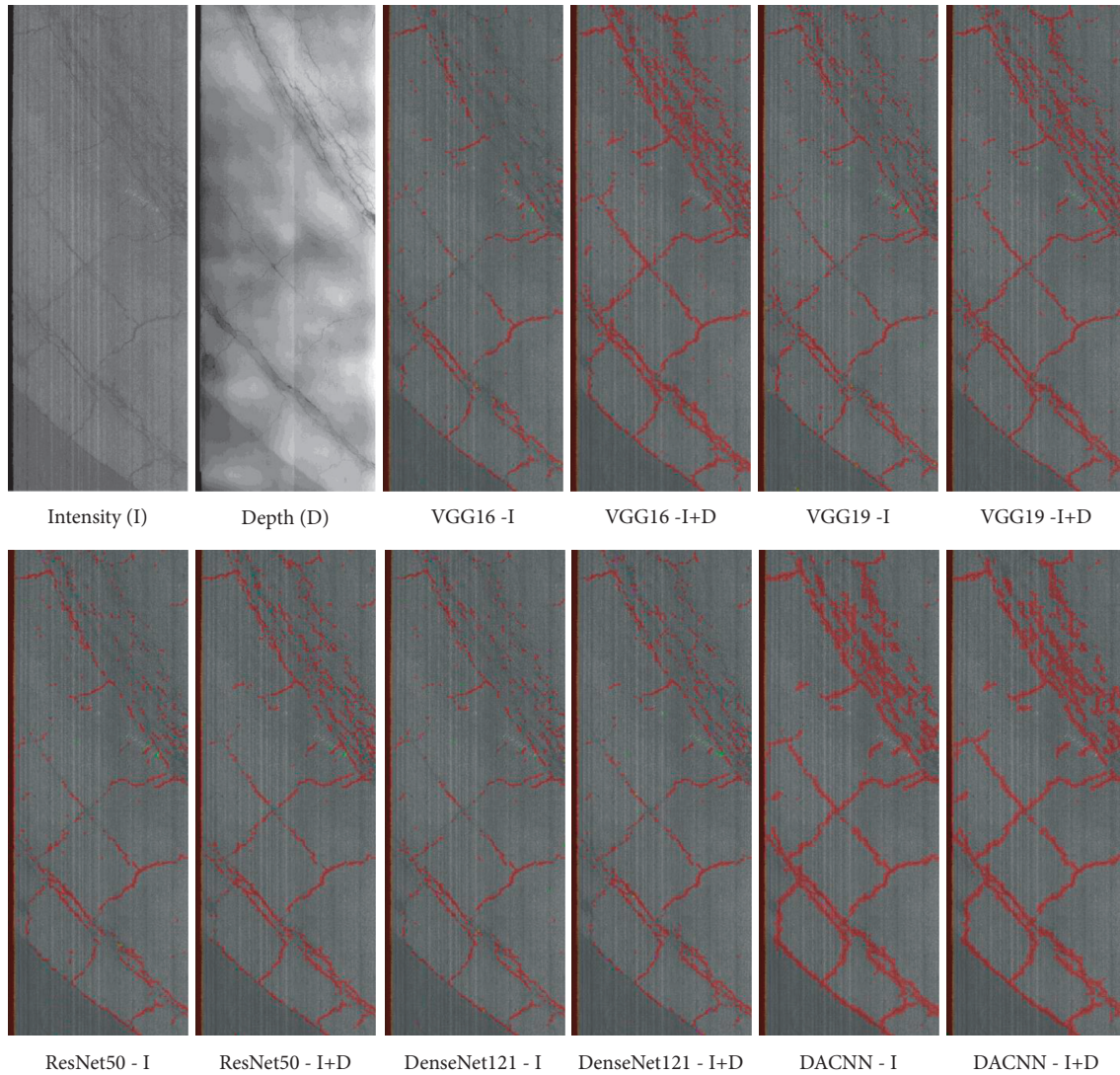


FIGURE 10: Classification results of road cracks using different algorithms trained with intensity-only and intensity-range images. Segmentation masks are created by aggregating classification results of  $50 \times 50$  tiles.

both intensity and range images, as shown in Figure 11(b), the 3-channel image of each mode are merged at the input level (early fusion) and fed to the network.

Table 3 summarizes the performance of baseline models on the classification of 8 pavement classes in terms of precision, recall, and F-score. Comparing the results with the single-scale version of the networks, incorporating the contextual information into the networks improves the average F-score of VGG16, VGG19, ResNet50, and DenseNet121 by 28.3%, 29.3%, 24.3%, and 24.4%, respectively, when trained with intensity-only images. Furthermore, extracting depth features along with intensity features increases the average F-score of the VGG16, VGG19, ResNet50, and DenseNet121 by 4.1%, 3.4%, 4%, and 5.1%, respectively.

Although encoding the contextual information and incorporating the depth data into the network significantly enhances the performance of the baseline models, the DACNN classifies the objects more robustly by having an

effective mid-fusion strategy. The DACNN outperforms VGG16, VGG19, ResNet50, and DenseNet121 trained with multiscale multimodal features by 2.8%, 2.5%, 4.8%, and 2.2%, respectively, on average in terms of F-score. More specifically, the DACNN improves the crack classification (as one of the most important distress types in pavement condition assessment) by 8.8%, 7.2%, 8.7%, and 7% in terms of F-score compared to VGG16, VGG19, ResNet50, and DenseNet121, respectively. This demonstrates the effectiveness of attention modules for pavement object classification.

## 6. Discussion

### 6.1. Qualitative and Quantitative Analysis of DACNN.

One of the most important comparison metrics to evaluate the performance of multiclass classification models is their capability to distinguish between classes. AUC (Area under the Curve) of ROC (Receiver Operating Characteristics) is

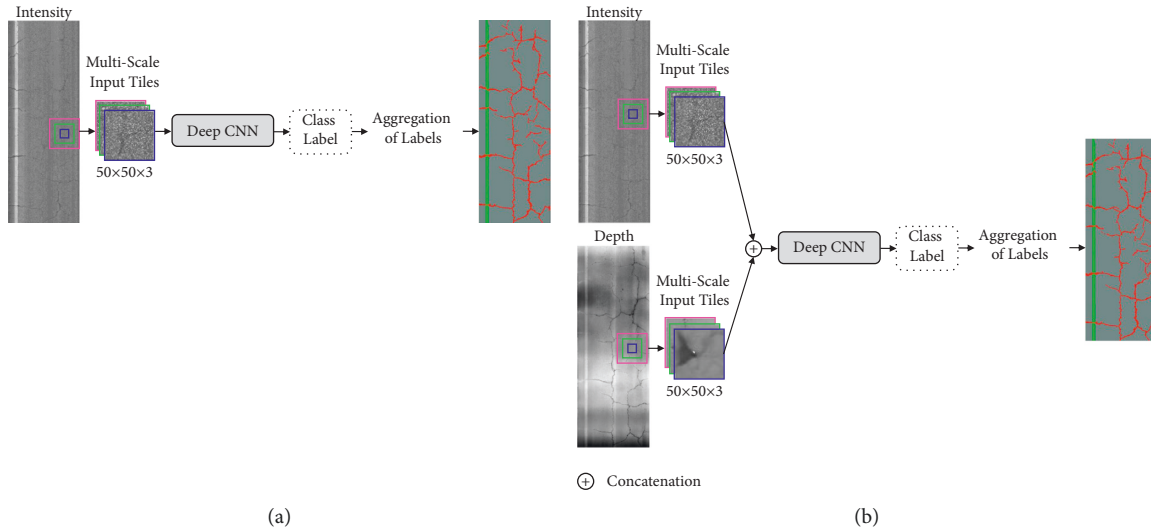


FIGURE 11: An overview of baseline classifiers trained with (a) multiscale intensity images and (b) multiscale intensity and range images.

TABLE 3: Comparison of deep CNNs for classification of pavement objects using multiscale intensity and range input tiles.

Metric	Method	Input image	Crack	Crack seal	Patch	Pothole	Marker	Manhole	Curbing	Asphalt	Avg	
Precision	M-VGG16	Intensity	0.755	0.850	0.874	0.865	0.949	0.930	0.982	0.977	0.898	
		Intensity + Range	0.824	0.966	0.929	0.920	0.958	0.933	0.990	0.982	0.938	
	M-VGG19	Intensity	0.775	0.883	0.859	0.902	0.959	0.959	0.910	0.984	0.977	0.912
		Intensity + Range	0.786	0.904	0.92	0.914	0.959	0.910	0.986	0.985	0.921	
	M-ResNet50	Intensity	0.772	0.894	0.798	0.885	0.952	0.952	0.976	0.975	0.901	
		Intensity + Range	0.781	0.940	0.932	0.898	0.949	0.917	0.980	0.984	0.923	
	M-DenseNet121	Intensity	0.756	0.839	0.869	0.883	0.960	0.934	0.981	0.975	0.900	
		Intensity + Range	0.811	0.889	0.972	0.920	0.961	0.933	0.983	0.984	0.932	
	DACNN (ours)	Intensity	0.864	0.897	0.965	0.947	0.966	0.919	0.983	0.986	0.941	
		Intensity + Range	<b>0.887</b>	<b>0.942</b>	<b>0.972</b>	<b>0.953</b>	<b>0.971</b>	<b>0.965</b>	<b>0.993</b>	<b>0.987</b>	<b>0.959</b>	
	Recall	M-VGG16	Intensity	0.640	0.771	0.838	0.854	0.965	0.910	0.985	0.984	0.868
			Intensity + Range	0.699	0.831	0.954	0.919	0.967	0.946	0.988	0.989	0.912
M-VGG19		Intensity	0.632	0.773	0.883	0.873	0.959	0.906	0.985	0.986	0.875	
		Intensity + Range	0.758	0.920	0.952	0.920	0.961	0.966	0.989	0.985	0.931	
M-ResNet50		Intensity	0.606	0.617	0.853	0.775	0.961	0.897	0.985	0.984	0.835	
		Intensity + Range	0.735	0.693	0.918	0.891	0.970	0.927	0.989	0.985	0.889	
M-DenseNet121		Intensity	0.636	0.766	0.824	0.810	0.952	0.906	0.984	0.985	0.858	
		Intensity + Range	0.741	0.929	0.917	0.943	0.962	0.946	0.990	0.988	0.927	
DACNN (ours)		Intensity	0.780	0.837	0.942	0.909	0.957	<b>0.937</b>	<b>0.990</b>	0.991	0.918	
		Intensity + Range	<b>0.805</b>	<b>0.947</b>	<b>0.958</b>	<b>0.956</b>	<b>0.966</b>	<b>0.937</b>	<b>0.990</b>	<b>0.993</b>	<b>0.944</b>	
F-score		M-VGG16	Intensity	0.693	0.808	0.856	0.859	0.958	0.920	0.984	0.981	0.882
			Intensity + Range	0.756	0.893	0.941	0.920	0.961	0.940	0.989	0.985	0.923
	M-VGG19	Intensity	0.696	0.824	0.871	0.887	0.959	0.931	0.985	0.982	0.892	
		Intensity + Range	0.772	0.912	0.936	0.917	0.960	0.937	0.988	0.985	0.926	
	M-ResNet50	Intensity	0.679	0.730	0.825	0.827	0.956	0.924	0.980	0.980	0.863	
		Intensity + Range	0.757	0.797	0.925	0.895	0.960	0.922	0.984	0.985	0.903	
	M-DenseNet121	Intensity	0.691	0.801	0.846	0.845	0.956	0.919	0.982	0.980	0.878	
		Intensity + Range	0.774	0.908	0.944	0.932	0.962	0.940	0.986	0.986	0.929	
	DACNN	Intensity	0.820	0.866	0.953	0.928	0.961	0.928	0.986	0.988	0.929	
		Intensity + Range	<b>0.844</b>	<b>0.944</b>	<b>0.965</b>	<b>0.954</b>	<b>0.969</b>	<b>0.951</b>	<b>0.992</b>	<b>0.990</b>	<b>0.951</b>	

a measure of how strongly the classifier separates the classes. Higher the AUC, the better the model is capable of predicting true classes. To evaluate the DACNN performance, ROC curves for all investigated methods are plotted in Figure 12. Comparing the AUC values, DACNN

demonstrates a stronger ability to separate classes while predicting the pavement objects.

Figure 13 shows segmentation samples of DACNN generated by integrating classified pavement tiles. The corresponding heatmaps for the pavement classes are also demonstrated for qualitative comparisons. A hotter color

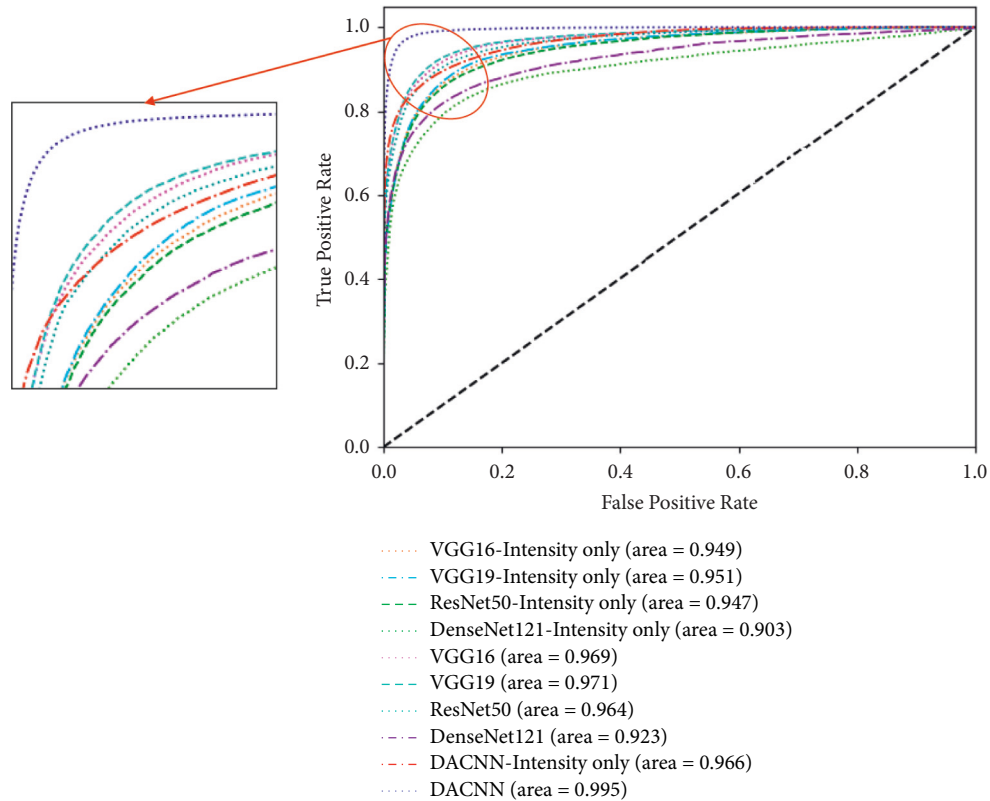


FIGURE 12: Receiver Operating Characteristic (ROC) curves. The presented DACNN achieves the highest area under the curve (AUC).

means a greater probability that the pixels belong to the corresponding class. The heatmaps reveal that the DACNN predicts the pavement object robustly with a strong separation from the rest of the objects.

Figure 14 visualizes the performance of the classifiers in terms of TP, TN, FP, and FN. Having the networks' predictions, we are able to analyze their performance in more detail. Especially in pavement applications, we care about not only increasing TPs but also decreasing FNs and FPs simultaneously. The reason is coming from: (i) having a high FN means that positive distresses are missed leading to an underestimation for road condition assessment, which is dangerous for safety considerations; (ii) having a high FP means that pavement tiles are misclassified as distresses leading to an overestimation, which is not cost-efficient for road assessment. As we can see in Figure 14, DACNN not only increases TPs but also significantly reduces FPs and FNs compared to all other methods. Other than DACNN which presents the best results, encoding depth information into all other networks also increases TPs and reduces FPs and FNs. For the pavement objects with a more distinctive representation in range images including cracks, crack seals, patches, potholes, and manholes, the improvements are more significant after combining the range data with intensity images. Figure 14 shows that DACNN generates the largest number of FPs and FNs for the crack classification. The reason mainly comes from the low contrast between cracks and the background within pavement images. Figure 15 demonstrates examples of DACNN predictions with FPs and FNs on crack classification.

**6.2. Contrast Enhancement.** As described in section 3.2, a histogram equalization technique, CLAHE, is employed to adjust the intensity values and improve the contrast in range images. CLAHE is a modified version of adaptive histogram equalization that limits the contrast to avoid over-amplification and noises in the images. Cliplimit value is the threshold defined to apply a limit over the image contrast. In this study, we conducted a grid search to optimize this hyperparameter for DACNN algorithm. Table 4 summarizes the DACNN performance while using different cliplimit values. Considering the F-score values, cliplimit = 4 is used as the threshold value for CLAHE.

**6.3. Computational Cost.** We compare the computational cost of investigated algorithms in this study in two cases: (i) The networks are trained with only intensity input tiles; (ii) The networks are trained with both intensity and range input images. This way, we can examine how encoding depth information to the networks affects the computational costs. To highlight the trade-off between performance and speed, our proposed method, DACNN is also compared to the baseline approaches. Table 5 summarizes the computational costs for different classification approaches used in this study, in terms of the number of trainable variables, training time per epoch, and inference time for 100 batches. While the first column presents the costs for intensity-only trained networks, including VGG16, VGG19, ResNet50, and DenseNet121, the second column presents the costs for the same networks trained with both intensity and range images.

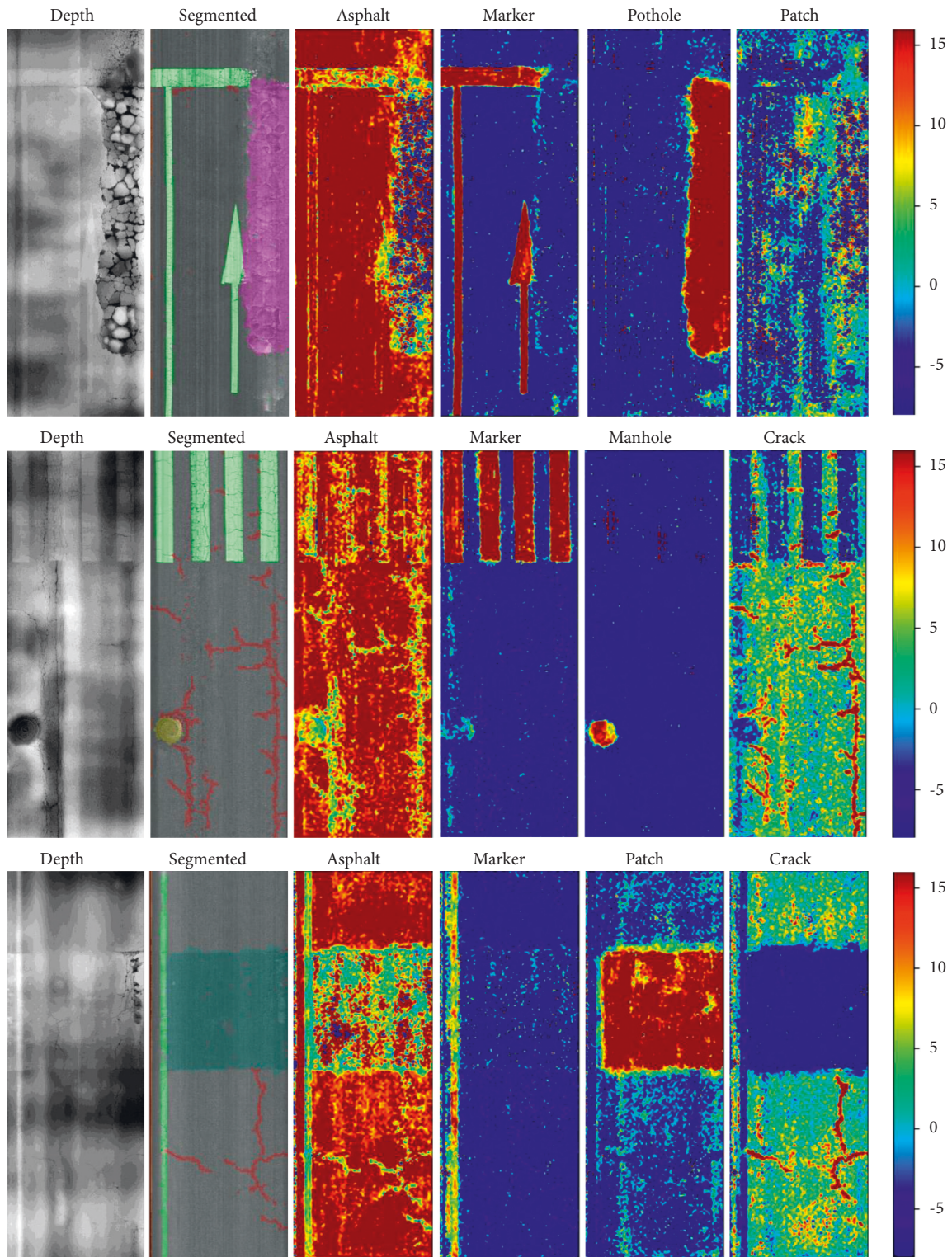
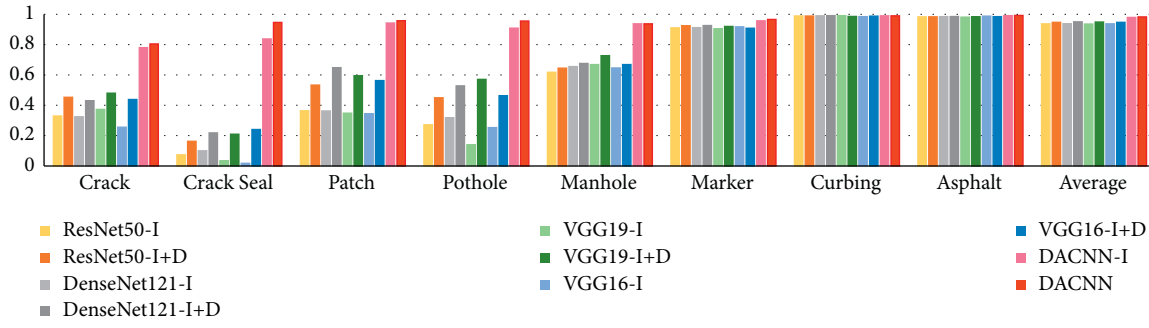


FIGURE 13: Classification results of the DACNN. Hotter colors mean a greater probability that the pixels belong to the specified class.

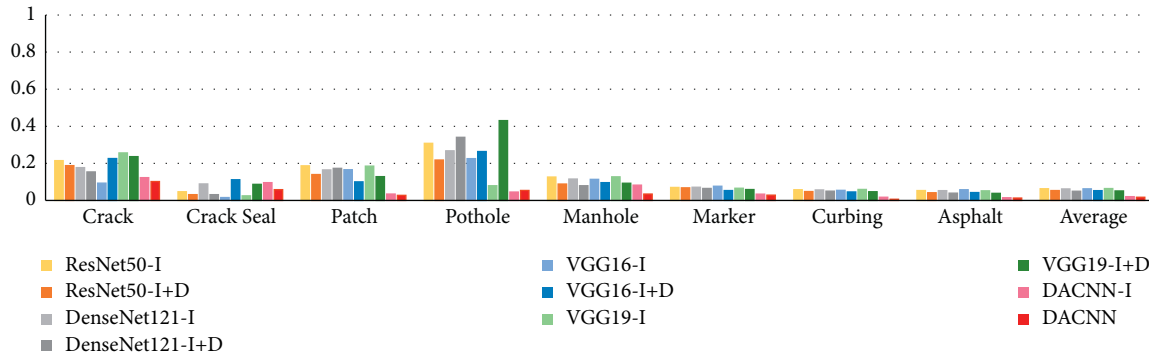
Comparing the first two columns reveals that the extra computational costs brought by encoding depth information to the baseline models were almost negligible. However, the average F-score increased by 16.5% for objects with discriminative features in the range of images (crack, crack seal, pothole, manhole, and patch). The third column shows the

computational costs for DACNN when the depth branch is removed, and the last column shows the cost for DACNN trained with both intensity and range images. It can be concluded that by providing a limited extra source of computations, we can improve the classification results. Training with intensity-only, DACNN enhances the

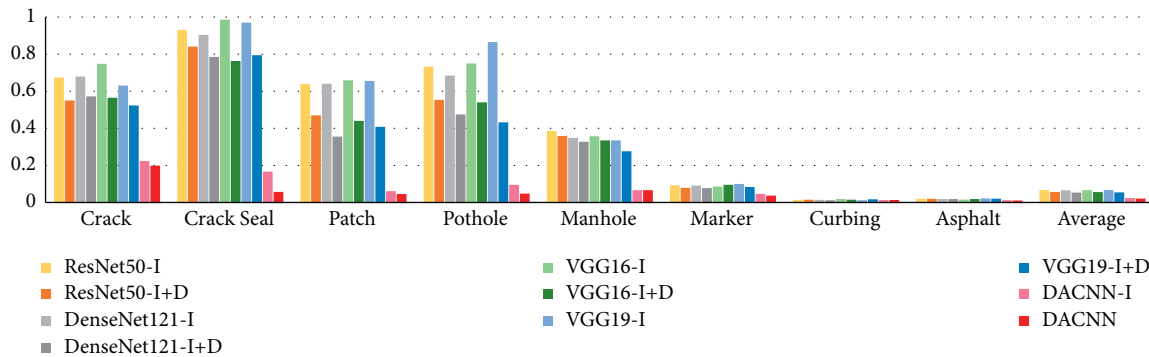




(a)



(b)



(c)

FIGURE 14: Normalized (a) TP, (b) FP, and (c) FN of each class using different algorithms.

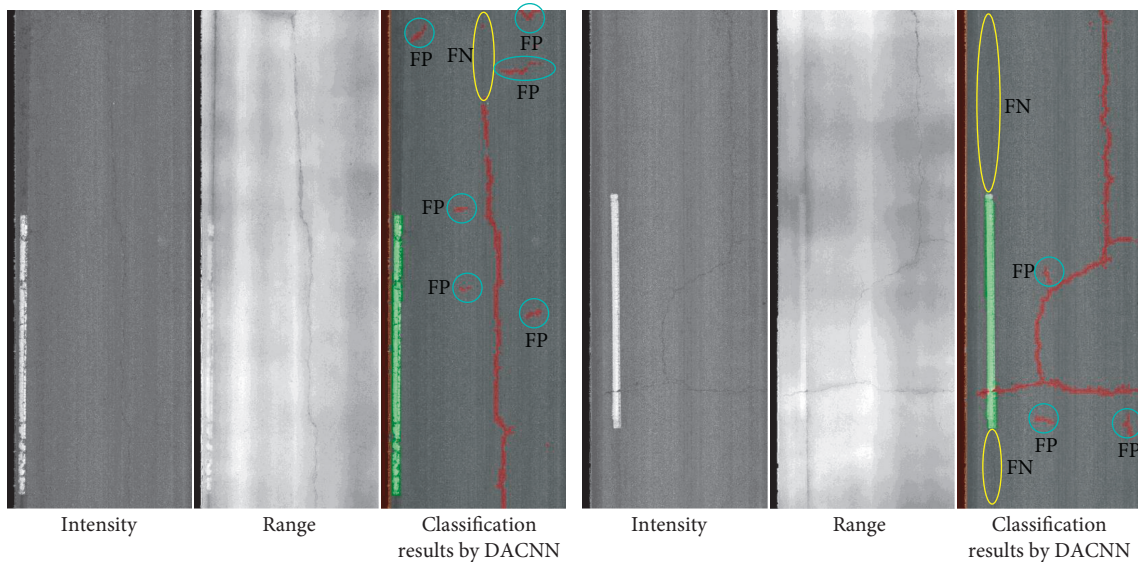


FIGURE 15: Examples of DACNN predictions with FPs and FNs on crack classification.

TABLE 4: Effect of different threshold values for histogram equalization on DACNN results.

Metric	Cliplimit = 2	Cliplimit = 3	Cliplimit = 4	Cliplimit = 5
Precision	0.924	0.973	<b>0.959</b>	0.967
Recall	0.913	0.878	<b>0.944</b>	0.906
F-score	0.918	0.923	<b>0.951</b>	0.936

TABLE 5: Comparison of computational costs for different classification approaches.

Computational costs	Baselines (intensity-only)	Baselines (intensity-range)	DACNN (intensity-only)	DACNN (intensity-range)
Number of parameters	51 M	51 M	61 M	63 M
Training time/epoch	67.1 s	82.6 s	175.6 s	247.5 s
Inference time/100 batches	4.2 s	4.3 s	8.7 s	10.2 s

classification results by capturing contextual information by 31.6% in F-score compared to the baseline methods (first vs. third column). Training with both intensity and range, DACNN improves the classification results by an adaptive fusion strategy by 23.3% in F-score compared to the baseline methods (second vs. fourth column). It should be noted that DACNN is not developed with the goal of having a real-time classification. In most practices, automated assessments of road conditions are performed offline where accuracy and robustness are the most important factors.

## 7. Conclusions

A deep learning-based model termed DACNN is presented to improve the performance of multiclass classification for road objects. Both intensity and range images are fed to the DACNN to enrich the image representation learned by the network. Discriminant feature representations obtained by encoding range images help the network to capture complex topology and to handle noises and illumination variances. Furthermore, feeding multiscale input images into the DACNN enables the network to catch both local and global fields of view, which is beneficial for classifying pavement objects with various sizes and shapes. We designed dual attention modules as an effective way to fuse scale-specific and mode-specific features to model the semantic interdependencies in spatial and channel dimensions. The position attention selectively aggregates the feature at each position by a weighted sum of the features at all positions, and channel attention selectively emphasizes interdependent channel maps by integrating associated features among all channel maps. This way, the network learns better the relevant content for each specific object at each scale and mode contributing to more precise classification results.

The effectiveness and feasibility of the DACNN were compared with four baseline CNN models. The comparison results showed that the DACNN outperforms all compared CNNs. The results also showed that encoding depth information into the networks improves the classification results of VGG16, VGG19, ResNet50, DenseNet121, and the DACNN by 11.9%, 13.2%, 7.7%, 9.3%, and 2.2% in terms of averaged F-score, respectively, compared to

when these models are trained with intensity-only images. The classification improvements are even more significant for pavement objects that are distinctive in range images by having height differences with neighboring pixels. For example, incorporating depth data with intensity information improves the crack classification by 17.9%, 10.2%, 10.9%, 11.2%, and 2.4% in terms of averaged F-score in VGG16, VGG19, ResNet50, DenseNet121, and the DACNN, respectively. In addition to encoding depth data, DACNN yields more improvements by capturing global context through multiscale input tiles, as well as focusing on the most important feature representations through attention modules. The DACNN outperforms VGG16, VGG19, ResNet50, and DenseNet121 by 23.3%, 22%, 25.4%, and 22.4%, respectively, in terms of averaged F-score, while they are all trained with range-intensity tiles.

Although the developed DACNN achieves great performance in pavement object classification, some limitations still exist in our model. Therefore, extra effort is required to make our model more practical and effective. Firstly, our model classifies  $50 \times 50$  pavement tiles into different categories. Although  $50 \times 50 \text{ mm}^2$  spatial resolution is acceptable in most road surveys, a pixel-level segmentation is required for some pavement applications such as crack width measurements. Secondly, quantifying the severity of pavement distresses is of necessity for road condition assessment, but it cannot be obtained directly from our model. Lastly, self-attention mechanisms capturing long-range dependencies in the network can be explored for further improvements. Furthermore, one can conduct hyperparameter studies for the training of the network and provide quantitative comparisons.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no potential conflicts of interest.

## Acknowledgments

This project was partially supported by Korea Institute of Civil Engineering and Building Technology (KICT). Data Transfer Solution (DTS) partially helped in the preparation of the ground-truth dataset used in this study.

## References

- [1] K. A. Zimmerman, "Pavement management systems: Putting data to work," vol. 20-05, 2017, <https://www.trb.org/Publications/Blurbs/175607.aspx>.
- [2] S. Mathavan, K. Kamal, and M. Rahman, "A review of three-dimensional imaging technologies for pavement distress detection and measurements," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2353–2362, 2015.
- [3] T. B. Coenen and A. Golroo, "A review on automated pavement distress detection methods," *Cogent Engineering*, vol. 4, no. 1, Article ID 1374822, 2017.
- [4] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14531–14544, 2020.
- [5] A. Ragnoli, M. R. De Blasiis, and A. Di Benedetto, "Pavement distress detection methods: a review," *Infrastructure*, vol. 3, no. 4, 2018.
- [6] N. S. P. Peraka and K. P. Biligiri, "Pavement asset management systems and technologies: a review," *Automation in Construction*, vol. 119, Article ID 103336, 2020.
- [7] H. S. Munawar, A. W. A. Hammad, A. Haddad, C. A. P. Soares, and S. T. Waller, "Image-based crack detection methods: a review," *Infrastructure*, vol. 6, no. 8, p. 115, 2021.
- [8] Y.-A. Hsieh and Y. J. Tsai, "Machine learning for crack detection: review and model performance comparison," *Journal of Computing in Civil Engineering*, vol. 34, no. 5, Article ID 04020038, 2020.
- [9] K. Gopalakrishnan, "Deep learning in data-driven pavement image analysis and automated distress detection: a review," *Data*, vol. 3, no. 3, p. 28, 2018.
- [10] M. Pak and S. Kim, "A review of deep learning in image recognition," in *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pp. 1–3, IEEE, Kuta Bali, Indonesia, August 2017.
- [11] S. Zhou and W. Song, "Deep learning-based roadway crack classification using laser-scanned range images: a comparative study on hyperparameter selection," *Automation in Construction*, vol. 114, Article ID 103171, 2020.
- [12] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.
- [13] A. Abdollahi and B. Pradhan, "Integrating semantic edges and segmentation information for building extraction from aerial images using unet," *Machine Learning with Applications*, vol. 6, Article ID 100194, 2021.
- [14] J. Tang and Y. Gu, "Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis," in *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3026–3030, IEEE, Manchester, UK, October 2013.
- [15] A. Ayenu-Prah and N. Attoh-Okine, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP Journal on Applied Signal Processing*, vol. 2008, pp. 861701–861707, 2008.
- [16] S. Chambon, P. Subirats, and J. Dumoulin, "Introduction of a wavelet transform based on 2d matched filter in a Markov random field for fine structure extraction: application on road crack detection," *Image Processing: Machine Vision Applications II*, vol. 7251, p. 72510A, 2009.
- [17] L. Sun and Z. Qian, "Multi-scale wavelet transform filtering of non-uniform pavement surface image background for automated pavement distress identification," *Measurement*, vol. 86, pp. 26–40, 2016.
- [18] Y. Hu and C.-x. Zhao, "A novel lbp based methods for pavement crack detection," *Journal of pattern Recognition research*, vol. 5, no. 1, pp. 140–147, 2010.
- [19] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [20] A. Vouloimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, 2018.
- [21] J. Chai, H. Zeng, A. Li, and E. W. Ngai, "Deep learning in computer vision: a critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, Article ID 100134, 2021.
- [22] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, "A deep learning approach for road damage detection from smartphone images," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pp. 5201–5204, IEEE, Seattle, WA, USA, December 2018.
- [23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [24] 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.
- [25] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Germany, September 2016.
- [26] L. Song and X. Wang, "Faster region convolutional neural network for automated pavement distress detection," *Road Materials and Pavement Design*, vol. 22, no. 1, pp. 23–41, 2021.
- [27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
- [28] B. Li, K. C. P. Wang, A. Zhang, E. Yang, and G. Wang, "Automatic classification of pavement crack using deep convolutional neural network," *International Journal of Pavement Engineering*, vol. 21, no. 4, pp. 457–463, 2020.
- [29] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322–330, 2017.
- [30] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2014, <https://arxiv.org/abs/1409.1556>.
- [31] S. L. H. Lau, E. K. P. Chong, X. Yang, and X. Wang, "Automated pavement crack segmentation using u-net-based

- convolutional neural network,” *IEEE Access*, vol. 8, pp. 114892–114899, 2020.
- [32] O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Germany, November 2015.
- [33] 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*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [34] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: a deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [35] T. Chen, Z. Cai, X. Zhao et al., “Pavement crack detection and recognition using the architecture of segnet,” *Journal of Industrial Information Integration*, vol. 18, Article ID 100144, 2020.
- [36] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” 2014, <https://arxiv.org/abs/1409.0473?source>.
- [37] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, “An end-to-end spatio-temporal attention model for human action recognition from skeleton data,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.
- [38] Y. Tian, W. Hu, H. Jiang, and J. Wu, “Densely connected attentional pyramid residual network for human pose estimation,” *Neurocomputing*, vol. 347, pp. 13–23, 2019.
- [39] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, “Adding attentiveness to the neurons in recurrent neural networks,” in *Proceedings of the European Conference on Computer Vision*, pp. 135–151, (ECCV), Germany, October 2018.
- [40] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: a neural network for large vocabulary conversational speech recognition,” in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, IEEE, Shanghai, China, March 2016.
- [41] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, “Self-attentional acoustic models,” 2018, <https://arxiv.org/abs/1803.09519>.
- [42] K. Xu, J. Ba, R. Kiros et al., “Show, attend and tell: neural image caption generation with visual attention,” in *International Conference on Machine Learning*, pp. 2048–2057, PMLR, Breckenridge, Colorado, USA, 2015.
- [43] J. Lu, C. Xiong, D. Parikh, and R. Socher, “Knowing when to look: adaptive attention via a visual sentinel for image captioning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 375–383, San Juan, PR, USA, 2017.
- [44] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, “Attention-based transactional context embedding for next-item recommendation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [45] H. Ying, F. Zhuang, F. Zhang et al., “Sequential recommender system based on hierarchical attention network,” in *IJCAI International Joint Conference on Artificial Intelligence*, Stockholm, 2018.
- [46] V. Mnih, N. Heess, and A. Graves, “Recurrent models of visual attention,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 2204–2212, 2014.
- [47] M. Jaderberg, K. Simonyan, and A. Zisserman, “Spatial transformer networks,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 2017–2025, 2015.
- [48] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.
- [49] M. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber, “Deep Networks with Internal Selective Attention through Feedback Connections,” *Advances in neural information processing systems*, vol. 27, 2014.
- [50] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake City, UT, USA, June 2018.
- [51] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, “Attention augmented convolutional networks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3286–3295, Seoul, Korea (South), October 2019.
- [52] W. Song, G. Jia, D. Jia, and H. Zhu, “Automatic pavement crack detection and classification using multiscale feature attention network,” *IEEE Access*, vol. 7, pp. 171001–171012, 2019.
- [53] H. Wan, L. Gao, M. Su, Q. Sun, and L. Huang, “Attention-based convolutional neural network for pavement crack detection,” *Advances in Materials Science and Engineering*, vol. 2021, Article ID 5520515, 13 pages, 2021.
- [54] W. Qiao, Q. Liu, X. Wu, B. Ma, and G. Li, “Automatic pixel-level pavement crack recognition using a deep feature aggregation segmentation network with a scse attention mechanism module,” *Sensors*, vol. 21, no. 9, p. 2902, 2021.
- [55] W. Wang and C. Su, “Convolutional neural network-based pavement crack segmentation using pyramid attention network,” *IEEE Access*, vol. 8, pp. 206 548–206 558, 2020.
- [56] E. Eslami and H.-B. Yun, “Attention-based multi-scale convolutional neural network (a+ mcnn) for multi-class classification in road images,” *Sensors*, vol. 21, no. 15, p. 5137, 2021.
- [57] Q. Zhou, Z. Qu, and C. Cao, “Mixed pooling and richer attention feature fusion for crack detection,” *Pattern Recognition Letters*, vol. 145, pp. 96–102, 2021.
- [58] Z. Qu, W. Chen, S.-Y. Wang, T.-M. Yi, and L. Liu, “A crack detection algorithm for concrete pavement based on attention mechanism and multi-features fusion,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.
- [59] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr, “Res2net: A New Multi-Scale Backbone Architecture,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, 2019.
- [60] Y. Pan, G. Zhang, and L. Zhang, “A spatial-channel hierarchical deep learning network for pixel-level automated crack detection,” *Automation in Construction*, vol. 119, Article ID 103357, 2020.
- [61] H. Liu, X. Miao, C. Mertz, C. Xu, and H. Kong, “Crackformer: transformer network for fine-grained crack detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3783–3792, Montreal, BC, Canada, October 2021.
- [62] A E1926-08, “Standard Practice for Computing International Roughness index of Roads from Longitudinal Profile Measurements,” *West Conshohocken, PA: ASTM International*, 2015.
- [63] ASTM, “Standard Test Method for Measuring the Longitudinal Profile of Traveled Surfaces with an Accelerometer

- Established Inertial Profiling Reference,” *Annual books of ASTM standards*, vol. 4, pp. 703–707, 1997.
- [64] I O for Standardization, *Characterization of Pavement Texture by Use of Surface Profiles: Determination of Megatexture*, International Organization for Standardization, 2009.
- [65] J. Laurent, D. Lefebvre, and E. Samson, “Development of a new 3d transverse laser profiling system for the automatic measurement of road cracks,” in *Symposium on Pavement Surface Characteristics*, vol. 6th, Portoroz, Slovenia, 2008.
- [66] J. Guan, X. Yang, L. Ding, X. Cheng, V. C. Lee, and C. Jin, “Automated pixel-level pavement distress detection based on stereo vision and deep learning,” *Automation in Construction*, vol. 129, Article ID 103788, 2021.
- [67] H. Lang, J. J. Lu, Y. Lou, and S. Chen, “Pavement cracking detection and classification based on 3d image using multi-scale clustering model,” *Journal of Computing in Civil Engineering*, vol. 34, no. 5, Article ID 04020034, 2020.
- [68] Y. C. J. Tsai and A. Chatterjee, “Pothole detection and classification using 3d technology and watershed method,” *Journal of Computing in Civil Engineering*, vol. 32, no. 2, Article ID 04017078, 2018.
- [69] Y.-C. J. Tsai, Y. Zhao, B. Pop-Stefanov, and A. Chatterjee, “Automatically detect and classify asphalt pavement raveling severity using 3d technology and machine learning,” *International Journal of Pavement Research and Technology*, vol. 14, no. 4, pp. 487–495, 2021.
- [70] Y. J. Tsai, Z. Wang, and F. Li, “Assessment of rut depth measurement accuracy of point-based rut bar systems using emerging 3d line laser imaging technology,” *Journal of Marine Science and Technology*, vol. 23, no. 3, p. 8, 2015.
- [71] Y. J. Tsai, Y. Wu, C. Ai, and E. Pitts, “Critical assessment of measuring concrete joint faulting using 3d continuous pavement profile data,” *Journal of Transportation Engineering*, vol. 138, no. 11, pp. 1291–1296, 2012.
- [72] F. Hong and Y. R. Huang, “Measurement and characterization of asphalt pavement surface macrotexture using three dimensional laser scanning technology,” *Journal of Testing and Evaluation*, vol. 42, no. 4, pp. 20130147–20130890, 2014.
- [73] R. Ghosh and O. Smadi, “Automated Detection and Classification of Pavement Distresses Using 3d Pavement Surface Images and Deep Learning,” *Transportation Research Record*, vol. 2675, pp. 1359–1374, 2021.
- [74] Z. Yang, X. Zhang, Y. Tsai, and Z. Wang, “Quantitative Assessments of Crack Sealing Benefits by 3d Laser Technology,” *Transportation Research Record*, vol. 2675, pp. 103–116, 2021.
- [75] Y. Fei, K. C. P. Wang, A. Zhang et al., “Pixel-level cracking detection on 3d asphalt pavement images through deep-learning-based cracknet-v,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 273–284, 2020.
- [76] B. Li, K. C. P. Wang, A. Zhang, Y. Fei, and G. Sollazzo, “Automatic segmentation and enhancement of pavement cracks based on 3d pavement images,” *Journal of Advanced Transportation*, vol. 2019, Article ID 1813763, pp. 1–9, 2019.
- [77] A. Zhang, K. C. P. Wang, Y. Fei et al., “Automated pixel-level pavement crack detection on 3d asphalt surfaces with a recurrent neural network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 3, pp. 213–229, 2019.
- [78] R. Gui, X. Xu, D. Zhang, and F. Pu, “Object-based crack detection and attribute extraction from laser-scanning 3d profile data,” *IEEE Access*, vol. 7, pp. 172728–172743, 2019.
- [79] A. Zhang, K. C. P. Wang, B. Li et al., “Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.
- [80] A. Zhang, K. C. P. Wang, Y. Fei et al., “Deep learning-based fully automated pavement crack detection on 3d asphalt surfaces with an improved cracknet,” *Journal of Computing in Civil Engineering*, vol. 32, no. 5, Article ID 04018041, 2018.
- [81] Q. Li, D. Zhang, Q. Zou, and H. Lin, “3d laser imaging and sparse points grouping for pavement crack detection,” in *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, pp. 2036–2040, IEEE, Kos, Greece, September 2017.
- [82] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, “Entropy rate superpixel segmentation,” in *Proceedings of the CVPR 2011*, pp. 2097–2104, IEEE, Colorado Springs, CO, USA, June 2011.
- [83] W. Sultani, S. Mokhtari, and H.-B. Yun, “Automatic pavement object detection using superpixel segmentation combined with conditional random field,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2076–2085, 2018.
- [84] L. Wu, S. Mokhtari, A. Nazef, B. Nam, and H.-B. Yun, “Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment,” *Journal of Computing in Civil Engineering*, vol. 30, no. 1, Article ID 04014118, 2016.
- [85] S. K. Shome and S. R. K. Vadali, “Enhancement of diabetic retinopathy imagery using contrast limited adaptive histogram equalization,” *International Journal of Computer Science and Information Technologies*, vol. 2, no. 6, pp. 2694–2699, 2011.
- [86] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.

## Research Article

# Stacking Ensemble Learning Process to Predict Rural Road Traffic Flow

Arash Rasaizadi  and Seyedehsan Seyedabrishami 

Department of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran

Correspondence should be addressed to Seyedehsan Seyedabrishami; [seyedabrishami@modares.ac.ir](mailto:seyedabrishami@modares.ac.ir)

Received 23 January 2022; Revised 24 April 2022; Accepted 12 May 2022; Published 1 June 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Arash Rasaizadi and Seyedehsan Seyedabrishami. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

By predicting and informing the future of traffic through intelligent transportation systems, there is more readiness to avoid traffic congestion. In this study, an ensemble learning process is proposed to predict the hourly traffic flow. First, three base models, including K-nearest neighbors, random forest, and recurrent neural network, are trained. Predictions of base models are given to the XGBoost stacking model and bagged average to determine the final prediction. Two groups of models predict traffic flow of short-term and mid-term future. In mid-term models, predictor features are cyclical temporal features, holidays, and weather conditions. In short-term models, in addition to the mentioned features, the observed traffic flow in the past 3 to 8 hours has been used. The results show that for both short-term and mid-term models, the least prediction error is obtained by the XGBoost model. In mid-term models, the root mean square error of the XGBoost for the Saveh to Tehran direction and Tehran to Saveh direction is 521 and 607 (veh/hr), respectively. For short-term models, these values are decreased to 453 and 386 (veh/hr). This model also brings less prediction error for predicting the first and fourth quartiles of the observed traffic flow as rare events.

## 1. Introduction

Intelligent transportation systems are one of the leading efficient tools for transportation network traffic management. The result of using these systems is achieving or maintaining the balance between transportation supply and demand with low cost [1]. Intelligent transportation systems include various subsystems which one of the most important of them is the advanced traveler information system. By this system, available information about the transportation network is given to travelers to plan their travels with more awareness. This information can be informed for the current state of the network, but its effectiveness becomes more if it is predicted and informed for the future of the transportation network [2]. In such circumstances, the traveler is more prepared to choose the appropriate route and departure time and even to choose to have a trip or cancel it. Generally, traffic parameters such as traffic volume [3], average speed [4], and travel time [5] are predicted and informed by intelligent systems. As the time horizon of these

predictions is limited to the near future compared to the time horizon of classical 4-step transportation planning prediction, they are short-term predictions.

Prediction of traffic parameters is made by analyzing the past observations and discovering effective features on the variation of traffic parameters. For this purpose, the use of time-series models as a tool based on statistics and probability has more antiquity in previous studies. In time-series models, each traffic parameter's variation is a function of that parameter's previously observed values, independent effective features, and random term. For example, Kumar and Vanajakshi [6] have predicted the traffic flow using the seasonal autoregressive integrated moving average (SARIMA). Results show that the model is more accurate than the historical average models. In Yan et al.'s study [7], autoregressive integrated moving average (ARIMA) has been used to predict subway passengers' flow. Time-series models are only capable of considering linear relationships between independent and dependent variables. On the other hand, by increasing the number of observations and features, traffic

data are converted to big data. These models are not compatible with big data characteristics, including volume, velocity, and variety [8].

Another approach to predict traffic parameters is the machine learning (ML) approach. ML models are compatible with big data characteristics and can depict linear and nonlinear relationships. Lack of interpretability and disability in discovering causal relationships are the main weaknesses of ML models, and time-series models are superior to ML models in this regard [9]. ML models are diverse and artificial neural network (ANN) [10], support vector machine [11], and decision tree [12] are some of the widely used ML models. To predict the traffic flow, Ma et al. [13] use ANN optimized by genetic algorithms and exponential smoothing. The results show that the optimization of the artificial neural network improves prediction accuracy. Simple ANN considers consecutive observations independently. To capture the relationship between successive observations, Lu et al. [14] have used a recurrent neural network (RNN) model. The RNN model emphasizes the importance of the time-series nature of data by forming neural network blocks at different time intervals. Each block's input is the output of another block related to past times and predictive features. Also, long short-term memory (LSTM) model is another type of RNN model that considers the dependency of observations for both short-term (near past observations) and long-term (far past observations) pasts. This algorithm is used in Farahani et al. [15] and Chen [16] studies. Wang et al. [17] focus their research on ANN models' weakness in interpreting results. After training a deep neural network model for traffic flow prediction, the proposed model is interpreted in two different ways: first, justifying the number of layers and nodes; second, explaining the causality between historical data and future state of traffic.

The ML models used to predict traffic variables are not limited to the neural network-based modes and traffic flow prediction problem. As an example of other ML models and other traffic parameters, Xu et al. [18] predict the nominal traffic state by using the Kalman filter, Zheng et al. [19] predict traffic speed by K-nearest neighbours (KNNs), Liu et al. [20] predict traffic congestion by random forest (RF), and Yang et al. [21] predict travel time by Markov chain method.

Variety of short-term prediction methods, and on the other hand, lack of a technique that has the highest accuracy for all situations has led researchers to the use of ensemble learning process. In this process, the base models' output is used and to provide one unique final prediction. In general, the ensemble learning process is divided into three categories: bagging, boosting, and stacking. In the bagging process, the base models are trained with the same training dataset, and by averaging or voting, the final prediction is determined. In the boosting process, the base models are trained sequentially to improve the old model's prediction accuracy in the current model. In the stacking process, predictions of base models are introduced as inputs of a supermodel that can be an ML model, the supermodel's output is the final prediction [22]. By using bagging ensemble modeling, Moretti et al. [23] combine predictions of statistical and neural network models to predict traffic flow. Yenru and Haghani [24] use a gradient boosting regression

tree model to predict travel time. Ma et al. [25] use a contextual convolutional recurrent neural network to recognize inter- and intra-day traffic patterns. Lin et al. [26] propose a stacking ensemble learning process to predict public bicycle traffic flow. In all of these three studies, using ensemble learning modeling leads to more accuracy of predictions than base models.

In this study, hourly traffic flow is predicted using three ML base methods, including KNN, RF, and RNN. Outputs of these models are given to XGboost as a stacking supermodel and bagged averaging to predict the final output in the ensemble learning process. The predictive models are divided into two categories: short-term and mid-term. In the short-term models, in addition to the external predictive features including cyclical temporal features, holidays, and weather conditions, the observed traffic flow in the previous 3 to 8 hours has also been used, and these models can only predict the traffic flow only for one and two hours of the future. In mid-term models, only use external predictive features, and there is no time horizon limitation. Finally, the accuracy of these two sets of models is evaluated and compared. The data used in this study are related to traffic data of Tehran-Saveh, a rural road in Iran, for both directions. In general, identifying the dominant pattern of traffic parameters in rural roads is more complicated than the urban roads because in contrast to urban trips, a significant part of rural trips is nonroutine.

This study's contribution is to propose a stacking and bagging ensemble learning process consisting of three base ML algorithms, including KNN, RF, and RNN, alongside the XGboost as a supermodel that puts predictions of base models together. Although previous studies use ensemble learning process for traffic parameter prediction, but designed architecture used in this study is unique. XGboost is a significant part of this structure which is recommended to be used as a stacking supermodel which is not used in the architecture of previous studies related to traffic parameter prediction. Also, short-term and mid-term models with different time horizons and different predictive features are trained and evaluated in this paper for rural road that less investigated before. Finally, employing cyclical feature which are related to temporal features is another novel idea for traffic flow prediction.

## 2. Data

This study's traffic data is collected for one section of the Saveh-Tehran rural road for both directions by loop detectors. Data collection has been carried out for about three years, from 21 March 2017 to 10 March 2020. Data are divided into three sections: first, two years of observations are used to train base models, the next six months, and related predictions of base models are used to train the stacking model. The last six months are used to test the base models and stacking model performances. We called these datasets train 1, train 2, and test. Also, total observations for the ensemble learning process, including train 1 and train 2 datasets, are named train datasets. The raw data includes hourly traffic flow and date. After exploring the relationship between hourly traffic flow and calendar attributes such as

holidays and their type, new features related to the calendar are added to the dataset. Since holidays in Iran are based on two lunar and solar calendars, and as these two calendars are not fixed together, both of them are considered. Also, many passengers start their trips before the holiday and continue it until after the holiday, so it is necessary to consider the effects of holidays on the traffic flow of the days before and after it. Weather condition is another important factor affecting the traffic flow, which is extracted and added to the features. Table 1 describes the candidate features to predict traffic flow in the dataset.

In Table 1, season, solar and lunar months, day of solar and lunar months, day of the week, and time of day (temporal features) are essentially cyclical and varied in particular intervals. For instance, hour 23 and hour 0 are close to each other. This also refers to the spring and winter, the first month of the year and the last month of the year and the first day of the week, and the last day of the week. The biggest problem is letting the algorithms know that these features varied in cycles. Calculating the components of the sinus and cosine and introducing cyclical characteristics is the best way to deal with this problem. For this purpose, the following sinus and cosine transformations are used [27].

$$\begin{aligned} x_{\sin} &= \sin\left(\frac{2 * \pi * x}{\max(x)}\right), \\ x_{\cos} &= \cos\left(\frac{2 * \pi * x}{\max(x)}\right). \end{aligned} \quad (1)$$

The scatter graph of temporal features after these transformations is shown in Figure 1.

Season, solar and lunar months, day of solar and lunar months, day of the week, and time of day are used cyclically in this study.

The introduced features in Table 1 are used to train the mid-term models with the unlimited prediction time horizon. In short-term models, in addition to the features in Table 1, the traffic flow observed at intervals 3 to 8 hours ago is also used as predictor features and these models are only able to predict one and two hours of the future.

Figure 2 depicts the traffic flow histogram for the ensemble learning train dataset (train 1 + train 2) and the test dataset. Table 2 presents a statistical summary description of traffic flow.

In current study, to prepare and select predictive features, cyclical features have been used. There are several input data selection methods for this purpose. For example, genetic algorithm, forward or backward feature selection, and recursive feature elimination [28]. In the rest of this paper, the effect of using cyclical features have been presented.

### 3. Methods

This study proposes a stacking and bagging ensemble learning process consisting of three base ML algorithms, including KNN, RF, and RNN, alongside the XGboost as a supermodel. We choose base models based on their accuracy and selected

base model outperforms other models. For example, we tried to employ LSTM algorithm as a deep learning base model but the resulted predictions have not enough accuracy to consider LSTM in ensemble learning process.

**3.1. K-Nearest Neighbors.** The KNN model is an ML method used for both classification and regression problems. The main objective of the KNN is to find some labeled observations in the training dataset which have the smallest distance with nonlabeled observations in the test data. Using the averaging or voting, the new label is assigned to new data [29]. The four main steps of this approach are as following:

- Step 1: the train dataset is given in an  $n$ -dimensional coordinate system ( $n$  is the number of features).
- Step 2: Euclidean distance between any new observation and training data observations is calculated.
- Step 3:  $k$  is the number of observations that have the smallest distance from any new observation.
- Step 4: the average of  $K$  observation labels is selected as the new observation label.

Euclidean distance between observations  $p$  and  $q$  is defined according to equation (2) [30].

$$\begin{aligned} d(p, q) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \end{aligned} \quad (2)$$

**3.2. Random Forest.** Similar to the KNN, the RF is an ML model used for regression and classification problems. The RF consists of a large number of decision trees. In this model, the training data are divided between decision tree models, and after training them, predictions are made for each decision tree. The average of predictions is determined as the RF's final prediction [31]. The following steps indicate how the algorithm works.

- Step 1: start with the select random samples from the training dataset
- Step 2: using each sample to train a decision tree.
- Step 3: the prediction of each decision tree model is made for the test data.
- Step 4: the average of predictions is selected as the final prediction.

RF starts with a node and branches to another node. This paper uses the entropy formula to determine how the dataset branches from each node. Equation (3) presents the entropy formula [31].

$$\text{Entropy} = \sum_{i=1}^c -p_i * \log_2(p_i), \quad (3)$$

where  $p_i$  is the relative frequency of label  $i$ ,  $i$  is the index of labels, and  $c$  is the total number of labels.



TABLE 1: Description of candidate features to predict traffic flow.

Feature name	Description
Season	Including spring, summer, fall, and winter
Solar month	Including 12 solar months
Lunar month	Including 12 lunar months
Day of a solar month	Including 29–31 days of a solar month
Day of a lunar month	Including 29–30 days of a lunar month
Time of day	Including 24 hours a day
6 hour before the holidays	Equal to 1 if it is 1 to 6 hour before holidays
6 hour after the holidays	Equal to 1 if it is 1 to 6 hour after holidays
Day or night	Including day and night
Number of holidays	The number of sequential holidays
Holidays	Includes 1 for holidays and 0 for other days
Holiday type	Type of holidays
Holiday in three days later	Equal to 1 if three days later is a holiday
Type of holidays in three days later	Including the holiday type of three days later if it is a holiday, otherwise equals 0.
Holiday in three days ago	Equal to 1 if three days ago is a holiday
Type of holidays in three days ago	Including the holiday type of three days ago if it is a holiday, otherwise equals 0.
Holiday in two days later	Equal to 1 if two days later is a holiday
Type of holidays in two days later	Including the holiday type of two days later if it is a holiday, otherwise equals 0.
Holiday in two days ago	Equal to 1 if two days ago is a holiday
Type of holidays in two days ago	Including the holiday type of two days ago if it is a holiday, otherwise equals 0.
Holiday in a day later	Equal to 1 if a day later is a holiday
Type of holidays in a day later	Including the holiday type of a day later if it is a holiday, otherwise equals 0.
Holiday in a day ago	Equal to 1 if a day ago is a holiday
Type of holidays in a day ago	Including the holiday type of a day ago if it is a holiday, otherwise equals 0.

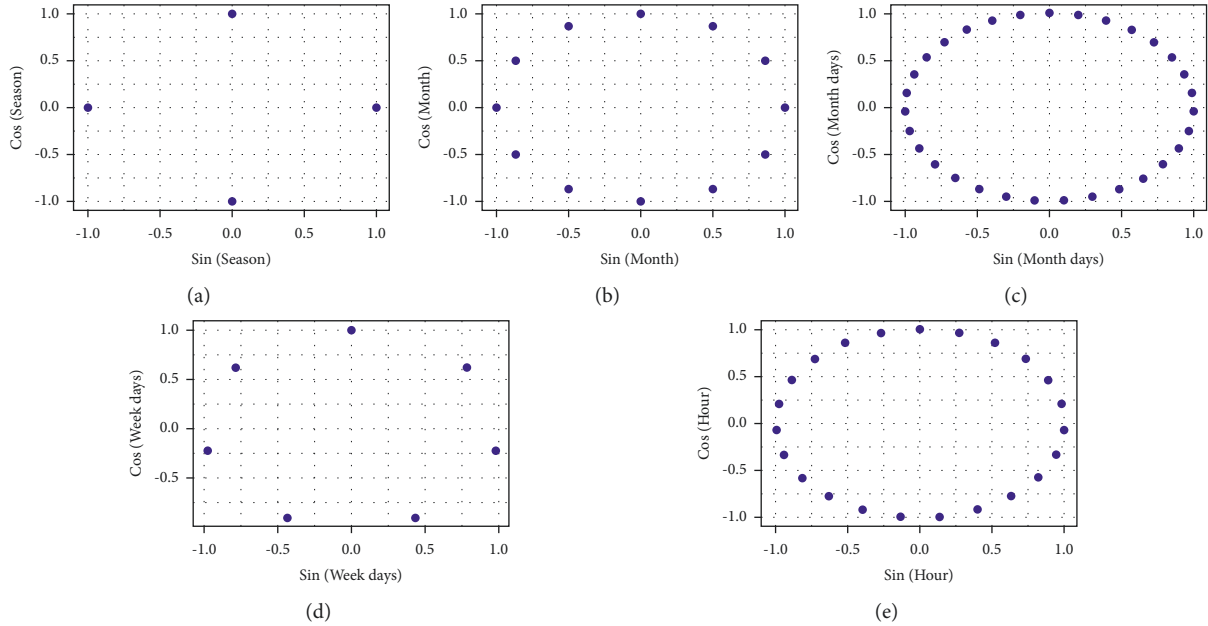


FIGURE 1: Cyclical features in the dataset. (a) Cyclical seasons, (b) Cyclical solar and lunar months, (c) Cyclical days of a solar or a lunar month, (d) Cyclical days of a week, (e) Cyclical hours of a day.

**3.3. Recurrent Neural Network.** RNN is a kind of deep neural network. Since the successive observations are dependent on each other, the use of the RNN can help improve the accuracy of predictions. These ANNs are particularly useful for time-series analysis, where each neuron can maintain internal information of the connected nodes. This attribute of maintaining the internal state or the memory capability helps the network to understand and discover the link between different successive observations [32].

Let denote the input time series with  $D$  variables of length  $T$  as  $X = (X_1, X_1, \dots, X_T)$ , where  $X_t$  is the  $t$ -th observation.  $c_t$  is a memory cell, contains information at time step  $t$ , and is controlled by three gates. These gates control whether to forgot the current cell value (forget gate  $f_t$ ) to read its input (input gate  $i_t$ ) and to output the new cell value (output gate  $o_t$ ) [33]. Also,  $\tilde{c}_t$  is an input modulation gates. All these gates, cell update, and output are computed in the following formulas [34]:

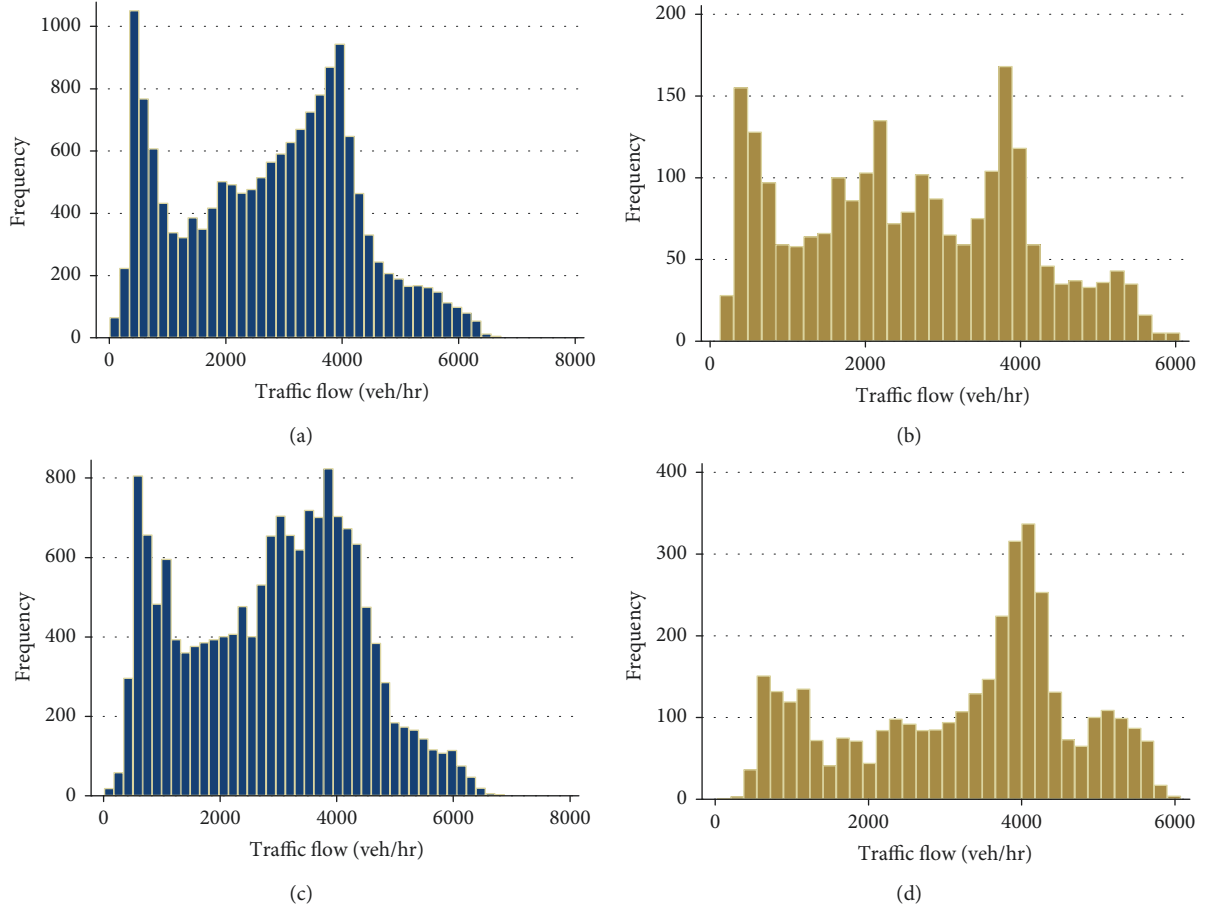


FIGURE 2: Traffic flow histograms for train and test datasets of Saveh–Tehran road. (a) Histogram of train dataset for Tehran to Saveh. (b) Histogram of test dataset for Tehran to Saveh. (c) Histogram of train dataset for Saveh to Tehran. (d) Histogram of test dataset for Saveh to Tehran.

TABLE 2: Statistical summary description of traffic flow.

Direction	Dataset	Number of observations	Average (veh/hr)	Standard deviation (veh/hr)	Minimum (veh/hr)	Maximum (veh/hr)
Tehran to Saveh	Train	19908	2750	1500	8	7069
	Test	3658	2578	1447	130	6058
Saveh to Tehran	Train	19901	2907	1455	19	6880
	Test	3686	3270	1439	29	6079

$$\begin{aligned}
 i_t &= \sigma(W_{xi}X_t + W_{hi}h_{t-1}), \\
 f_t &= \sigma(W_{xf}X_t + W_{hf}h_{t-1}), \\
 o_t &= \sigma(W_{xo}X_t + W_{ho}h_{t-1}), \\
 \tilde{c}_t &= \phi(W_{xc}X_t + W_{hc}h_{t-1}), \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
 h_t &= o_t \odot \phi(c_t),
 \end{aligned} \tag{4}$$

where  $\odot$  indicates scalar product,  $W$  s are the network parameters matrices,  $h_t$  is the hidden state,  $\phi$  is the hyperbolic tangent function, and  $\sigma$  denotes the standard logistics sigmoid transfer function.

**3.4. Bagged Averaging.** After training KNN, RF, and RNN, the predicted traffic flow is given to the ensemble learning

algorithms to determine the final prediction. Bagged averaging is one of these algorithms that can be done weighted or simple. In the weighted method, each model's prediction weight is inversely related to the model's root mean square error (RMSE). Equation (5) shows how weights in bagged averaging are calculated.

$$W_i = \frac{1/\text{RMSE}_i}{\sum_{i=1}^I \text{RMSE}_i}, \tag{5}$$

where  $W_i$  is prediction weight of model  $i$ ,  $I$  is the total number of models, and  $\text{RMSE}_i$  is the root mean square error of model  $i$ .

**3.5. Stacking XGBoost.** XGBoost is an optimized variant of the ensemble learning model that has improved and expanded from the tree model of gradient boosting. Under the

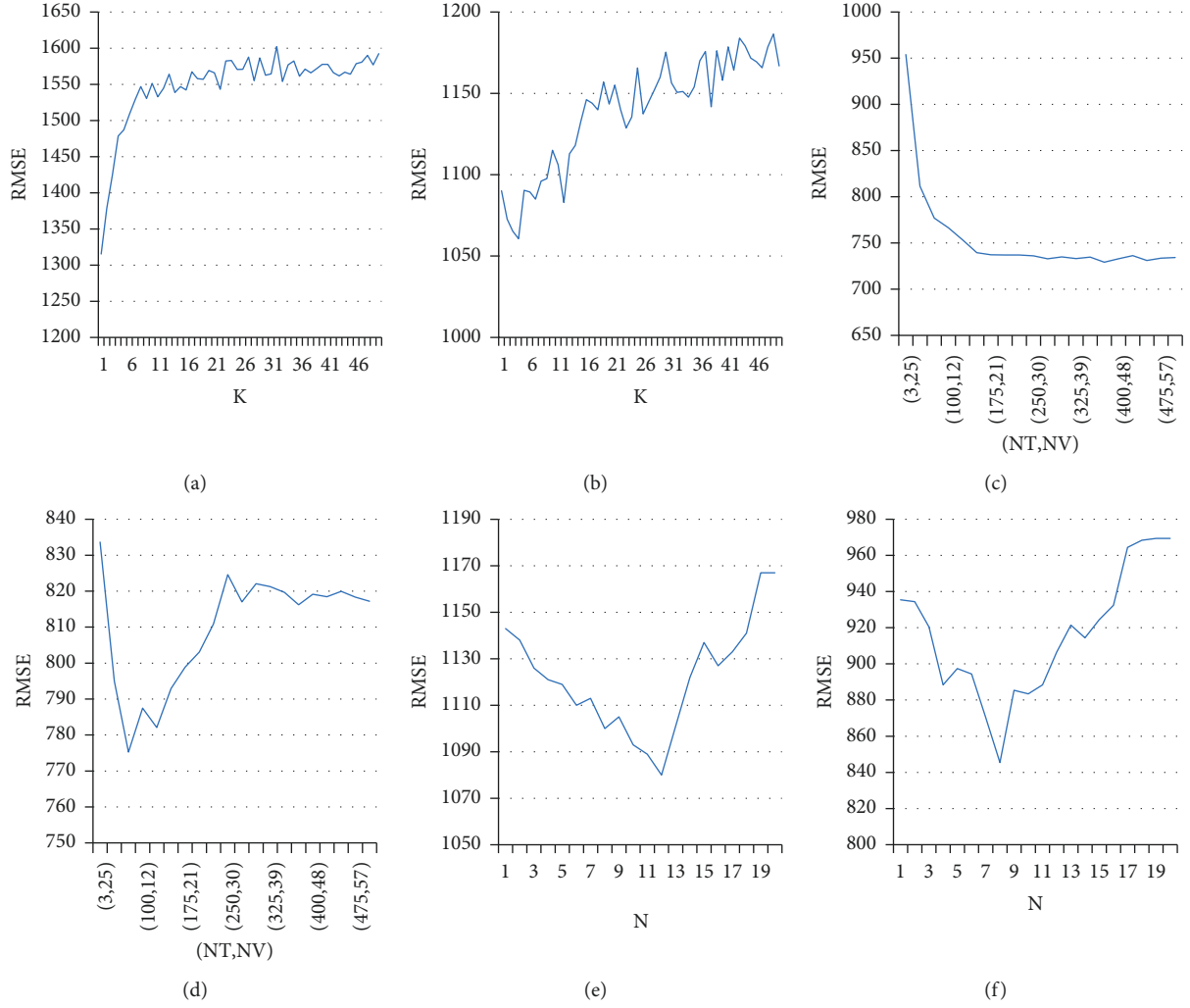


FIGURE 3: Parameter tuning of short-term models. (a)  $K$  in KNN for Saveh to Tehran. (b)  $K$  in KNN for Tehran to Saveh. (c) NT and NV in RF for Saveh to Tehran. (d) NT and NV in RF for Tehran to Saveh. (e)  $N$  in RNN for Saveh to Tehran. (f)  $N$  in RNN for Tehran to Saveh.

gradient boosting paradigm, it applies ML algorithms. XGBoost offers a parallel tree boost that easily and reliably addresses several data science issues [35]. The boosting tree is defined as follows:

$$\hat{y}_i = \sum_{k=1}^n f_k(x_i), f_k \in F, \quad (6)$$

where  $F$  is the set of decision trees,  $\hat{y}_i$  is model prediction,  $x_i$  is a set of predictor features, and  $n$  is the number of trees. The loss function of the model is as follows:

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^k \Omega(f_k), \quad (7)$$

where  $L$  is the difference between the predicted and actual values, named differentiable function. Popular loss functions include square, logarithmic, and exponential function functions.  $\Omega$  is used to regulate the complexity of the model.

$$\Omega = \gamma T + \lambda, \quad (8)$$

where  $\gamma$  and  $\lambda$  are penalty coefficients. XGBoost aims to minimize the differentiable function. By rewriting the differentiable function and Taylor expansion, the formula is as follows:

$$\begin{aligned} L^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i + f_t(x_i)) + \sum_{k=1}^k \Omega(f_k), \\ &\approx \sum_{i=1}^n l\left[(y_i + \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \sum_{k=1}^k \Omega(f_k), \end{aligned} \quad (9)$$

where  $g_i$  and  $h_i$  are the first and second derivatives of the loss function, respectively [36].

## 4. Results and Discussion

**4.1. Base Models Results.** In the first step to train the KNN, RF, and RNN, selecting proper values for model parameters has a significant effect on the final accuracy of prediction. These parameters include the number of neighbors ( $K$ ) in the

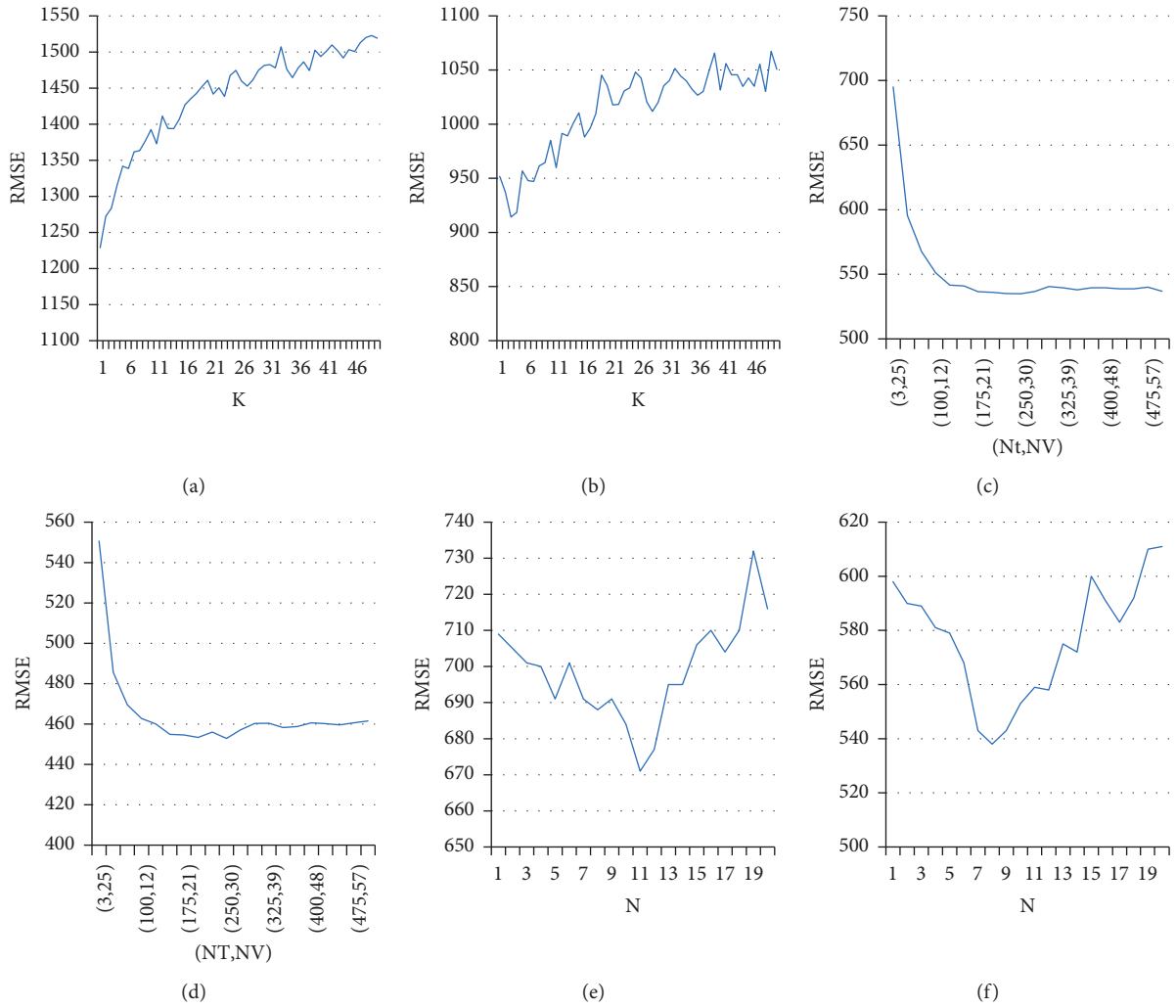


FIGURE 4: Parameter tuning of mid-term models. (a)  $K$  in KNN for Saveh to Tehran. (b)  $K$  in KNN for Tehran to Saveh. (c)  $NT$  and  $NV$  in RF for Saveh to Tehran. (d)  $NT$  and  $NV$  in RF for Tehran to Saveh. (e)  $N$  in RNN for Saveh to Tehran. (f)  $N$  in RNN for Tehran to Saveh.

TABLE 3: Selected optimal values for models parameters.

Time horizon	Model	Road	Optimal parameters
Mid-term	RF	Tehran to Saveh	$NV = 9$ $NT = 75$
		Saveh to Tehran	$NV = 45$ $NT = 375$
	KNN	Tehran to Saveh	$K = 4$
		Saveh to Tehran	$K = 1$
	RNN	Tehran to Saveh	$N = 8$
		Saveh to Tehran	$N = 12$
Short-term	RF	Tehran to Saveh	$NV = 30$ $NT = 250$
		Saveh to Tehran	$NV = 30$ $NT = 250$
	KNN	Tehran to Saveh	$K = 3$
		Saveh to Tehran	$K = 1$
	RNN	Tehran to Saveh	$N = 8$
		Saveh to Tehran	$N = 11$

TABLE 4: Error metrics for the final models.

Time horizon	Model	Road	RMSE	MAPE
Mid-term	RF	Tehran to Saveh	775	27.14
		Saveh to Tehran	729	21.23
	KNN	Tehran to Saveh	1060	33.58
		Saveh to Tehran	1315	33.58
	RNN	Tehran to Saveh	845	42.22
		Saveh to Tehran	1080	34.12
Short-term	RF	Tehran to Saveh	453	16.61
		Saveh to Tehran	535	15.25
	KNN	Tehran to Saveh	914	34.59
		Saveh to Tehran	1229	34.47
	RNN	Tehran to Saveh	538	25.03
		Saveh to Tehran	671	21.60

KNN, the number of trees ( $NT$ ), and the number of variables randomly sampled as candidates at each split ( $NV$ ) in RF, and the number of hidden layers ( $N$ ) in the neural network model. To find the optimal value of these parameters after assigning

different values to them, models are trained. Accuracy for the test dataset is evaluated based on the RMSE. Equation (10) represents how to calculate the RMSE. Figures 3 and 4 show

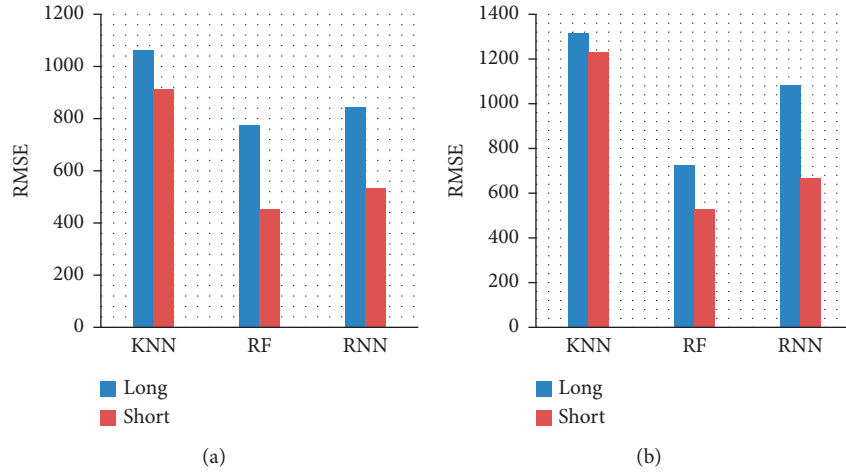


FIGURE 5: RMSE of short-term and mid-term models. (a) Saveh to Tehran. (b) Tehran to Saveh.

TABLE 5: Error metrics for the ensemble learning methods and the most accurate base model.

Time horizon	Model	Road	RMSE	MAPE
Mid-term	RF	Tehran to Saveh	775	27.14
		Saveh to Tehran	729	21.23
	Max	Tehran to Saveh	687	35.20
		Saveh to Tehran	578	21.92
	Min	Tehran to Saveh	1054	33.09
		Saveh to Tehran	1386	42.49
	Bagged simple averaging	Tehran to Saveh	705	27.03
		Saveh to Tehran	868	26.96
	Bagged weighted averaging	Tehran to Saveh	683	26.62
		Saveh to Tehran	832	26.05
	XGBoost	Tehran to Saveh	607	24.59
		Saveh to Tehran	521	20.48
Short-term	RF	Tehran to Saveh	453	16.61
		Saveh to Tehran	535	15.25
	Max	Tehran to Saveh	696	41.03
		Saveh to Tehran	639	26.46
	Min	Tehran to Saveh	957	30.74
		Saveh to Tehran	1214	35.79
	Bagged simple averaging	Tehran to Saveh	638	26.86
		Saveh to Tehran	748	24.52
	Bagged weighted averaging	Tehran to Saveh	595	26.11
		Saveh to Tehran	683	22.57
	XGBoost	Tehran to Saveh	386	14.07
		Saveh to Tehran	453	14.61

the sensitivity analysis performed to find optimal values of the short-term and mid-term models' parameters.

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}, \quad (10)$$

where  $\hat{y}_t$  and  $y_t$  are predicted and actual values, and  $n$  is the number of observations.

Table 3 shows selected optimal values for final models.

After training the final models to assess the accuracy of predictions on the test dataset, in addition to the RMSE, the mean absolute percentage error (MAPE) is used. Equation (11) shows how MAPE is calculated. Table 4 presents the obtained values of error metrics for the final models.

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{\hat{y}_t - y_t}{y_t} \right|. \quad (11)$$

Results in Table 4 shows that for both short-and mid-term models and both directions of Saveh–Tehran road, the lowest error prediction is achieved by the RF, and then the KNN has the highest prediction error. The MAPE of the mid-term RF for the Saveh to Tehran and Tehran to Saveh is 21.23 and 27.14, respectively. Also, in the short-term model, the MAPE of RF for the Saveh to Tehran and Tehran to Saveh is 15.25 and 16.61.

Figure 5 shows the difference between the RMSE of the short-term and mid-term models. The accuracy of the short-term models is higher than the mid-term models, and using previously observed traffic flows had increased the accuracy

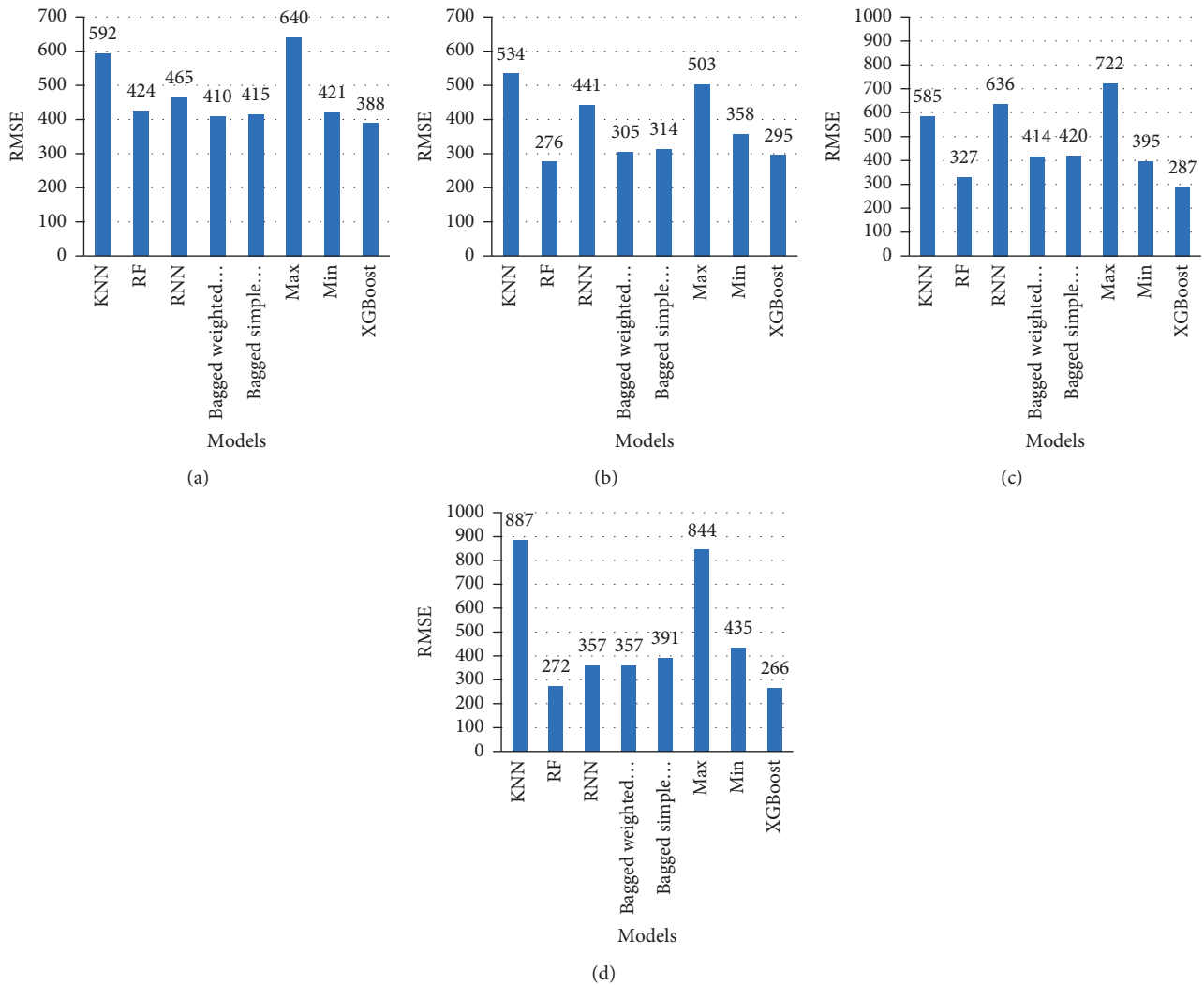


FIGURE 6: RMSE of base, bagging, and stacking models for the first quartiles of the observed traffic flow. (a) Mid-term models of Tehran to Saveh. (b) Mid-term models of Saveh to Tehran. (c) Short-term models of Tehran to Saveh. (d) Short-term models of Saveh to Tehran.

of the prediction. The limited-time horizon of these models is considered as their weakness.

**4.2. Bagging and Stacking Ensemble Models Results.** After receiving the base models' predictions, the ensemble learning process is performed by using the bagging and stacking methods, and the final results are obtained. In addition to the bagging and stacking methods, the maximum and minimum predicted traffic flow values are analyzed as the final prediction. Like the base models, the ensemble learning process has also been examined for short-term and mid-term predictions that their inputs are the short-term and mid-term output of base models. Table 5 shows the results obtained by ensemble learning and the RF as the most accurate base model.

Table 5 indicates that for both the short-term and mid-term models and both directions of Tehran–Saveh road, based on the RMSE and the MAPE, using the XGBoost, decreases the prediction error and stacking ensemble

learning by using XGBoost has the lowest prediction error. Based on the RMSE, in the mid-term model, the predictions through maximum and minimum values of the predicted traffic flow values have higher and lower accuracy compared to the RF, respectively. It can be concluded that the base models underestimate traffic flow. Bagged averaging only increases the accuracy of predicting for Tehran to Saveh. In the short-term models, only the XGBoost model has reduced the traffic volume prediction error, and other methods have no positive effects on the accuracy of traffic flow prediction.

Another critical point in the traffic flow prediction is predicting maximum and minimum traffic flow values that indicate rare traffic events. Generally, informing hours with high and low traffic flow is more worthwhile for users and system operators than normal traffic flows. To determine the models' performance in predicting rare events, the RMSE has been calculated separately for the first and fourth quartiles of the observed traffic flow and presented in Figures 6 and 7.

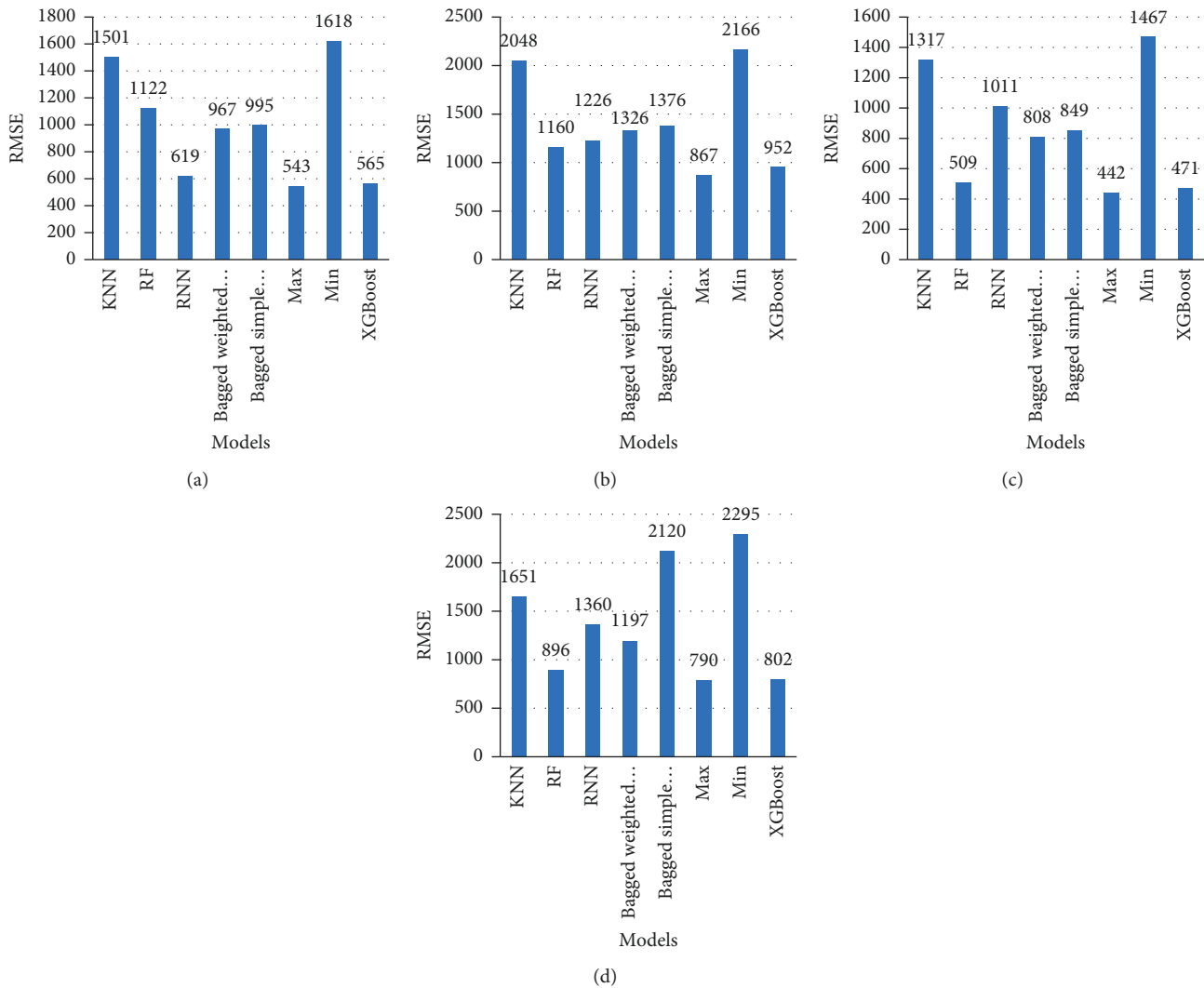


FIGURE 7: RMSE of the base, bagging, and stacking models for the fourth quartiles of the observed traffic flow. (a) Mid-term models of Tehran to Saveh. (b) Mid-term models of Saveh to Tehran. (c) Short-term models of Tehran to Saveh. (d) Short-term models of Saveh to Tehran.

Figures 6 and 7 show the lowest RMSE for the first and fourth quartiles are achieved by XGBoost and Max methods, respectively. The exciting point is less prediction error of the XGBoost than the Min method in predicting the first quartile. The XGBoost could predict both the first and fourth quartile more accurately than the base models, whereas the Max method only predicts the fourth quarter more accurately than the base models. Among the base models, the RF model predicts the traffic flow for two quadrants more accurately than the two other base models.

## 5. Conclusion

One of the applications of intelligent transportation systems is predicting the future state of traffic while the traveler will have more proper planning to choose travel, departure time, and route choice. Also, the transportation network operator will be more prepared to deal with traffic congestion. In this study, traffic flow as a parameter shows

the state of traffic is predicted using three base methods based on ML, including KNN, RF, and RNN for a rural road in Iran for both directions. Then, using the bagging and stacking methods, the most important of them is the XGBoost, and the final traffic flow is predicted. Pre-processing is performing by adding predictor features related to cyclical temporal features, holidays, types of holidays, and weather in the first step. In the second step, to find optimal values of the parameters of short-term and mid-term models, models are trained by different values of parameters, and optimal values are selected based on the accuracy of prediction on the test data. After training the base models with optimal values of parameters, the initial predictions are evaluated and compared. In the next step, by using base models' predictions, the ensemble learning process is applied to make the final prediction, which is expected to be more accurate than base models predictions. The results show that the highest accuracy of prediction for both short-term and mid-term is achieved

using the XGBoost model in the stacking learning process. This model predicts the first and fourth quartiles of the observed traffic flow more accurately than the base models. In general, the prediction error of short-term models is lower than the mid-term models. However, these models can only predict the traffic flow of one and two hours of the future.

In the end, the predicted traffic flow by short-term and mid-term models can be informed to passengers via advanced traveler information systems. To use the prediction accuracy of the short-term models and have the prediction time horizon of mid-term models, future one and two hours will be predicted by short-term models, and for the next hours, prediction by mid-term models can be used.

## Data Availability

The traffic data used in this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

A. R. contributed to software analysis, validation, formal analysis, data curation, original draft preparation, and visualization. S. S. contributed to conceptualization, methodology, supervision, and review and editing.

## References



- [1] A. Merenkov, "Digital economy: transport management and intelligent transportation systems," *E-management*, vol. 1, no. 1, pp. 12–18, 2018.
- [2] A. Boukerche and J. Wang, "Machine learning-based traffic prediction models for intelligent transportation systems," *Computer Networks*, vol. 181, Article ID 107530, 2020.
- [3] S. Manne, E. L. Lydia, I. V. Pustokhina, D. A. Pustokhin, V. S. Parvathy, and K. Shankar, "An intelligent energy management and traffic predictive model for autonomous vehicle systems," *Soft Computing*, vol. 25, pp. 1–13, 2021.
- [4] M. Simunek and Z. Smutny, "Traffic information enrichment: creating long-term traffic speed prediction ensemble model for better navigation through waypoints," *Applied Sciences*, vol. 11, p. 315, 2021.
- [5] N. Chiabaut and R. Faitout, "Traffic congestion and travel time prediction based on historical congestion maps and identification of consensual days," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102920, 2021.
- [6] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.
- [7] D. Yan, J. Zhou, Y. Zhao, and B. Wu, "Short-term subway passenger flow prediction based on ARIMA," in *Proceedings of the International Conference on Geo-Spatial Knowledge and Intelligence*, pp. 464–479, Chiang Mai, Thailand, December 2017.
- [8] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, vol. 50, pp. 148–163, 2018.
- [9] J. Barros, M. Araujo, and R. J. Rossetti, "Short-term Real-Time Traffic Prediction Methods: A Survey," in *Proceedings of the 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 132–139, IEEE, Budapest, Hungary, June 2015.
- [10] R. More, A. Mugal, S. Rajgure, R. B. Adhao, and V. K. Pachghare, "Road traffic prediction and congestion control using Artificial Neural Networks," in *Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pp. 52–57, IEEE, Pune, India, December 2016.
- [11] C. Luo, C. Huang, J. Cao et al., "Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm," *Neural Processing Letters*, vol. 50, no. 3, pp. 2305–2322, 2019.
- [12] W. Alajali, W. Zhou, S. Wen, and Y. Wang, "Intersection traffic prediction using decision tree models," *Symmetry*, vol. 10, no. 9, p. 386, 2018.
- [13] C. Ma, L. Tan, and X. Xu, "Short-term traffic flow prediction based on genetic artificial neural network and exponential smoothing," *Promet - Traffic & Transportation*, vol. 32, no. 6, pp. 747–760, 2020a.
- [14] S. Lu, Q. Zhang, G. Chen, and D. Seng, "A combined method for short-term traffic flow prediction based on recurrent neural network," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 87–94, 2021.
- [15] M. Farahani, M. Farahani, M. Manthouri, and O. Kaynak, "Short-term Traffic Flow Prediction Using Variational LSTM Networks," 2020, <https://arxiv.org/abs/2002.07922>.
- [16] X. Chen, "Research on short-term traffic flow forecasting model based on LSTM," *World Scientific Research Journal*, vol. 6, pp. 191–200, 2020a.
- [17] W. Wang, H. Zhang, T. Li et al., "An interpretable model for short term traffic flow prediction," *Mathematics and Computers in Simulation*, vol. 171, pp. 264–278, 2020.
- [18] D.-W. Xu, Y.-D. Wang, L.-M. Jia, Y. Qin, and H.-H. Dong, "Real-time road traffic state prediction based on ARIMA and Kalman filter," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 2, pp. 287–302, 2017.
- [19] L. Zheng, H. Huang, C. Zhu, and K. Zhang, "A tensor-based K-nearest neighbors method for traffic speed prediction under data missing," *Transportation Business: Transport Dynamics*, vol. 8, no. 1, pp. 182–199, 2020.
- [20] X. Li, T. Pang, B. Xiong, W. Liu, P. Liang, and T. Wang, "Convolutional Neural Networks Based Transfer Learning for Diabetic Retinopathy Fundus Image classification," in *Proceedings of the 2017 10th International congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI)*, pp. 1–11, Shanghai, China, October 2017.
- [21] H. Yang, Y. Zou, Z. Wang, and B. Wu, "A hybrid method for short-term freeway travel time prediction based on wavelet neural network and Markov chain," *Canadian Journal of Civil Engineering*, vol. 45, no. 2, pp. 77–86, 2018.
- [22] M. H. D. M. Ribeiro and L. Dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Applied Soft Computing*, vol. 86, Article ID 105837, 2020.



- [23] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.
- [24] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 308–324, 2015.
- [25] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter-and intra-day traffic patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2627–2636, 2020b.
- [26] F. Lin, J. Jiang, J. Fan, and S. Wang, "A stacking model for variation prediction of public bicycle traffic flow," *Intelligent Data Analysis*, vol. 22, no. 4, pp. 911–933, 2018.
- [27] A. Rasaizadi, A. Ardestani, and S. Seyedabrishami, "Traffic management via traffic parameters prediction by using machine learning algorithms," *International Journal of Human Capital in Urban Management*, vol. 6, pp. 57–68, 2021a.
- [28] A. Rasaizadi, S. Seyedabrishami, and M. Saniee Abadeh, "Short-term prediction of traffic state for a rural road applying ensemble learning process," *Journal of Advanced Transportation*, vol. 2021, Article ID 3334810, 14 pages, 2021b.
- [29] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, "kNN algorithm with data-driven k value," in *Proceedings of the International Conference on Advanced Data Mining and Applications*, pp. 499–512, Guilin, China, December 2014.
- [30] A. Rasaizadi, E. Sherafat, and S. Seyedabrishami, "Short-term Prediction of Traffic State," *Statistical Approach versus Machine Learning Approach*, 2021c.
- [31] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.
- [32] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," 2014, <https://arxiv.org/abs/1409.2329>.
- [33] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.
- [34] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, 818 pages, 2017.
- [35] Y. Chen, "Short-Term Traffic Flow Prediction Based on Multi-Model by Stacking Ensemble Learning," in *Proceedings of the 20th COTA International Conference of Transportation Professionals*, Xi'an, China, December 2020.
- [36] A. Rasaizadi, I. Farzin, and F. Hafizi, "Machine learning approach versus probabilistic approach to model the departure time of non-mandatory trips," *Physica A: Statistical Mechanics and Its Applications*, vol. 586, Article ID 126492, 2022.

## Research Article

# Resilience of Urban Road Network to Malignant Traffic Accidents

Yiding Lu <sup>1</sup>, Zhan Zhang <sup>2</sup>, Xinyi Fang,<sup>1</sup> Linjie Gao,<sup>1</sup> and Linjun Lu<sup>1</sup>

<sup>1</sup>Department of Traffic Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>2</sup>School of Design, Shanghai Jiao Tong University, Shanghai 200240, China

Correspondence should be addressed to Zhan Zhang; zhanzhang@sjtu.edu.cn

Received 7 February 2022; Revised 30 March 2022; Accepted 18 April 2022; Published 6 May 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Yiding Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Malignant traffic accidents are typical devastating events suffered by the urban road network. They cause severe functional loss when loading on the urban road network is high, exerting a significant impact on the operation of the city. The resilience of a road network refers to its ability to maintain a certain level of capacity and service when disturbed by external factors and to recover after a disturbance event, which is a crucial factor in the construction of transportation infrastructure systems. A comprehensive understanding of the adverse effects of malignant traffic accidents on the urban road network is imperative, and resilience is a concept employed to systematically explain this. This study investigates the impact of malignant traffic accidents on the resilience of the urban road network. A simulation is carried out focusing on an ideal urban road network, describing the temporal and spatial distribution of the average speed of road sections in the network. Inspired by the simulation experiment results, the ideal resilience curve is summarized, and the theory of resilience concept portrayal is innovatively developed into “6R” (redundancy, reduction, robustness, recovery, reinforcement, and rapidity). Combining the topological and “6R” resilience attributes of the urban road network, the urban road network resilience evaluation system is constructed, which yields an all-round and full-process evaluation for the urban road network with malignant traffic accidents. Results show that under malignant traffic accidents, the resilience of high-class surface roads, such as primary roads, is the poorest, suggesting that more attention and resources must be devoted to high-class surface roads. This study on the urban road network deepens the understanding and portrayal of its resilience and proposes an evaluation method to analyze its performance under disruption events.

## 1. Introduction

The continuous development of the urban road network has significantly boosted the construction of transportation infrastructure systems, and at the same time, it has inevitably caused numerous traffic problems. The occurrence of unanticipated disruptive events often has serious consequences [1, 2]. Malignant traffic accidents are typical devastating events, which often cause loss of life and property. Although malignant traffic accidents are not very common, once they occur, they provoke considerable adverse impacts. For example, on January 16, 2014, a large traffic accident occurred on China's Beijing-Shenyang Expressway. More than one hundred vehicles collided, and a flour truck exploded into flames, killing two people, injuring five others, and damaging forty-five vehicles. With the growing development of

the urban road network, the impact of malignant traffic accidents on it is attracting increasing interest.

Resilience originally refers to the ability of a material to absorb energy during plastic deformation and rupture. Holling [3] introduced the concept of resilience to the field of ecology in 1973. Since, the concept has been gradually extended to various fields, such as socio-economy and engineering [4, 5]. Hansen and Sutter's [6] study of the effects of road closures caused by the Loma Prieta earthquake initiated the research field of road traffic system resilience. In recent years in the field of transportation, resilience has become a hot research topic. The urban road network forms the basic network for urban transportation activities. The resilience of the urban road network refers to the ability to maintain a certain level of capacity and service when disturbed by external factors and to recover after a disturbance

event. Research on the resilience of the urban road network helps us better manage the operation of urban road traffic.

The urban road network with better resilience has a stronger ability to resist the adverse impacts of malignant traffic accidents and quickly recover from them. Improving the ability to respond to and handle malignant traffic accidents enhances the resilience of the urban road network, and in turn, developing the portrayal of the resilience concept helps us better understand the mechanism of malignant traffic accidents. The impact of malignant traffic accidents on the resilience of the urban road network is a problem that involves both temporal and spatial dimensions. Its propagation mechanism is difficult to capture, and the spatiotemporal mechanism is difficult to portray, which reveals the lack of research about the impact of malignant traffic accidents on the urban road network. Furthermore, it is difficult to make an all-round and full-process evaluation of the resilience of the urban road network with malignant traffic accidents. Therefore, the study of the impact of malignant traffic accidents on the resilience of the urban road network is extremely necessary.

This study combines traffic simulation technology with the study of the resilience of the urban road network. Focusing on an ideal urban road network, the entire process is simulated from before the occurrence of malignant traffic accidents to after the implemented recovery measures, obtaining temporal and spatial distributions of the average speed of road sections in the network. Inspired by the results of simulation experiments, the ideal resilience curve is obtained, and the theory of resilience concept portrayal is innovatively developed. Based on the topological and resilience attributes of the urban road network, the resilience evaluation system is established, yielding a full-process evaluation of the resilience under malignant traffic accidents. From the perspective of the government, based on the spreading characteristics of the congestion phenomenon, corresponding policy suggestions are made to improve the resilience of the urban road network when facing malignant traffic accidents.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature and highlights the identified research gaps. The parameters and flow of the simulation experiments are described in detail in Section 3. Section 4 shows and analyses the results of the simulation experiment and proposes the ideal resilience curve. Section 5 innovatively discusses the resilience evaluation system and provides corresponding policy suggestions for the urban road network from the perspective of the government. Finally, conclusions are presented in Section 6. The structure of the paper is shown in Figure 1.

## 2. Literature Review

*2.1. Impact of Emergency Events on Traffic Networks.* Research on the impact of emergencies on urban road networks have attracted the interest and attention of researchers in the past decade. The day-long traffic jams in New York City after 9/11 caused massive disruptions in the city's transportation network, demonstrating its vulnerabilities and their negative consequences and prompting research on this topic.

Berdica [7] was the first to define the degree of impact on the traffic network of such sudden events that lead to a significant reduction in the network service capacity as the vulnerability of the traffic network. Vulnerability focuses on the weaknesses of the network and the consequences of failure. According to [8–10], the vulnerability of urban road networks could be viewed in a similar manner to measure the risk. The concept of vulnerability can be separated into two parts, using both the product of the probability of an event occurring and the outcome. Murray et al. [11] classified the current methods for assessing disturbance events on road networks into four broad categories: the scenario-specific, attack strategy-specific, simulation, and mathematical model.

The scenario-specific assessment method examines the impact on the road network in the case of a specific edge or node disruption and evaluates its possible consequences [12–15]. The attack strategy-specific assessment method involves the study of the sensitivity of different network structures to external stimuli and attacks. The method sets up an attack strategy and then applies statistical methods to study the vulnerability of the infrastructure network [16, 17]. The simulation assessment method analyses the factors affecting network performance using simulation software to simulate macro or micro road networks, such that the consequences of the impact can be evaluated [7, 18]. The mathematical model assessment method focuses on the methodology and model and uses mathematical expressions or models to determine the consequences of network unit failure [19].

The above four methods do not conflict with each other and can be used to study the impact of unexpected events on the road network, identify critical points and key locations, and assess the vulnerability of the network. In practice, several of these methods can be used in combination.

The emergency event that is the focus of current studies is a broad and abstract concept. It has not been refined to specific events. Furthermore, because the consequences of malignant traffic accidents are generally measured in terms of human or property damage, there are little data about the impact of malignant traffic accidents on the urban road network. Moreover, few studies have been conducted on the impact of malignant traffic accidents on the urban road network.

*2.2. Resilience of Road Traffic System.* The resilience of the road traffic system refers to the ability of the road traffic system to maintain a certain capacity and level of service when disturbed by external factors and to recover after a disturbance event. Research related to road traffic system resilience began in 1990 with Hansen and Sutter's [6] study of the effects of road closures caused by the Loma Prieta earthquake. This study inaugurated the field of road traffic system resilience research. Research on traffic system resilience has gone through three main stages: the stage of conceptual framework research, the stage of metrics, and the stage of quantitative evaluation.

In the stage of conceptual framework research, Bruneau and Chang [20] proposed the classical conceptual

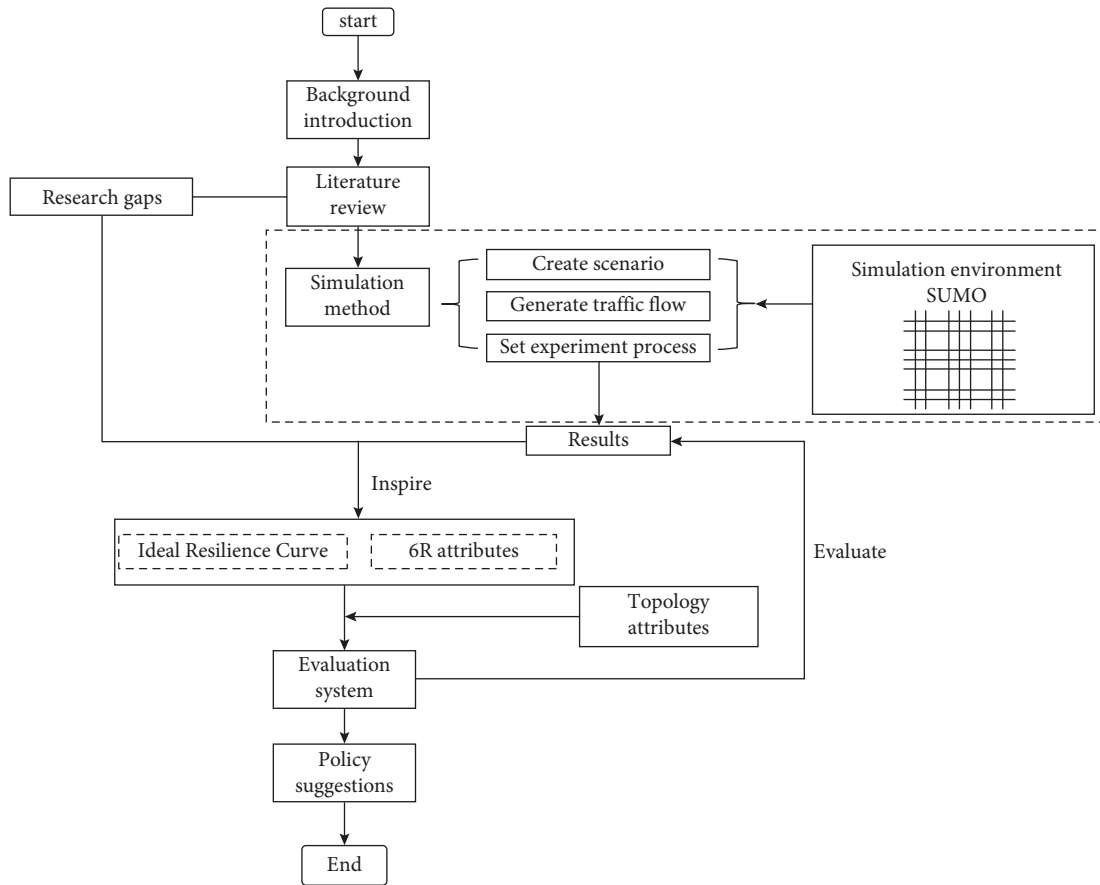


FIGURE 1: Structure of the paper.

framework of resilience, which is defined as the ability of a system to mitigate the effects of a disaster and maintain its own function. This was measured by a resilience triangle consisting of robustness, rapidity, redundancy, and resourcefulness. Cutter and Barnes [21] defined resilience as the ability of a system to cope with and recover from a disaster, including the ability of the system to absorb and resist its adverse effects. Mattsson and Jenelius [22] argued that the concept of resilience aimed to capture the ability to maintain its functionality and the speed to return to a normal state after a large-scale disruption or disaster. According to Qiliang et al. [23], resilient city construction requires that cities not only have the ability to absorb disaster disturbances by engineering measures but also have the ability of self-adaptation and rapid recovery. Based on the existing research results, three consensus points of the concept of road traffic system resilience are summarized. First, resilience refers to the ability of the system to withstand, respond to, and recover from disasters, rather than a state or outcome of the system. Second, resilience emphasizes the adaptability of the system in the face of disasters rather than stability. Third, resilience emphasizes the capacity of the system throughout the process of disaster occurrence and encompasses numerous aspects, rather than one attribute.

In the stage of metrics, there has been significant growth in the metric study of road traffic system resilience in recent

years. Faturechi and Miller-Hooks [24] provided a comprehensive overview of the performance of transportation infrastructure systems in disasters and found that resilience was often measured by characteristics such as risk, vulnerability, reliability, robustness, mobility, and survivability. Murray-Tuite [25] classified traffic resilience into ten metrics: cooperation, redundancy, diversity, efficiency, safety, self-organization, strength, adaptability, mobility, and ability to recover quickly. In the stage of quantitative evaluation, Bruneau and Chang's research [20] on resilience provided a solid basis for quantitative evaluation of the resilience of urban road traffic systems. Ip and Wang [26] defined the independent paths between every two nodes of an urban road traffic network and measured the resilience of the road network in terms of the number of independent paths across the whole network. Wang et al. [27] introduced the degree of nodes as a measure of node resilience with the help of the entropy method in physics. Furthermore, some researchers use mathematic models to perform the evaluation and optimization work in transportation systems from the perspective of resilience [28, 29].

Notably, most current research on resilience involves static evaluation studies, focusing on proposing a metric system and conducting a static evaluation for an urban road system under a certain state. Few studies have been performed on the whole dynamic process of perturbation events.

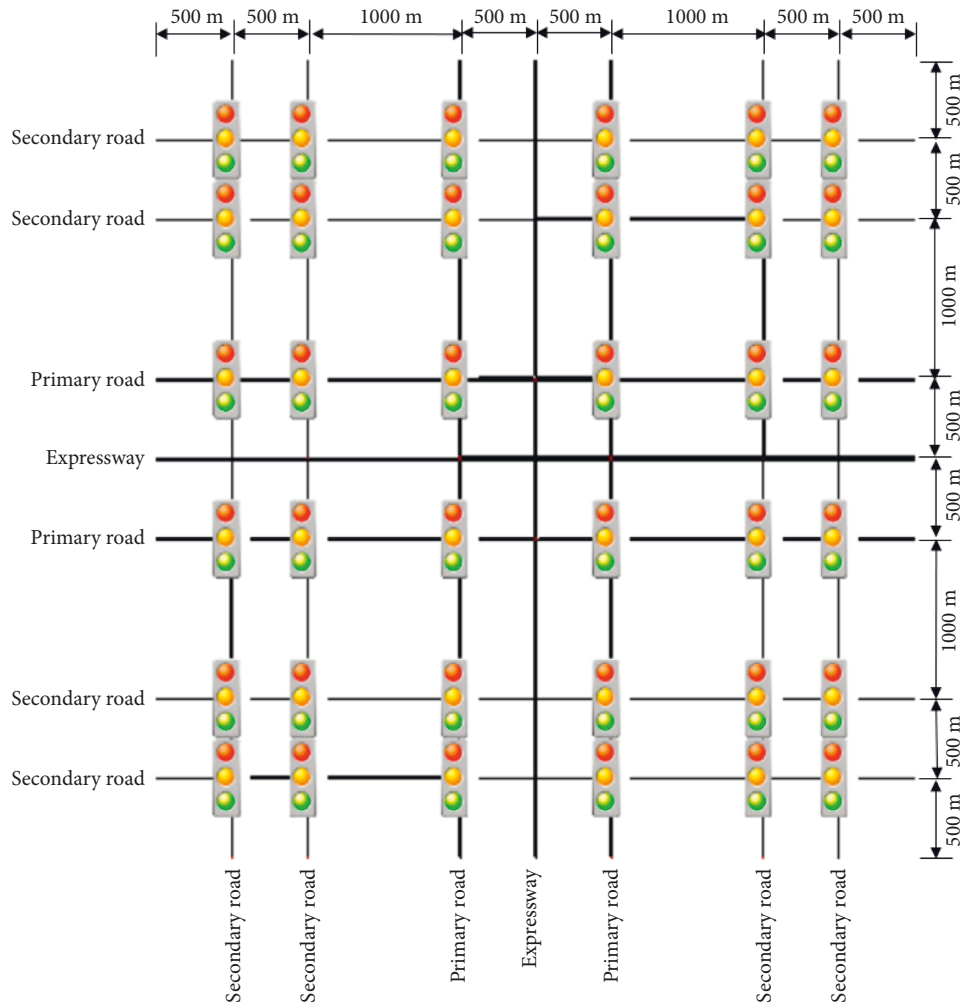


FIGURE 2: Ideal urban road network in SUMO.

To solve the research gaps mentioned above, this study specifies the unexpected event as malignant traffic accidents and uses traffic simulation technology to obtain data about the impact of malignant traffic accidents on the road network. The dynamic impact on the urban road network resilience of the whole process from before the occurrence of malignant traffic accidents to after the recovery measures have been implemented is explored. Based on the topological attributes and resilience attributes of the urban road network, the urban road network resilience evaluation system is established, which renders a full-process evaluation of the resilience of the urban road network under malignant traffic accidents conditions.

### 3. Simulation Method

In this study, SUMO (Simulation of Urban MObility), a microscopic traffic simulation software, was used to conduct simulation experiments. SUMO is an open-source traffic simulation software that enables the control of traffic flow.

In SUMO, we draw an ideal urban road network with a scale of  $5 \times 5$  km, including three types of urban roads, namely secondary roads, primary roads, and expressways.

The three types of urban roads have a regular distribution, as shown in Figure 2. Each intersection on secondary and primary roads is a level intersection with traffic signals within it. The expressways intersection is treated as a separate grade to simulate three-dimensional crossings in real scenarios. Connections are set between primary roads and expressways to simulate ramps.

As shown in Figure 2, there are eight secondary roads, four primary roads, and two expressways in the ideal urban road network. An ideal traffic flow with a uniform and symmetrical distribution of vehicles was added to these fourteen roads to make them operate in the network. The specific traffic flow setting is as follows. Each of the twenty-eight endpoints of the fourteen roads is used as the origin of the traffic flow. Each origin corresponds to all twenty-eight endpoints, that is, the twenty-eight endpoints are the destinations of this origin. The interval of traffic generation and traffic flow volume is well designed and adjusted according to pre-repeated experiments such that the traffic generated in the network and the arriving traffic are basically in a level state, making it possible for the traffic to be evenly and symmetrically distributed in the network in a stable manner. We set the volume based on this principle: the traffic runs

TABLE 1: Parameters and model settings of simulation experiment.

Parameter/model	Scenario	Description
Number of lanes	Secondary roads	Two lanes in both directions
	Primary roads	Four lanes in both directions
	Expressways	Six lanes in both directions
Speed limit	Secondary roads	60 km/h
	Primary roads	60 km/h
	Expressways	100 km/h
Traffic volume between each OD (origin-destination) pair	All	24 vehicles
Car-following model	All	Krauss
Lane-changing model	All	LC2013

smoothly in normal conditions, and when malignant traffic accidents happen, there will occur congestion.

Some specific parameters and model settings of the simulation experiment are shown in Table 1.

In the simulation experiment, there are three scenarios where malignant traffic accidents occur: at secondary and primary roads intersections and at the expressways intersection. The process of the simulation experiment is described as follows.

During the first 30 min, the traffic flow is gradually distributed to the whole network, reaching a balanced and stable state. According to our repeated tests, it takes more than 20 minutes for the traffic flow to spread throughout the network and reach a stable running state. In order not to interfere with the later experiments, the traffic flow runs freely for the first 30 minutes of the simulation. At 30 min, malignant traffic accidents occur at an intersection, causing damage to the intersection and making it impassable. Vehicles are stalled on the roads connected to the intersection, but other road users in the network are unaware of the malignant traffic accidents and continue to follow the originally planned path, resulting in gradual congestion on the roads connected to the intersection where the accidents occur. According to daily experience, the road repair department needs some time to arrive at the scene of the accident, and it also takes some time to notify other road users of the accident through the news and radio broadcast. Here, we set the time interval as 15 minutes. So, at 45 min, the roads connected with the intersection are closed for road repair. At the same time, initial recovery measures are taken. There are two initial recovery measures. First, when the road is closed, the original signalized intersection becomes a T-intersection. The capacity of the intersection can be optimized by extending the green signal ratio on the long side of the T-intersection and shortening the green signal ratio on the short side. Second, repairing or opening emergency lanes on roads around the disrupted intersection improves the capacity. In the simulation, each vehicle in the network chooses the path with the shortest total travel time at every iteration. After the roads connected to the intersection are closed, each vehicle will reroute between the OD and choose the route with the shortest travel time among all the remaining alternative routes according to the calculation results at every iteration. Statistics show that the average time to deal with a traffic accident in the city is about 30

minutes. Therefore, we set the time for the roads closed for emergency repairs as 30 minutes. At 75 min, the accidents are handled, the road repair is completed; the intersection damaged by the accident resumes traffic; and the traffic light phase returns to normal. To relieve the congestion on the surrounding roads due to the malignant traffic accident, the emergency lane remains open. After the accidents are handled, it takes another 30 minutes for the traffic flow to reach a new stable state, so the simulation is allowed to run for another 30 min to reach a new equilibrium stable state, after which it ends. The flow of the whole simulation experiment is shown in Figure 3.

## 4. Results and Analysis

From the SUMO data output, we select the average speed of each road section in the network during each one-minute period as a measure of its function. The ideal road network in SUMO is a directed graph. To facilitate data visualization, we process it to be an undirected graph. The average speed of road sections closed due to malignant traffic accidents is set to zero, and the average of the road sections' speed in two directions is calculated. Then, we obtain the average speed of each road section in the road network per minute (undirected graph).

*4.1. Time Distribution.* With the horizontal axis depicting time and the vertical axis depicting the network average speed, we plot a graph of network average speed changing with time. Then, we use a polynomial to fit the curve and obtain the time distribution of network average speed for the whole process under the three accident scenarios.

The temporal distribution of the network average speed under three accident scenarios is shown in Figure 4. Figure 4(a) represents the temporal distribution of the network average speed when malignant traffic accidents occur at the secondary roads intersection. Figure 4(b) shows the temporal distribution of the network average speed when malignant traffic accidents occur at the primary roads intersection. Figure 4(c) shows the temporal distribution of the network average speed when malignant traffic accidents occur at the expressways intersection. The blue line in Figure 4 depicts the processed simulation experiment data, and the red dashed line depicts the polynomial fitted curve.

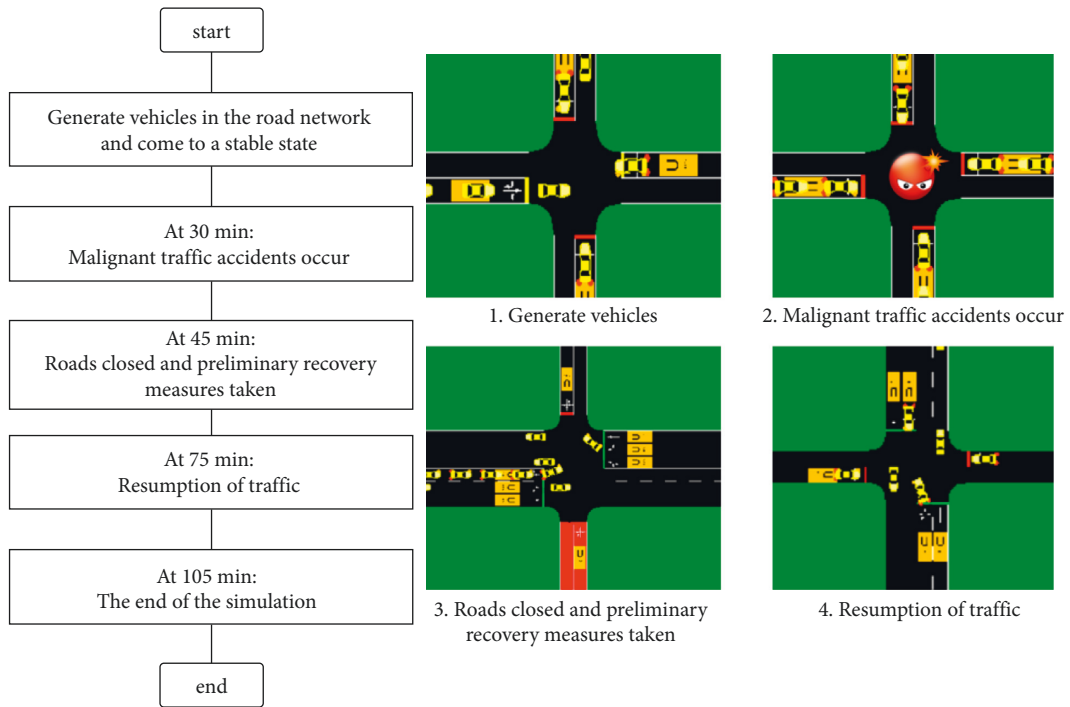


FIGURE 3: Flow of simulation experiment.

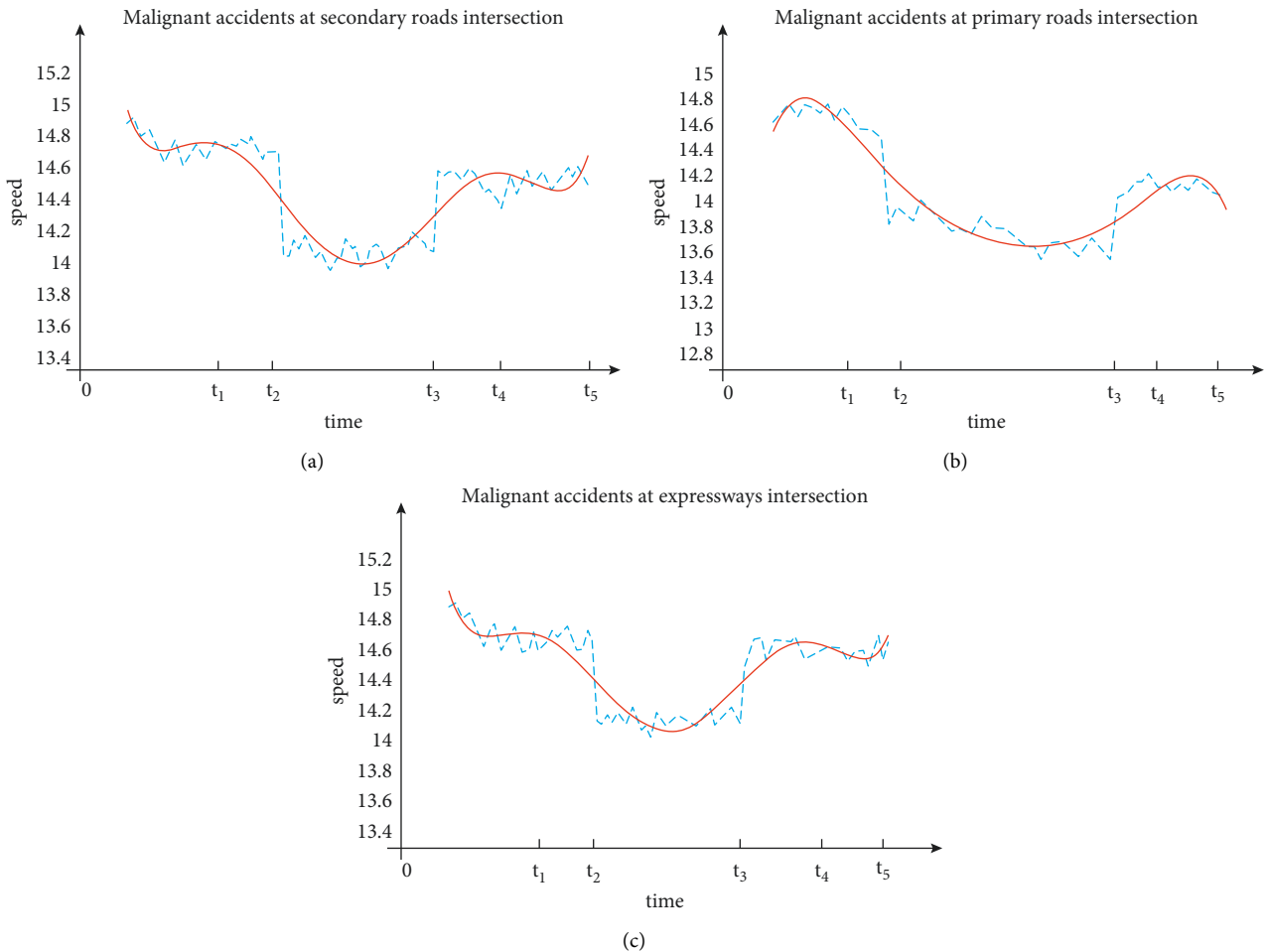


FIGURE 4: Time distribution of network average speed with malignant accidents at (a) secondary roads intersection, (b) primary roads intersection, and (c) expressways intersection.

Figure 4 shows that the time distribution of the network average speed can be divided into five periods: the initial stable period ( $0-t_1$ ), disruption period ( $t_1-t_2$ ), stable period after disruption ( $t_2-t_3$ ), recovery period ( $t_3-t_4$ ), and stable period after recovery ( $t_4-t_5$ ). Before the accidents (initial stable period), the entire network is in a normal operating state, and the network average speed is generally stable with slight fluctuations. After malignant traffic accidents occur (disruption period), the network average speed gradually decreases. Dropping to the lowest value (stable period after a disruption), it reaches a stable period with a low service level, and the network average speed is at a low, slightly fluctuating stable period. After implementing the recovery measures (recovery period), the network average speed gradually increases. A new stable period (stable period after recovery) is reached after the network average speed rises to a certain value. Because we simulate a short-term recovery, the network average speed in the stable period after recovery is lower than that in the initial stable stage.

**4.2. Spatial Distribution.** Each road section is assigned a color according to its average speed, such that the spatial distribution of the average speed of the whole network can be plotted.

We select the spatial distribution graphs of the average speed of each road section at a representative moment of each period and plot them on the time axis. Thus, the spatial distribution pattern of road sections' average speed over time is revealed, as shown in Figure 5.

Figures 5(a)–5(c) show the spatial distribution of the average speed of road sections over time under the scenarios of malignant traffic accidents occurring at the secondary roads, primary roads, and expressways intersections, respectively.  $t_1$  denotes a certain moment in the initial stable period. The traffic runs smoothly throughout the network, and the speed of the expressways is higher than that of the surface roads. Due to the phase of the traffic lights, the average speed on the road sections near the signalized intersection is slightly lower.  $t_2$  is a certain moment in the disruption period. Malignant traffic accidents occur at a certain intersection (circled part in Figures 5(a)–5(c)). Herein, we define the road sections whose average speed is lower than 80% of the speed limit, that is, 13 m/s, as congested road sections. At this time, traffic on the four road sections connected to the intersection generates congestion, whereas the rest of the network is not affected.  $t_3$  denotes a certain moment in the stable period after a disruption. When malignant traffic accidents occur at the secondary roads intersection, the roads connected to the intersection close. Because traffic volume is not very high on secondary roads, the traffic congestion only gradually spreads to several roads around this intersection. When malignant traffic accidents occur at the primary roads intersection, the roads connected to the intersection are closed, and the traffic congestion gradually spreads along with the horizontal and vertical directions northwest and southeast of the network. The entire network is at a low level of service. When malignant traffic accidents occur at the expressways intersection, the

roads connected to the intersection close. However, because expressways are equipped with ramps, their good connectivity with primary roads stops the traffic congestion phenomenon from spreading, making the congestion only generates on the roads connected to the intersection.  $t_4$  denotes a certain moment in the recovery period. Owing to the completion of road repair, the intersection suffering malignant traffic accidents resumes traffic, and the congestion on the connected and impacted roads is eased. However, some impact on the network remains.  $t_5$  marks a certain moment in the stable period after recovery. The entire network enters a new state of stable operation, but the overall level of service is slightly lower than that in the initial stable stage.

**4.3. Ideal Resilience Curve.** Using the network average speed per minute as a functional measure of the road network resilience, we obtain the time distribution of road network resilience in the whole process of malignant traffic accidents. The ideal resilience curve of the road network is proposed according to the time distribution of network average speed, which is shown above. Based on the classical “4R” resilience theory [20], the conceptual portrayal of resilience is further extended to “6R.”

The ideal curve of road network function with respect to time is shown in Figure 6. The horizontal axis represents time, and the vertical axis represents the functional attributes that can characterize the resilience of the road network. The ideal resilience curve can be divided into five periods: initial stable period, disruption period, stable period after disruption, recovery period, and stable period after recovery. Before the malignant traffic accidents, the road network is in a stable state with slight fluctuations. After the malignant traffic accidents, the functional level of the road network gradually decreases and then enters a lower functional level with slight fluctuations. After the recovery measures are taken, the functional level of the road network gradually rises and then enters a new stable state with slight fluctuations. Notably, the duration of the three stable periods can be extended or shortened, mainly depending on when malignant traffic accidents happen and when recovery measures are taken.

Compared with the resilience curve proposed by Bruneau and Chang [20], the ideal resilience curve presented here has several differences. First, we believe that the three stable periods are not completely stable states but stable states with slight fluctuations. Second, we suggest that the damage caused by malignant traffic accidents to the function of the road network is not instantaneous but takes some time to gradually impose on the road network. Therefore, the road network function does not decline instantaneously when the damage occurs but experiences a gradual decline process. Third, we suggest that the road network function does not recover immediately after the disruption but enters a stable state with a low level of service. It gradually recovers only after the intervention of recovery measures. Fourth, in the ideal resilience curve, we describe a short-term recovery such that the functional level of the road network after recovery is lower than that in the initial stable period.



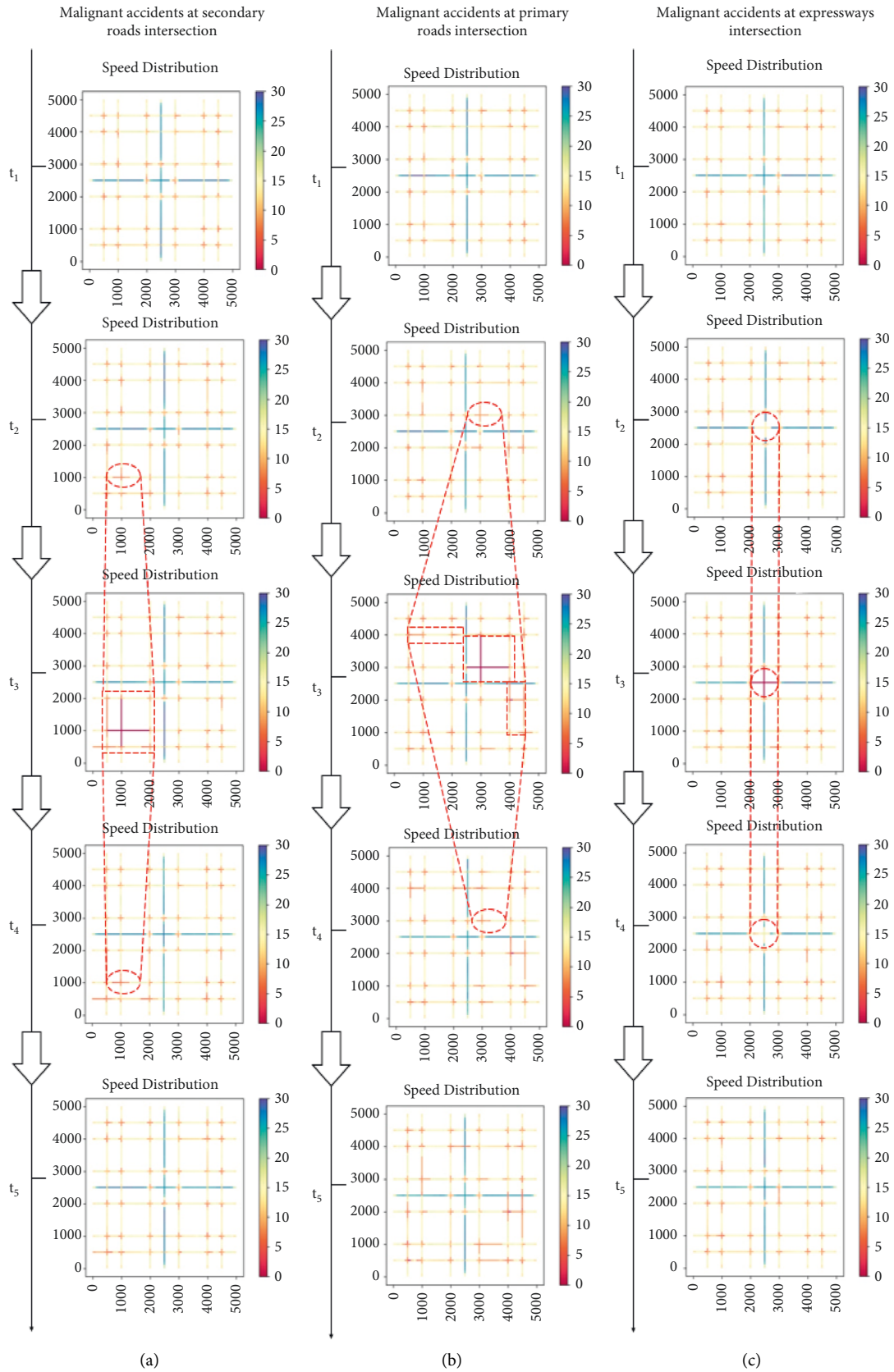


FIGURE 5: Spatial distribution of average speed with malignant accidents at (a) secondary roads intersection, (b) primary roads intersection, and (c) expressways intersection.

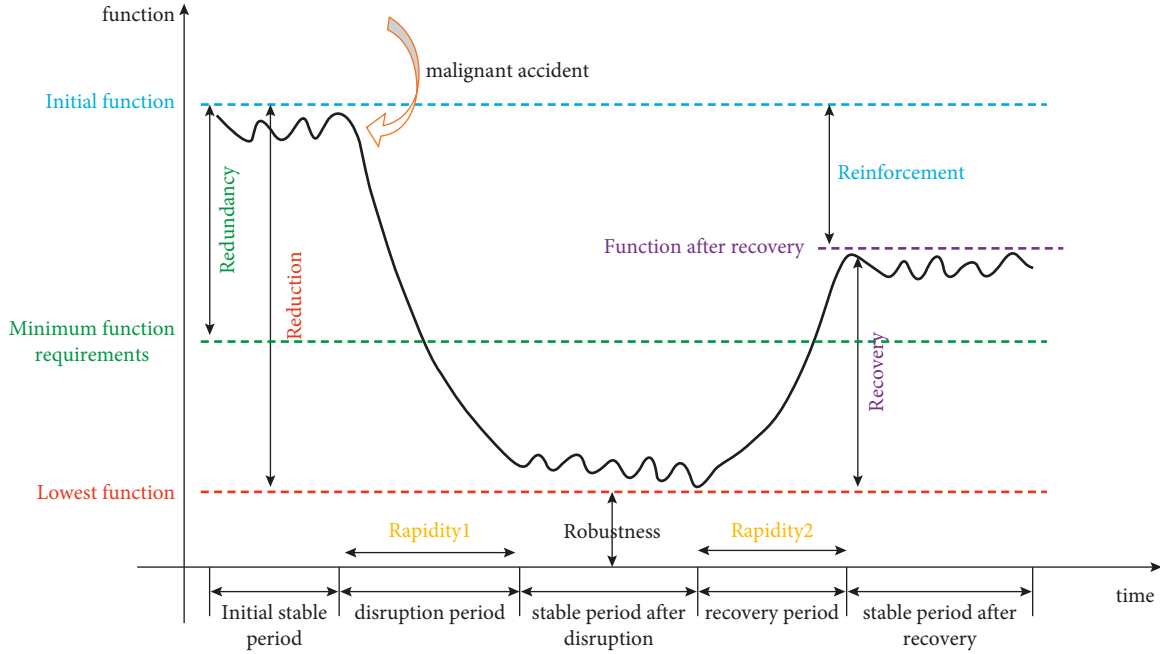


FIGURE 6: Ideal resilience curve.

After deriving the ideal resilience curve, we can define relevant “6R” resilience attributes. There are four baselines in Figure 6. The blue dashed line indicates the initial function; the green dashed line indicates the minimum functional requirements; the red dashed line indicates the lowest function, and the purple dashed line indicates the function after recovery. Redundancy refers to the difference between the initial function and the minimum function requirements, reflecting the part of the initial function that is higher than the requirement. Reduction refers to the difference between the initial function and the lowest function, indicating the loss of the road network’s function due to the damage. Robustness refers to the lowest function, indicating the remaining function of the road network after the damage. Recovery refers to the difference between the function after recovery and the lowest function, indicating the degree of road network recovery. Reinforcement refers to the difference between the initial function and the function after recovery, indicating the lack of function in the stable period after recovery compared to the initial stable period. Rapidity refers to the rate at which the function level of the road network decreases and increases. To facilitate the calculation and subsequent use, we conduct normalization for the “6R” resilience attributes. The calculation of the “6R” resilience attributes is shown in the following equations:

$$\text{Redundancy} = \frac{IF - MFR}{IF}, \quad (1)$$

$$\text{Reduction} = \frac{IF - LF}{IF}, \quad (2)$$

$$\text{Robustness} = \frac{LF}{IF}, \quad (3)$$

$$\text{Recovery} = \frac{FAR - LF}{IF}, \quad (4)$$

$$\text{Reinforcement} = \frac{IF - FAR}{IF}, \quad (5)$$

$$\text{Rapidity1} = \frac{ToD}{TT}, \quad (6)$$

$$\text{Rapidity2} = \frac{ToR}{TT}, \quad (7)$$

where IF denotes the initial function, depicted by the blue dashed line in Figure 6; MFR denotes the minimum function requirements, depicted by the green dashed line; LF denotes the lowest function, depicted by the red dashed line; FAR denotes the function after recovery, depicted by the purple dashed line; ToD denotes the time of disruption period; ToR denotes the time of recovery period; and TT denotes the total time of the whole process of malignant traffic accidents.

Among the “6R” attributes, redundancy, robustness, recovery, and rapidity1 are positively correlated with resilience, that is, the larger the Rs the better the resilience. Reduction, reinforcement, and rapidity2 are negatively correlated with resilience, that is, smaller Rs indicate better resilience. The “6R” attributes together reflect the resilience of the road network under the scenarios of malignant traffic accidents.

Based on the results obtained from the simulation experiments, we can calculate the values of “6R” under three scenarios of malignant traffic accidents, as shown in Table 2.

The three classes of roads have their own advantages and disadvantages in terms of “6R” attributes under the scenario of malignant traffic accidents. According to the correlation between “6R” and resilience, three classes of roads intersections

TABLE 2: Values of “6R” under three scenarios of malignant traffic accidents.

	Secondary roads intersection (%)	Primary roads intersection (%)	Expressways intersection (%)
Redundancy	6.10	5.18	6.10
Reduction	6.41	8.32	5.98
Robustness	93.59	91.68	94.02
Recovery	4.37	4.60	4.47
Reinforcement	2.04	3.72	1.51
Rapidity1	7.77	6.67	4.44
Rapidity2	7.77	8.33	11.11

under the condition of malignant traffic accidents are ranked based on the simulation experiment data. The higher the ranking, the higher the score, indicating better resilience performance under a certain R attribute. The radar diagram shows the resilience performance of three classes of roads intersections under the “6R” attribute in the event of malignant traffic accidents, as shown in Figure 7. The secondary roads intersection performs best in terms of redundancy and rapidity and worst in terms of recovery. The primary roads intersection performs best in recovery and worst in redundancy, reduction, robustness, and reinforcement. The expressways intersection performs best in redundancy, reduction, robustness, and reinforcement and worst in rapidity.

## 5. Discussion

In this section, by combining the topological and resilience attributes of the urban road network, the evaluation system of urban road network resilience is constructed. Based on the spreading characteristics of the congestion phenomenon, from the perspective of government management, we propose relevant policy suggestions about how the urban road network resists and responds to malignant traffic accidents.

**5.1. Evaluation System.** Combining the topological and “6R” resilience attributes of the urban road network, its resilience evaluation system can be established.

Among the topological attributes, degree centrality, closeness centrality, and betweenness centrality, which characterize the importance of node locations, are selected as the topological evaluation metrics of the network. Degree centrality is the most direct metric to portray the centrality of a node. A large degree of a node indicates a higher centrality. In the ideal urban road network, the degree centrality of a node is characterized by the number of lanes of one road connected to it. Closeness centrality indicates the closeness between a node and other nodes in the network. The inverse of the sum of the shortest path distance from a node to all other nodes indicates closeness centrality. Betweenness centrality is the sum of the ratio of the number of times a node lies on the shortest path between any two other nodes to the number of all paths between the two nodes. The calculations of the three topological attributes are shown in the following equations:

$$C_d(i) = \frac{d(i)}{n-1} * nl, \quad (8)$$

$$C_c(i) = \frac{n-1}{\sum_j d_{ij}}, \quad (9)$$

$$C_b(i) = \frac{2\sum_{j<k} g_{jk}(i)/g_{jk}}{(n-1)(n-2)}, \quad (10)$$

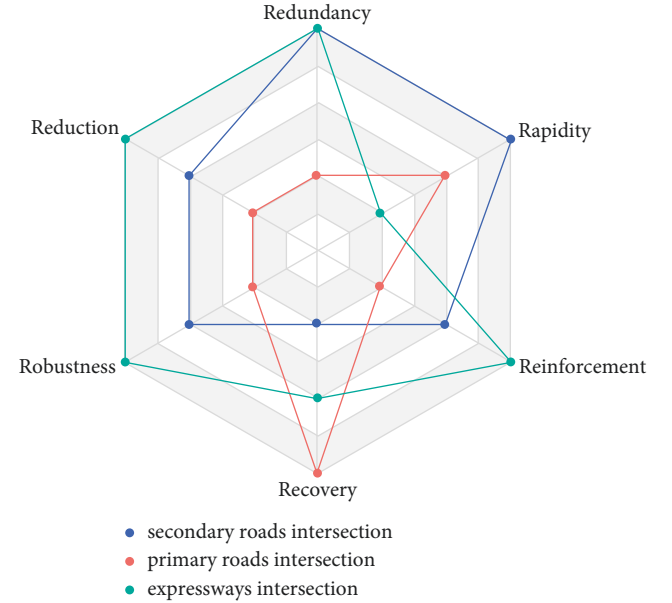


FIGURE 7: Radar diagram of “6R” resilience performance for three classes of roads.

where  $n$  is the number of nodes,  $C_d(i)$  is the degree centrality of node  $i$ ,  $d(i)$  is the degree of node  $i$ ,  $nl$  is the number of lanes on one road connected to node  $i$ ,  $C_c(i)$  is the closeness centrality of node  $i$ ,  $d_{ij}$  is the shortest path distance between node  $i$  and  $j$ ,  $C_b(i)$  is the betweenness centrality,  $g_{jk}$  is the number of shortest paths between node  $j$  and  $k$ , and  $g_{jk}(i)$  is the number of shortest paths between node  $j$  and  $k$  through node  $i$ .

Among “6R” resilience attributes, since the reduction is negatively correlated with robustness and reinforcement is negatively correlated with recovery, four attributes that are independent of each other are selected. Redundancy, robustness, recovery, and rapidity1, which are positively correlated with resilience, and rapidity2, which is negatively correlated with resilience, are selected as the resilient

TABLE 3: Resilience score for three scenarios of malignant traffic accidents.

	Secondary roads intersection	Primary roads intersection	Expressways intersection
Topology attributes	Degree centrality	0.006	0.012
	Closeness centrality	0.029	0.0398
	Betweenness centrality	0.0968	0.207
Resilience index	0.958	0.936	0.96
Resilience attributes	Redundancy	0.061	0.052
	Robustness	0.936	0.9168
	Recovery	0.0437	0.046
	Rapidity1	0.0777	0.0667
	Rapidity2	-0.0777	-0.833
Resilience score	2.54974282	1.027532398	2.685798309

evaluation metrics. The calculation of redundancy, robustness, recovery, rapidity1, and rapidity2 is depicted as equations (1), (3), (4), (6), and (7).

The resilience index refers to the integral of the resilience function curve over time during the whole process of malignant traffic accidents, reflecting the overall resilience performance of the urban road network under a certain damage scenario. The calculation of the resilience index is improved from the calculation of resilience loss in the literature [20]. The resilience index is positively related to resilience, as shown in the following equation:

$$RI = \frac{\int_0^T F(t)dt}{T * F_0}, \quad (11)$$

where  $RI$  is the resilience index,  $T$  is the duration of the whole process of malignant traffic accidents,  $F(t)$  is the resilience function curve with respect to time, and  $F_0$  is the initial function of the road network.

The resilience evaluation system includes the above-mentioned topological attributes, resilience attributes, and the resilience index. The resilience evaluation system is given in the form of a resilience score of the road network under a certain damage scenario, which is shown in the following equation:

$$RS = RI * e^{C_d(i)+C_c(i)-C_b(i)} * e^{\text{Redundancy}+\text{Robustness}+\text{Recovery}+\text{Rapidity1}-\text{Rapidity2}}, \quad (12)$$

where  $RS$  denotes the resilience score.

Equation (12) indicates that the road network resilience score is approximately between the interval  $(0, e^2)$ . A higher resilience score indicates better resilience performance of the road network. Theoretically, equation (12) can be used to evaluate the resilience of any road network under the occurrence of any event.

Based on the data obtained from simulation experiments, the resilience scores under three scenarios of malignant traffic accidents can be calculated, as shown in Table 3.

Table 3 indicates that under this evaluation system, the road network resilience is optimal when malignant traffic accidents occur at an expressways intersection, and the road network resilience is poorest when malignant traffic accidents occur at the intersection of the primary roads. The

main reason is that in this ideal urban road network, the betweenness centrality of primary roads intersection is relatively high, resulting in more vehicles passing through this intersection. Furthermore, this has a low resilience index and relatively poor resilience attributes, thus resulting in the lowest resilience score.

**5.2. Policy Suggestions.** According to “6R” resilience attributes and the resilience evaluation system, based on the spreading characteristics of the congestion phenomenon, we propose policy suggestions for the urban road network to resist and cope with malignant traffic accidents from the perspective of government managers.

First, the resilience performance of low-class surface roads, such as secondary roads, is good under the conditions of malignant traffic accidents. The congestion on low-class surface roads has a tendency to spread to surrounding roads in the event of malignant traffic accidents. Small-scale spreading must be prevented, and traffic control and congestion dispersal on surrounding roads must be strengthened. Consistently, the current level of service on low-class surface roads must be maintained and steadily improved.

Second, the resilience performance of high-class surface roads, such as primary roads, is poor under the conditions of malignant traffic accidents. The congestion on high-class surface roads has a tendency to spread to the entire network. Large-scale spreading must be prevented, and the traffic control and congestion dispersal on roads in horizontal and vertical directions, where the accidents occur, must be strengthened to prevent the proliferation of congestion. The investment of human and material resources for high-class surface roads must be increased; the traffic control must be strengthened; and the resistance and recovery ability of high-class surface roads against damage caused by malignant traffic accidents must be improved.

Finally, expressways, urban interchanges, elevated roads, and other high-class roads are more resilient under the conditions of malignant traffic accidents owing to good connectivity. There is no tendency for congestion on high-class roads to spread. However, the speed at which the high-class roads recover from malignant traffic accidents is relatively low. The efficiency of the handling of traffic accidents must be improved to accelerate the speed of recovery from malignant traffic accidents. The capacity construction must

be strengthened, enabling high-class roads to better withstand the impact of reduced capacity due to the occurrence of malignant traffic accidents. The current management efforts must be maintained without slackening, and the maintenance of high-class roads must be strengthened to ensure their connectivity and smoothness.

## 6. Conclusion

Focusing on an ideal urban road network, we conducted traffic simulations to evaluate the impact of malignant traffic accidents on the resilience of the urban road network. The simulation experiments simulate the whole process from before the occurrence of malignant traffic accidents to after the recovery and obtain the temporal and spatial distributions of the average speed of road sections in the road network. The results show that in terms of temporal distribution, the network average speed under the scenario of malignant traffic accidents goes through five periods: initial stable period, disruption period, stable period after disruption, recovery period, and stable period after recovery. In terms of spatial distribution, with the passage of time in five periods, congestion due to malignant traffic accidents first becomes aggravating, then gradually eases, and finally returns to normal. It has the tendency to spread outward during the disruption period.

Inspired by simulation experiments, the ideal resilience curve is summarized as follows. It can be divided into five periods: initial stable period, disruption period, stable period after disruption, recovery period, and stable period after recovery. Based on the classical "4R" resilience theory [20], we further refine it into the "6R" resilience theory, including redundancy, reduction, robustness, recovery, reinforcement, and rapidity, improving the portrayal of the resilience concept. The "6R" resilience attributes together reflect the resilience of the road network under malignant traffic accidents. Combining the topological attributes and resilience attributes of the urban road network, we establish an urban road network resilience evaluation system to obtain an all-round evaluation of the urban road network under malignant traffic accidents. Results show that when malignant traffic accidents occur at expressways intersection, the road network's resilience performance is optimal, and when malignant traffic accidents occur at primary roads intersection, the road network's resilience performance is the poorest. Therefore, more resources and attention must be devoted to high-class surface roads to help improve the resilience in dealing with malignant traffic accidents. Finally, from the perspective of government management, based on the spreading characteristics of the congestion phenomenon, we propose relevant policy suggestions for urban road networks to resist and respond to malignant traffic accidents.

The major contributions of this work are as follows. First, we propose the ideal resilience curve for the urban road network under the scenario of malignant traffic accidents to describe the time distribution of the road network resilience, which is more detailed compared with previous studies. We plot the spatial distribution of the average speed of the whole network and find the propagation characteristics of

congestion when malignant traffic accidents occur at different classes of intersections. Second, we develop the classical resilience theory into "6R" (redundancy, reduction, robustness, recovery, reinforcement, and rapidity), improving the theory of resilience concept portrayal. Third, we establish an evaluation system based on the topological and resilience attributes of the urban road network, which can make an all-round and full-process resilience evaluation under the scenario of malignant traffic accidents.

Several limitations persist in this work. We use an ideal urban road network and set ideal traffic flow in the simulation, which differs from actual situations. Furthermore, we only qualitatively describe the spreading characteristics of the congestion phenomenon without any quantitative expression. Subsequent research will focus on the real road network in cities and quantify the propagation of the congestion phenomenon.

## Data Availability

The data generated by the simulation experiment are used in the paper. The data can be obtained from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors thank the Transportation Research Centre of Shanghai Jiao Tong University for its help and support. The authors also would like to thank Lian Zhu for her help in the simulation experiment.

## References

- [1] L. Wesemann and T. S. G. Hamilton, "Cost-of-delay studies for freeway closures caused by Northridge earthquake," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1559, no. 1, pp. 67–75, 1996.
- [2] S. Zhu and D. H. X. K. Levinson, "The traffic and behavioral effects of the I-35W Mississippi River bridge collapse," *Transportation Research Part A: Policy and Practice*, vol. 44, no. 10, pp. 771–784, 2010.
- [3] C. S. Holling, "Resilience and stability of ecological systems," *Annual Review of Ecology and Systematics*, vol. 4, no. 1, pp. 1–23, 1973.
- [4] R. L. Wears, "Resilience engineering: concepts and precepts," *Quality and Safety in Health Care*, vol. 15, no. 6, pp. 447–448, 2006.
- [5] C. S. Holling, *Engineering Resilience versus Ecological Resilience*, Engineering Within Ecological Constraints, Washington, DC, USA, 1996.
- [6] M. Hansen and J. Sutter, *Studies on the Loma Prieta Earthquake No. 1. The Shake with Freight: The Impact of the Loma Prieta Earthquake on bay Area Truckers* University of California, Sacramento, CA, USA, 1990.
- [7] K. Berdica, "An introduction to road vulnerability: what has been done, is done and should be done," *Transport Policy*, vol. 9, no. 2, pp. 117–127, 2002.

- [8] M. G. H. Bell, "A game theory approach to measuring the performance reliability of transport networks," *Transportation Research Part B: Methodological*, vol. 34, no. 6, pp. 533–545, 2000.
- [9] J. Husdal, "Reliability/vulnerability versus Cost/benefit," in *Proceedings of the conferences ETC 2004*, Strasbourg, France, October 2004.
- [10] E. Jenelius, T. Petersen, and L.-G. Mattsson, "Importance and exposure in road network vulnerability analysis," *Transportation Research Part A: Policy and Practice*, vol. 40, no. 7, pp. 537–560, 2006.
- [11] A. T. Murray, T. C. Matisziw, and T. H. Grubestic, "A methodological overview of network vulnerability analysis," *Growth and Change*, vol. 39, no. 4, pp. 573–592, 2008.
- [12] N. Afzal, S. Sohn, S. Abram, H. Liu, I. J. Kullo, and A. M. Arruda-Olson, "Identifying peripheral arterial disease cases using natural language processing of clinical notes," in *Proceedings of the 3rd IEEE International Conference on Biomedical and Health Informatics (BHI 2016)*, pp. 126–131, Las Vegas, NV, USA, February 2017.
- [13] A. Chen and C. S. M. Yang, "Network-based accessibility measures for vulnerability analysis of degradable transportation networks," *Networks and Spatial Economics*, vol. 7, no. 3, pp. 241–256, 2007.
- [14] E. Jenelius and L.-G. Mattsson, "Developing a methodology for road network vulnerability analysis," *Nectar Cluster*, vol. 1, no. 1, pp. 1–9, 2006.
- [15] M. A. P. Taylor, S. V. C. Sekhar, and G. M. D'Este, "Application of accessibility based methods for vulnerability analysis of strategic road networks," *Networks and Spatial Economics*, vol. 6, no. 3-4, pp. 267–291, 2006.
- [16] A. Quium and S. Hoque, "The completeness and vulnerability of road network in Bangladesh," *Engineering Concerns of Flood*, vol. 1, pp. 59–75, 2002.
- [17] P. Zhong and B. Shuai, "Simulation analysis on hazardous materials transportation network survivability under terrorist attack," *Computer Engineering*, vol. 39, no. 9, pp. 20–24, 2013.
- [18] P. Zhong, *Topology Properties, Mechanism of Cascading Failure and Invulnerability for Hazardous Materials Transportation Network Based on Complex Network*, Southwest Jiaotong University, Sichuan, China, 2015.
- [19] T. C. Matisziw and A. T. Murray, "Modeling - path availability to support disaster vulnerability assessment of network infrastructure," *Computers & Operations Research*, vol. 36, no. 1, pp. 16–26, 2009.
- [20] M. Bruneau and S. Chang, "A framework to quantitatively assess and enhance the seismic resilience of communities," *Earthquake Spectra*, vol. 19, no. 4, pp. 733–752, 2003.
- [21] S. L. Cutter and L. Barnes, "A place-based model for understanding community resilience to natural disasters," *Global Environmental Change*, vol. 18, no. 4, pp. 598–606, 2008.
- [22] L.-G. Mattsson and E. Jenelius, "Vulnerability and resilience of transport systems - a discussion of recent research," *Transportation Research Part A: Policy and Practice*, vol. 81, pp. 16–34, 2015.
- [23] B. Qiliang, L. Xin, and L. Yan, "Urban disaster prevention plan with resilient city theory," *The Planner*, vol. 33, no. 8, 2017.
- [24] R. Faturechi and E. Miller-Hooks, "Measuring the performance of transportation infrastructure systems in disasters: a comprehensive review," *Journal of Infrastructure Systems*, vol. 21, no. 1, Article ID 04014025, 2015.
- [25] P. M. Murray-Tuite, "A comparison of transportation network resilience under simulated system optimum and user equilibrium conditions," in *Proceedings of the 2006 Winter Simulation Conference*, Monterey, CA, USA, March 2006.
- [26] W. H. Ip and D. Wang, "Resilience and friability of transportation networks: evaluation, analysis and optimization," *IEEE Systems Journal*, vol. 5, no. 2, pp. 189–198, 2011.
- [27] L. Wang, X. Xue, and X. Zhou, "A new approach for measuring the resilience of transport infrastructure networks," *Complexity*, vol. 2020, no. 1, Article ID 7952309, 16 pages, 2020.
- [28] X. Mao and J. C. D. Zhou, "Resilience-based optimization of postdisaster restoration strategy for road networks," *Journal of Advanced Transportation*, vol. 2021, no. 8, Article ID 8871876, 15 pages, 2021.
- [29] X. Lai and J. P. L. Teng, "Resilient schedule coordination for a bus transit corridor," *Journal of Advanced Transportation*, vol. 2020, no. 1, Article ID 5398298, 12 pages, 2020.

## Research Article

# Cloud Update of Geodetic Normal Distribution Map Based on Crowd-Sourcing Detection against Road Environment Changes

Chansoo Kim <sup>1</sup>, Sungjin Cho <sup>2</sup>, Myoungcho Sunwoo <sup>3</sup>, Paulo Resende,<sup>4</sup>  
Benazouz Bradai,<sup>4</sup> and Kichun Jo <sup>5</sup>

<sup>1</sup>Department of Intelligent Mobility, Chonnam University, Gwangju 61186, Republic of Korea

<sup>2</sup>Smart Mobility R&D Center, ACELAB, Seoul 06222, Republic of Korea

<sup>3</sup>Department of Automotive Convergence, Korea University, Seoul 02841, Republic of Korea

<sup>4</sup>Driving Assistance Research, Valeo, Créteil 94000, France

<sup>5</sup>Department of Smart Vehicle Engineering, Konkuk University, Seoul 05029, Republic of Korea

Correspondence should be addressed to Kichun Jo; [kichun@konkuk.ac.kr](mailto:kichun@konkuk.ac.kr)

Received 1 January 2022; Accepted 30 March 2022; Published 25 April 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Chansoo Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

LiDAR-based localization has been widely used for the pose estimation of autonomous vehicles. Since the localization requires a sustainable map reflecting environment changes, a map update framework based on crowd-sourcing measurements has been researched. Unfortunately, a point cloud map occupies too large data size to transmit data in the uploading and downloading of the map update framework. To realize the LiDAR map update framework by reducing the data size, we proposed a novel map update framework using a Geodetic Normal Distribution (GND) map that compresses the point cloud to the normal distributions. The proposed GND map update framework comprises two parts: map change detection based on crowd-sourcing vehicles and map updating based on a map cloud server. GND map changes are detected based on an evidence theory considering geometric relationships between the GND map and crowd-sourcing measurements and uploaded to the map cloud server. Uploaded map changes reproduce representative map changes based on a similarity-based clustering, which are updated into the GND map. The proposed framework was evaluated in simulations and real environments on construction sites. As a result, although partial map changes occurred, the GND map was kept up-to-date through the proposed framework and the localization for autonomous driving was performed successfully.

## 1. Introduction

Recently, map-matching localization has been globally researched for localization of autonomous vehicles [1–3]. After a high-definition (HD) map has been constructed by a mapping vehicle equipped with a mobile mapping system (MMS), autonomous vehicles with affordable sensors can estimate their poses (position and heading) by matching their measurements with HD map information. One of the widely used sensors for precise positioning based on map-matching is an LiDAR sensor [4–6]. Localization can be achieved by matching geometric shapes measured by the LiDAR sensor with a LiDAR map which is constructed by the mapping vehicle. Since the LiDAR sensor provides a

three-dimensional precise point cloud within 10 cm accuracy and the LiDAR map provides precise surrounding information within 20 cm accuracy, a localization accuracy within 30 cm can be achieved [7].

Although LiDAR sensors can offer very accurate localization performance, they do encounter a critical problem. The problem is an adaptation to changes in real environments. High-accuracy LiDAR-based localization can be achieved in environments with no changes in their geometric shapes. However, it is easy for the geometric shapes in the environments to be changed. For example, vegetation, such as trees, plants, and grass, can be grown; semistatic objects, such as parked vehicles, can be moved; and construction sites can appear. Accordingly, for LiDAR-based

localization, it is essential for environment changes to be periodically updated into the existing HD map for minimization of the differences between the environments and the map.

However, change updates based on conventional mapping methods using a mapping vehicle with precise sensors have several problems: (1) Mapping vehicles must be driven on all roads to acquire new data, which has a substantial cost. (2) Latency necessarily occurs in map updates because it takes lots of time for a few mapping vehicles to acquire all necessary data. Therefore, a novel approach to update road environment changes is required.

To solve the problems for the adaptation to environment changes, numerous companies (HERE [8], TomTom [9], Mobileye [10], Bosch [11], and Daimler [12]) and academics ([13–15]) have researched a map update framework based on crowd-sourced features (i.e., lanes and traffic control devices), which are measured from camera sensors of intelligent vehicles driven on roads. After crowd-sourced features are uploaded into a map cloud server, the features are merged into the representative changes, and the representative changes are updated into the existing HD map. As a result, the updated HD map information is downloaded and used in intelligent vehicles. The sequential map update framework has tried to overcome the inaccuracy of the features measured by camera sensors through the merging of crowd-sourced information. As a result, the framework based on crowd-sourcing vehicles on roads can provide some advantages such as reducing costs and rare latency for map updates.

Unfortunately, while concepts of map update frameworks based on visual features such as lanes and traffic control devices have been open to the public, most frameworks have not clearly provided performances and results for the HD map and vehicle pose estimation based on the cameras yet. In addition, the camera-based map update frameworks cannot be performed either in the alleys or in the large intersections without lanes and traffic control devices.

Different from the camera-based localization and map update framework, the LiDAR-based localization and map update framework has two advantages. The LiDAR-based approach provides good positioning performances and map accuracy due to the accuracy of the LiDAR sensors. Next, the approach enables vehicle positioning wherever regardless of the absence of visual features such as lanes and traffic control devices. However, it is difficult for the framework to be directly applied for an update of the LiDAR map because map-relevant LiDAR data are too large to be uploaded into the map cloud server and downloaded to intelligent vehicles. While map-relevant information based on camera sensors occupies 10 kB/km [10], raw point cloud map data occupies over 500 MB/km on real roads [7]. Accordingly, a LiDAR map structure for the map update framework based on crowd-sourced measurements is required to occupy a low data size. In addition, researches such as change detection and merging of crowd-sourced measurements are required for the map update framework to adopt the map structure.

To compress the map-relevant LiDAR data between the crowd-sourcing vehicles and the map cloud server, a concept of a Normal Distribution Transform (NDT) map is used in the paper. Especially, a Geodetic Normal Distribution (GND) map structure [16], which is extended from the NDT map in our previous research, is adopted because it supports a unified map structure for multiple vehicles. Based on the worldwide management property of the GND map, the map data are interpreted in the same manner in individual vehicles regardless of the coordinate conversion errors. In order to update the GND map periodically, the paper proposes a cloud update framework of the GND map based on crowd-sourcing detection of road environment changes. There are two problems for the GND map update framework to be considered for realization in the real roads. First, map changes must be detected in crowd-sourcing vehicles. In order to detect map changes, each change probability in each cell of the GND map is estimated based on the evidence theory and the ray-casting approach between the measured points and the normal distributions. Second, it is essential to merge and reproduce representative changes from crowd-sourced map changes, and then the representative changes are updated into the existing GND map. In order to merge the crowd-sourced map changes, a similarity-based clustering algorithm is applied.

The objective of this paper is to keep the GND map up-to-date for continuous localization of autonomous driving. To achieve the objective, the contributions of the paper are the following:

- (i) The proposed framework reduces the map-relevant data size by applying the GND map structure in the overall map update process
- (ii) The change detection algorithm based on the evidence theory and the ray-casting approach between the measured points and the normal distributions detects the GND map changes from crowd-sourcing vehicles
- (iii) After the GND map changes detected by crowd-sourcing vehicles are merged into representative map changes based on the clustering algorithm using a similarity of map changes, the representative map changes update the existing GND map in the map server

To explain the framework of the GND map update, this paper is organized as follows. Section 2 explains some works related to map update system based on the LiDAR sensor. Section 3 describes an overall framework to update the existing GND map. In Section 4, crowd-sourcing vehicles can detect changes from GND map structure. Section 5 explains extraction of a representative change from the crowd-sourcing map changes in the map cloud server. In Section 6, the performance of the proposed framework is evaluated in simulation environments. In Section 7, experiments are performed for analyzing the performance of the proposed framework in real environments. Finally, the paper is concluded with Section 8.



## 2. Related Works

In an autonomous driving field, there are mainly three LiDAR map structures: point cloud (PCD) map [17–20], grid map [21–24], and geometric map [25–28]. Since the PCD map is constructed easily by accumulating point clouds acquired by LiDAR sensors based on the vehicle moving trajectory, it is widely applied as a fundamental map type for LiDAR. The grid map is also widely used as the map structure for LiDAR. In particular, since the occupancy grid map consists of lots of grids with occupancy probabilities distinguishing based on the probability theory and the LiDAR ray-casting, the map type can remove the moving objects in the mapping process easily. Finally, the geometric map structure, such as the NDT map [25, 26] and Gaussian mixture map [27, 28], can be used for the LiDAR map. To reduce the map size, after points in the PCD map are split into lots of voxels similar to the grid map, the points are converted to the normal distribution or the Gaussian mixture distribution.

Map update approaches can be generally determined based on the properties of the map structure. As shown in Table 1, the approaches to updating map changes can be split into two categories: standalone-based and crowd-sourcing-based. In researches on standalone-based map update, there are two main parts: map change detection/update and instant map update. The PCD map structure can be generally updated after changes have been detected because the PCD map structure has definite point information around environments. The first approach to map change detection for the PCD map involves determining the minimum distances between sensor points and map points [29, 30]. The other approach determines changes by checking whether map points are traversed by sensor points [31–34]. Among them, Xiao et al. applied the Dempster–Shafer theory to integrate multiple inferences from multiple rays to estimate the change states of map points precisely [33, 34].

An occupancy grid map can be updated based on the instant map update strategy because the map structure models the occupied probabilities in spaces that do not contain definite information. In addition, the occupancy grid map structure supports a basic function to instantly update the occupied and free probabilities of cells based on the ray-casting of LiDAR measurements. In this way, the occupancy grid map can be easily updated in a map-changing environment [35]. Particularly noteworthy is the frequency map enhancement platform proposed by Krajník et al., which provides and updates the occupancy grid map frequently [36, 37]. However, the occupancy grid map cannot distinguish the unknown grids (no measurement region) and conflict grids (different measurements in the same region) explicitly. In order to solve the problem, Trehard et al. applied the evidence theory to the occupancy grid map [38, 39].

A NDT map has been researched for both the change detection/update strategy and instant map update strategy. Katsura et al. detected changes in the NDT map by comparing the normal distributions in the NDT map with the normal distributions constructed by measurements [40]. To

TABLE 1: Categorization for researches of LiDAR map update.

Agent	Map type	Methodology	Research
Standalone	PCD	Change detection/update	[29–34]
	Grid	Instant map update	[35–37]
	NDT	Change detection/update Instant map update	[40] [41–43]
Crowd-sourcing	PCD	Change detection/update	[44, 45]
	Grid	Map merging	[46–49]

determine the differences of distributions from two sources, they compared geometric shapes consisting of sheets, planes, and lines determined by eigendecomposition. However, this approach cannot distinguish deleted cells and unmeasured cells because the changes cannot distinguish free space based on ray-casting. On the other hand, an NDT map integrated with an occupancy grid map can be updated using the instant map update strategy [41–43]. This approach instantly updates the occupied probability in the cells of the NDT map based on ray-casting. Simultaneously, the normal distribution in the cell is updated based on the recursive covariance sample update. However, in this approach, the cell information is not definite within the transient state, which is different from the deterministic state in a real environment, where one state must be allocated. In addition, because all measurements are required to update the map, they may occupy too large size.

On the other hand, a crowd-sourced map update system for LiDAR has been widely researched. The approach used crowd-sourced sensor information transmitted to a cloud server. To update the PCD map structure, Kim et al. applied both probabilistic and evidential theories to the map update based on the ray-casting approach [44]. In addition, Xue et al. used the infrastructure units for point cloud map update [45]. However, the PCD map structure still occupies a too large size to upload and download the map data. For the probabilistic occupancy grid map structure, there are several ways to update the map based on crowd-sourced data [46–48]. However, the occupancy grid map cannot explicitly deal with unknown states that are not measured from the sensors. To overcome this problem, Jo et al. proposed updating a worldwide 3D environment based on an evidential occupancy grid map using multiple vehicles [49]. However, to the best of our knowledge, no crowd-sourced data-based map update systems currently exist for other geometric map structures.

## 3. Framework of GND Map Update Based on Crowd-Sourcing Detection

Figure 1 illustrates an overall framework of the GND map update system based on crowd-sourced data in changing environments. The system has two physical parts: a collection of intelligent vehicles and a map cloud server. Intelligent vehicles detect differences between a GND map and point clouds measured by in-vehicle LiDAR sensors, which are denoted as map changes. The map changes detected by multiple vehicles are uploaded to the map cloud server. The

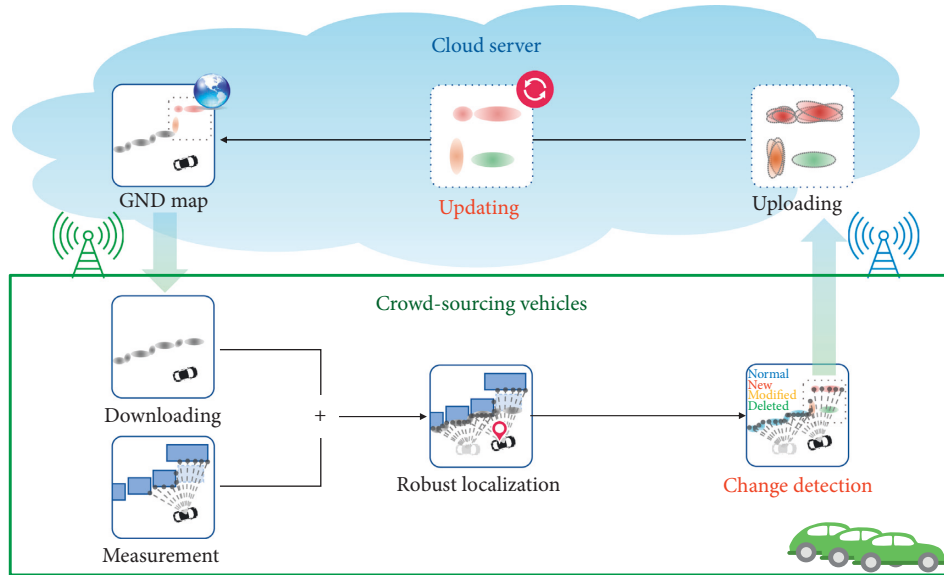


FIGURE 1: Overall framework of map update system based on crowd-sourced data.

server merges the crowd-sourced map changes into representative map changes, which update the existing GND map, denoted as the base GND map. The process of the crowd-sourced map update framework has five steps, which are conducted as follows:

- (i) *Download*. The GND map, which stores the geometric shape information about road environments as multiple normal distribution models, is used as the LiDAR map for the localization of intelligent vehicles. Because the vehicle approximately knows its location from its GNSS, the GND map information around the location of the ego-vehicle can be downloaded to the vehicle in advance before the vehicle enters into the region through vehicle wireless networks. By the characteristics of the GND map structure compressing the map data, the burden on the wireless network is reduced over the PCD map widely used.
- (ii) *Robust Localization*. The pose of the vehicle is essential for detecting the map changes between the GND map and the measured point cloud. The vehicle's pose can be estimated based on map-matching localization using the geometric matching relationship between measurements and the base GND map. However, changes in the environments may cause poor performance of geometric matching for localization. To reduce the degradation of localization, this paper applies an off-line hierarchical approach based on a submap concept into the GND map-matching process [44]. Because the map update framework does not require the vehicle pose in real-time but the precise vehicle pose, the off-line approach is used for the elaborate vehicle localization.
- (iii) *Change Detection*. Based on the pose estimated from robust localization, point clouds measured by in-vehicle LiDARs are used to detect changes within the downloaded GND map. To detect these changes, this paper considers the geometric relationship between LiDAR beam characteristics and normal distributions in the GND map. To estimate the normal distribution changes precisely, an evidence theory is employed. The process of map change detection is explained in Section.
- (iv) *Upload*. The map changes detected by multiple intelligent vehicles are uploaded to the map cloud server by vehicle wireless networks. To reduce network costs by minimization of the transmitted data, only changed parts are uploaded. In addition, because the changed parts are formatted by the GND map structure, the data size can be reduced more than the PCD map changes [44].
- (v) *Update*. The map changes uploaded by multiple intelligent vehicles merge representative map changes, which update the base GND map in the map cloud server. Because the map changes detected by the vehicles may have some errors due to their low-cost sensors and inaccurate positioning (tens of centimeter level), rules for merging the crowd-sourced data are required. This paper proposes similarity check methods and merging rules between crowd-sourced data based on linear algebra; this is explained in Section.

To implement the proposed GND map update system based on crowd-sourced data, the initial GND map must be constructed by a mapping vehicle equipped with an MMS in advance. The inaccurate crowd-sourcing data measured by affordable sensors cannot be used for the precise mapping but for the map update framework. In addition, dynamic points reflected against dynamic objects (including moving pedestrians, bikes, and vehicles) are classified. In order to classify the moving states of the points, a LiDAR point

motion segmentation algorithm based on a combination of probabilistic and evidential approaches is used [50]. In addition, it is assumed that real-time updates to the GND map are not necessary; the map update is performed periodically (e.g., daily). It is also assumed that Partial map changes in the interval between periodic map updates can be supported by robust localization. Based on preliminaries and assumptions, the paper mainly focuses on steps of *Change detection* and *Update* processes in section and, respectively.

#### 4. GND Map Change Detection in Individual Vehicles

The major objective of this paper is to keep the base GND map up-to-date based on crowd-sourced data from multiple intelligent vehicles. The map change update algorithm has two parts: map change detection in individual vehicles and map change updating based on crowd-sourced data in the cloud. First of all, differences between the base GND map and the real environment, which are defined as map changes, are detected from the intelligent vehicles to keep the base GND map up-to-date. In this section, the change detection algorithm for the GND map structure based on ray-casting is presented.

*4.1. Voxel-to-Voxel (V2V) Comparison-Based Map Change Detection.* As shown in Figure 2(a), a GND map structure,  $m$ , is composed of multiple GND tiles  $\{m^q | 1 \leq q \leq Q\}$  split by the same angle in geodetic coordinates, where  $Q$  represents the maximum tile id. As shown in Figure 2(b), the tile  $m^q$  is the union of GND voxels  $\{v^{q,i} | 1 \leq i \leq I\}$ , where  $I$  is the maximum voxel index within the tile  $m^q$ . The GND voxel  $v^{q,i}$  includes the number of points  $n^{q,i}$ , and the normal distribution  $N(p|\mu^{q,i}, \Sigma^{q,i})$  with mean  $\mu^{q,i}$  and covariance  $\Sigma^{q,i}$ . Because each GND voxel  $v^{q,i}$  is split by each spatial boundary, the probability of a point  $p^{q,i}$  being located in the voxel  $v^{q,i}$  is only related to  $N(p|\mu^{q,i}, \Sigma^{q,i})$ . Accordingly, all voxels in the GND map are independent of each other. Due to the independency of each voxel, the change of the GND map can be determined within each voxel. Accordingly, in this section, the GND voxel  $v^{q,i}$  is simply denoted as  $v_{\text{map}}$  with  $n_{\text{map}}$ ,  $\mu_{\text{map}}$ , and  $\Sigma_{\text{map}}$ .

As shown in Figure 3, the map change class  $c_{\text{map} \rightarrow \text{real}}$  can be defined by five classes: *normal*, *empty*, *new*, *modified*, and *deleted* by comparison between GND map voxel  $v_{\text{map}}$  and real environment voxel  $v_{\text{real}}$ . The blue *normal* class indicates that normal distributions of the voxels in the GND map and real environment are the same. The gray *empty* class indicates that the voxel does not have any confident normal distributions in both the map and real environment. The *normal* and *empty* classes are not map changes between the GND map and real environment. On the other hand, the red-colored *new* class indicates that the voxel has a new normal distribution that is not in the map. The orange-colored *modified* class indicates that the voxel has a normal distribution with a different shape to that of the normal distribution in the GND map. The green *deleted* class indicates that the normal distribution in the GND map is

deleted in the real environment. The *new*, *modified*, and *deleted* classes can be determined as map changes from the GND map.

The voxel-to-voxel (V2V) comparison-based map change detection may detect the map change classes  $c_{\text{map} \rightarrow i}$  by comparing voxels of the GND map with voxels of the real environment constructed by point clouds of an intelligent vehicle  $i$  [40]. Unfortunately, in contrast to the definition of the map change class, the algorithm encounters some problems in the real road environment because it does not consider the measurement limitations. In a moving and changing environment with moving and parked vehicles, point clouds measured by an intelligent vehicle can be occluded by other objects, as shown in Figure 4(c). The occlusion of measurements in Figure 4 can cause two problems: (1) incomplete normal distributions and (2) ambiguity in unknown regions. First, even with no changes in the real environment, the occluded measurements can construct incomplete normal distributions, as shown in (1) of Figure 4. The incomplete normal distributions distinguish voxels as being modified incorrectly. In addition, unknown voxels (not measured by LiDAR in (2) of Figure 4) and *deleted* voxels (passed by the rays in (3) of Figure 4) cannot be distinguished from each other in the algorithm because both types of voxels have no points measured by LiDAR. This problem causes the *deleted* class to be assigned incorrectly, as shown in Figure 4(b).

*4.2. Map Change Detection Algorithm Based on Ray-Casting.* To solve two problems with V2V comparison-based map change detection, a novel change detection algorithm based on ray-casting is proposed in this paper. Different from the previous map change algorithm comparing with the constructed map, the new change detection algorithm estimates the map change state  $s_{\text{map} \rightarrow i, j}$  using vehicle  $i$ 's individual point  $p_j$ .

The change state  $s_{\text{map} \rightarrow i, j}$  can be inferred by a geometric relationship between the voxel in the GND map  $v_{\text{map}}$  and the individual point  $p_j$  measured by vehicle  $i$ . When the point is measured in the voxel by being blocked by a normal distribution, the normal distribution may be estimated as *sustained*, as shown in voxels (1) of Figure 5. Next, when the ray is traversed inside the normal distribution in the voxel, the geometric relationship can provide an inference for the voxel to be *changed*, as shown in voxels (2) of Figure 5. In addition, as the point is measured in the empty voxel, the voxel can be estimated as *changed*, as shown in voxels (3) of Figure 5. Accordingly, the geometric relationship between the voxel and the point can provide inference to determine whether the state is *changed* or *sustained*.

As shown in Figure 6, there are six cases for the geometric relationship between a point and a voxel, which are distinguished based on three criteria: whether there is a normal distribution or not in the voxel, whether a point is blocked or passed in the voxel, and whether the ray or point intersects the normal distribution or not. The first criterion can be checked by finding the normal distribution in the voxel. The second criterion, which is about whether the point

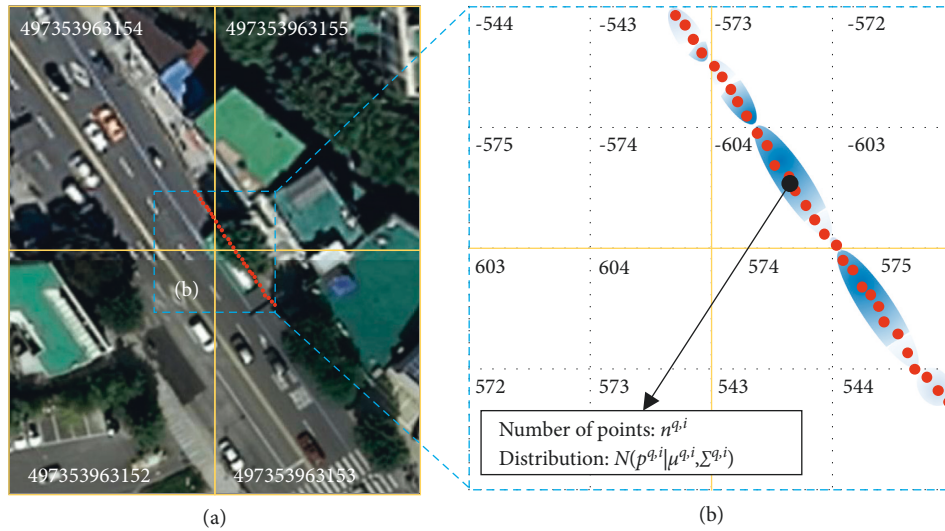


FIGURE 2: GND map structure in (a) geodetic tiles and (b) Cartesian voxels at the sea-level floor (1209 voxels).

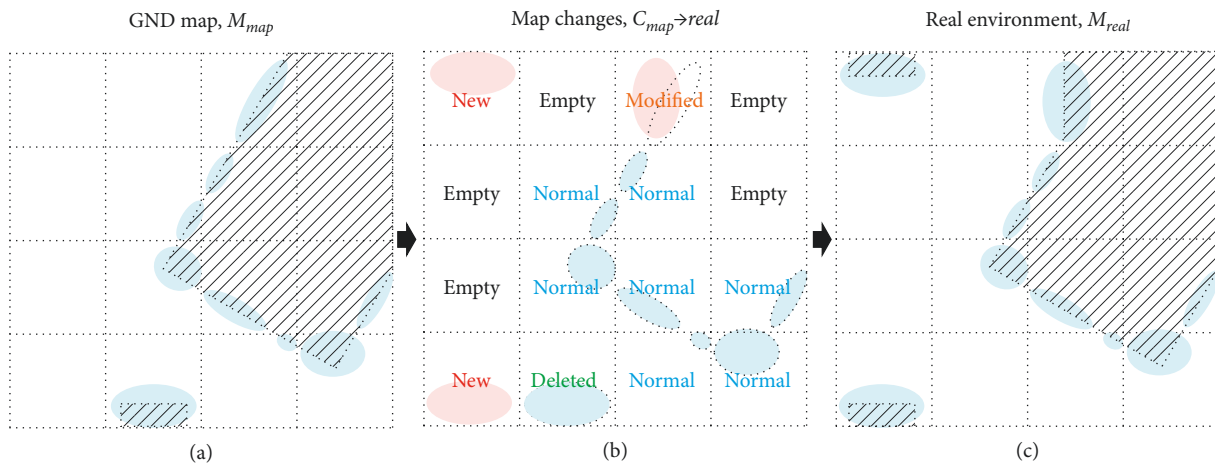


FIGURE 3: Definition of five map change classes including *normal*, *empty*, *new*, *deleted*, and *modified*.

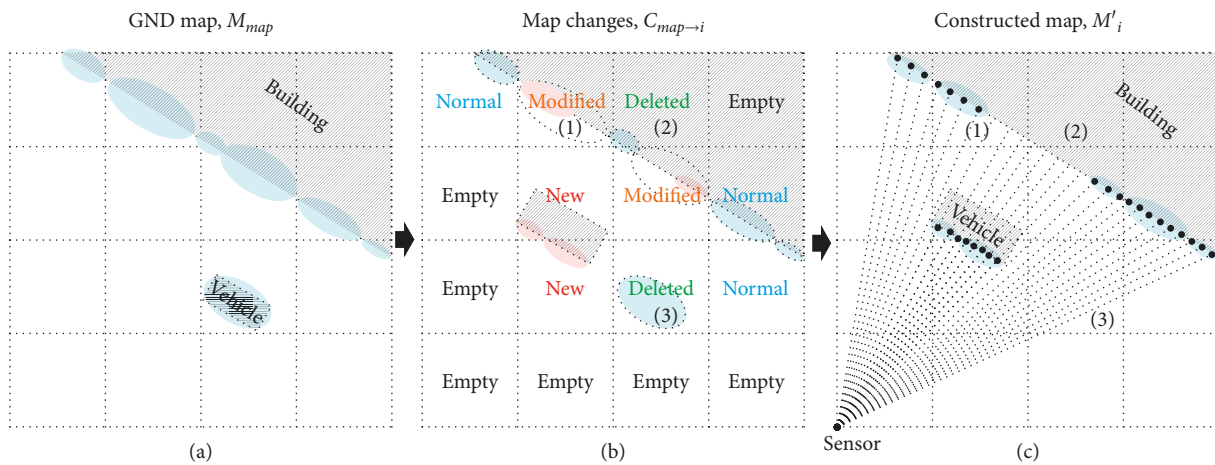


FIGURE 4: V2V comparison-based map change detection compared with map constructed by intelligent vehicles.

is blocked or passed in the voxel, is straightforwardly determined by localizing the point in the voxel and ray-casting towards that point. In the blocked case, the point  $p_j$  is stored

in the temporary point storage  $P$  with the same size as the map. The final criterion related to the intersection between a point and a normal distribution is explained as follows.

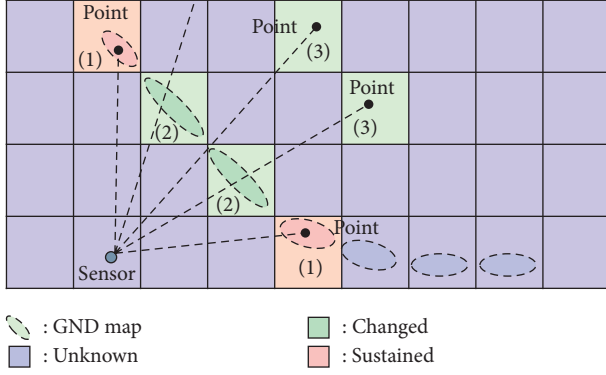


FIGURE 5: Map change inference by ray-casting.

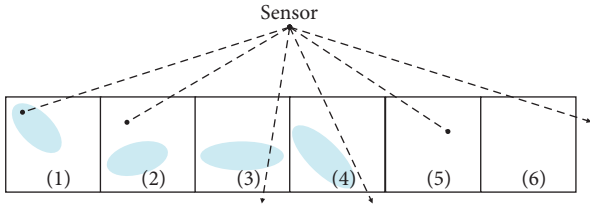


FIGURE 6: Geometric relationship between a measured point and a voxel.

#### 4.2.1. Intersection between Point and Normal Distribution.

To evaluate the intersection between a point and a normal distribution, we can check whether the measured point  $p_j$  is located inside or outside the normal distribution  $N(p|\mu_{\text{map}}, \Sigma_{\text{map}})$ . Unfortunately, the normal distribution does not have a boundary because it is a three-dimensional probability density function, as shown in Figure 7. To determine whether the point  $p_j$  is located inside or outside the normal distribution, a confidence interval is employed. The confidence interval  $\eta_{\text{bound}}$  to determine the boundary is the physically expected probability that a sample in the voxel is located within the normal distribution. When the confidence interval  $\eta_p$  computed by the point  $p_j$  is smaller than  $\eta_{\text{bound}}$ , the point  $p_j$  is located within the boundary of the normal distribution determined by the confidence interval  $\eta_p$ , as described by

$$\begin{cases} \eta_{\text{bound}} \geq \eta_p: \text{ point is located inside distribution,} \\ \eta_{\text{bound}} < \eta_p: \text{ point is located outside distribution.} \end{cases} \quad (1)$$

To compute the confidence interval  $\eta_p$  at the point  $p_j$ , the cumulative distribution function of the normal distribution is defined as the probability that a sample lies inside the ellipsoid determined by its Mahalanobis distance  $r$  from the normal distribution in (2). Accordingly, the problem of finding the confidence interval  $\eta_p$  is converted to the problem of finding the cumulative function of  $r^2$ . Because the square of the Mahalanobis distance  $r^2$  is represented as the sum of the squares of three independent normally distributed Gaussian variables, the cumulative function of  $r^2$  can be converted to a chi-square cumulative distribution function, as shown in (3).

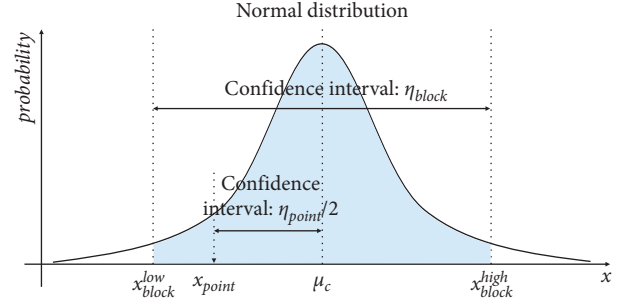


FIGURE 7: Confidence interval in normal distribution.

$$r^2 = (p_j - \mu_{\text{map}})^T \sum_{\text{map}}^{-1} (p_j - \mu_{\text{map}}). \quad (2)$$

$$\eta_p = \int_0^{r^2} \frac{t^{(k-2)/2} \cdot e^{-t/2}}{2^{k/2} \cdot \Gamma(k/2)} dt, \quad (3)$$

where  $\Gamma(\cdot)$  is the Gamma function, and  $k$  is set to 3 in accordance with the three-dimensional normal distribution. The confidence interval  $\eta_p$  is determined by the chi-square cumulative distribution function, as described by equation (3). As a result, it is determined whether the point  $p_j$  is located inside or outside the normal distribution.

#### 4.2.2. Intersection between Ray and Normal Distribution.

To check whether the ray by the measured point  $p_j$  is traversed into the normal distribution  $N(p|\mu_{\text{map}}, \Sigma_{\text{map}})$ , a maximum likelihood point  $p_{\text{ML}}$  is used. As shown in Figure 8, the maximum likelihood point  $p_{\text{ML}}$  indicates a point with the maximum likelihood on the ray against the normal distribution  $N(p|\mu_{\text{map}}, \Sigma_{\text{map}})$ . The maximum likelihood point  $p_{\text{ML}}$  can be calculated analytically [41]. When the confidence interval  $\eta_{\text{ML}}$  computed by the maximum likelihood point is smaller than the boundary confidence interval  $\eta_{\text{bound}}$ , the ray is confirmed to have passed into the normal distribution, as shown in

$$\begin{cases} \eta_{\text{bound}} \geq \eta_{\text{ML}}: \text{ ray is passing the distribution,} \\ \eta_{\text{bound}} < \eta_{\text{ML}}: \text{ ray is not passing the distribution.} \end{cases} \quad (4)$$

**4.3. Evidence Modeling for Geometric Relationship.** As shown in Figure 6, the six cases from the geometric relationship between the measured point  $p_j$  and the voxel  $v_{\text{map}}$  can provide inference to determine whether the change state  $s_{\text{map} \rightarrow i,j}$  of the voxel  $v_{\text{map}}$  is *changed* (C) or *sustained* (S). However, if the voxel  $v_{\text{map}}$  is not measured by any points and passed by any rays, the map change state of the voxel cannot be estimated. To handle the *unknown* state issue explicitly, evidence theory is applied. In evidence theory, the two states form a frame of discernment  $\Omega = S, C$ . Additional states  $\Omega, \phi$  can be managed explicitly by extending the frame of discernment  $\Omega$  to the power set  $2^\Omega = S, C, \Omega, \phi$ , which is the set of all subsets of  $\Omega = S, C$ . The state  $\Omega$  means that the voxel is *sustained* (S) or *changed* (C). However, because the state

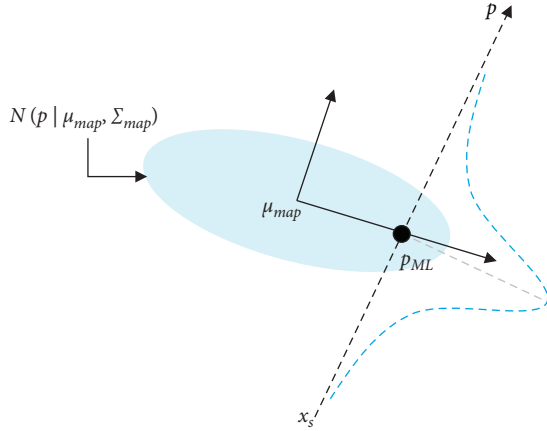


FIGURE 8: Maximum likelihood point in normal distribution.

cannot be *sustained* ( $S$ ) and *changed* ( $C$ ) simultaneously, the state  $\Omega$  represents an *unknown* state. The state  $\phi$  represents that the voxel is not both *sustained* ( $S$ ) and *changed* ( $C$ ). However, because this scenario is physically impossible, the state  $\phi$  represents a *conflict* situation. To quantify the evidence of each element of the power set, a mass function, denoted by  $m$ , is applied. Given a point  $p_j$  measured by intelligent vehicle  $i$ , the mass functions of  $m_{\text{map} \rightarrow i, j}(S)$  and  $m_{\text{map} \rightarrow i, j}(C)$  represent the beliefs of the voxel being *sustained* ( $S$ ) and *changed* ( $C$ ), respectively. The mass function  $m_{\text{map} \rightarrow i, j}(\Omega)$  is the union of the beliefs of *sustained* and *changed*, and the mass function  $m_{\text{map} \rightarrow i, j}(\phi)$  represents the belief that the voxel is conflicted by different measurements. The sum of all masses in the power set must be one based on its definition in evidence theory. The mass function  $m_{\text{map} \rightarrow i, j}(\text{state})$ , including  $m_{\text{map} \rightarrow i, j}(S)$ ,  $m_{\text{map} \rightarrow i, j}(C)$ ,  $m_{\text{map} \rightarrow i, j}(\Omega)$ , and  $m_{\text{map} \rightarrow i, j}(\phi)$ , can be modeled by the geometric relationship between the voxel  $v_{\text{map}}$  and point  $p_j$  measured by intelligent vehicle  $i$ .

#### 4.3.1. Case 1: Blocking inside Normal Distribution (BI).

In the BI case, the measured point in the voxel  $v_{\text{map}}$  is blocked inside a normal distribution. When the point is judged to be located inside the normal distribution by (1), the map change class can be either *normal* or *modified*, as shown in Figure 9. The measurement can provide a hypothesis for the map change state to be converted as *sustained* ( $S$ ) or *changed* ( $C$ ) for the map change model. The mass  $m_{\text{map} \rightarrow i, j}^{\text{BI}}(\text{state})$  in the BI case can be modeled by

$$\begin{aligned} m_{\text{map} \rightarrow i, j}^{\text{BI}}(\phi) &= 0, \\ m_{\text{map} \rightarrow i, j}^{\text{BI}}(S) &= \lambda_{s, \text{block}}, \\ m_{\text{map} \rightarrow i, j}^{\text{BI}}(C) &= \lambda_{c, \text{block}}, \\ m_{\text{map} \rightarrow i, j}^{\text{BI}}(\Omega) &= 1 - \sum_{A \notin \Omega} m_{\text{map} \rightarrow i, j}^{\text{BI}}(A), \end{aligned} \quad (5)$$

where the parameters  $\lambda_{s, \text{block}}$  and  $\lambda_{c, \text{block}}$  are configuration parameters.  $\lambda_{s, \text{block}}$  represents the probability that the map change state is *sustained*, and  $\lambda_{c, \text{block}}$  represents the

probability that the voxel is changed by the blocked point information. Practically,  $\lambda_{s, \text{block}}$  is higher than  $\lambda_{c, \text{block}}$ .

#### 4.3.2. Case 2: Blocking outside Normal Distribution (BO).

In the BO case, the point is blocked outside a distribution. If the point is judged not to be located inside the distribution by (1), the situation is determined as the BO case. When the point is located outside the distribution, there is only the case that the map change class is *modified*, as shown in Figure 9. Accordingly, the measurement can provide a hypothesis on the map change state to be *changed*. The mass  $m_{\text{map} \rightarrow i, j}^{\text{BO}}(\text{state})$  can be modeled by

$$\begin{aligned} m_{\text{map} \rightarrow i, j}^{\text{BO}}(\phi) &= 0, \\ m_{\text{map} \rightarrow i, j}^{\text{BO}}(S) &= 0, \\ m_{\text{map} \rightarrow i, j}^{\text{BO}}(C) &= \lambda_{c, \text{block}}, \\ m_{\text{map} \rightarrow i, j}^{\text{BO}}(\Omega) &= 1 - \sum_{A \notin \Omega} m_{\text{map} \rightarrow i, j}^{\text{BO}}(A), \end{aligned} \quad (6)$$

where the parameter  $\lambda_{c, \text{block}}$  is equal to the equivalent parameter in the BI case.

#### 4.3.3. Case 3: Passing inside Normal Distribution (PI).

In the PI case, the point is not located in the voxel and the ray is passing the distribution, as shown in Figure 9. The voxels passed by the ray are detected by the modified ray-casting algorithm [51]. Because Bresenham's algorithm [51] assumes that the width, length, and height of all voxels are same, the algorithm is modified for variant voxel sizes in the GND map. In detected voxels passed by the ray, the passing of the distribution can be confirmed by (4). If the ray is judged to intersect the distribution, the voxel is treated as the PI case. In the PI case, there are two cases, where the map change class can be *modified* or *deleted*, as shown in Figure 9. Because both cases cause the map change state to be *changed*, the measurement can provide a hypothesis for the voxel to be *changed*. Therefore, the mass  $m_{\text{map} \rightarrow i, j}^{\text{PI}}(\text{state})$  in the PI case can be derived by

$$\begin{aligned} m_{\text{map} \rightarrow i, j}^{\text{PI}}(\phi) &= 0, \\ m_{\text{map} \rightarrow i, j}^{\text{PI}}(S) &= 0, \\ m_{\text{map} \rightarrow i, j}^{\text{PI}}(C) &= \lambda_{c, \text{pass}}, \\ m_{\text{map} \rightarrow i, j}^{\text{PI}}(\Omega) &= 1 - \sum_{A \notin \Omega} m_{\text{map} \rightarrow i, j}^{\text{PI}}(A), \end{aligned} \quad (7)$$

where parameter  $\lambda_{c, \text{pass}}$  represents a probability for a passed voxel to be *changed*.

#### 4.3.4. Case 4: Passing outside Normal Distribution (PO).

In the PO case, the point is not located in the voxel and the ray is not passing the distribution, as represented by Figure 9. This case can be determined by (4). This case causes three candidates: *normal*, *modified*, and *deleted*. The three cases are the same as all possible changes in a voxel with a normal distribution. This means that the PO case cannot

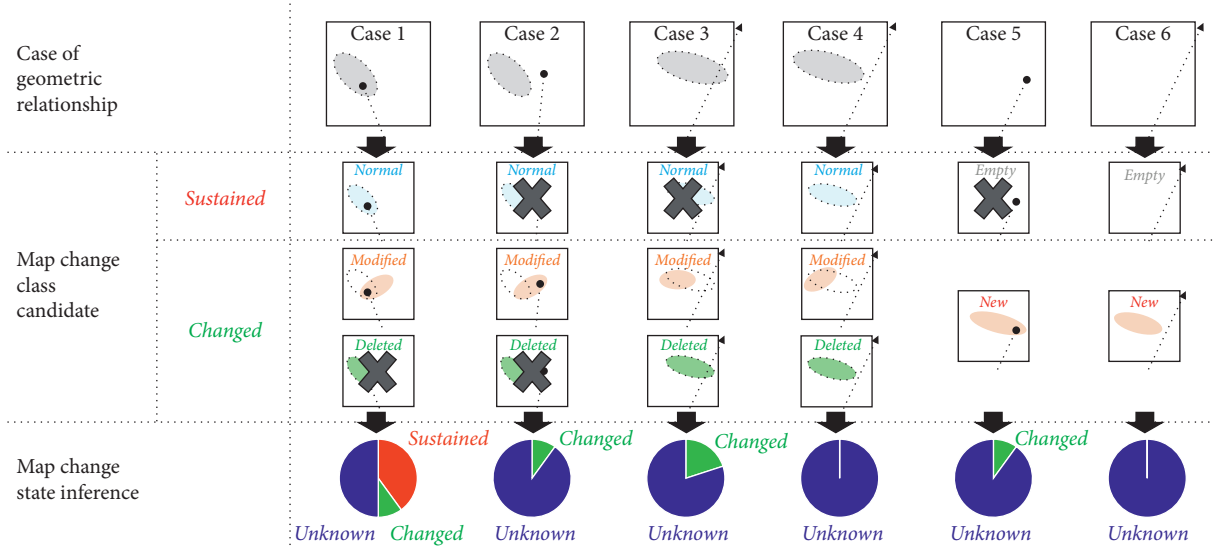


FIGURE 9: Evidence modeling based on six cases for geometric relationship.

provide any inference to detect map changes based on the measurement. Accordingly, the mass  $m_{\text{map} \rightarrow i,j}^{\text{PO}}$  (state) in the PO case is modeled by

$$\begin{aligned} m_{\text{map} \rightarrow i,j}^{\text{PO}}(\phi) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{PO}}(S) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{PO}}(C) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{PO}}(\Omega) &= 1. \end{aligned} \quad (8)$$

4.3.5. *Case 5: Blocking in Empty Voxel (BE)*. The point is measured in the empty voxel in the BE case, as shown in Figure 9. In this case, only the *new* class is inferred. Accordingly, the mass  $m_{\text{map} \rightarrow i,j}^{\text{BE}}$  (state) of the BE case is described by

$$\begin{aligned} m_{\text{map} \rightarrow i,j}^{\text{BE}}(\phi) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{BE}}(S) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{BE}}(C) &= \lambda_{c,\text{block}}, \\ m_{\text{map} \rightarrow i,j}^{\text{BE}}(\Omega) &= 1 - \sum_{A \notin \Omega} m_{\text{map} \rightarrow i,j}^{\text{BE}}(A), \end{aligned} \quad (9)$$

where the parameter  $\lambda_{c,\text{block}}$  represents the probability for the voxel to be *changed*. The parameter is the same as the parameter in the BI case because it is modeled such that the effect of the blocked point provides the same inference for the map change.

4.3.6. *Case 6: Passing through Empty Voxel (PE)*. When the ray is passing the empty voxel, the voxel is placed in the PE case. As shown in Figure 9, the candidates of the map change class are composed of the *empty* or *new* classes. Because the

map change class of the voxel without a normal distribution must be either *empty* or *new*, the measurement does not provide any inference for the voxel. As a result, the mass  $m_{\text{map} \rightarrow i,j}^{\text{PE}}$  (state) of the PE case can be represented by

$$\begin{aligned} m_{\text{map} \rightarrow i,j}^{\text{PE}}(\phi) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{PE}}(S) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{PE}}(C) &= 0, \\ m_{\text{map} \rightarrow i,j}^{\text{PE}}(\Omega) &= 1. \end{aligned} \quad (10)$$

4.4. *Map Change Detection Based on Integrating Masses of Map Changes*. The six cases for the geometric relationship provide the mass  $m_{\text{map} \rightarrow i,j}$  (state) for map change states as shown in Figure 5. Because the voxel  $v_{\text{map}}$  is measured iteratively by points  $\{p_j | 1 \leq j \leq N_j\}$  from intelligent vehicle  $i$ , the voxel  $v_{\text{map}}$  has multiple masses  $m_{\text{map} \rightarrow i,j}$  (state) to provide inference for the map change. To detect the map change precisely, the measured masses  $m_{\text{map} \rightarrow i,j}$  (state) are integrated into one mass. To integrate the masses, Dempster-Shafer's combination rule is applied, as described by

$$\begin{aligned} m_{1 \cap 2}(A) &= \sum_{B \cap C = A | B, C \subseteq \Omega} m_1(B) \cdot m_2(C), \\ m_{1 \oplus 2}(A) &= \frac{m_{1 \cap 2}(A)}{1 - m_{1 \cap 2}(\phi)}, \quad \forall A \subseteq \Omega, A \neq \phi, \\ m_{1 \oplus 2}(\phi) &= 0. \end{aligned} \quad (11)$$

To merge the masses to the change state in each voxel, the masses in each voxel must be initialized. All voxels are initialized as unknown states as (12) because voxels do not have any inferences at first.

$$\begin{aligned}
m_{\text{map}}(\phi) &= 0, \\
m_{\text{map}}(S) &= 0, \\
m_{\text{map}}(C) &= 0, \\
m_{\text{map}}(\Omega) &= 1.
\end{aligned} \tag{12}$$

The result of all mass integration measured by intelligent vehicle  $i$  is denoted by  $m_{\text{map} \rightarrow i}(\text{state})$ , which includes  $m_{\text{map} \rightarrow i}(S)$ ,  $m_{\text{map} \rightarrow i}(C)$ ,  $m_{\text{map} \rightarrow i}(\Omega)$ , and  $m_{\text{map} \rightarrow i}(\phi)$ . The mass  $m_{\text{map} \rightarrow i}(\phi)$  must be 0 by (11). Finally, the maximum value within three masses  $m_{\text{map} \rightarrow i}(\text{state})$  determines the ray-casting-based map change state  $s_{\text{map} \rightarrow i}$  composed of *sustained*, *changed*, and *unknown*.

Figure 10 shows the examples of six map change classes (*unmeasured*, *empty*, *normal*, *new*, *modified*, and *deleted*), which can appear in the situation for the map change detection. The *unmeasured* class is added into the basic five map change classes to consider the unmeasured situation due to limited LiDAR ranges. The *unmeasured* class is classified as the *unknown* state by the initial masses (12) because any rays do not reach the voxel. For the *empty* class, there is no normal distribution and no points are blocked in the voxel. It means that all measurements are allocated to the PE case. Accordingly, after the merging (11), the merged mass can be propagated to the *unknown* state. The two map change classes are not distinguished as map changes because the change states of the two cases are *unknown* state. On the other hand, for the *normal* class, the measurements from the LiDAR points are mostly the BI cases. Although there may be some mis-measurements such as the PE and BO cases, the merged mass can be propagated to the *sustained* state by evidence merging (10). Therefore, the *normal* class is not also distinguished as map changes. Different from the *unmeasured*, *empty*, and *normal* classes, the remained three classes (*new*, *modified*, and *deleted*) must be detected as map changes. For the *new* class, most of points are blocked at the voxel (BE case); therefore, the voxel can be classified as the *changed* state. The voxel with the *modified* case has three BO cases, one BI case, and two PI cases as measurements. The result of merging of six measurements propagates the voxel to the *changed* state. The *deleted* case can be estimated as the *changed* state based on four PI cases and two PE cases.

As shown in Figure 11, the proposed ray-casting-based algorithm can solve two problems caused by the V2V comparison-based change detection algorithm in Figure 4. Voxel (1) is estimated as the *sustained* state updated by the measurements in the BI case, which means that the voxel is classified as the *normal* class different to voxel (1) of Figure 4. On the other hand, voxels (2) and (3) can be distinguished based on the ray-casting approach. Voxel (2) is classified as the *unmeasured* class because no measurement is reached in the voxel. Voxel (3) is estimated as *changed* state due to the measurement update in the PI case. Since there are no blocked points in the voxel, the voxel can be classified as the *deleted* class.

**4.5. Uploading of Map Changes.** When the voxel  $v_{\text{map}}$  is estimated as the *changed* state (*new*, *modified*, or *deleted* classes), the voxel  $v_{\text{map}}$  is uploaded to the map cloud server as map change information. The map change voxels are categorized into two groups: a new normal distribution group and no normal distribution group. If the voxel is classified as *changed* state by the blocking points as shown in *new* and *modified* cases of Figure 10, a new normal distribution is constructed and uploaded to the map cloud server with the voxel index information. On the other hand, if the voxel does not have any blocked points, the only voxel index is uploaded to the map server as the no normal distribution part. The regions to be uploaded are determined by the position of the ego-vehicle. When the vehicle has escaped from the boundary of a geodetic tile of the GND map, the map changes are serialized and stored in a file based on Google Protocol Buffers. The file with the map changes can be uploaded to the map cloud server through network protocols such as the http, FTP, WebDAV, and SAMBA.

## 5. GND Map Update Based on Crowd-Sourced Changes in Cloud Server

**5.1. Clustering Based on Crowd-Sourced Data.** When the differences between the environments and the GND map, the  $N_i$  intelligent vehicles can detect and upload the map change voxels  $\{v_i | 1 \leq i \leq N_i\}$  for the voxel  $v_{\text{map}}$ . The map change voxel  $v_i$  detected by  $i$ -th vehicle consists of the existence of the normal distribution  $\varepsilon_i$ , the number of points  $n_i$ , mean  $\mu_i$ , and covariance  $\Sigma_i$ . If the map change voxel includes a normal distribution as a new normal distribution group,  $n_i$ ,  $\mu_i$ , and  $\Sigma_i$  are defined. Otherwise, the parameters are not defined. However, because the map change voxels have been detected by low-cost sensors, they are not always precise, and inaccurate information can be uploaded. Therefore, it is essential to avoid inaccurate voxels from crowd-sourced voxels and to merge crowd-sourced voxels into an accurate voxel to update the base map. To process two functions simultaneously, density-based spatial clustering of applications with noise (DBSCAN) is applied [52]. The DBSCAN algorithm generates clusters based on the distance between two points. Unfortunately, because the targets of the algorithm in the map change merging process are not points but crowd-sourced map change voxels (*new*, *modified*, and *deleted* classes), the distance-based DBSCAN algorithm cannot be directly applied. To solve this problem, the similarity between two voxels is applied instead of the distance between two points.

The similarity  $d_{i,j}$  between two voxels  $v_i$  and  $v_j$  is composed of three parts: existence similarity  $d_{i,j}^\varepsilon$ , L2 distance-based similarity  $d_{i,j}^{L2}$ , and geometric similarity  $d_{i,j}^g$ . The existence similarity  $d_{i,j}^\varepsilon$  evaluates the existences between two voxels. The L2 distance-based similarity  $d_{i,i+1}^{L2}$  and geometric similarity  $d_{i,i+1}^g$  evaluate the similarities between two distributions of two voxels. If at least one of the two distributions does not exist, the L2 distance-based similarity  $d_{i,i+1}^{L2}$  and geometric similarity  $d_{i,i+1}^g$  are not calculated. Since the three similarities take values from 0 to 1, the total similarity



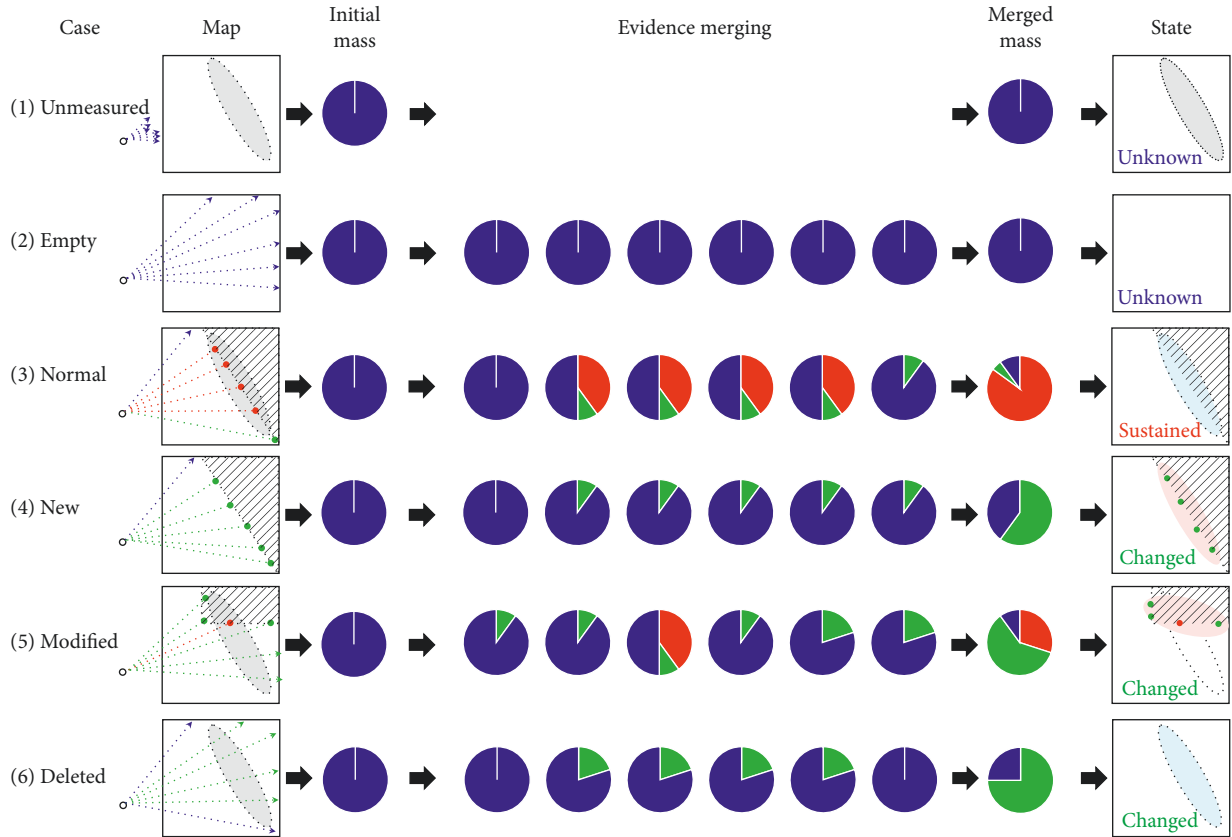


FIGURE 10: Examples of results of evidence merging.

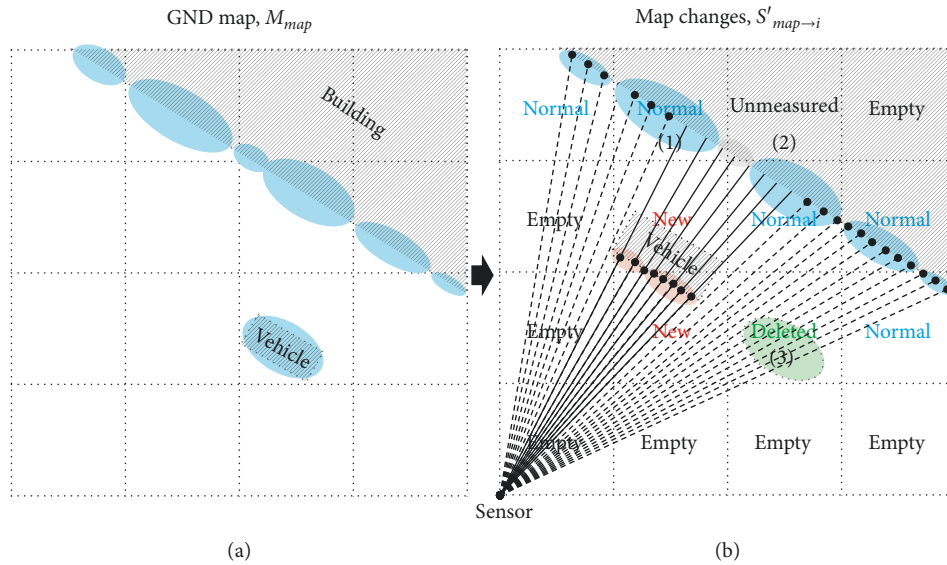


FIGURE 11: Map change detection based on ray-casting approach.

is represented as the product of three similarities to apply the DBSCAN algorithm in

$$d_{i,j} = d_{i,j}^\epsilon \times d_{i,j}^{L2} \times d_{i,j}^g. \quad (13)$$

The voxel without a normal distribution has an existence  $\epsilon_i$  of false. To reflect the voxel without a normal distribution,

the existence similarity is considered. The existence similarity compress the two existences  $\epsilon_i$  and  $\epsilon_j$  between two uploaded voxels  $v_i$  and  $v_j$  according to

$$d_{i,j}^\epsilon = \begin{cases} 1, & \text{if } \epsilon_i = \epsilon_j = \text{true}, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

The L2 distance-based similarity is calculated only in case that the two uploaded voxels have two distributions. The L2 distance-based similarity, used as a constraint for optimization of an distribution-to-distribution matching [25], is represented by the likelihood of the Mahalanobis distance between two distributions in

$$d_{i,j}^{L2} = \exp\left(\frac{-(\mu_j - \mu_i)^T (\Sigma_j + \Sigma_i)^{-1} (\mu_j - \mu_i)}{2}\right). \quad (15)$$

The similarity  $d_{i,j}^{L2}$  takes a value from 0 to 1, with the value 1 representing the best similar relationship between two distributions.

Although the L2 distance-based similarity  $d_{i,j}^{L2}$  can provide a straightforward distance between two distributions, it encounters a problem. Although the two means  $\mu_i$  and  $\mu_j$  are located near each other and two shapes are very different from each other, the L2 distance-based similarity  $d_{i,j}^{L2}$  is approximately 1 because the difference  $\mu_j - \mu_i \approx 0$  makes the similarity to be 1. To solve this problem, geometric similarity, which is based on geometric shape evaluation, is applied [53]. Based on linear algebra, the geometric similarity can classify a distribution as one of three types: sphere, plane, and line. By the eigenvalues  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  ( $\lambda_1 < \lambda_2 < \lambda_3$ ) of normal distribution  $N(\mu_i, \Sigma_i)$  derived by eigendecomposition of the covariance  $\Sigma_i$ , the type  $t_i$  can be determined by

$$t_i = \begin{cases} \text{SPHERE,} & \lambda_3 \approx \lambda_2 \approx \lambda_1 \gg 0, \\ \text{PLANE,} & \lambda_3 \approx \lambda_2 \gg \lambda_1 \approx 0, \\ \text{LINE,} & \lambda_3 \gg \lambda_2 \approx \lambda_1 \approx 0. \end{cases} \quad (16)$$

When the ratio of  $\lambda_2$  to  $\lambda_1$  is larger than 10,  $\lambda_2 \gg \lambda_1$  is achieved. If two distributions have different types, the geometric similarity must be 0. Even if distributions have the same type, the geometric shapes are compared. The normal vector  $n_i$  at the *PLANE* type and the direction vector  $d_i$  at the *LINE* type are used as feature vectors to compare geometric shapes. When the shapes of two distributions are different, the dot product of feature vectors of two distributions is lower. Using this concept, the geometric similarity can be derived through

$$d_{i,j}^g = \begin{cases} 0, & t_i \neq t_j, \\ n_i \cdot n_j, & t_i = t_j = \text{PLANE}, \\ d_i \cdot d_j, & t_i = t_j = \text{LINE}. \end{cases} \quad (17)$$

**5.2. Merging of Crowd-Sourced Map Information.** The DBSCAN algorithm based on the similarity in (13) derives multiple clusters  $\{c_j | 1 \leq j \leq N_j\}$  from multiple map change voxels  $\{v_i | 1 \leq i \leq N_i\}$ , as shown in Figure 12. Each cluster  $c_j$  is a union of multiple map change voxels. There are two types of clusters: empty type and distribution type. An empty type cluster example is  $c_1$  in Figure 12, which means that all map change voxels in the cluster are empty voxels, i.e.,  $\varepsilon_i = \text{false}$ . There can be only one cluster grouped as the empty type in a voxel because the empty voxel always has similarity 1 with

other empty voxels. Accordingly, the representative cluster with the empty type is always set to an empty voxel. On the other hand, the distribution type clusters represented by clusters  $c_2$  and  $c_3$  in Figure 12 mean that clusters are grouped by the map change voxels with the distributions. The representatives of the clusters with distribution type can be derived by the recursive update of sample mean and covariance [54].

**5.3. Publication of Definite Changed Map Information.** There are several representatives constructed from each cluster, as shown in Figure 12(c). To publish the updated map information to the base map, there are several processes. First, the numbers of map change voxels in the clusters are compared, as shown in Figure 13. As a result, the representative normal distribution with the maximum number of merged map change voxels is selected as the final map candidate to be updated. Secondly, validation based on the ratio of uploaded information is evaluated. The ratio of the number of merged voxels to the number of information uploads must be higher than the configured parameter  $\xi_{\text{update}}$  in Figure 13(b) to update the base map. The final process is a comparison of the final map candidate with the base map. The voxel in the base map and the voxel in the final map candidate are compared based on the similarity  $d_{\text{map,candidate}}$  in (13), as shown in Figure 13. When the similarity is larger than  $d_{\text{update}}$ , as shown in Figure 13(b), the map candidate is published to the base map.

## 6. Simulations

**6.1. Simulation Environments.** To evaluate the map update framework, the environments with large changes such as construction sites are required. The site must also make the high-precision GNSS/INS information provide the precise position without any noises due to evaluation of robust localization in changing environments. Unfortunately, it is difficult to find the sites to satisfy both conditions for evaluation in practice. Therefore, the map update framework was evaluated in the simulations to construct the changing environments.

The simulation environments consisted of virtual perception data and real data measured by a test vehicle. The test vehicle installed with on-board sensors (yaw rate and wheel speed sensors), a low-cost GNSS receiver within 2.5 m positioning errors (U-Blox EVK-6T), and a high-precision GNSS/INS within 0.01 m positioning errors (OXTS RT3002) acquired the vehicle motion and positioning data as real data. The motion and positioning information measured by the on-board sensors and the low-cost GNSS receiver was used as input data of a vehicle part performing the map change detection algorithm in the proposed framework. In order to construct virtual perception data, an in-house LiDAR simulator generated some point cloud measurements based on a ray-casting approach from the vehicle pose to the virtually modeled environments. The positioning information measured by the high-precision GNSS was used as the ground truth of the vehicle pose to construct the LiDAR

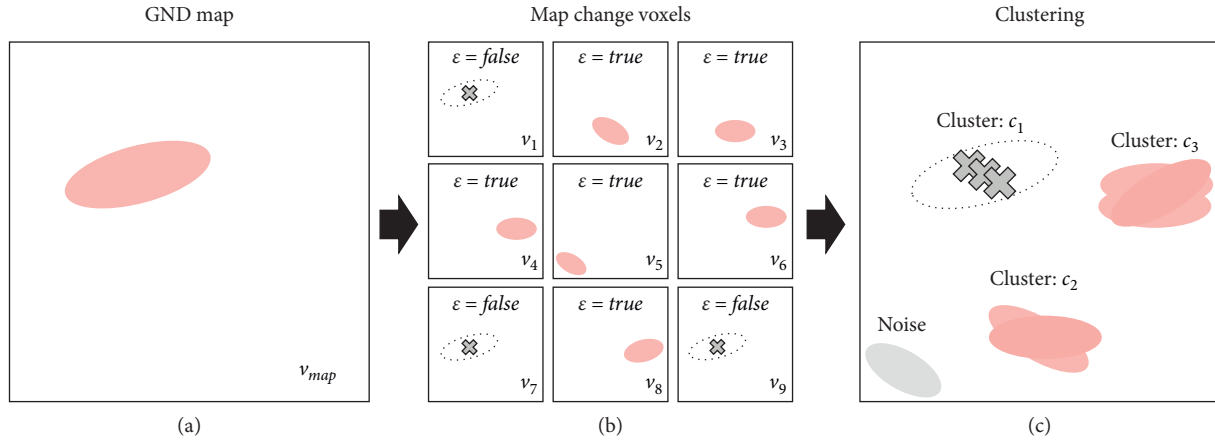


FIGURE 12: Clustering of map change voxels.

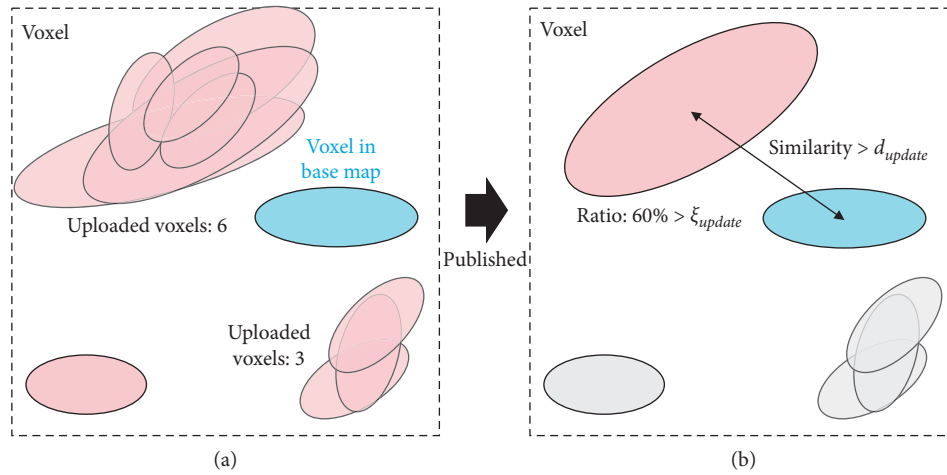


FIGURE 13: Publication of map changes into the base map.

measurements. As shown in Figure 14(a), the base GND map was constructed by the simulated LiDAR measurements along with the vehicle poses. In order to simulate the changing environments, the virtually modeled environments were modified by manual editing including adding/removing/transforming some objects. Accordingly, the editing of the environments provided the changed GND map after environment changes in Figure 14(b).

In the simulator, the LiDAR measurements were modeled considering the LiDAR specifications such as the number of layers, the horizontal resolution, and the vertical resolution. In order to simulate the crowdsourcing information measured from various vehicles, various sensor configurations including Velodyne VLP-16, HDL-32E, and HDL-64E, Robosense RS-LiDAR-16, RS-LiDAR-32, and Valeo Scala were applied in the simulations. For evaluating the accuracy of the GND map updated by the proposed framework, the simulations were accomplished in a local computer without a cloud computing system.

### 6.2. Robust Localization in Map-Changing Environments.

For detecting map changes in the GND map, the exact pose of the vehicle is very important. In order to estimate the vehicle pose in changing environments, we applied a hierarchical algorithm [44, 55]. In the first process of the hierarchical architecture, a submap, which models the present environments in real-time, is constructed based on the Graph SLAM algorithm. Since the submap plays a role as a single LiDAR measurement measured from larger areas by summing continuously sequential measurements, the performance of the map matching in the second process of the hierarchical architecture can be robust in changing environments. Since our previous work [44] evaluated the performance of the algorithm in the PCD map only, we evaluated the performance of the localization with the submap approach in the GND map. To estimate the precise pose, we applied a Graph SLAM algorithm because the algorithm generally provides better performances than the Kalman filter algorithm and the map update does not require real-time processing.

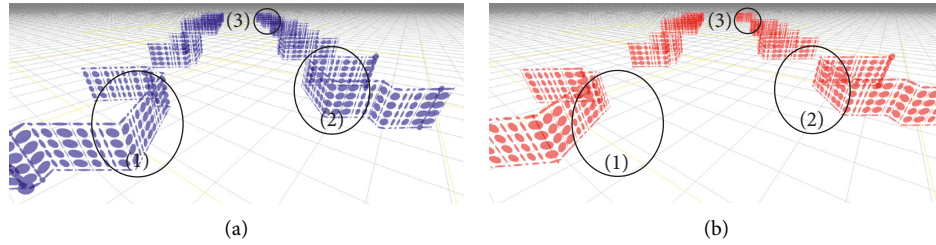


FIGURE 14: GND maps (a) before environment changes and (b) after environment changes. The circles (1)–(3) indicate the differences between two maps.

Figure 15 shows the localization performances of the general Graph SLAM algorithm and the Graph SLAM with the submap concept through errors to the longitudinal, lateral, and heading directions. The errors were evaluated by the ground truth of the vehicle pose measured by the high-precision GNSS/INS. The differences between the GND maps as shown in circles (1)–(3) of Figure 14 are represented from 9 to 17 seconds in Figure 15. The hierarchical Graph SLAM algorithm with the submap concept had better performances than the general Graph SLAM algorithm without the submap concept because a matching ratio, which represents the ratio of matched regions over the measurements in the map-matching process, was higher by the submap.

To judge whether the localization can be used in the map update framework or not, localization performances were validated through localization requirements for autonomous driving proposed by Ford [56]. The Ford localization requirements consist of two criteria in three directions: maximum error boundary and 95% error boundary. In order to satisfy the maximum error boundary criterion, the longitudinal, lateral, and heading errors have to be located within the 0.44 m, 0.44 m, and  $0.50^\circ$  represented by yellow rectangles of Figure 15, respectively. On the other hand, to satisfy the 95% error boundary, the longitudinal, lateral, and heading errors, located in 95% error position from ascending ordering arrays, must be lower than the 0.15 m, 0.15 m, and  $0.17^\circ$  represented by green rectangles of Figure 15, respectively.

The maximum and 95% errors to the longitudinal, lateral, and heading directions from the general Graph SLAM and the hierarchical Graph SLAM are represented in Table 2. If the criterion is satisfied, the blank of the table is filled with the green color. Otherwise, the black is filled with the red color. Since the general Graph SLAM without the submap concept cannot satisfy the lateral and heading requirements, the algorithm cannot be used to the map update framework. On the other hand, the hierarchical Graph SLAM with the submap concept can be used to the map update framework because all criteria in Ford requirements are satisfied.

**6.3. Map Change Detection and Map Update.** The proposed map change detection algorithm can detect map changes from the base GND map using the relationship between the map and the measurements. The map changes, which are estimated as *change* states including *new*, *modified*, and

*deleted* classes, are uploaded to the map cloud server. The map change information makes the changed map information be inferred as shown in Figure 16(a). The *new*, *modified*, and *deleted* classes are represented as red, yellow, and green ellipsoids, respectively. The *normal* class, represented by blue ellipsoids, is not uploaded to the map cloud server. The uploaded crowd-sourced map changes, detected by crowd-sourcing vehicles, are merged into the representative map changes through the map update algorithm in the proposed framework. As a result, the updated map can be represented in Figure 16(b). In order to validate the performance, the ground truth of map changes are represented in Figure 16(c), which is constructed by comparing the previous GND map (a) and the present GND map (b) of Figure 15.

By comparing with the ground truth of map changes, the performances of proposed algorithms are represented in confusion matrices, which are generally used to evaluate a classification problem, as shown in Table 3. The confusion matrices to represent performances of map changes include five map change classes: *normal*, *new*, *modified*, *deleted*, and *empty*. Since the outputs of the algorithm are compared with the ground truth of map changes, higher diagonal values mean better performances. Table 3 includes three confusion matrices, comparisons with (1) the base map, (2) results of map change detection, and (3) results of map update. The first one represents the confusion matrix compared with the base map without map change consideration, as shown in Figure 15(a). Since the base map has been not updated, there is no prediction for map changes (*new*, *modified*, and *deleted*) in the confusion matrix. The second one means the confusion matrix compared with the map inferred by the map change detection based on single driving, as shown in Figure 16(a). Although map changes can be detected, some errors still occur due to two reasons. The parked vehicles cause the inaccurate estimate from the true *empty* class to the predicted *new* class. Also, occlusion regions by parked vehicles cause misclassification from the true *deleted* class to the predicted *normal* class. The third one is the confusion matrix compared with the map updated by the map change update, as shown in Figure 16(b). The errors caused in the second confusion matrix are reduced to zero based on the crowd-sourcing measurements.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (18)$$

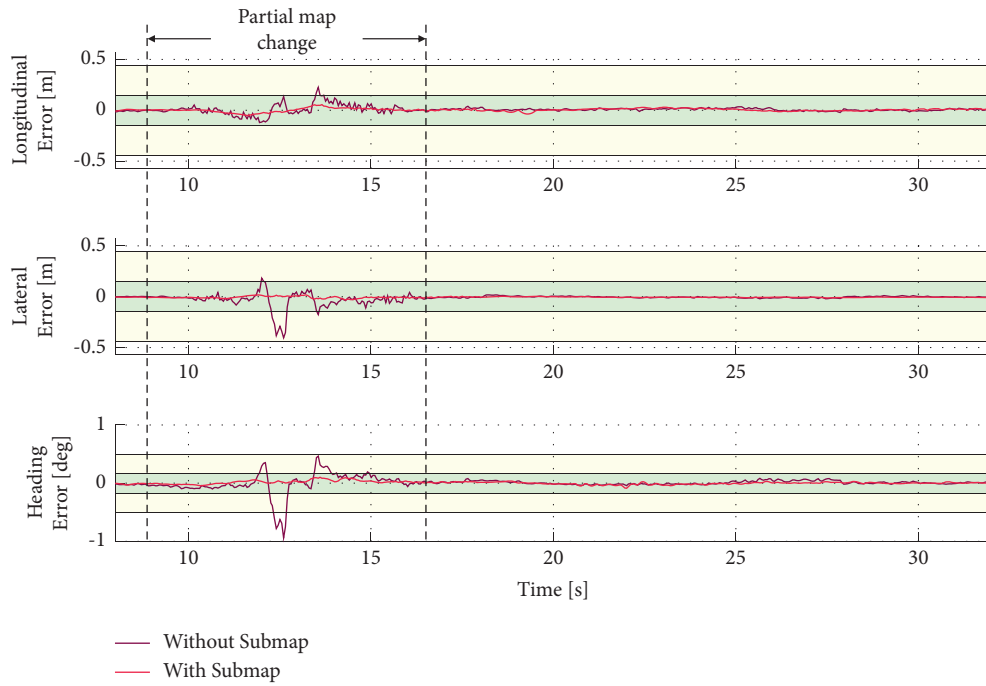


FIGURE 15: Performance of robust localization in map-changing simulations.

TABLE 2: Table of robust localization performance in simulations.

Experiments	Longitudinal (m)		Lateral (m)		Heading (°)	
	95%: 0.15	Max: 0.44	95%: 0.15	Max: 0.44	95%: 0.17	Max: 0.5
Without submap	0.08412	0.2002	0.30124	1.4518	1.1388	6.2191
With submap	0.031013	0.054553	0.012499	0.030573	0.052715	0.094034

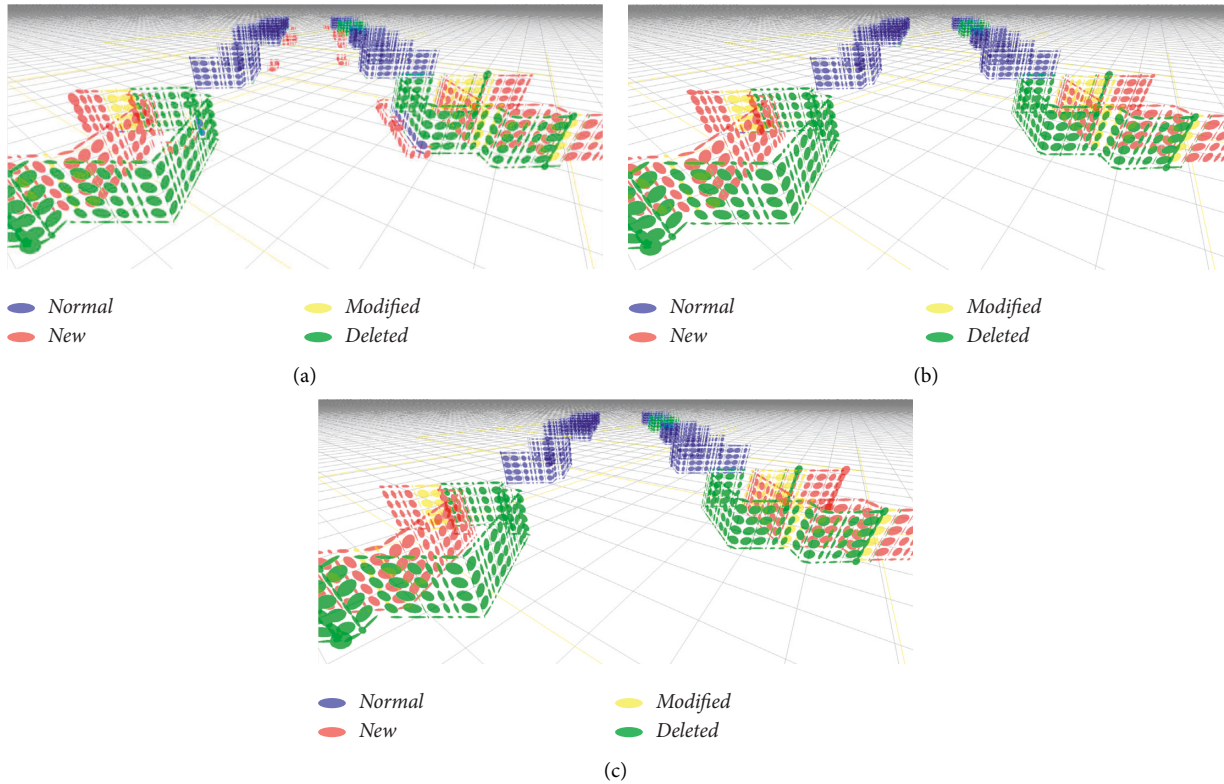


FIGURE 16: Map change information (a) detected by the map change detection algorithm based on single driving, (b) updated by the proposed framework based on crowd-sourced data, and (c) representing ground truth information.



To evaluate the overall performance based on a unit index, we adopt the F1-score (18). The F1-score is widely used in the situation that the number of voxels in each class is not balanced (i.e., *empty*  $\gg$  *normal*  $\gg$  *new, modified, deleted*). The F1-score by the base map, results of map change detection, and results of map update are represented to 37.36%, 88.05%, and 99.62%, respectively.

Figure 17 represents the effects of the number of vehicles to join the map update. The blue line of Figure 17 means the F1-score by the base map. The yellow line represents the F1-score by results of the map change detection based on single driving. The green line represents the results of the map update based on the number of crowd-sourcing vehicles. More crowd-sourcing vehicles construct a more precise map by considering more map change information, and the performance of the map update is saturated after 9 vehicles.

## 7. Experiments

**7.1. Experimental Environments.** The test vehicle, used in the simulations, was also used in the experiments. Differently with the usage of the virtual perception sensors in simulations, the experiments used two real LiDAR sensors (Velodyne VLP-16). Although two LiDAR sensors were used to construct the base map, only one LiDAR sensor was used to update the map information based on the proposed system. The high-precision GNSS/INS information was not used for the proposed system but used for the base map construction and ground truth positioning. The low-cost GNSS was only used to solve the kidnapped problem for localization.

As the paper said in Section 6.1, there were two requirements from a test site for evaluation of the proposed map update system. In order to evaluate the map update system, the environment must be changed from the base map. In addition, to use the high-precision GNSS/INS as ground truth information, the quality of the GNSS must be good in the test site. The chosen place that met the requirements was Wangsimni-ro in Korea, as shown in Figure 18. In the test site, the test vehicle was driven to construct the base map on August 1, 2019. Next, the test vehicle was driven 38 times on the same road to update the base map on October 12, 2019.

The raw PCD map was constructed by accumulating point clouds measured by two LiDAR sensors based on the trajectory of the ego-vehicle measured by high-precision GNSS/INS. To convert the PCD map to the GND map, the points in the PCD map were allocated to each voxel with 1 m basic size. Based on the points, the normal distribution in each voxel could be constructed to form the GND map structure.

**7.2. Robust Localization in Map-Changing Environment.** The real environment confronts the changes of the road over time, such as the construction site, growing trees, and parked vehicles. Accordingly, it is essential to estimate the precise pose considering the map-changing situations. To reflect the map-changing environments, postprocessing localization

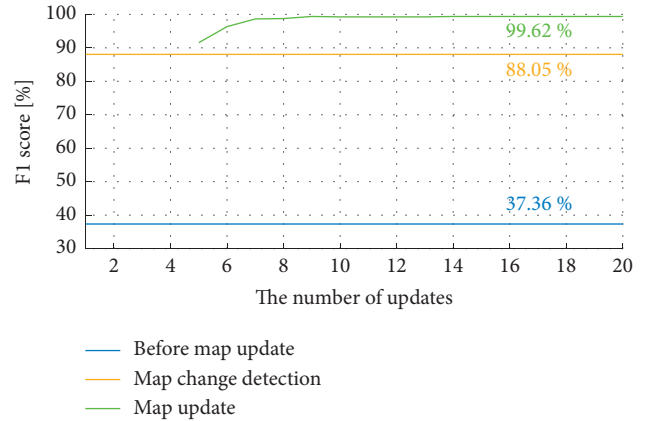


FIGURE 17: F1-scores by the number of crowd-sourcing vehicles in simulations.

based on hierarchical Graph SLAM with the submap concept is applied [44].

To evaluate variations regarding the submap, the errors of localization algorithms without the submap and with the submap are shown in Figure 19. The brown and red lines represent the localization performance without and with the submap, respectively. The yellow and green rectangles in Figure 19 represent the maximum and 95% error boundaries for the localization requirements, respectively. As shown in Figure 19, the notable environment changes occur from 20 to 31 seconds.

Localization without the submap has particularly varying errors in the partial map changes because incorrect map-matching constraints can be constructed by mismatching between the LiDAR measurements and the unmodified GND map. Although precise accuracy of localization is essential for map change detection that meets the localization requirements, the general Graph SLAM without the submap does not satisfy the requirements. To relieve the LiDAR mismatching between measurements and the GND map, the hierarchical Graph SLAM algorithm is applied with the submap concept. After applying hierarchical Graph SLAM, localization error can be reduced. The maximum and 95% errors are represented in Table 4.

The 95%/maximum errors of hierarchical Graph SLAM to the longitudinal and lateral directions are 0.1478/0.19742 m and 0.13104/0.20154 m, respectively. Since the errors are located within 0.15/0.44 m, the localization with the submap can satisfy the localization requirements. In addition, the 0.14799/0.2351° of 95%/maximum heading errors satisfy the localization requirements. Therefore, the robust localization algorithm based on hierarchical SLAM with the submap meets the requirements for localization, and it means that the localization algorithm can be used for the proposed framework.

**7.3. Map Change Detection in GND Map Structure.** The proposed map change detection and map update processes were evaluated in real environments. Different from simulations, the real environments, which have more noise, can

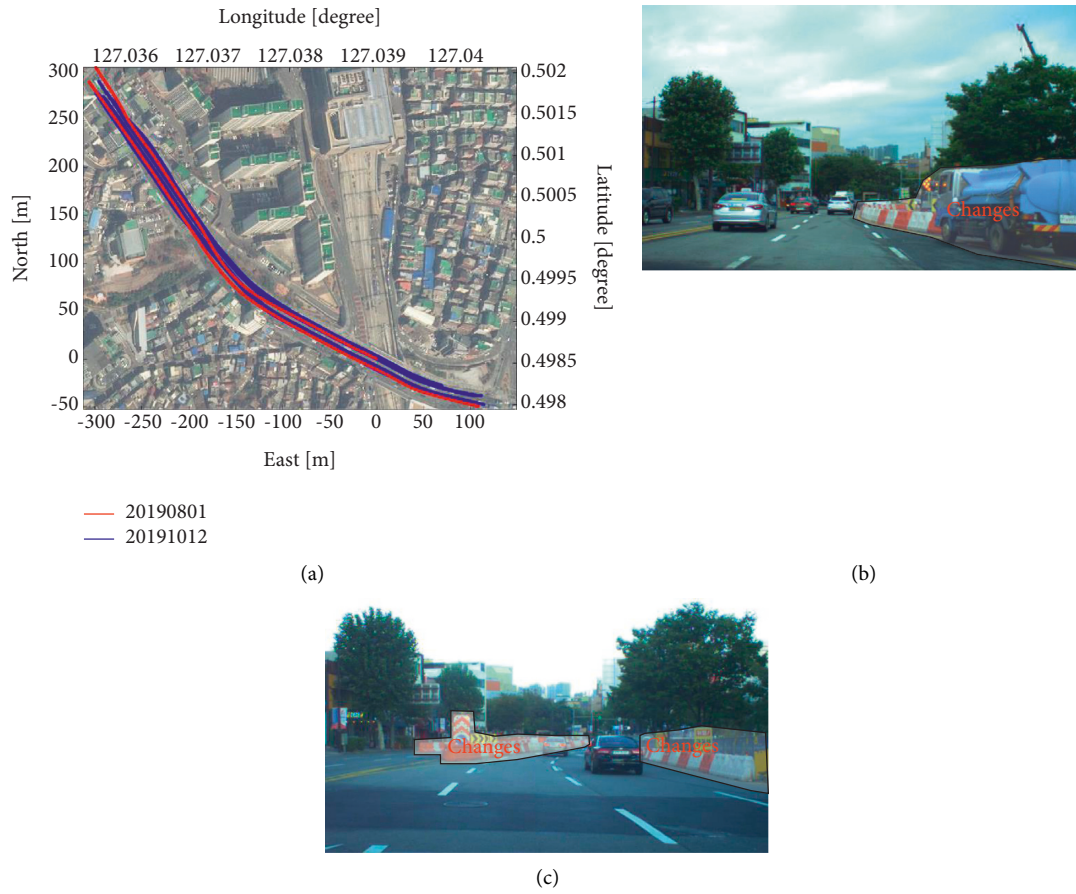


FIGURE 18: Test site for experiment: Wangsimni-ro in Korea. (a) Global trajectories, (b) base map environment at August 1, 2019, and (c) present map environment at October 12, 2019.

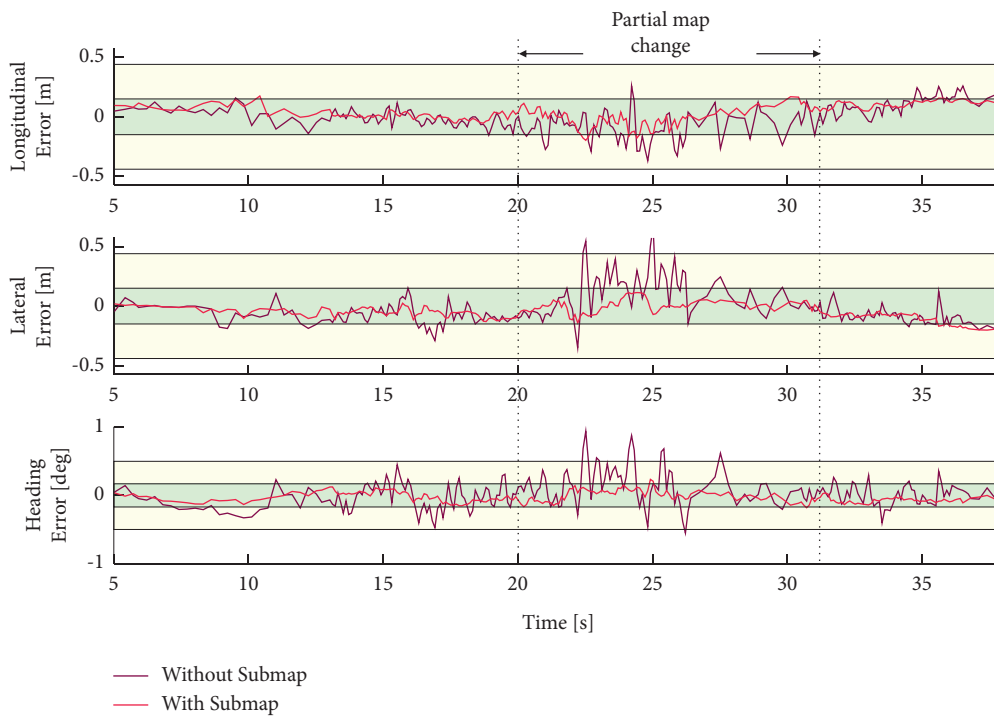


FIGURE 19: Performance of robust localization in map-changing experiments.



TABLE 4: Table of robust localization performance in experiments.

Experiments	Longitudinal (m)		Lateral (m)		Heading ( $^{\circ}$ )	
	95%: 0.15	Max: 0.44	95%: 0.15	Max: 0.44	95%: 0.17	Max: 0.5
Without submap	0.2299	0.36875	0.28506	0.66626	0.41892	0.94196
With submap	0.1478	0.19742	0.13104	0.20154	0.14799	0.2351

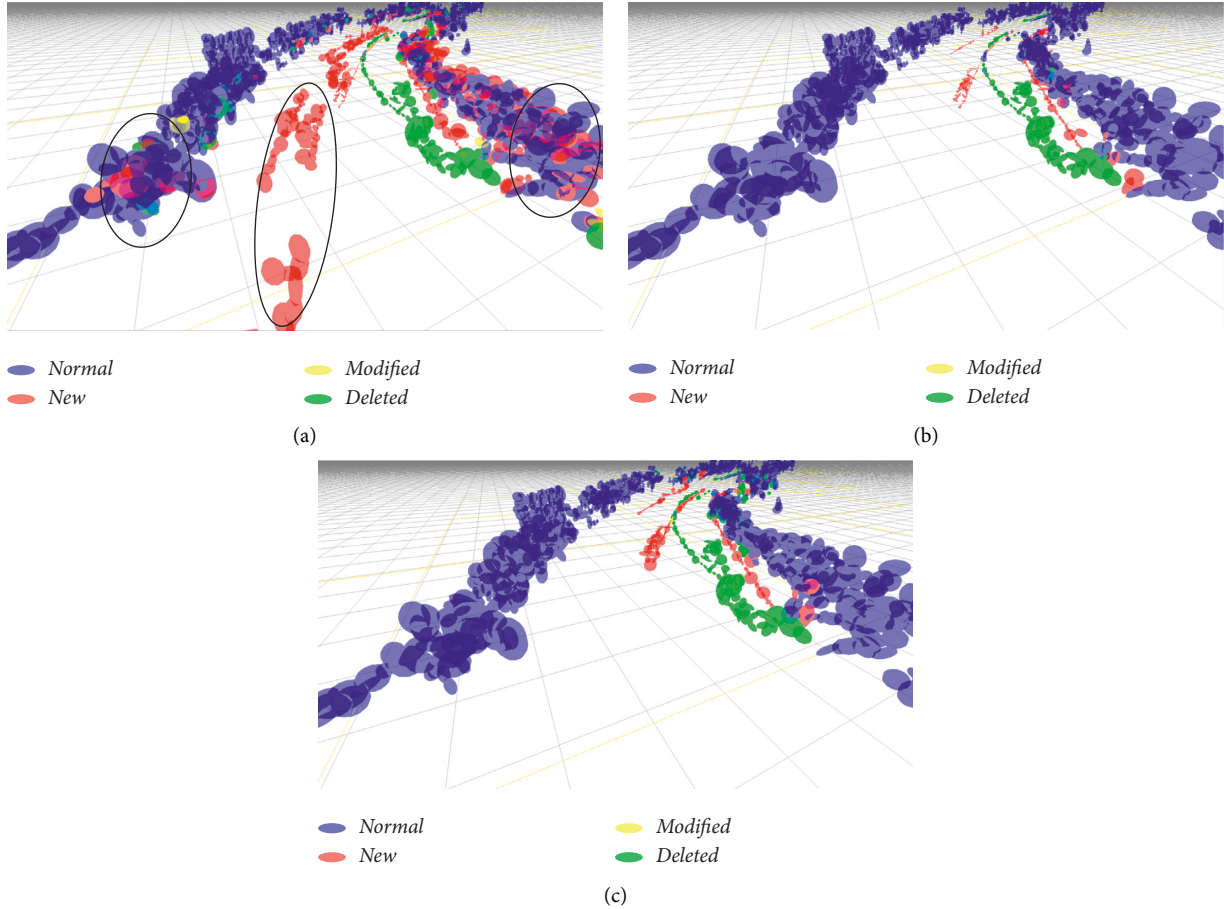


FIGURE 20: Map change information (a) detected by the map change detection algorithm based on single driving, (b) updated by the proposed framework based on crowd-sourced data, and (c) representing ground truth information.

generate more positioning errors in the robust localization process. Especially, the rotation errors can affect the performance of the map change detection and the map update processes rather than the translation errors. In order to relieve effects of rotation errors, the map change detection is performed within 50 m from the ego-vehicle. The results based on the map change detection and the map update are represented in Figures 20(a) and 20(b). Figure 20(c) represents the ground truth of map changes constructing by comparison between the previous GND map and the present GND map.

The performances of proposed algorithms are evaluated using the confusion matrices compared with the ground truth of map changes (Figure 20(c)), as shown in Table 5. The evaluation was performed in the region from the ground to a height of 1.5 m because most of the changes are in that region. The first confusion matrix represents the results of

comparison with the base map. Because the base map does not have any map change information, the precision rates and the recall rates related with the map change classes (*new*, *modified*, and *deleted*) were set to 0. Therefore, the F1-score based on the base map was 37.36%, which is a very low value. The second confusion matrix means the results of the map inferred by the map change detection based on single driving, as shown in Figure 20(a). The F1-score of map change detection was 78.34%. There are several reasons for degrading the accuracy of the map change detection algorithm. First, the precision rate of the *new* class is 35.92% because the dynamic points that are misclassified as static are propagated to the *new* class, as shown in the circles of Figure 20(a). Secondly, the precision rate of the *modified* class is 25.37% because the true *normal* class is misclassified as *modified* class. Because low precision rate is caused by incorrectly constructed new normal distributions in the map

TABLE 5: Confusion matrices about nonupdate, change detection based on single driving, and map update based on crowd-sourced data in experiments.

	Base map			Recall (%)			Map change detection			Recall (%)			Map update			Recall (%)	
	Normal	New	Modified	Deleted	Empty	1999	0	0	43	51	0	0	1998	0	0		1
Truth	1999	0	0	0	0	100	1905	0	43	51	0	0	1998	0	0	1	0
	New	0	0	0	345	0	0	0	0	0	33	0	0	305	0	0	40
	Modified	20	0	0	0	0	3	0	17	0	0	3	0	0	17	0	0
	Deleted	291	0	0	0	0	22	0	7	262	0	25	0	0	4	262	0
	Empty	0	0	0	38507	100	0	610	0	0	37897	0	0	51	0	0	38490
Precision (%)	86.53	0	0	0	99.11	F1-score	98.70	35.92	25.37	83.70	99.99	98.61	85.67	80.95	99.61	99.89	F1-score
						38.51					78.34						92.80

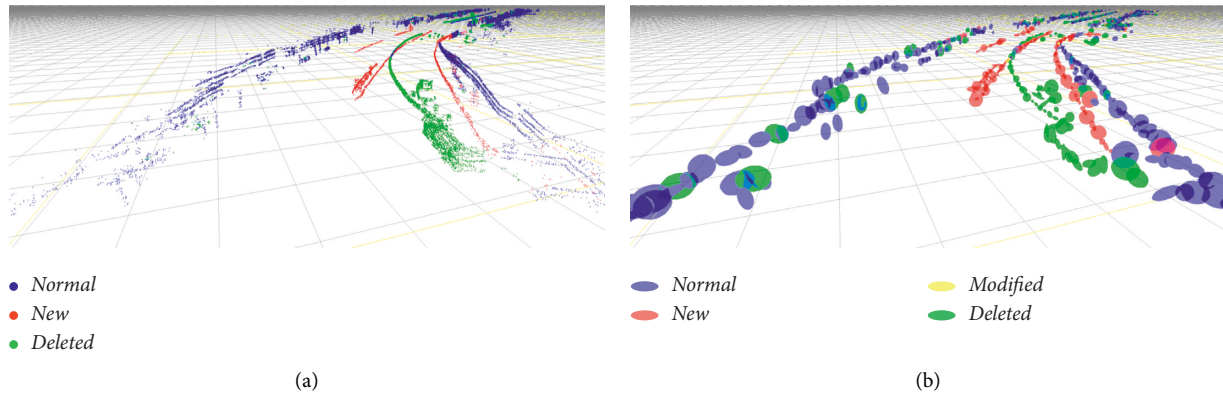


FIGURE 21: The framework of PCD map update constructed (a) the updated PCD map. The map was converted to (b) the updated GND map.

change detection process, the misclassification can be overcome after the crowd-sourced data-based evaluation process in the map cloud server. The third reason is the 83.7% precision rate of the *deleted* class, which is low. This means that the normal states in the previous base map are incorrectly removed. The misclassified voxels can be overcome by applying the change ratio  $\xi_{\text{update}}$  to the merging process.

Based on the crowd-sourcing detection, the confusion matrix, representing the results of the map update based on crowd-sourcing detection (Figure 20(b)), made the F1-score be 92.8%. The precision rate of the *new* class increases dramatically from 32.92% to 85.67% due to the merging process in the map update. Similarly, the precision rate of the *modified* class compared with the precision rate of the map change detection algorithm increases from 25.37% to 80.95%. Because the class predicted as *modified* by the map change detection algorithm is compared with each other, only clear voxels remain. The precision rate of the *deleted* class also increases from 83.7% to 99.61%.

**7.4. Comparison with the PCD Map Update.** In order to compare the performance of the GND map update over the other approach, the framework of the PCD map update based on crowd-sourcing detection was used [44]. The PCD map, used for constructing the base GND map, was used as the base map. The changed information can be detected by the ray-casting approach in each vehicle. The detected map changes, which were composed of only changed points (*new* and *deleted* points), were uploaded to the map cloud server. Using the map change information, the PCD map was finally updated, as shown in Figure 21(a). For the fair evaluation, the same indicator, F1-score based on comparison of normal distributions, was used. Accordingly, the updated PCD map was converted to the updated GND map, as shown in Figure 21(b). The updated GND map was evaluated by comparing with the ground truth of map changes (Figure 20(c)).

Figure 22 shows the results of the PCD map update and the GND map update. The blue line represents the F1-score by the base map. The F1-scores of the base PCD map and the base GND map are the same. The yellow and green lines

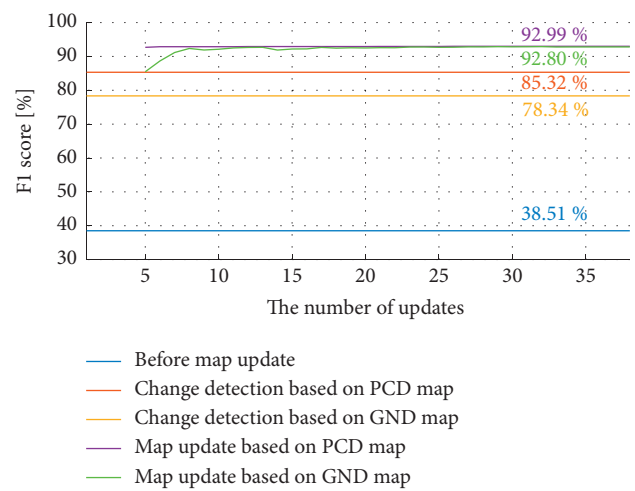


FIGURE 22: F1-scores by the number of crowd-sourcing vehicles in experiments.

represent the F1-scores from the map change detection and the map update based on the GND map. On the other hand, the orange and purple lines represent the F1-score detected by the map change detection and the map update based on the PCD map. In using single driving data, the F1-score based on PCD map update is higher than the F1-score based on the GND map update because the ray-casting based on the PCD map can check the changes more in detail than the ray-casting based on the GND map. Although the degradation of the change detection based on the single driving data, the results of the map update were similar to each other after 16 vehicles due to the crowd-sourcing data.

**7.5. Traffic of Wireless Network Communications.** The uploaded size is dramatically reduced by 96.06% in the proposed framework.

The analysis of the traffic of the wireless networks is represented in Figure 23. The blue, orange, yellow lines in Figure 23 mean the data size which can be transmitted during the vehicle driving through average speeds of 3G, 4G, and 5G wireless networks in Korea. The purple and green lines in Figure 23(a) show the downloaded sizes of the PCD

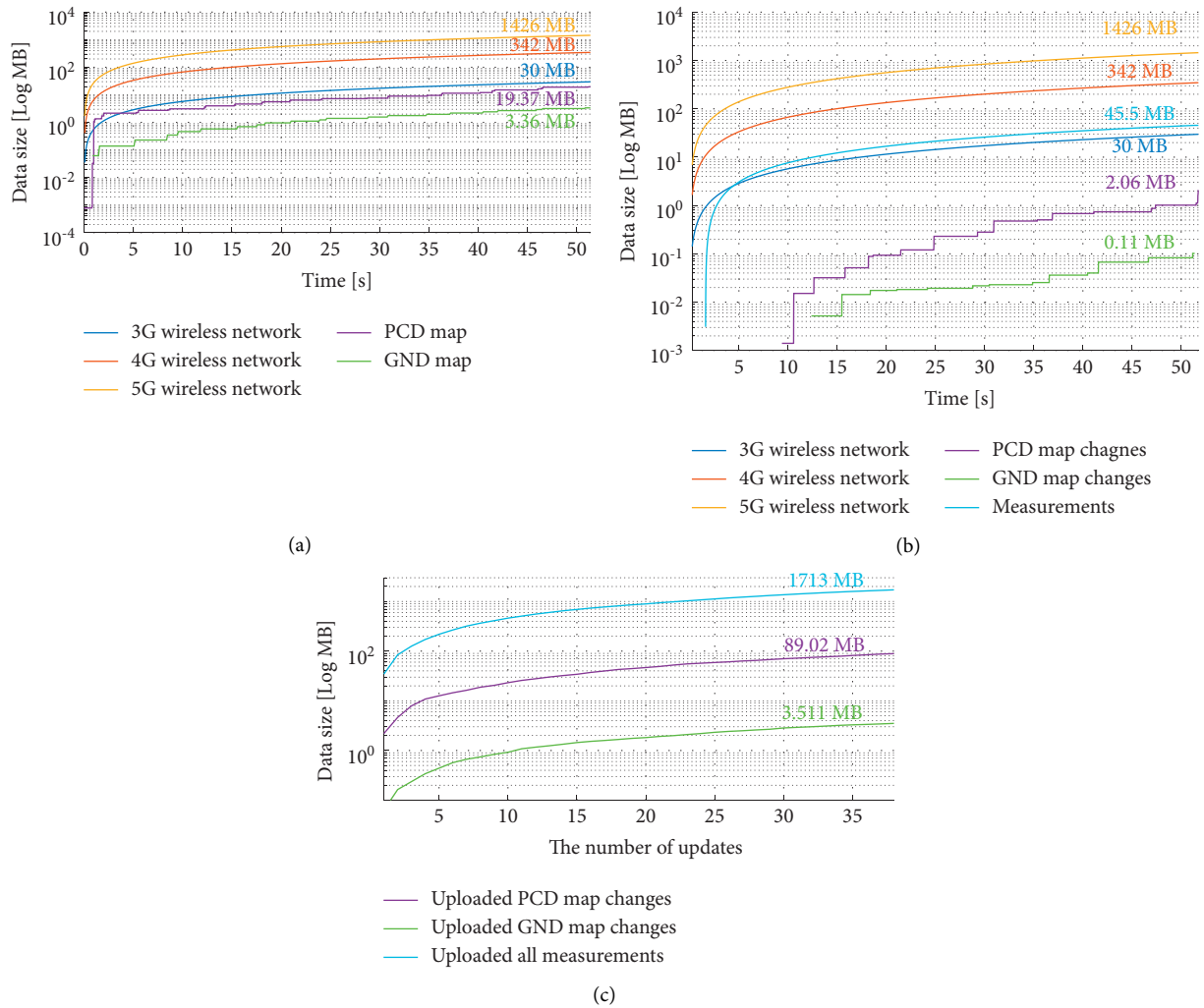


FIGURE 23: Traffic of wireless networks about (a) downloading the base map, (b) uploading map changes from a single vehicle, and (c) uploading map changes from crowd-sourcing vehicles.

map and the GND map during the vehicle driving. While the downloaded PCD map size was 19.37 MB, the downloaded GND map size is 3.36 MB, due to the compression based on the normal distributions. The map size was compressed by 82.6% using the GND map. Since the green line for the GND map is always lower than the blue line of the 3G wireless network, the 3G network can be used for downloading the GND map. The purple and green lines in Figure 23(b) represent the uploaded size of map changes based on the PCD map and the GND map during the vehicle driving. The sky-colored line represents all measurement data (in-vehicle sensors, the low-cost GNSS, and the LiDAR measurements) used to update the map changes similar to the conventional camera-based map update approaches. The size of uploaded PCD map changes, which were composed of *new* and *deleted* points, was 2.06 MB, which was lower than all measurements. In the map change uploading, the adaptation of normal distributions with *new*, *modified* and *deleted* classes dramatically reduced 2.06 MB to 0.11 MB. A similar tendency was represented in Figure 23(c), which shows the total uploaded data from crowd-sourcing vehicles. While the data

size for uploading all measurements and PCD map changes were used until 1713 MB and 89.24 MB during driving of 38 vehicles, the proposed framework used only 3.511 MB. Compared with the PCD map update framework, the uploaded size is dramatically reduced by 96.06% in the proposed framework.

**7.6. Localization Performance after Map Update.** The final goal of the GND map update is to perform precise online localization by keeping the GND map up-to-date. To evaluate the online localization performance, two data are compared, as shown in Figure 24. The brown line is the same as the brown line in Figure 19. Online localization with the previous map cannot satisfy the localization requirements. On the other hand, the red line represents the online localization performance based on the matching with the GND map after the map update. The results of localization before and after the map update were reinterpreted as Table 6. The maximum longitudinal and lateral errors were 0.17868 m and 0.20133 m,

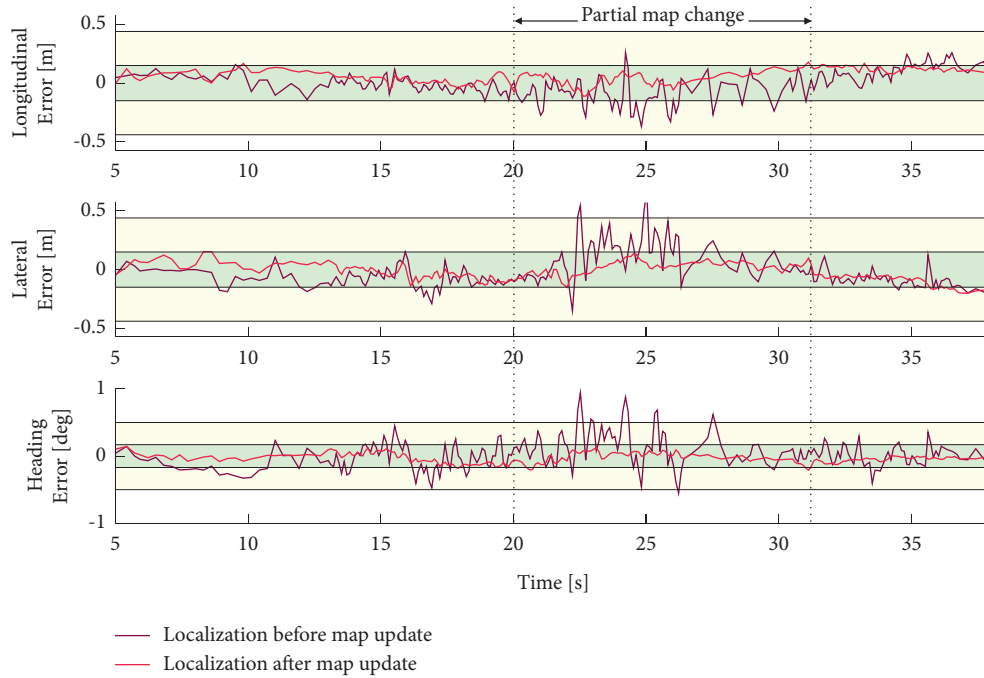


FIGURE 24: Online localization performance after GND map update.

TABLE 6: Table of localization performance before and after map update.

Experiments	Longitudinal (m)		Lateral (m)		Heading (°)	
	95%: 0.15	Max: 0.44	95%: 0.15	Max: 0.44	95%: 0.17	Max: 0.5
Before map update	0.2299	0.36875	0.28506	0.66626	0.41892	0.94196
After map update	0.11053	0.17868	0.14431	0.20133	0.13136	0.20966

respectively, which were within the maximum boundary of 0.44 m. The maximum heading error was  $0.20966^\circ$ , which was within the maximum boundary of  $0.5^\circ$ . In addition, the longitudinal, lateral, and heading 95% errors were 0.11053 m, 0.14431 m, and  $0.20966^\circ$ , respectively. Since the online localization algorithm with the updated map can satisfy the localization requirements, the localization based on the proposed framework can be used for autonomous driving.

## 8. Conclusion

There are some problems that LiDAR-based localization has a critical issue to be solved in order to realize autonomous driving in the future. Since LiDAR-based localization uses the LiDAR map modeling static environments, it can be fragile by the environment changes of static information (i.e., parked vehicle and construction site). Accordingly, it is essential for changes in the real environment to be periodically updated in the LiDAR map for minimization of the difference between the environment and the LiDAR map. However, it is difficult to apply directly to the periodic update because the LiDAR map is too large to be downloaded or uploaded.

In order to solve the problems, the paper proposed the GND map update framework based on crowd-sourcing data.

The proposed framework consists of five steps: downloading, robust localization, change detection, uploading, and map update. The main contributions of the proposed framework are summarized as follows:

- (1) The paper proposed a GND map update framework based on crowd-sourcing detection. Based on the framework, the LiDAR map can be kept up-to-date to support the localization for autonomous driving. While the performance of the proposed framework is similar to the performance of the state-of-the-art framework based on crowd-sourcing data [44], the data size transmitted in the proposed framework is dramatically reduced by 82.6% in the downloading process and 96.06% in the uploading process. Using the updated map, the online LiDAR localization was successfully performed in changing environments.
- (2) The proposed change detection algorithm finds the probability that the environments are changed based on the relationship between the normal distribution and the LiDAR measurements. Accumulation of change information based on the evidence approach detects the deterministic map changes to be uploaded into the map server. Since the only map changes (*new*, *modified*, and *deleted* normal distributions) are uploaded into the map server, the uploaded data size

is reduced than the data size of uploading the changed points.

- (3) The proposed map merging algorithm detects the representative map changes from the crowd-sourcing change information through a DBSCAN algorithm based on the proposed normal-distribution similarity. The representative map changes can update the GND map in the map server by replacing the changing parts.

This framework keeps map information up-to-date without the additional costs generated by professional mapping vehicles with mobile mapping systems. In addition, the map update framework reduced the wireless network burdens dramatically. Despite these innovations, there are some limitations that should be resolved as future works. First, the data size to update the map based on a camera (10 kB/km) is still less than the uploaded data size of the GND map changes (56.32 kB/km). To compress the size of the GND map more, the authors plan to research the GND map compression based on the deep learning approach. Second, the map update framework does not consider the semantic segmentation information of point clouds researched in deep learning fields. To improve the performance of the map update framework, the authors plan to research a map update approach considering the semantic information.

## Data Availability

The dataset cannot be opened because the data are company properties.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the BK21 Plus Program (22A2013000045) under the Ministry of Education, South Korea, in part by the Industrial Strategy Technology Development Program under Grant 10039673, Grant 10060068, and Grant 10079961, the International Collaborative Research and Development Program (N0001992) under the Ministry of Trade, Industry, and Energy (MOTIE Korea), and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MEST) under Grant 2011-0017495 and 2020R1C1C1007739. Following are results of a study on the “Future Transportation Equipment Development of Regional Innovation Platform for Gwangju and Jeollanamdo” Project, supported by the Ministry of Education and National Research Foundation of Korea.

## References

- [1] N. Mattern, R. Schubert, and G. Wanielik, “High-accurate vehicle localization using digital maps and coherency images,”

- in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 462–469, La Jolla, CA, USA, June 2010.
- [2] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo, “Precise localization of an autonomous car based on probabilistic noise models of road surface marker features using multiple cameras,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3377–3392, 2015.
- [3] X. Chen, W. Hu, L. Zhang, Z. Shi, and M. Li, “Integration of low-cost gnss and monocular cameras for simultaneous localization and mapping,” in *Sensors*, vol. 18, no. 7, p. 2193, Multidisciplinary Digital Publishing Institute, 2018.
- [4] J. Levinson, M. Montemerlo, and S. Thrun, “Map-based precision vehicle localization in urban environments,” in *Robotics: Science and Systems*, IEEE, 2007.
- [5] J. Levinson and S. Thrun, “Robust vehicle localization in urban environments using probabilistic maps,” in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4372–4378, IEEE, Anchorage, AK, USA, May 2010.
- [6] J. Choi, “Hybrid map-based SLAM using a Velodyne laser scanner,” in *Proceedings of the 2014 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 3082–3087, IEEE, Qingdao, China, October 2014.
- [7] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, “A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, IEEE, 2018.
- [8] S. Sood, “Self-healing maps for autonomous driving,” in *Proceedings of the 2017 NVIDIA GPU Technology Conference*, San Jose Convention Center, Silicon Valley, San Jose, CA, USA, 2017.
- [9] TomTom, “Qualcomm and TomTom to Crowdsource High-Definition Map Data,” <https://www.roadtraffic-technology.com/news/newsqualcomm-and-tomtom-to-crowd-source-high-definition-map-data-5752324/>.
- [10] Mobileye, “Road experience management (REM),” [Online]. Available: <https://www.mobileye.com/our-technology/rem/>.
- [11] Bosch, “Bosch road signature,” <https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/automated-driving/road-signature/>.
- [12] Daimler, “Daimler and HERE to bring HD Live Map to future Mercedes-Benz models,” <https://media.daimler.com/marsMediaSite/en/instance/ko/Daimler-and-HERE-to-bring-HD-Live-Map-to-future-Mercedes-Benz-models.xhtml?oid=33472763>.
- [13] C. Kim, K. Jo, B. Bradai, and M. Sunwoo, “Multiple vehicles based new landmark feature mapping for highly autonomous driving map,” in *Proceedings of the 2017 14th Workshop on Positioning, Navigation and Communications (WPNC)*, vol. 2018, pp. 1–6, Bremen, Germany, 2017.
- [14] C. Kim, S. Cho, M. Sunwoo, and K. Jo, “Crowd-sourced mapping of new feature layer for high-definition map,” in *Sensors*, vol. 18, no. 12, p. 4172, Multidisciplinary Digital Publishing Institute, 2018.
- [15] K. Jo, C. Kim, and M. Sunwoo, “Simultaneous localization and map change update for the high definition map-based autonomous driving car,” *Sensors*, vol. 18, no. 9, p. 3145, 2018.
- [16] C. Kim, S. Cho, and M. Sunwoo, “A geodetic normal distribution map for long-term LiDAR localization on earth,” *IEEE Access*, vol. 9, pp. 470–484, Bremen, Germany, 2020.
- [17] A. D. Stewart and P. Newman, “Laps-localisation Using Appearance of Prior Structure: 6-dof Monocular Camera Localisation Using Prior Pointclouds,” in *Proceedings of the 2012 IEEE International Conference on Robotics and*

- Automation (ICRA)*, pp. 2625–2632, Saint Paul, MN, USA, May 2012.
- [18] W. Maddern, G. Pascoe, and P. Newman, “Leveraging experience for large-scale LIDAR localisation in changing cities,” in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1684–1691, Seattle, WA, USA, May 2015.
- [19] Y. Xu, V. John, and S. Mita, “3D point cloud map based vehicle localization using stereo camera,” in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 487–492, Los Angeles, CA, USA, June 2017.
- [20] Z. Liu, H. Chen, H. Di et al., “Real-time 6d lidar slam in large scale natural terrains for ugv,” in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 662–667, Changshu, China, June 2018.
- [21] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping,” in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 321–328, San Francisco, CA, USA, April 2000.
- [22] G. Grisetti, C. Stachniss, W. Burgard, and others, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, p. 34, 2007.
- [23] C. Kim, S. Cho, C. Jang, M. Sunwoo, and K. Jo, “Evidence filter of semantic segmented image from around view monitor in automated parking system,” *IEEE Access*, vol. 7, Article ID 92791, 2019.
- [24] S. Cho, C. Kim, K. Jo, and M. Sunwoo, “A GPU accelerated particle filter based localization using 3D evidential voxel maps,” *SAE Technical Paper 2019-01-0491*, 2019.
- [25] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, “Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations,” *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, Sage Publications Sage UK, London, England, 2012.
- [26] E. Javanmardi, M. Javanmardi, Y. Gu, and S. Kamijo, “Adaptive resolution refinement of NDT map based on localization error modeled by map factors,” in *Proceedings of the 2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2237–2243, IEEE, Maui, HI, USA, November 2018.
- [27] R. W. Wolcott and R. M. Eustice, “Fast LIDAR localization using multiresolution Gaussian mixture maps,” in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2015, pp. 2814–2821, IEEE, Seattle, WA, USA, May 2015.
- [28] R. W. Wolcott and R. M. Eustice, “Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving,” *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017.
- [29] D. Girardeau-Montaut, M. Roux, R. Marc, and Guillaume Thibault, “Change detection on points cloud data acquired with a ground laser scanner,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, 2005.
- [30] L. Wellhausen, D. Renaud, G. Abel, S. Roland, and C. Cesar, “Reliable real-time change detection and mapping for 3D LiDARs,” in *Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp. 81–87, IEEE, Shanghai, China, October 2017.
- [31] J. P. Underwood, D. Gillsjo, T. Bailey, and V. Vlaskine, “Explicit 3D change detection using ray-tracing in spherical coordinates,” in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4735–4741, Karlsruhe, Germany, May 2013.
- [32] F. Ferri, M. Gianni, M. Menna, and F. Pirri, “Dynamic obstacles detection and 3D map updating,” in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5694–5699, IEEE, Hamburg, Germany, October 2015.
- [33] W. Xiao, B. Vallet, and N. Paparoditis, “Change detection in 3D point clouds acquired by a mobile mapping system,” *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II, pp. 331–336, 2013.
- [34] X. Wen, B. Vallet, M. Brédif, and N. Paparoditis, “Street environment change detection from mobile laser scanning point clouds,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 107, pp. 38–49, Elsevier, 2015.
- [35] W. Burgard, C. Stachniss, and D. Hähnel, “Mobile robot map learning from range data in dynamic environments,” *Autonomous Navigation in Dynamic Environments*, vol. 35, pp. 3–28, 2007.
- [36] T. Krajník, J. P. Fentanes, M. Hanheide, and T. Duckett, “Persistent localization and life-long mapping in changing environments using the frequency map enhancement,” in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4558–4563, Daejeon, Republic of Korea, October 2016.
- [37] T. Krajník, J. P. Fentanes, J. M. Santos, and T. Duckett, “Fremen: frequency map enhancement for long-term mobile robot autonomy in changing environments,” *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 964–977, IEEE, 2017.
- [38] G. Trehard, Z. Alsayed, E. Pollard, B. Bradai, and F. Nashashibi, “Credibilist simultaneous localization and mapping with a LIDAR,” in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2699–2706, Iros, Chicago, IL, USA, September 2014.
- [39] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi, “On line mapping and global positioning for autonomous driving in urban environment based on evidential SLAM,” in *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 814–819, Montreal, QC, Canada, May 2015.
- [40] U. Katsura, K. Matsumoto, A. Kawamura, and T. Ishigami, “Spatial change detection using voxel classification by normal distributions transform,” in *Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2953–2959, 2019.
- [41] J. P. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “3D normal distributions transform occupancy maps: an efficient representation for mapping in dynamic environments,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1627–1644, 2013.
- [42] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, “Normal distributions transform occupancy maps: application to large-scale online 3D mapping,” in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2233–2238, IEEE, Karlsruhe, Germany, May 2013.
- [43] E. Einhorn and H. Michael Gross, “Generic NDT mapping in dynamic environments and its application for lifelong SLAM,” in *Robotics and Autonomous Systems*, vol. 69, no. 1, pp. 28–39, Elsevier, 2015.
- [44] C. Kim, S. Cho, M. Sunwoo, P. Resende, B. Bradai, and K. Jo, “Updating point cloud layer of high definition (HD) map

- based on crowd-sourcing of multiple vehicles installed LiDAR,” *IEEE Access*, vol. 9, p. 1, 2021.
- [45] D. Xue, N. Yang, X. Zhao, and Z. Wang, “Point-cloud map update for connected and autonomous vehicles based on vehicle infrastructure cooperation: framework and field experiments,” in *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 2062–2067, Indianapolis, IN, USA, September 2021.
- [46] F. Abrate, B. Bona, M. Indri, S. Rosa, and F. Tibaldi, “Multi-robot map updating in dynamic environments,” *Distributed Autonomous Robotic Systems*, vol. 83, pp. 147–160, Springer, Berlin, Germany, 2012.
- [47] M. Labbé and F. Michaud, “Long-term online multi-session graph-based SPLAM with memory management,” in *Autonomous Robots*, vol. 42, pp. 1133–1150, Springer US, no. 6, 2018.
- [48] N. Shaik, T. Liebig, C. Kirsch, and H. Müller, “Dynamic map update of non-static facility logistics environment with a multi-robot system,” in *Advances in Artificial Intelligence*, G. Kern-Isberner, J. Fürnkranz, and M. Thimm, Eds., Lecture Notes in Computer Science, vol. 10505, pp. 249–261, Springer, Cham, 2017.
- [49] K. Jo, S. Cho, C. Kim et al., “Cloud update of tiled evidential occupancy grid maps for the multi-vehicle mapping,” in *Sensors*, vol. 18, no. 12, p. 4119, Multidisciplinary Digital Publishing Institute, 2018.
- [50] K. Jo, S. Lee, C. Kim, and M. Sunwoo, “Rapid motion segmentation of LiDAR point cloud based on a combination of probabilistic and evidential approaches for intelligent vehicles,” *Sensors*, vol. 19, no. 19, p. 4116, 2019.
- [51] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, IBM, 1965.
- [52] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, pp. 226–231, Portland Oregon, August 1996.
- [53] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, “Appearance-based loop detection from 3D laser data using the normal distributions transform,” in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 23–28, Kobe, Japan, May 2009.
- [54] T. F. Chan, “Gene Howard Golub and Randall J LeVeque. “Updating formulae and a pairwise algorithm for computing sample variances”” in *Proceedings of the COMPSTAT 1982 5th Symposium held at Toulouse*, pp. 30–41, Toulouse, Franch, 1982.
- [55] S. Cho, C. Kim, M. Sunwoo, and K. Jo, “Robust localization in map changing environments based on hierarchical approach of sliding window optimization and filtering,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 1–7, 2020.
- [56] T. G. R. Reid, S. E. Houts, R. Cammarata et al., “Localization requirements for autonomous vehicles,” pp. 1–16, 2019, 10.4271/12-02-03-0012 <https://arxiv.org/abs/1906.01061>.



## Research Article

# Yolov4 High-Speed Train Wheelset Tread Defect Detection System Based on Multiscale Feature Fusion

Changfan Zhang <sup>1</sup>, Xinliang Hu <sup>2</sup>, Jing He <sup>1</sup> and Na Hou<sup>1</sup>

<sup>1</sup>School of Electrical and Information Engineering, Hunan University of Technology, Zhuzhou 412007, Hunan, China

<sup>2</sup>School of Railway Transportation, Hunan University of Technology, Zhuzhou 412007, Hunan, China

Correspondence should be addressed to Jing He; hejing@263.net

Received 24 September 2021; Revised 20 January 2022; Accepted 28 January 2022; Published 27 March 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Changfan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Yolov4 detection algorithm does not sufficiently extract local semantic and location information. This study aims to solve this problem by proposing a Yolov4-based multiscale feature fusion detection system for high-speed train wheel tread defects. First, multiscale feature maps are obtained from a feature extraction backbone network. The proposed multiscale feature fusion network then fuses the underlying features of the original three scales. These fused features contain more defect semantic information and location details. Based on the fused features, a path aggregation network is used to fuse feature maps at different resolutions, with an improved loss function that speeds up the convergence of the network. Experimental results show that the proposed method is effective at detecting defects in the wheel treads of high-speed trains.

## 1. Introduction

High-speed train fault diagnosis is crucial for the safe operation and maintenance of high-speed trains. As an important support and running component of high-speed trains, wheelsets incur tread wear, scratches, and other damages caused by rolling contact between the wheel and rail. The deterioration can easily cause serious damage, such as wheelset tread fractures. Therefore, it is essential to diagnose wheelset tread damage to ensure the safety of high-speed trains.

Current defect detection methods include magnetic particle detection, ultrasonic detection, and machine vision detection [1]. In recent years, many scholars have studied machine vision methods [2–5], owing to their wide application range and high precision. Such methods are efficient at inspecting the damage and unaffected by the contour of the inspection part. Moreover, machine vision inspection can proceed automatically. Traditional defect detection methods based on machine vision for railway components begin from the perspective of image processing. On the one hand, they use low-level grayscale features [6], textures,

colors, frequencies, and other features to detect defects, yet such artificial features must be selected manually and require rich expert knowledge. On the other hand, the damage is automatically located after image enhancement [7–9], although the images are susceptible to noise. Given the complex operating scenarios of high-speed rail, changing working conditions, and the nonlinearity of the sensor itself, the signals collected by the sensors often contain foreground interference, noisy backgrounds, and nonlinear characteristics such as corrosion, stains, uneven reflection, a low signal-to-noise ratio, excessive illumination, and uneven illumination. It is difficult for traditional machine vision defect detection methods to effectively extract small fault features under foreground interference and noise. Furthermore, the differences between wheelset tread defects are not obvious under these two conditions.

*1.1. Defect Detection Based on a Deep Convolutional Neural Network (DCNN).* Considering that deep learning has exhibited superiority in feature extraction and pattern recognition, an increasing number of scholars have

attempted to apply deep learning methods to defect detection [10]. Deep learning based on convolutional neural networks is widely used to detect railway component damage [11–13]. Faghih-Roohi et al. [14] proposed a DCNN with multiple structures and activation properties for rail damage detection. High-speed train wheelset tread defect detection requires scene analysis at the regional level. Current regional-level target detection methods based on DCNNs are generally divided into two types. One type is two-stage detection based on the region, such as the Regions with CNN (R-CNN) features [15] and Faster R-CNN [16]. Liu et al. [12] integrated the feature extraction module isoelectric line network (ILNET) in the Faster R-CNN and used the segmentation method intersecting the cortical model/maximization of the posterior marginal based on a Markov random field to locate and divide loose strands of isoelectric lines. He et al. [17] proposed an end-to-end two-step defect detection method for steel rolling defects and explored the trade-off between detection speed and accuracy for different numbers of regions. However, the detection speed of two-step algorithms is still slower than that of single-step detection algorithms, and it is difficult to deal with the problem of the short maintenance operation time for China's high-speed railways. The other type is a series of single-step detection algorithms, including the you only look once (YOLO) series [18], which includes single-shot multibox detector (SSD) [19] and RetinaNet [20]. Kou et al. [21] introduced the DenseNet module in Yolov3 for strip defects and proposed a strip defect detection method based on Yolov3. Based on RetinaNet, Cheng et al. [22] proposed a retinal network (DEA RetinaNet) defect detection model based on differential channel attention and adaptive spatial feature fusion. Cui et al. [23] proposed an SSDNet for defect detection, which solved the problem of large texturing and small size defect detection by introducing feature retention blocks and skip dense connection modules. Considering that the YOLO series of target detection algorithms use fast detection frameworks, this study focuses on the application of Yolov4 [24] in wheelset tread defect detection.

The YOLO series includes Yolov4. Yolov1 locates targets based on the last convolution map. Yolov2–Yolov4 [25,26] only locate large, medium, and small targets from three-scale high-level feature maps. Generally, a CNN acts as a filter in deep neural networks. Semantic information and location details of the wheelset tread image change layer by layer with the filter. Shallow features are rich in location information, but their discrimination is inadequate. Deep features contain ample semantic information but at the cost of location details.

*1.2. Multiscale Feature Fusion (MFF).* Some scholars have proposed using high-quality feature learning for networks by combining the features of different scales. Yang et al. [27] designed a multiscale channel compression deep surface defect detection algorithm that generates multiscale features through the convolutional layers of different sizes of cores to address messy backgrounds and defects of various scales. The added convolutional layer is compressed to increase the

speed of the network. Hu et al. [28] proposed a spatio-temporal segmentation model with a hybrid multidimensional feature fusion structure for automatic thermal imaging defect detection. An attention module was designed that encourages local interaction between adjacent pixels and calibrates the feature map self-adaptively to lighten the model. Gao et al. [29] used feature acquisition and a compression network for multiscale feature fusion of IBD defect detection and used Gaussian weighted pooling instead of ROI pooling. Their method provides more accurate defect location information.

*1.3. Transfer Learning.* Deep neural networks require a large number of datasets as a drive. For wheelset treads, building a large target detection dataset is very difficult. If the dataset is too small, then the performance of the deep neural network will be limited. Therefore, pretraining the network or transfer learning is commonly used for small samples. Yang et al. [30], Zhang et al. [31], Badmos et al. [32], and Sun et al. [33] used transfer learning to detect Mura defects in LCD panels, PCB defects, electrode defects in lithium batteries, and surface defects in metal parts. Kim et al. [34] compared the effects of fine tuning-based transfer learning and training the network from scratch on the DAGM defect dataset. They demonstrated that transfer learning outperforms training the network from scratch.

The contributions of this study are as follows:

- (1) This study proposes an MFF-Yolov4 high-speed train wheelset tread defect detection algorithm. Yolov4 uses only three high-level features to perform detection tasks, resulting in low detection accuracy. To address this limitation, we propose a multiscale feature fusion module. The proposed module provides rich semantic information and location details after the feature extraction stage. Low-level features are integrated into high-level features, and the fused features effectively improve the classification and localization capabilities of the detection network.
- (2) A dataset that contains 277 high-speed train wheelset images is built by collecting data to fine-tune our pretrained model. In the case of small samples, the proposed MFF-Yolov4 achieves competitive performance on this dataset.
- (3) Based on the two-class detection problem—and considering that noisy background candidate frames that are not related to wheelset tread defects contribute to most of the loss—the classification loss function is optimized, and the adjustment factor  $\alpha$  is added, thereby achieving a better anti-interference performance of the improved network.

## 2. Construction of Defect Detection Model for Wheelset Treads

The proposed MFF-Yolov4 algorithm is described in this section (see Figure 1). A single wheelset tread image of any size is processed by CSPDarknet53 and SPP for feature

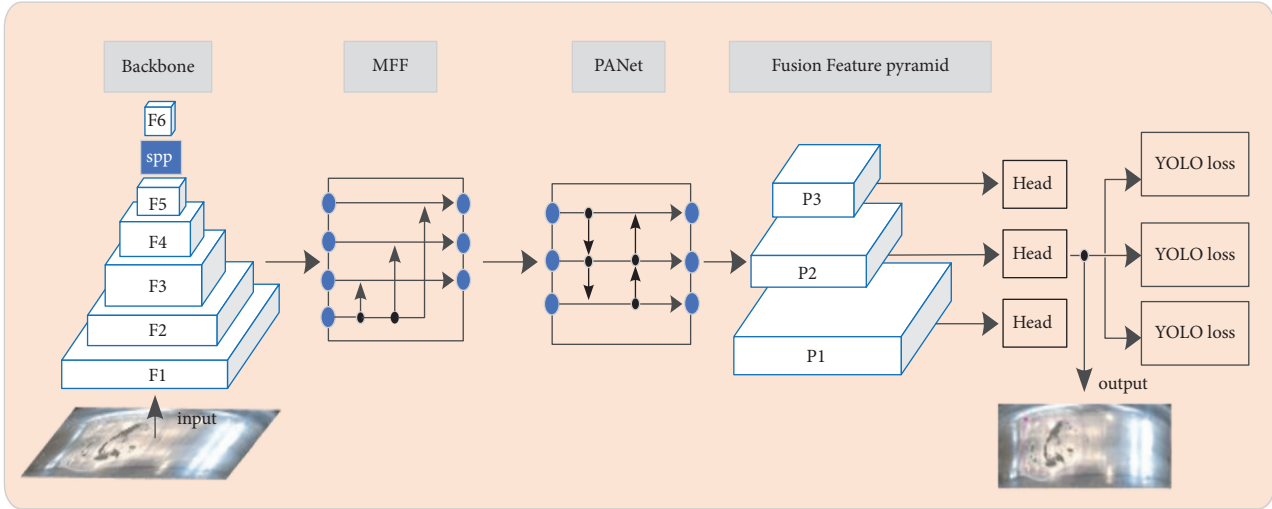


FIGURE 1: MFF-Yolov4 framework.

extraction, and a convolutional feature map at each stage is generated. Four feature maps are extracted and then merged into three-dimensional feature outputs through MFF. In this way, MFF contains both the bottom-level location information of the wheelset tread image and the high-level classification semantics. Subsequently, a path aggregation network (PANet) [35] is used to perform a secondary fusion of the three fusion features with underlying location information. Next, the three fusion features at different scales are divided into grids to predict the bounding box for each grid and then return to the basic truth box. Each scale feature map stores the defect categories (“classes”), regression parameters of prior boxes (“location”), and confidence scores (“confidence”) that correspond to the three prior boxes of each grid.

**2.1. Backbone Network.** Preprocessing the model on the VOC2007 dataset can improve the performance of the deep neural network. The preprocessing model can be fine-tuned on a smaller wheelset tread defect dataset. Given that MFF-Yolov4 is based on the Yolov4 target detection algorithm, the use of CSPDarknet53 and SPP as the backbone has the following characteristics:

- (1) According to research by the author of Yolov4, CSPDarknet53 has the following advantages over the CSPResNeXt50 network: a higher input network size, which is conducive to the detection of small objects; a larger receptive field that covers a larger input network; more parameters to improve the model with single images; and the ability to detect multiple objects of different sizes.
- (2) SPP can significantly increase the receptive field and isolate important context features. Furthermore, it will not increase the operating speed of the network.

In this study, CSPDarknet53 and SPP are selected as the backbone. The detailed structure of the network is shown in Table 1 and the output feature of the last layer of each CSP module is expressed as {F1, F2, F3, F4, F5, F6}.

TABLE 1: Feature extraction backbone network.

Block name	Type	Filter size	Output size
CBM	Conv1	$3 \times 3, 32, \text{stride} = 1$	$416 \times 416$
	Conv2	$3 \times 3, 64, \text{stride} = 2$	$208 \times 208$
Csp1(F1)	RS1	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 64 \end{bmatrix} \times 1$	$208 \times 208$
Csp2(F2)	Conv3	$3 \times 3, 128, \text{stride} = 2$	$104 \times 104$
	RS2	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$104 \times 104$
Csp3(F3)	Conv4	$3 \times 3, 256, \text{stride} = 2$	$52 \times 52$
	RS3	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 256 \end{bmatrix} \times 8$	$52 \times 52$
Csp4(F4)	Conv5	$3 \times 3, 512, \text{stride} = 2$	$26 \times 26$
	RS4	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 512 \end{bmatrix} \times 8$	$26 \times 26$
Csp5(F5)	Conv6	$3 \times 3, 1024, \text{stride} = 2$	$13 \times 13$
	RS5	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 1024 \end{bmatrix} \times 4$	$13 \times 13$
CBL	Conv7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 1024 \\ 1 \times 1, 512 \end{bmatrix} \times 1$	$13 \times 13$
SPP	MaxPool	$\begin{bmatrix} 1 \times 1 \\ 5 \times 5 \\ 9 \times 9 \\ 13 \times 13 \end{bmatrix} \times 1$	$13 \times 13$
CBL(F6)	Conv8	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 1024 \\ 1 \times 1, 512 \end{bmatrix} \times 1$	$13 \times 13$

**2.2. Multiscale Feature Fusion.** Yolov4 uses only three high-level scale features for feature extraction. The evolution of wheelset tread damage is a coupled development process. The network proceeds from shallow to deep, features are mapped to high-dimensional space, and the overall semantic information is gradually strengthened in abstraction. However, the hidden positioning information and local semantic features are gradually weakened layer by layer. In order to strengthen the detection ability of Yolov4

to detect wheel tread damage, we need to extend each single-scale feature to a dual-scale fusion feature. The method we use here is to fuse the high-resolution underlying features with high-level features through convolution transformation. Two basic conditions must be met: the features must be nonadjacent, because adjacent features must have high similarity [36], and the features of each scale generated by the backbone network should be taken into account.

From Table 1, we know that when the image flows through the feature extraction backbone network,  $\{F1, F2, F3, F4, F5, F6\}$  multigrid scale features are generated. The original Yolov4 network uses only three high-level feature maps,  $\{F3, F4, F6\}$ . In order to integrate multiple scale features to obtain more comprehensive semantic information and location details, the proposed MFF module performs a fusion strategy on the underlying feature F2. In particular, F2 is connected to the same F3, F4, and F6 after L2 normalization. By modifying the number of filters in the  $1 \times 1$  convolution, most MFFs reduce the required parameters. This operation may affect the accuracy but will prevent overfitting in the case of insufficient training data. Finally, by stitching the features together, multiscale fusion features  $\{FF1, FF2, FF3\}$  are generated, as shown in Algorithm 1. Its structure is shown in Figure 2.

**2.3. Path Aggregation Network (PANet).** For  $FF_i$  with more comprehensive feature information, the three fusion feature layers of the input are stacked through PANet's step-by-step upsampling and downsampling. A second fusion is performed to obtain three effective feature layers, namely, P1, P2, and P3, as shown in Figure 3.

**2.4. Yolo Head.** In particular, Yolo head is integrated through a  $3 \times 3$  convolution, and then, a  $1 \times 1$  convolution is used to obtain an  $S \times S \times 3(4 + 1 + K)$  tensor, where the four coordinates of the bounding box, namely,  $b_x$ ,  $b_y$ ,  $b_w$ , and  $b_h$ , are stored in "4," the confidence of the detected object is stored in "1," "k" is the detected object category, and the calculation formula is presented as follows:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x, \\ b_y &= \sigma(t_y) + c_y, \\ b_w &= p_w e^{t_w}, \\ b_h &= p_h e^{t_h}, \end{aligned} \quad (1)$$

$$\text{Confidence} = P(\text{object}) \times \text{IoU}_{\text{pred}}^{\text{truth}}, \quad P(\text{object}) \in \{0, 1\}. \quad (2)$$

The coordinate system is established with the upper-left corner of the sample as the origin;  $c_x$  and  $c_y$  represent the coordinates from the upper-left corner of the bounding box to the origin;  $p_w$  and  $p_h$  are the width and height of the

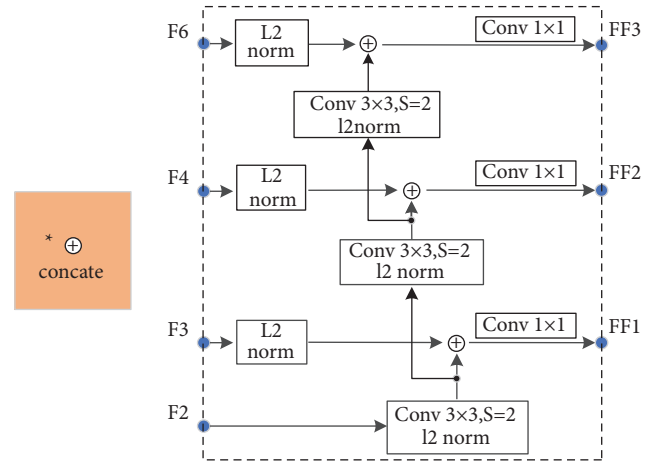


FIGURE 2: MFF structure.

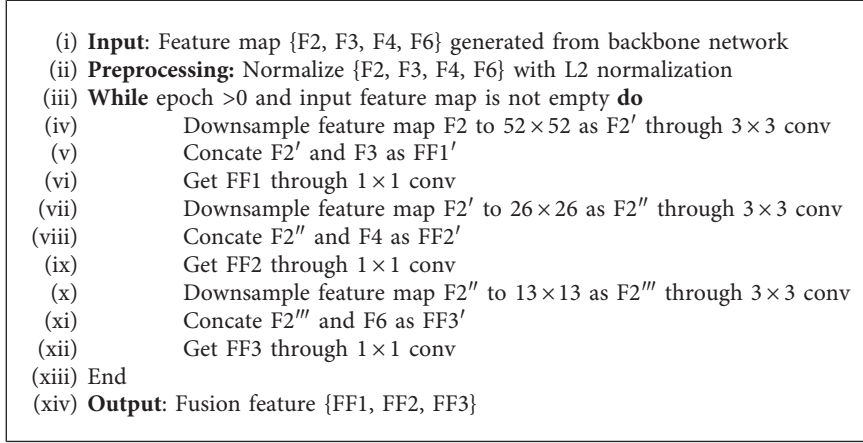
anchor box, respectively;  $t_x$  and  $t_y$  are the distance from the center point of the bounding box to the origin;  $t_w$  and  $t_h$  are the width and height of the bounding box, respectively;  $b_x$  and  $b_y$  are the coordinates of the center point of the prediction box;  $b_w$  and  $b_h$  are the width and height of the prediction box, respectively; and  $\sigma(\cdot)$  is the sigmoid activation function. In equation (2),  $P(\text{object})$  represents the probability that the prediction frame contains the target object, and  $\text{IoU}_{\text{pred}}^{\text{truth}}$  is the intersection and union ratio of the truth frame and the prediction frame. Given that the Yolo head can extract thousands of prediction boxes, greedy nonmaximum suppression (NMS) is often used to eliminate areas with high overlap. The threshold of NMS is set to 0.5, and bounding boxes below 0.5 are discarded. After NMS, the remaining prediction boxes are used to fine-tune our MFF-Yolov4 network.

**2.5. Improved Loss Function.** The MFF-Yolov4 loss function can be divided into three parts, as shown in Figure 4. Hence, the loss function is calculated as follows:

$$L = E_{\text{ciou}} + E_{\text{coord}} + E_{\text{cls}}, \quad (3)$$

where  $E_{\text{ciou}}$  is the location error,  $E_{\text{coord}}$  is the confidence error, and  $E_{\text{cls}}$  is the classification error.

The proposed algorithm is designed to detect defects in the tread surface of high-speed train wheels. In the actual operating environment of high-speed trains, however, the wheelsets are exposed to natural light, stains, rust, and other interference in long-term operation under various working conditions, and the background unrelated to the detected target contributes more classification loss  $E_{\text{cls}}$  in the total loss. As such, we draw on the idea of focal loss [20] and add an index adjustment factor  $\alpha$  to the classification loss to improve its ability to distinguish the foreground and background. The improved classification loss is expressed as follows:



ALGORITHM 1: Multiscale feature fusion.

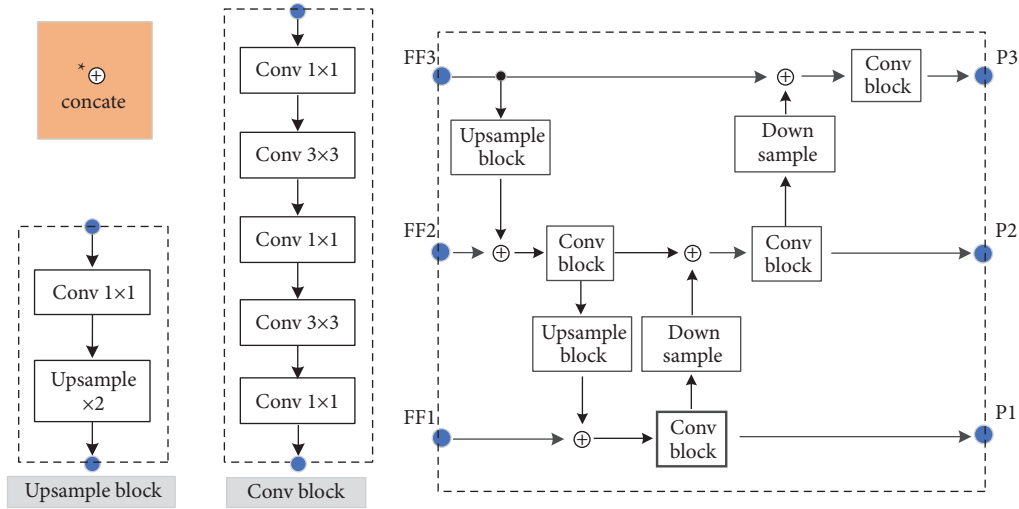


FIGURE 3: PANet structure.

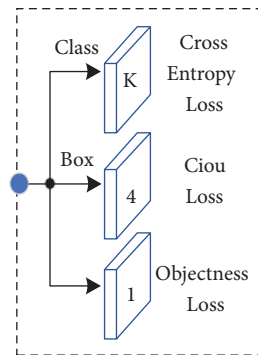


FIGURE 4: Yolo loss.

$$E_{cls} = \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{\text{defect}} \sum_{c \in \text{classes}} [p_i(c) \log(\hat{p}_i(c)) + (1 - p_i(c))^\alpha \log(1 - \hat{p}_i(c))], \quad (4)$$

where  $S \times S$  represents the number of grid cells divided by the input image; B represents the number of bounding boxes

generated by the cell; and  $I_{ij}^{\text{defect}}$  is the locked activation parameter item. To judge whether the  $j$ -th prior box in the  $i$ -

th grid is responsible for predicting the target,  $I_{ij}^{\text{defect}} = 1$  when the wheelset tread damage target falls to the  $j$ -th bounding box generated by the  $i$ -th mesh; otherwise,  $I_{ij}^{\text{defect}} = 0$ . Therefore, only the grid responsible for predicting the target needs to punish the classification error.  $p_i(c)$  represents the true probability of category  $c$ , and  $\hat{p}_i(c)$  represents the predicted probability of category  $c$ .

### 3. Experiments

MFF-Yolov4 was evaluated on our self-built dataset, called WT-DET. The implementation results show that the model we designed is feasible and effective.

#### 3.1. Wheelset Tread Defect Dataset

**3.1.1. Dataset Collection.** The image samples needed in the experiment were collected from the wheel axle workshop of CRRC Group Co., Ltd., Zhuzhou City, Hunan Province, China. The collection equipment included a wheel delivery track, CCD area camera, magnetic steel sensor, and computer for image storage (Figure 5). When the wheelset was sent into the acquisition area by the wheel delivery track, the camera waited for the hardware control signal, the locomotive wheelset triggered the magnetic steel sensor, and the camera began collecting images. The camera collected one or more wheelset images each time. The computer was used to save the collected images and generate the dataset.

**3.1.2. Dataset Production.** We collected wheelset tread defect images according to actual needs, including defects and normal samples, with 204 images of defect samples and 74 normal samples. An example of a defect is shown in Figure 6. Note that a single image may have multiple defects. Our dataset thus contains 278 samples, of which 204 are wheelset tread defect samples, 74 are normal samples, and the number of wheelset tread defects is 218, as shown in Figure 7:

**3.2. Defect Detection on WT-DET.** For the pretrained Yolov4, the MFF module is new. Thus, we trained Yolov4 and MFF to share the same convolutional features. The backbone is essentially a feature extraction network that generates a single multiscale feature  $F_i$ . The multiscale features generated by MFF can be fed into the Yolo head after the second fusion of PANet. Therefore, the pretrained backbone network was jointly trained with MFF and PANet for end-to-end training. In particular, the training model was divided into two stages. In the first stage, the shared convolutional layer (backbone) was frozen, and on this basis, the nonshared layer (others) was trained. In the second stage, the shared convolutional layer was unfrozen and the network was globally trained.

A defect detection experiment on the WT-DET dataset was conducted. The GPU used in the experiment was RTX 2080Ti, the Python version was 3.6, and it was carried out in Keras 2.1.5 and TensorFlow 1.13.2 environments. The

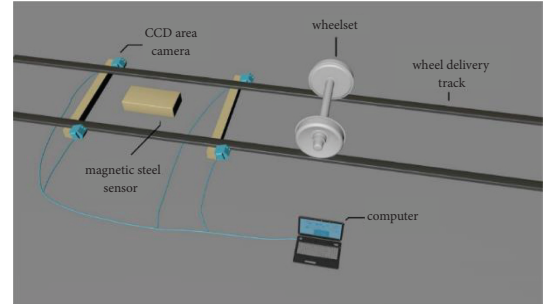


FIGURE 5: Schematic diagram of wheelset tread defect collection system.

improved Yolov4 model was used in this study. In the formal training, the training and the test sets were divided according to a 7 : 3 ratio. Among them, 193 were training sets and 84 were test sets. To ensure the reliability of model training, a ten-fold cross-validation method was used, and the 193 wheelset tread defects in the training set in one epoch were randomly divided into ten parts: nine parts as the training set, and one part as the verification set, to avoid model overfitting caused by an unreasonable data division in the case of too few samples. For MFF-Yolov4, the image input was adjusted to a uniform size of  $416 \times 416 \times 3$ . We used the Adam optimizer and adopted the freezing-based training procedure described above. The total number of epochs was 100. In the first 50 epochs, the backbone network training fusion network part (MFF and PANet) was frozen. At this stage, the batch size was set to eight, and the learning rate was set to 0.001. In the last 50 epochs, the entire network was trained. At this stage, the batch size was set to two, and the learning rate was set to 0.0001.

We used the above-divided dataset to fine-tune MFF-Yolov4. The MFF-Yolov4 model obtained after the improvement of Yolov4 was trained and tested, and an ablation experiment was carried out for each step of the improvement. Details of this are given in the following sections. The training results of the model are given, as shown in Figure 8.

As the epochs of the model increase, the loss value gradually decreases and finally reaches convergence, indicating that the model is effectively fitting the data. The loss is large at the beginning of model training. Thus, the first iteration is ignored when drawing the loss curve. Figure 8 shows that the loss value after 20 epochs before the start of training significantly drops, and when the training reaches a certain stage, the curve tends to stabilize. After 50 epochs, the loss value is maintained at about 1.3. The neural network learning effect is ideal, and the hyperparameter settings are reasonable in the training phase.

A comparative experiment was carried out on the self-built dataset with the current mainstream single-step and two-step target detection models (Table 1). In addition, unlike defect classification, in the case of defect detection, only the F1-score is not a suitable performance metric. Therefore, the accuracy, recall rate, and average precision (AP) were used to evaluate the results of the detection experiment. These indicators are defined as follows:

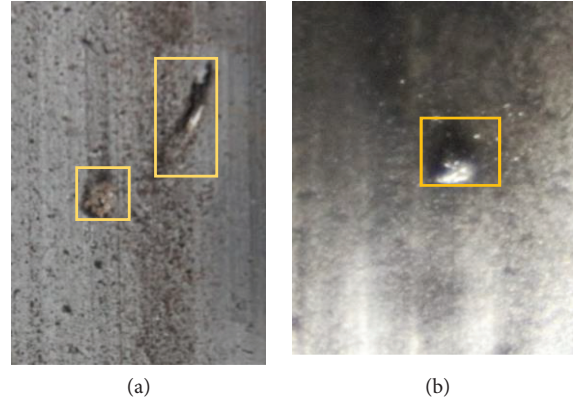


FIGURE 6: Sample wheelset tread defect dataset, the yellow frame is the ground truth box. (a) Multiple defects. (b) Single defect.

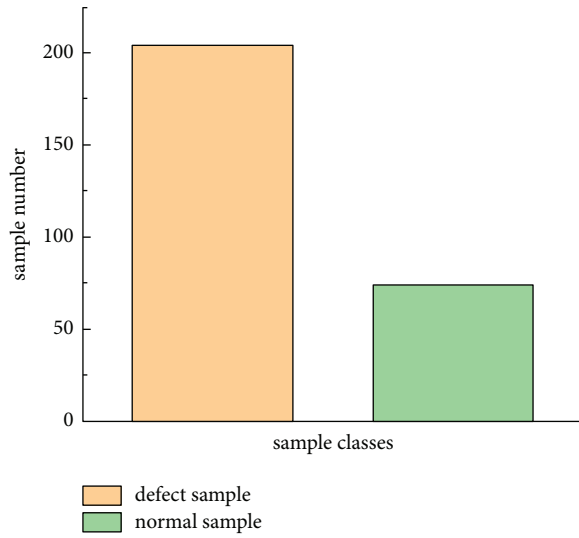


FIGURE 7: Sample situation of wheelset tread.

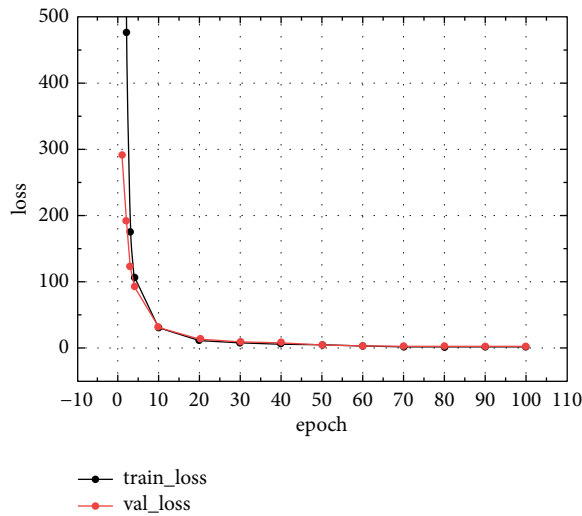


FIGURE 8: Loss curve during training.

TABLE 2: Detection results on WT-DET.

Method	Recall (%)	Precision (%)	mAP (%)	Fps	F1
SSD [19]	69.70	97.87	77.39	47.11	0.81
CenterNet [37]	65.15	95.56	84.82	63.95	0.77
Faster R-CNN [16]	72.73	48.48	73.36	2.05	0.58
RetinaNet [20]	75.76	90.91	82.67	45.28	0.83
RFB Net [38]	74.24	89.09	78.38	39.96	0.81
M2Det [39]	75.76	98.04	84.57	39.69	0.85
Our method ( $\alpha = 1.1$ )	78.79	94.55	86.25	37.05	0.86

$$\text{precision} = \frac{TP}{TP + FP},$$

$$\text{recall} = \frac{TP}{TP + FN},$$

$$\text{AP} = \frac{\text{precision} + \text{recall}}{2},$$

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

(5)

where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively. The mean average precision (mAP) was also calculated to evaluate the overall performance. Table 2 shows the experimental results of defect detection. Under the same conditions, all aspects of the Yolov4 model's data are different from those of other models with our self-built dataset. However, Yolov4 with the embedded MFF module has higher a recall, mAP, and F1-score than the other models. The results comprehensively show that the wheelset tread features extracted from the multiscale features have more comprehensive semantic features and location details. Yolov4 itself is a multiscale feature detector, but the multiscale features fused by our method have a more comprehensive feature representation, as discussed in detail below. The detection example of WT-DET is shown in Figure 9.

TABLE 3: Results of ablation experiments.

Method	Recall (%)	Precision (%)	mAP (%)	Fps	F1
Yolov4	66.67	88.00	76.06	39.09	0.76
MFF-Yolov4	77.27	94.44	86.21	36.38	0.85
Our method ( $\alpha = 1.1$ )	78.79	94.55	86.25	37.05	0.86

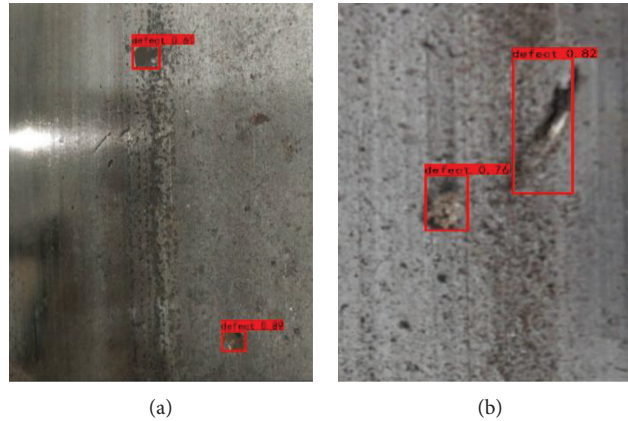


FIGURE 9: A visual example of WT-NET’s test results. The red box is the bounding box indicating its location, and the black word is its category score.

The above ablation experiment and detection method comparison experiment show that MFF can effectively improve the defect detection mAP of Yolov4. However, it remains to be shown that the mAP improvement of MFF-Yolov4 benefits from the location information contained in the multiscale fusion features extracted by MFF. The positioning accuracy performance of MFF is evaluated in the next section.

**3.3. MFF Semantic Analysis.** To evaluate the impact of the MFF module on classification, the results of defect classification are first reported to show that our method offers improved accuracy compared to the competition. Table 3 shows the results of the ablation experiments before and after the optimization of Yolov4. According to F1-score in Table 3, we can draw the following conclusions: compared with the original single-scale algorithm, MFF has better classification capabilities. Thus, multiscale fusion features still have strong semantic capabilities. When the improved loss function  $\alpha = 1.1$ , the performance index of the proposed method is further improved on the basis of MFF-Yolov4. In Table 4, we added a comparative experiment, which replaced the MFF module part in MFF-Yolov4 with two fusion modules, FPN [40] and ASFF [41], and conducted two sets of comparative experiments to demonstrate the performance of the proposed method.

**3.4. MFF Positioning Analysis.** To verify that MFF improves the positioning accuracy, Yolov4 and MFF-Yolov4 were compared. If multiscale fusion features have more location details, MFF-Yolov4 should have a higher recall

rate under the same IoU. Based on this, different IoU thresholds were used to evaluate the recall rate on the self-built dataset, WT-DET. IoU represents the ratio of the intersection and union of the prediction box and the underlying true value. Figure 10 shows the defect recall rate of Yolov4 with different IoU thresholds with and without the MFF module. The higher the IoU threshold, the higher the quality of the prediction box. As expected, Yolov4 with MFF is better than the original Yolov4. When  $0 < \text{IoU} < 0.87$ , the recall rate of Yolov4 is significantly lower than that of MFF-Yolov4. The original Yolov4 only regressed three high-level features of different scales, and the position information of wheelset tread defects was filtered by the layers in front of the network, which reduced the quality of the features. Our MFF selectively combines the previous layers with less interference information and rich location information, as discussed in Section 4. This gives MFF-Yolov4 stronger positioning capabilities.

## 4. Discussion

In this part, to prove the effectiveness of MFF, several hidden factors that affect the proposed wheelset tread defect detection module are discussed.

**4.1. Determining the Connection Layer of MFF.** MFF combines features from different levels into multiscale features, and this effectively improves detection. In Section 3.2, the kinds of layers that should be combined are briefly discussed. In MFF-Yolov4, two layers belong to the bottom-layer features, namely, the last layer of F1 and F2. We discuss how to integrate these two features into the three



TABLE 4: Results of different fusion modules in Yolov4.

Method	Recall (%)	Precision (%)	mAP (%)	Fps	F1
FPN-Yolov4	72.73	96.00	80.63	35.85	0.83
ASFF-Yolov4	43.94	72.50	59.87	36.37	0.55
Our method ( $\alpha = 1.1$ )	78.79	94.55	86.25	37.05	0.86

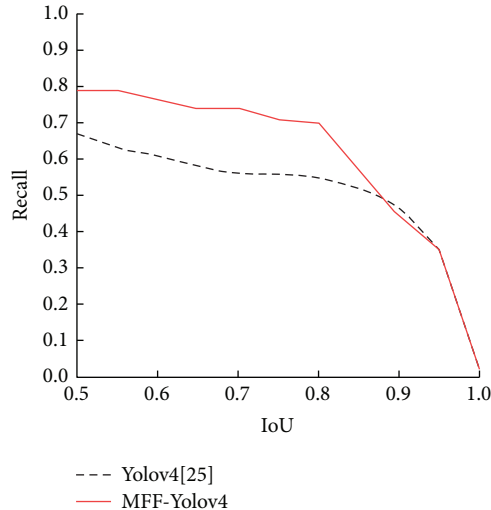


FIGURE 10: Recall rate under different IoU thresholds.

single-scale advanced features: the last layer of F3, F4, and F6 and whether other combinations of these five layers will lead to better performance. Therefore, on the WT-DET dataset, MFF-Yolov4 is trained in three different combinations, as shown in Figure 11, where  $\oplus$  denotes a splicing operation.

As shown in Table 5, integrating the second layer into the other three high-level feature layers is significantly better than other methods, thereby showing that multiscale fusion features are effective at improving detection accuracy.

At the same time, the features of the lower level are integrated into F1. MFF should have stronger detection performance, but the quantitative indicators show that the fusion of low-level features in F1 leads to a decline in the model mAP. In Figure 12, we visualize the feature map F1 (Figure 12(a)) and F2 (Figure 12(b)) flowing in the network in Table 5 and analyze the reasons that lead to the degradation of the model's performance. Although the F1 feature has more locational details than the F2 feature, the F1 feature has more noise and interference because our image comes from an actual industrial environment, which causes difficulty in the network learning defect details.

MFF unifies the features of different levels of resolution and channels through  $3 \times 3$  convolution and  $1 \times 1$  convolution, respectively. To maintain consistency in the number of channels, a simple method uses  $1 \times 1$  convolution to increase and decrease the number of channels. This  $1 \times 1$  convolution method can be conducted in two ways. A placement strategy is selected by comparing the two ways of connecting the multiscale features: before and after placing the  $1 \times 1$  convolution, and before and after multiscale feature

fusion. Front mode refers to placing  $1 \times 1$  convolution before multiscale feature connections, and rear mode refers to placing  $1 \times 1$  convolution after multiscale feature fusion. We adopted postlocation. That is, after each two-scale feature is spliced, the channel is adjusted through a  $1 \times 1$  convolution. Although the use of rear mode increases the number of parameters, it merely results in a slight drop in network detection speed (2 FPS), whereas the rear-mode mAP is higher, as shown in Table 6. Multiple  $3 \times 3$  convolutional cascaded downsampling forms are used to deal with the fusion between the underlying F2 feature and the three different scale features, namely, F3, F4, and F6. Compared with direct downsampling, step-by-step downsampling can retain more image details.

*4.2. Case Analysis of Missed Inspections.* Although the improved model is generally better than other methods on the self-built dataset, there are a few cases of missed detections. As shown in Figure 13, we analyze some failure cases and the reasons for the failure of detection. On the one hand, it is difficult to correctly identify damage in the initial stage of wheelset tread abrasion using MFF-Yolov4, mainly because of the lack of sample data and insufficient examples of wheelset tread defects. Thus, the network cannot fully learn the characteristics of wheelset tread damage. As shown in Figure 13(a), wheelset tread damage is easily confused with the complex background environment, and even experienced people cannot accurately distinguish them from the background. On the other hand, in a complex environment, the wheelset tread is

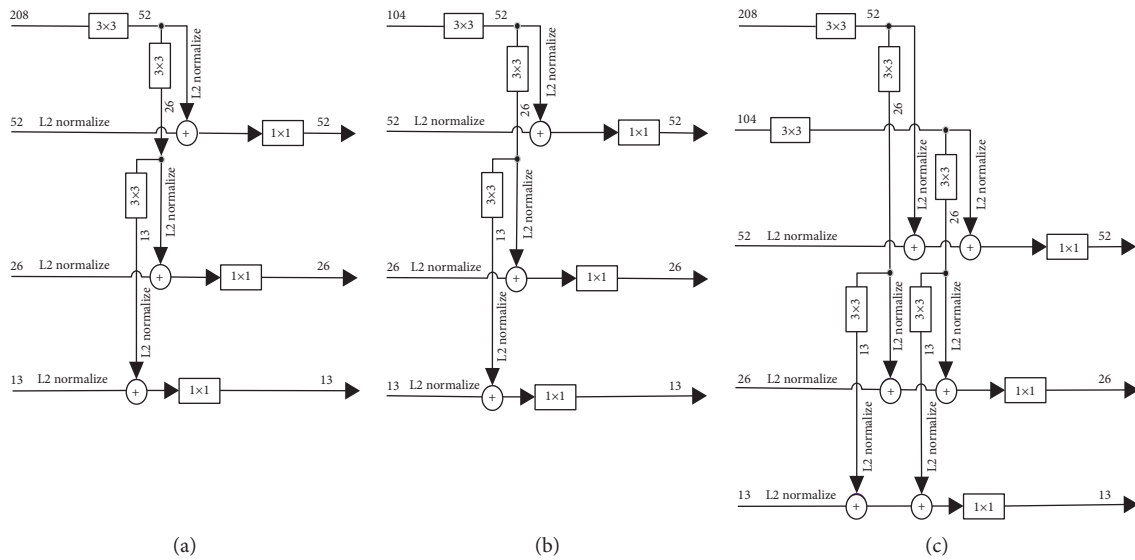


FIGURE 11: Multiscale feature fusion methods. (a) Fusion F1 layer. (b) Fusion F2 layer. (c) Fusion F1 and F2 layers.

TABLE 5: mAP of MFF-Yolov4 on WT-DET: use L2 normalization.

F1	The connected layer comes from:				F6	mAP (%)	
	F2	F3	F4	Without L2 norm		L2 norm	
✓		✓	✓	✓	77.19	81.86	
	✓	✓	✓	✓	83.85	86.21	
✓	✓	✓	✓	✓	76.70	75.45	

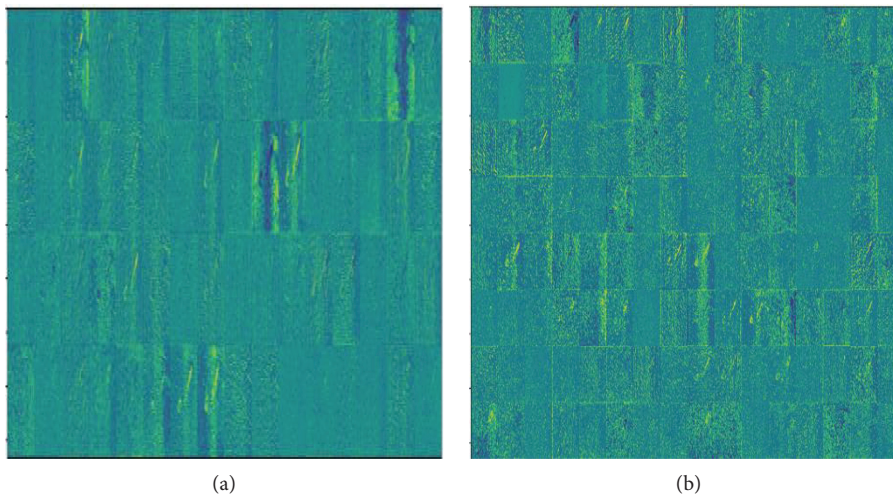


FIGURE 12: Feature layer visualization. (a) F1 feature map. (b) F2 feature map.

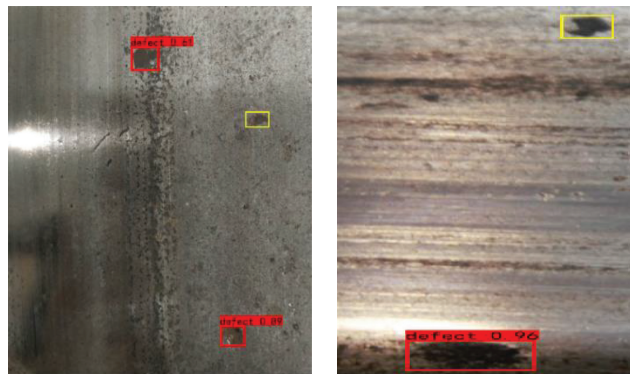
accompanied by oil, sand, and rust during the operation of the wheelset. These interferences are attached to the surface of the wheelset tread in blocks, and the dataset is incorrectly marked as a wheelset tread defect, as shown in Figure 13(b). Although the neural network does not predict the defect as a defect after learning the characteristics of the wheelset tread defect, in the quantitative

statistics, the missed detection rate of each model is classified as a classification error, which affects the detection accuracy. Here, we compare the false detection rate of our model with the other models.

The log-average miss rate of the proposed model is 19%, which is lower than that of the six other models, as shown in Figure 14.

TABLE 6: Parameters and mAP in two different placement modes, and parameters and accuracy in two different placement modes.

Front mode	Rear mode
1,861,248	Parameters (KB) 6,822,144
76.23	mAP (%) 86.21



(a) (b)

FIGURE 13: Missing case. (a) Missed detection. (b) Wrong label.

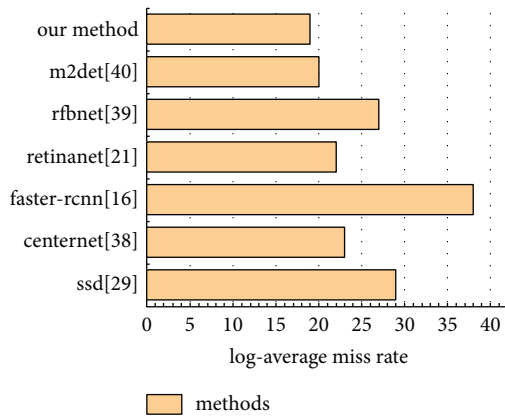


FIGURE 14: Comparison of missed detection rate of each model.

## 5. Conclusions

It is difficult for wheelset tread defect detection algorithms based on Yolov4 to consider local semantics and location details while ensuring real-time performance. Therefore, this study proposed an improved Yolov4 high-speed train wheelset tread detection algorithm based on multiscale feature fusion. To obtain more wheelset tread defect categories, and semantic and location details, we embedded a multiscale feature fusion module in the Yolov4 model that improves its detection accuracy. In addition, a valuable wheelset tread defect detection dataset, WT-DET, was constructed. The wheelset tread detection performance of the proposed algorithm was compared to that of current single-step and two-step detection algorithms on the self-built dataset. The results showed that the F1-scores of this algorithm for wheelset tread defect classification reached 86%, and the mAP of wheelset tread defect detection reached 86.25%. Moreover, the model reached speeds of 37.05 FPS. Given that regional-level detection can only obtain the approximate area of the wheelset tread, it cannot reflect the contour of the damage. In future research, we will study wheelset tread defect segmentation technology based on deep learning to obtain finer contour boundaries of wheelset tread defects.

## Data Availability

The data used to support the findings of this study are not applicable because the data interface cannot provide external access temporarily.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the Natural Science Foundation of China (52172403 and 62173137) and the Project of the Hunan Provincial Department of Education (19A137).

## References

- [1] S. Li, J. Yang, Z. Wang, S. D. Zhu, and G. C. Yang, "Review of development and application of defect detection technology," *Acta Automatica Sinica*, vol. 46, no. 11, pp. 2319–2336, 2020.
- [2] Y. Chen, L. Liu, J. Tao et al., "The improved image inpainting algorithm via encoder and similarity constraint," *The Visual Computer*, vol. 37, no. 7, pp. 1691–1705, 2021.
- [3] Y. Chen, H. Zhang, L. Liu et al., "Research on image inpainting algorithm of improved total variation minimization method," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2021.
- [4] Y. Chen, L. Liu, J. Tao et al., "The image annotation algorithm using convolutional features from intermediate layer of deep learning," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4237–4261, 2021.
- [5] S. Mao, M. Ye, X. Li, F. Pang, and J. Zhou, "Rapid vehicle logo region detection based on information theory," *Computers & Electrical Engineering*, vol. 39, no. 3, pp. 863–872, 2013.
- [6] C. Sun and Y. Zhang, "Research on automatic early warning method for rail flaw based on intelligent identification and periodic detection," *Journal of the China Railway Society*, vol. 40, no. 11, p. 140, 2018.
- [7] J. Gan, Q. Li, J. Wang, and H. Yu, "A hierarchical extractor-based visual rail surface inspection system[J]," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7935–7944, 2017.
- [8] X. Ni, H. Liu, Z. Ma, C. Wang, and J. Liu, "Detection for rail surface defects via partitioned edge feature," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–17, 2021.
- [9] M. Nieniewski, "Morphological detection and extraction of rail surface defects," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6870–6879, 2020.
- [10] T. Xian, W. Hou, and D. Xu, "A survey of surface defect detection methods based on deep learning," *Acta Automatica Sinica*, vol. 47, no. 5, pp. 1017–1034, 2021.
- [11] J. Chen, Z. Liu, H. Wang, A. Nunez, and Z. Han, "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 2, pp. 257–269, 2018.
- [12] Z. Liu, L. Wang, C. Li, Z. Han, and Z. W. Han, "A high-precision loose strands diagnosis approach for isoelectric line in high-speed railway," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1067–1077, 2018.
- [13] G. Krummenacher, C. S. Ong, S. Koller, S. Kobayashi, and J. M. Buhmann, "Wheel defect detection with machine learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1176–1187, 2018.
- [14] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, and R. Babuska, "Deep convolutional neural networks for detection of rail surface defects," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2584–2589, IEEE, Vancouver, Canada, July 2016.
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.

- [17] 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, 2020.
- [18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.
- [19] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: Single Shot Multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Amsterdam, Netherlands, October 2016.
- [20] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, Venice, Italy, 22–29 October 2017.
- [21] X. Kou, S. Liu, K. Cheng, and Y. Qian, "Development of a YOLO-V3-based model for detecting defects on steel strip surface," *Measurement*, vol. 182, no. 1–4, Article ID 109454, 2021.
- [22] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2020.
- [23] L. Cui, X. Jiang, M. Xu, W. Li, P. Lv, and B. Zhou, "SDDNet: a fast and accurate network for surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, no. 99, pp. 1–13, 2021.
- [24] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal Speed and Accuracy of Object detection," 2020, <https://arxiv.org/abs/2004.10934>.
- [25] J. Redmon and A. Farhadi, "Yolov3: An Incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>.
- [26] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, Honolulu, HI, USA, July 2017.
- [27] J. Yang, G. Fu, W. Zhu, Y. Cao, Y. Cao, and M. Y. Yang, "A deep learning-based surface defect inspection system using multiscale and channel-compressed features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 8032–8042, 2020.
- [28] B. Hu, B. Gao, W. L. Woo et al., "A lightweight spatial and temporal multi-feature fusion network for defect detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 472–486, 2021.
- [29] Y. Gao, J. Lin, J. Xie, and Z. Ning, "A real-time defect detection method for digital signal processing of industrial inspection applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3450–3459, 2020.
- [30] H. Yang, S. Mei, K. Song, B. Tao, and Z. Yin, "Transfer-learning-based online Mura defect classification," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 1, pp. 116–123, 2018.
- [31] C. Zhang, W. Shi, X. Li, H. Zhang, and H. Liu, "Improved bare PCB defect detection approach based on deep feature learning," *Journal of Engineering*, vol. 2018, no. 16, pp. 1415–1420, 2018.
- [32] O. Badmos, A. Kopp, T. Bernthaler, and G. Schneider, "Image-based defect detection in lithium-ion battery electrode using convolutional neural networks," *Journal of Intelligent Manufacturing*, vol. 31, no. 4, pp. 885–897, 2020.
- [33] J. Sun, P. Wang, Y.-K. Luo, and W. Li, "Surface defects detection based on adaptive multiscale image collection and convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 12, pp. 4787–4797, 2019.
- [34] S. Kim, W. Kim, Y. K. Noh, and F. C. Park, "Transfer Learning for Automated Optical inspection," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2517–2524, IEEE, Anchorage, AK, USA, May 2017.
- [35] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Salt Lake City, UT, USA, June 23 2018.
- [36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [37] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: keypoint triplets for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569–6578, Seoul Korea, October 2019.
- [38] L. Deng, M. Yang, T. Li, Y. He, and C. Wang, "RFBNet: Deep Multimodal Networks with Residual Fusion Blocks for RGB-D Semantic segmentation," 2019, <https://arxiv.org/abs/1907.00135>.
- [39] Q. Zhao, T. Sheng, Y. Wang et al., "M2Det: a single-shot object detector based on multi-level feature pyramid network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 1, pp. 9259–9266, Honolulu, HI, USA, February 2019.
- [40] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.
- [41] S. Liu, D. Huang, and Y. Wang, "Learning Spatial Fusion for Single-Shot Object detection," 2019, <https://arxiv.org/abs/1911.09516>.

## Review Article

# An Overview of Pavement Degradation Prediction Models

Amir Shtayat <sup>1</sup>, Sara Moridpour <sup>1</sup>, Berthold Best <sup>2</sup>, and Shahriar Rumi<sup>1</sup>

<sup>1</sup>Civil and Infrastructure Engineering Discipline, RMIT University, Melbourne, Australia

<sup>2</sup>Faculty of Civil Engineering, Nuremberg Institute of Technology, Nuremberg, Germany

Correspondence should be addressed to Amir Shtayat; [amir.shtayat@rmit.edu.au](mailto:amir.shtayat@rmit.edu.au)

Received 17 September 2021; Revised 25 December 2021; Accepted 27 December 2021; Published 31 January 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Amir Shtayat et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pavement management systems (PMSs) have a primary role in determining pavement condition monitoring and maintenance strategies. Moreover, many researchers have focused on pavement condition evaluation tools, starting with data collection, followed by processing, analyzing, and ultimately reaching practical conclusions regarding pavement condition. The analysis step is considered an essential part of the pavement condition evaluation process, as it focuses on the tools used to find the most accurate results. On the other hand, prediction models are important tools used in pavement condition evaluation to determine the current and future performance of the road pavement. Therefore, pavement condition prediction has an effective and significant role in identifying the appropriate maintenance techniques and treatment processes. Moreover, pavement performance indices are commonly used as key indicators to describe the condition of pavement surfaces and the level of pavement degradation. This paper systematically summarizes the existing performance prediction models conducted to predict the condition of asphalt pavement degradation using pavement condition indexes (PCI) and the international roughness index (IRI). These performance indices are commonly used in pavement monitoring to accurately evaluate the health status of pavement. The paper also identifies and summarizes the most influencing parameters in road pavement condition prediction models and presents the strength and weaknesses of each prediction model. The findings show that most previous studies preferred machine learning approaches and artificial neural networks forecasting and estimating the road pavement conditions because of their ability to deal with massive data, their higher accuracy, and them being worthwhile in solving time-series problems.

## 1. Introduction

Road infrastructure facilities have essential and active roles in the advancement of cities and communities. Road infrastructure is considered the most significant factor for the welfare and comfort of people and roadway users. Also, it is one of the sectors that determine the socioeconomic development of countries [1]. Pavement management systems (PMSs) play an efficient role in monitoring, planning, evaluating, managing, and implementing capable recommendations to keep road pavement conditions in an acceptable health condition [1, 2].

However, in terms of monitoring, high-precision equipment must be used to monitor changes and any existing distress or damage on road surfaces. Pavement monitoring plays an essential role in assessing pavement conditions. Therefore, the monitoring results and in-filed

collected data are used in formulating prediction models. After monitoring the pavement condition, pavement assessment strategies should be applied, and field surveys should be conducted for data collection to evaluate pavement infrastructure. Then, a decision will be made based on the relevant information of pavement conditions, and pavement maintenance procedures will be carried out based on the condition of the paving surfaces and expectations of pavement performance [1–3].

Moreover, PMSs concern the condition of road pavements after implementing maintenances and rehabilitation. Therefore, modeling the pavement performance is essential to transport agencies and governments at all management levels [4]. Lytton [4] mentioned that the future monitoring of pavement condition is called “prediction” or “forecasting,” which measures the future performance of pavement condition over time. After the prediction stage,

recommendations will be taken regarding the appropriate maintenance and treatment to be implemented [5]. However, the challenge is to build the best prediction model by combining all road pavement and environment parameters and variables. Thus, building any performance model requires a predefined dataset that is divided into three groups, including (1) training data, (2) testing data, and (3) validation data [6]. Moreover, the prediction of pavement performance has been studied extensively by many researchers over the last decade, combined with great efforts from transport agencies to find and disclose the most accurate evaluation and forecast of pavement performance [7–9].

The prediction performance of pavement surfaces has been developed using field evaluation and experimental tests. American Association of State Highway and Transportation Officials focused on predicting pavement distresses and the future failure of the pavement. In addition to the experiments, prediction models are also required to assess pavement degradation patterns and possible future maintenance plans [4]. Many studies have used different types of prediction models, such as mechanistic models, empirical models, mechanistic-empirical models, machine learning models, and neural network models, to predict the future condition of road pavement [10]. Machine learning models are the most popular prediction models used to estimate the current and future conditions of road pavement degradation. Developing any accurate road pavement performance prediction model depends on two main factors, including accessing accurate databases and correctly identifying the influencing variables on road pavement degradation.

Developing accurate prediction models mainly depends on the precision and consistency of the monitoring and evaluation data. Many monitoring techniques have been used to evaluate the pavement condition and collect the information and details on the pavement health statuses. These techniques include vibration-based methods, vision-based methods, walk and look, and scanning techniques. Besides, international standard performance indices are used to inspect and evaluate the pavement condition under different scenarios, such as pavement condition index, international roughness index, present serviceability rating, and structural index. Each type of performance indices has a different way to conduct the data of pavement health status. Pavement condition index (PCI) and international roughness index (IRI) have significant contributions in pavement monitoring and condition estimation. Therefore, many researchers use the outcomes of these indices in building and developing their pavement performance prediction models. PCI is a subjective monitoring index that depends mainly on the visual inspection and the inspector's experience. The PCI rating system consists of a scale from 0 to 100, where the worst pavement surface is at 0, while the excellent pavement condition is at 100. IRI is an indication of the level of surface smoothness. It can be measured using a profilometer. Also, there is a part of the IRI that depends on vibration-based methods and is called IRI (Proxy).

This paper is structured as follows: the subsequent section provides a general layout of the paper. It is followed by a general

overview of data sources, while Section 4 reviews the existing pavement performance prediction models, depending on PCI and IRI. Section 5 presents the discussion and limitations of the existing pavement prediction models, followed by the future direction of the pavement performance prediction.

## 2. Data Source

This review paper presents different prediction models based on the database used. Some of the past research papers have focused on using the results of pavement performance indices as a database to build their prediction models, while others focused on using filed measurements or other intelligent techniques, such as image processing and vibration data, to collect appropriate databases. Several studies have divided the database of pavement degradation models into two categories, including an observation database and an online database. In the observation category, the data collection is conducted using visual inspections by equipped modes of transport, e.g., automobile, bicycle [11–16], and intelligent monitoring techniques [17, 18]. In addition, many pavement prediction models used the long-term pavement performance (LTPP) or short-term pavement performance (STPP) dataset to predict the future pavement performance [19–21].

Pavement performance indices are commonly used as key indicators to describe the condition of pavement surfaces and the level of pavement degradation. Thus, government and transport agencies use these performance indicators to define the required maintenance and rehabilitation measures. Moreover, since the last decade, many efforts have been made to develop pavement performance assessment procedures to be more accurate, cost-effective, and straightforward [2]. Many studies have been conducted to investigate the status and the level of pavement degradation using pavement performance indexes, including PCI, IRI, pavement serviceability index (PSI), and pavement condition rating (PCR). Moreover, pavement condition indices can be considered time-dependent variables [10]. To achieve the optimal goals of the high-precision rating system, IRI and PCI indices are used as main variables in developing pavement performance prediction models. Figure 1 shows the field data source of the pavement condition.

## 3. Applied Methodology

Researchers agreed that the optimum method to accurately monitor the pavement condition performance is by forecasting and using prediction models [22]. These models have the ability to describe the minimum and maximum changes in road pavement performance. Different types of performance prediction models are used to provide ultimate accuracy and precision. The subsequent sections describe the prediction models that are used to estimate the performance of pavement conditions. In the subsequent sections, there are time-series models that are used to predict pavement condition performance, which can be divided into two main categories, including probabilistic reasoning and shallow machine learning models. Figure 2 shows the selected time-series modeling.

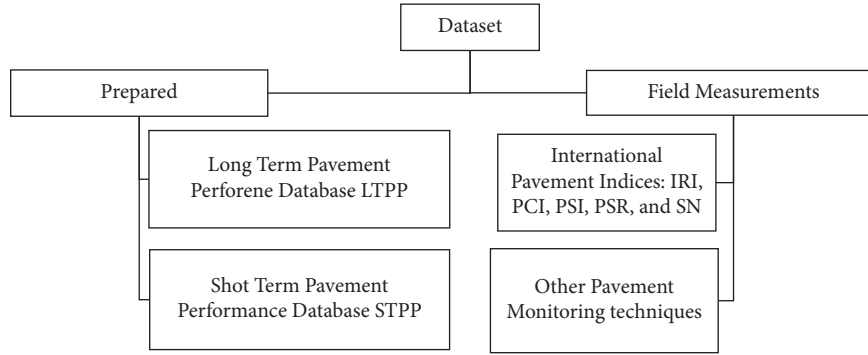


FIGURE 1: Data sources of monitoring road pavement condition.

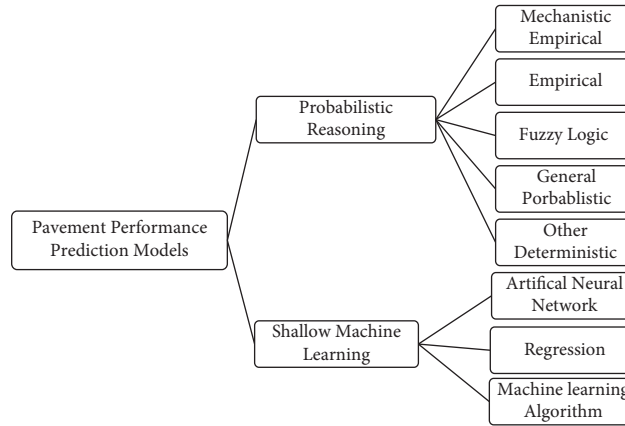


FIGURE 2: Time-series models in predicting road pavement performance.

**3.1. Probabilistic Reasoning.** Probabilistic reasoning is a way of logic exploration and representation according to a series of uncertain events and situations that depends on probabilities. In prediction performance, probabilistic reasoning algorithms have been widely used to predict pavement conditions' performance for short- and long-term statuses.

**3.1.1. Mechanistic Empirical Models.** These models can be used for the prediction of existing and future pavement degradation and maintenance activities. Also, mechanistic-empirical models are able to provide more reliable predictions with the future pavement condition. In addition, these models focus on the properties and qualities of pavement material. PCI and IRI are used to provide valuable information on pavement health status in this model type [23].

In 1989, George et al. [23] used a mechanistic-empirical model to predict future pavement performance. An empirical mechanistic model was developed based on PCI values. They used the PCI values over two years and approximately 2000 miles of road in Mississippi, USA, for three categories of asphalt surface state (flexible pavement with no overlay, with overlay, and composite pavement). Furthermore, the model focused on assessing the different types of degradation and distress of pavement surfaces and how they affect maintenance plans. Their study used six main parameters to develop a road pavement performance model,

including traffic volume, pavement age, pavement structural number, material quality, and surface deflection measurements. A performance indicator was developed in their research to describe the interaction between the pavement roughness (PR) data and distress rating (DR) as follows:

$$PCR = PR^{0.6} DR^{0.4}. \quad (1)$$

The following condition performance prediction models include time-series pavement condition data with no overlay, with overlay, and the composite pavement is shown from equations (2) to (4), respectively [23].

$$PCR(t) = 90 - a \left[ \exp(\text{Age}^b) - 1 \right] \log \left[ \frac{ESAL}{SN^c} \right], \quad (2)$$

$$PCR(t) = 90 - a \left[ \exp(\text{Age}^b) - 1 \right] \log \left[ \frac{ESAL}{SN^c * T} \right], \quad (3)$$

$$PCR(t) = 90 - a \left[ \exp \left( \frac{\text{Age}^b}{T} \right) - 1 \right] \log[ESAL], \quad (4)$$

where  $a$ ,  $b$ , and  $c$  are constants and are equivalent single axle loads ESAL, structural number SN, and the thickness of the last overlay  $T$ , respectively. In their study, George et al. [23] used the most significant and most effective variables like the pavement age.



Sidess et al. [24] proposed a model based on the combination of the empirical-mechanistic and the regressive empirical approach to predict IRI. Data were collected from the pavements, and a total of 165 road segments of 287.5 km of data were used in this model. The IRI degradation model was calculated as follows:

$$IRI(t > t_{ini}) = 1.10 + K * (W_0 + W_t)^\gamma, \quad (5)$$

where  $K$  and  $\gamma$  are regression coefficients, which are the functions of subgrade modulus, structural number at the time of pavement construction, and asphalt thickness. ( $W_0$ ) is the cumulative number of (130 kN) equivalent single axle loads ESAL applications leading to the increase of the IRI from (1.10 m/km) to ( $IRI_{ini}$ ), where  $IRI_{ini}$  is IRI of the section at the time ( $t_{ini}$ ), and  $W_t$  is a cumulative number of (130 kN) equivalent single axle loads ESAL applications applied until time  $t$ . ( $R^2 > 0.9$ ) for the predicted and measured data. Sidess et al. [24] had developed a similar model for predicting PCI degradation. The characteristics of mechanistic models are summarized in Table 1.

**3.1.2. Empirical Models.** Empirical models mainly depend on the results of experiments or field observations. Empirical models are known as models that relate the causes and effects. These models are more accurate at network-level analysis. In terms of evaluating future pavement performance, many studies have been conducted to predict the performance of pavement conditions. The online data and field observation are suitable for developing empirical models. The specifications of empirical models are summarized in Table 1. Figure 3 shows the categories of empirical models.

**(1) Statistical Models.** Statistical models use data from experiments or field measurements to make statements about the future changes of the experiment outcome. These statistical methods provided real-time solutions to complex problems. Attoh-Okine [25] and Marcelino et al. [26] studied statistical prediction models to measure and evaluate the future performance of pavement conditions. Also, the accuracy of statistical models was compared with artificial neural networks (ANN) [25]. The results show that statistical models are capable of generalizing and providing accurate road pavement performance models. The  $R^2$  value obtained was approximately 40%, and the standard error of IRI was 1.88.

**(2) Recursive Partitioning.** It is a part of statistical methods and nonparametric modeling. It is also used to determine a group of field measurements with similar parameter values. This method uses a decision tree to correctly classify the number of variables, such as the pavement age, traffic condition, weather condition, and pavement structure details. Inkoom et al. [27, 28] performed a model to predict the cracking condition on pavement surfaces using recursive partitioning and ANN. Approximately 5,814 pavement segments were selected in Florida, the U.S., and their eleven features. These features included the age of pavement,

average daily traffic, truck factor, asphalt thickness, maximum posted speed, the functional class of pavement, and previous five-year pavement condition rating. 70% of the dataset was considered to be the training dataset, and the rest was used as a test dataset. Two models were investigated, one with all these eleven variables and another without the time-series of a pavement condition rating. The first model showed more accurate pavement performance prediction results than the second model. For the regression tree,  $R^2$  was found to be 89%, and for ANN,  $R^2$  was found to be 41.4%.

**(3) Informative Feature for Prediction.** Piryonesi and EL-Diraby [29] found a computational system for performing PCI using informative features for prediction. The results showed that using more categories of prediction classes and levels of distress, the accuracy of the pavement performance prediction model decreases. Using the 7-class scale was less accurate than using the original 5-class scale in predicting PCI. The most accurate prediction model was for three years with an accuracy of approximately ( $78 \pm 4\%$ ), with a 5-class scale, and approximately ( $76 \pm 4\%$ ), with a 7-class scale. The study concluded that pavement age and climate conditions were the most effective variables in building this prediction model [30].

**3.1.3. Fuzzy Logic.** Several studies focused on developing an innovative IRI prediction model based on fuzzy-based time-series and particle swarm optimization (PSO) techniques [31, 32]. In their study, Li et al. [33] revealed the importance of using PSO techniques to enhance the results of the performance models and future IRI prediction models. Furthermore, in their study, Li et al. [33] used an LTTP database to extract the IRI values for some urban roads in Canada. The methodology of this study focused on dividing the IRI values into granular spaces. For more illustration, they divided the IRI data into factors and subfactors. The factors section used the average IRI values from the long-term pavement performance database, while the subfactors data were measured in the left and right wheel path [34]. Moreover, a second-order fuzzy trend model was used to predict the performance of the IRI factors and subfactors data. Consequently, the fuzzy trend model was defined as follows:

$$A = \frac{fA(U1)}{(U1)} + \frac{fA(U2)}{(U2)} + \dots + \frac{fA(Un)}{(Un)}, \quad (6)$$

where  $U = \{U1, U2, \dots, Un\}$  is defined as a universe of discourse,  $A$  is a fuzzy set,  $fA$  is a membership function of the fuzzy set  $A$ , and  $fA(Un)$  is a membership degree of  $Un$ .

By comparing the innovative IRI prediction method with other modeling approaches, such as polynomial fitting, autoregression integrated moving average (ARIMA), and backpropagation neural network (BPNN). The results showed that the IRI prediction model achieved high accurate forecasting compared with other modeling approaches. The IRI prediction error of the proposed model was identified using root mean square error (RMSE) and relative error (RE)

TABLE 1: Pavement performance prediction studies in probabilistic reasoning.

Technique used	Pavement indicator	Data sources	Metrics	References	Strength	Weakness
Markovian model	IRI	Observed		Porras-Alvarado et al. [40]	Capable of evaluating multiple hypotheses and accurate with first-order Markov property	Not suitable in case of higher-order correlation and missing data.
		LTPP	RMSE	Alimoradi et al. [41]		
Empirical mechanistic	PCR	Observed		George et al. [23]	It contains features from statistical and mathematical elements	Lake sights for pavement preservation and limited literature.
Fuzzy logic	IRI	LTPP	RMSE and RE	Li [33]; Nguyen, et al. [32]	Simple control, more robustness, and more efficiency with control systems	Have steady-state errors, weak in real-time response, and a limited number of input variables
Deterministic models (Al Omari-Darter model and Dubai model)	IRI	Observed	$R^2$	Al-Suleiman and Shiyab [34]	Good in decision making by providing clear information of the future trends and challenges.	Easy to misinterpret and hard to check the validity.
		LTPP	$R^2$	Chen and Zhang . [37]		

$R^2$ : coefficient of determination, RMSE: root mean squared error, MAPE: mean absolute presenting error, CF: correction factor, VAF: variance account for, MAE: mean absolute error, RE: relative error, MSE: mean squared error, SDMSE: standard deviation of mean squared error.

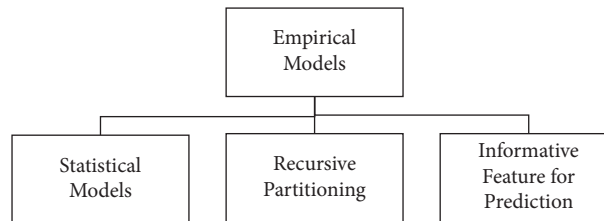


FIGURE 3: Categories of empirical models.

to evaluate the ability of each model to provide accurate performance prediction. The results revealed that the IRI prediction model was accurate enough with the smallest error values compared with other modeling approaches.

**3.1.4. Probabilistic Modeling.** Liu and Gharaibeh [35] focused on using probabilistic models to describe the change in pavement status and performance with time. They mainly used significant variables, such as average annual daily traffic, pavement layers thickness, layers air voids, layers liquid limits, layers asphalt content, and annual rainfall, to build an accurate prediction model. Abed et al. [9] developed a probabilistic prediction model of flexible pavement, where the thickness and stiffness of the pavement layers were used as variables. Besides, the mean values, standard deviations, and probability distribution functions of these two parameters were considered to be variables. For this study, a road section in Nottingham, U.K., was selected as the case study. This road had a four-layer pavement, including a surface course, base course, sub-base course, and a compacted subgrade. The layer thickness and stiffness variations and their probability distributions were collected from previous research. The random thickness values of each layer were calculated by the Monte Carlo method. Pavement

temperature and traffic volume were calculated for future predictions. In their study, KENLAYER software linked with MATLAB software was used to calculate the bottom-up fatigue cracking, top-down fatigue cracking, and pavement deformation as pavement responses at predefined critical locations of the pavement. The model was simulated for thirty years. It was found that the pavement layer thickness and stiffness had played a significant role in pavement performance. The mean values of the predicted performance indicators were increasing over time, however, the standard deviations of these were also increasing.

(1) *Markovian Models.* Different studies developed a probabilistic method using the Markov chain framework to characterize pavement conditions and predict pavement performance [36]. The prediction model was formed based on IRI data from the National Department of Transportation in Costa Rica. The IRI data were conducted for 2004, 2006, 2008, and 2010, and then, the prediction model was developed to predict the pavement performance based on the IRI data in 2020. The modeling process was divided into three stages, namely data collection and analysis, model development, and model validation. Transition probability matrix (TPM) was used based on the Markov chain process (MCP) to correlate pavement degradation with explanatory variables.

Moreover, the importance of using TPMs was to predict pavement performance in the subsequent specific years. At the same time, significant variables were used, including the thickness of pavement layers, structural number, and the number of wheel passes per unit strength of the pavement. The Markov prediction performance model results revealed that using the probabilistic model in predicting pavement performance during a specific time is reliable. Moreover, the TPM results showed more accurate pavement performance prediction, as the percentage of the errors will be minimized after applying the optimization techniques. One of the main advantages of using this probabilistic model is the ease of modeling pavement degradation and the ability of these models to help decision-makers for better planning and management (see Table 1).

**3.1.5. Other Deterministic Models.** Chen and Zhang [37] published a research paper on the evaluation of IRI based on the pavement degradation prediction model, which depends on four different deterministic models, including the Al Omari–Darter model, Dubai model, and the Transportation Research Board’s National Cooperative Highway Research Program (NCHRP) model. This comparison between the models was performed to identify the most accurate deterministic model in predicting pavement performance based on two main effective variables, including pavement age and thickness of pavement layers. Furthermore, Chen and Zhang [37] obtained the IRI data and other models related to data from the LTPP database in New Mexico. The IRI-based pavement degradation prediction model is divided into two main classifications, prediction of IRI (Al Omari–Darter and Dubai models), and prediction of other performance predictions based on IRI (NCHRP model). In the selected deterministic models, Al-Suleiman and Shiyab [34] developed a new prediction model (Dubai model) based on pavement age. The IRI data that was conducted in the left and right wheel path during vehicle movement, and the following equation (7) presents the Dubai model. The goodness of fit,  $R^2$ , was 0.801, which is relatively high and provides a good indication of the pavement condition.

$$IRI = 0.796 \exp(0.0539 \text{ age}). \quad (7)$$

Furthermore, Al Omari–Darter [38] found a prediction model based on IRI values and the Rut Depth (RD). Later on, they tried to elevate the model using the standard deviation (SD) of RD for higher accuracy.

The significance of work was measured depending on the  $R^2$  value, which was 0.93 for the IRI-RD model and 0.94 for the IRI-SD model. The models are shown in equations (8) and (9), respectively.

$$IRI = 57.56R D - 334.28, \quad (8)$$

$$IRI = 136.19S D - 116.36. \quad (9)$$

Moreover, the NCHRP model was developed using an exponential regression model to predict the pavement

serviceability index (PSI). The goodness of fit,  $R^2$ , of this model was relatively low, as it was 0.73. However, in 2008, the New Mexico Department of Transport [39] reported this model as follows:

$$PSI = 5 \exp(-0.26IRI). \quad (10)$$

Chen and Zhang [37] found that the Dubai and NCHRP models were accurate for pavement performance prediction regarding pavement age and thickness. The Al Omari–Darter model provided less capability to predict the performance of pavement conditions in terms of pavement thickness (Table 1).

Table 1 below presents the previous studies that used probabilistic reassuring to predict pavement performance. The table also shows the technique used for each type of pavement indices to perform the prediction. The data sources are provided with the standard matrices used to measure each developed model’s validation and accuracy. Besides, the strength and weaknesses of each model are presented and discussed.

**3.2. Shallow Machine Learning.** Shallow learning is a branch of machine learning algorithms that depends on expert-based descriptions. The datasets in shallow machine learning need to be preprepared and predefined with all required features. Regarding prediction performance, shallow learning algorithms have been widely used to predict and estimate the condition and performance of pavement health status.

**3.2.1. Artificial Neural Network (ANN).** The artificial neural network (ANN) is a complex model developed to simulate the thinking ways of the human brain and its ability to solve problems by offering various alternative solutions. The use of ANN in pavement performance prediction became widely known because of the accurate prediction results. The existing ANN models that are used in the literature for pavement performance prediction are presented in Figure 4.

Alsugair and Al-Qudrah [42] and Serin [31] measured the future performance of pavement conditions using ANN. This technique involves artificial intelligence, and many researchers favor its use in predicting pavement conditions. Besides, some researchers utilized a regression model and ANN to predict the probability of degradation on asphalt pavement and roughness distress level [43, 44]. The characteristics of ANN models are summarized in Table 2.

Moreover, some pieces of research focused on predicting PCI based on different optimizing techniques [45]. For instance, Shahnazari et al. [46] used ANN and genetic programming (GP). In their study, PCI data were collected based on field observation using an automated car for different urban roads in Iran. The data collection phase focused on measuring PCI values for most common pavement distresses, including cracking (alligator, longitudinal, edge, and transverse), potholes, patching, and bleeding. The type of pavement distress was used as an effective variable for the pavement performance prediction model. In their study, they used 80% of the dataset as a train set and 20% of the dataset as a test set.

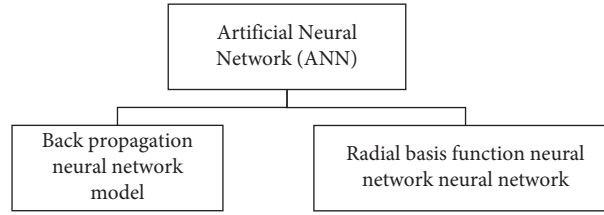


FIGURE 4: Artificial neural networks in pavement performance prediction.

TABLE 2: Pavement performance prediction studies in shallow machine learning.

Technique used	Pavement indicator	Data sources	Metrics	References	Strength	Weakness
Artificial neural network ANN	IRI	LTPP	$R^2$ , RMSE, MAE, MSE, CF, and VAF	Abdelaziz, et al. [64]	Able to work with vast amounts of data and most challenging problems, change the structure to the used parameters, suitable for time-series problems.	Expensive to train, requires long training time and massive data
		Observed	$R^2$ , RMSE, MAE	Lin et al. [50], Mallika, et al. [65]		
	PCI	Observed	MAE, RMSE, and $R^2$	Shahriazari et al. [46], Jalal, et al. [47]		
Neuro-fuzzy Model NFM	IRI	LTPP observed	$R^2$ and RMSE correlation factor R	Soncim et al. [66], Ngyeny, et al. [32]	Suitable for complex data interactions, easy to scale and have high converge	Requires huge data, complex and difficult to debug.
Regression	IRI	LTPP	$R^2$ , MSE, RMSE	Elhadidy et al. [22], Piryonesi and El-Diraby [29]	Simple, requires a minimum number of parameters, suitable in classification and recognition works	Expensive, not able to work with a multifeatures dataset and poor in presenting the extreme events.
	PCI	Observed	$R^2$	Ahmed, et al. [63]		
Support Vector machine	IRI	Observed	Ransom output error	Roberts and Attoh-Okine [57]	Training is simple and relatively easy, suitable in high-dimensional data	Requires high memory and more time for training the model.
	IRI	LTPP	MSE MAE and RMSE	Kargah-Ostadi and Stoffels [67]		

$R^2$ : coefficient of determination, RMSE: root mean squared error, MAPE: mean absolute presenting error, CF: correction factor, VAF: variance account for, MAE: mean absolute error, RE: relative error, MSE: mean squared error, and SDMSE: standard deviation of mean squared error.

In addition, Shahnazari et al. [46] assessed the accuracy of the previously mentioned models by determining  $R^2$ , RMSE, and mean absolute error (MAE). The results showed that the value of  $R^2$  for the ANN and GP models was 0.99. Therefore, the results indicate that these models are reliable for predicting pavement performance using PCI values.

Jalal et al. [47] also developed an ANN model to predict PCI based on observed and experimental measurements at different locations in the Texas University campus. They also applied an optimal ANN model to enhance the accuracy of the conventional ANN model. Three types of pavement, including asphalt concrete (AC), hot mixed asphalt (HMA), and Portland cement concrete (PCC), were evaluated during the period 2014 to 2016. Furthermore, two other main variables were used to build the model, namely the annual average daily traffic (AADT) and traffic loads. The study showed that the proposed ANN model was accurate for the selected types of pavement. After applying the optimal ANN, the results revealed that there were improvements and enhancements in model outcomes and limitations in errors.

The international roughness index prediction model is a time-series prediction performance model. Therefore, many effective variables, such as pavement thickness, cracking level, traffic volume, resilient deflection modulus, structure number, climate condition, must be carefully collected. In 2000, a report from the Highway Development and Management Series [48] stated that the previous variables are essential variables used to construct a degradation model, as shown in equation (11) for one year.

$$\Delta IRI = K_{gp} [\Delta IRI_s + \Delta IRI_c + \Delta IRI_r + \Delta IRI_t] + \Delta IRI_e, \quad (11)$$

where  $\Delta IRI$  represents the total rating changes in the IRI values,  $IRI_s$  rating changes because of structure deformation,  $IRI_c$  rating changes because of cracking,  $IRI_r$  rating changes because of rutting,  $IRI_t$  rating changes because of potholing, and  $IRI_e$  rating changes because of the environment during a year.

ANNs and a group method for data processing models were developed by Ziari et al. [49] to predict asphalt pavement in a short-term performance for a year and two

years. Also, the full pavement life cycle prediction was carried out as the long-term prediction performance. Furthermore, Ziari et al. [49] used the IRI values from the database of the PMS datasets in the U.S., and they selected nine effective variables to indicate the performance of pavement conditions. The nine variables were selected carefully to provide clear indications of the condition of pavement surfaces and the affected factors.

The  $R^2$  and RMSE were used to assess the quality and ability of the models to provide accurate and validated results. Furthermore, three more error indicators were examined, including mean absolute presenting error (MAPE), correction factor (CF), and variance account for (VAF) to identify errors in the proposed models and to provide optimum correlations for ANNs and group method for data processing models. The benefit of using the GMDH is that it focuses on predicting a complex system without the need for assumptions.

ANN models have specific features compared to other models. For instance, they have a high ability to work with and predict complex systems. Moreover, these models are more efficient and provide high-accuracy pavement condition predictions. Ziari et al. [49] mentioned that the ANN models always provide minimum error values compared with other models. Moreover, clear illustrations of the effect of each variable and parameter on the performance of pavement conditions are always provided in the modeling results, which represent one of the many advantages of using ANN models. Consequently, the results showed that the ANN model is important and accurate in predicting short- and long-term performance, while the group method for data processing model is unable to be used with the IRI values and the nine significant variables to predict the paving condition performance in neither the short-term nor the long-term pavement life cycle.

(1) *Back Propagation Neural Network Model*. Lin et al. [50] elevated the accuracy of using the backpropagation neural network model in pavement performance prediction. The model showed that there was a variation in correlation values, and the best value was approximately 0.94. Moreover, the results indicated that potholes, rutting, and patching presented the highest correlation coefficient, implying a clear correlation with IRI values. However, concerning other types of pavement distress, such as cracking, alligator cracking, and bleeding, they showed a low correlation with IRI values, which means less ability to correlate the types of pavement distress and IRI values. As Lin et al. [50] stated, this type of model is easy to implement and can simplify pavement inspection for transport agencies. It also provides clear information on the relationship between the type of distress and IRI values during long-term performance prediction. However, using this model was deficient in relating some type of pavement distresses with the conducted IRI values. The characteristics of the backpropagation neural network model are summarized in Table 2.

(2) *Radial Basis Function Neural Network (RBF)*. Karbalaeezadeh et al. [51] proposed a model to predict PCI from the falling weight deflectometer (FWD) deflection data.

FWD deflection data were collected from selected 236 pavement segments of the Tehran-Qom freeway in Iran. PCI was calculated in each segment by inspection. Data analysis were done using five different methods: multilayer perception neural network optimized by Levenberg–Marquardt (MLP-LM), multilayer perception neural network optimized by the scaled conjugate gradient (MLP-SCG), radial basis function neural network optimized by genetic algorithm (RBF-GA), radial basis function neural network optimized by the imperialist competitive algorithm (RBF-ICA), and merging these four with committee machine intelligent systems (CMIS). Results from these five methods were compared with four statistical parameters: average percent relative error (APRE), average absolute percent relative error (AAPRE), RMSE, and standard error (SE). However, it showed promising results for the five selected models but depended only on the accuracy of FWD data (Table 2).

3.2.2. *Machine Learning Algorithms*. Machine Learning (ML) methods are an area of artificial intelligence. ML techniques are widely used in pavement performance prediction because of high-precision results. ML techniques can be divided into two main categories, including support vector machine and hybrid machine learning (Figure 5).

Piryonesi and EL-Diraby [29] developed a cost-effective prediction model using a machine learning algorithm and LTPP database. This prediction model focused on estimating the pavement condition and surface distress using PCI over 2, 3, 5, and 6 years. In the study, different attributes were used to simplify the proposed model used by transport agencies and governments with minimum operating costs. Moreover, the researchers tried to change the PCI rating scale to be a 7-class scale instead of a 5-class scale. This attribute was applied as a trial to enhance the evaluation procedure of PCI. Furthermore, many attempts were made to measure the PCI values using the 7-class scale and a prediction model that was performed to evaluate the PCI measurements in both class scales over the selected years. The study also used influential variables, such as the age of pavement, type of pavement, AADT, average daily maximum and minimum temperature, climate condition, and functional class of the pavement (Table 2).

(1) *Support Vector Machine (SVM)*. Wang et al. [52] used a combination of grey relation analysis (GRA) and support the vector machine regression (SVR) for the prediction of asphalt pavement performance. GRA was conducted to select major factors affecting pavement performance, and SVR was done using those factors to predict pavement performance. Data were collected from Guangyun Expressway. Road temperature, humidity, and wind speed data were collected from the installed weather station. Temperature and humidity sensors were installed inside pavement layers and on the pavement surface. During the GRA analysis, twelve factors were found to be more influential than others. These were equivalent single axle loads, maintenance funds, pavement structure strength ratio, a mean value of soil moisture, the highest temperature in the middle surface, the highest temperature

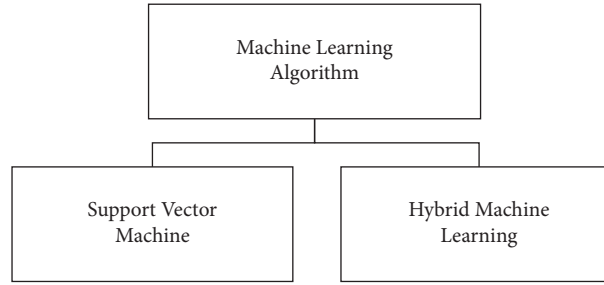


FIGURE 5: ML techniques in pavement performance prediction.

in the road surface, annual cumulative total radiation, annual average rainfall, the lowest temperature in the middle surface, the highest temperature in the upper surface, the lowest temperature of the upper surface, and the highest temperature in the lower surface. Finally, GRA-SVR, grey method (GM), genetic algorithm-back-propagation (GA-BP), and pavement performance index (PPI) models were applied to predict the rutting depth index (RDI). Compared with the other three, GRA-SVR was found highly accurate and time-independent though the modeling process was complex.

On the other hand, Ziari et al. [53] performed a support vector machine model to predict pavement performance conditions based on IRI measurements and LTPP, and a mathematical approach was used at the same time to prepare the existing data to validate the model and to investigate the interaction between the performance model and the model variables. Their research paper used the dataset consisting of five kernels types of the support vector machine algorithms and IRI data. The five kernels were tested, including the polynomial kernel with degrees 1 to 3, Pearson VII universal kernel, and the radial basis function. Moreover, the nine variables include the pavement layers thickness, equivalent single axle load, annual average daily traffic, average daily traffic, annual average daily truck traffic, environment changes, annual average temperature, pavement age, and annual average precipitation. They are formed to build the prediction model, see Table 2.

Three nonlinear kernel equations were applied to describe the prediction model equations (12)–(14). These equations represent the polynomial, radial basis function, and Pearson VII universal, respectively [54].

$$K(x, y) = (1 + (x, y))^d, \quad (12)$$

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{C}\right), \quad (13)$$

$$K(x, y) = \frac{1}{\left[1 + \left(\left(2\sqrt{x - y^2}\right)\sqrt{2^{(1/\omega)} - 1/\sigma}\right)^2\right]^\omega} \quad (14)$$

The RMSE and the correlation coefficient were examined to find an accurate performance model. They found that the Pearson VII universal kernel was the best and significant kernel of the support vector machine model. Additionally, it

matched the IRI measurements and the health status of pavement.

(2) *Hybrid Machine Learning*. Hoang [55] introduced a model to identify patches on asphalt pavement. Images were analyzed to get numerical features, and then, with these features, a hybrid machine learning model determines the output label as nonpatches and patches. A set of one thousand images were collected during a pavement survey in Danang City in Vietnam. The photos were fixed to be  $100 \times 100$  pixels. They were labeled as nonpatches and patches by human inspectors for training. From an image, a total number of thirty-four features were identified. The least-squares support vector machine (LSSVM) was used for training with differential flower pollination (DFP) as a fine tuner. LSSVM model had an accuracy of 95.3% in predicting the road pavement condition. Compared to previous models, it can work on color images, though one of the model's limitations was that the feature selection algorithms were not established during the model construction phase (Table 2).

*3.2.3. Regression Models*. The regression modeling measures the interaction between input (independent) variables and output (dependent) variables. It is a time-series forecasting model widely used to predict pavement performance conditions. There are various regression models, including random forest regression RFR, ordinary least squares OLS regression method, simplified regression model, and stepwise regression technique, see Figure 6. Madanat and Ibrahim [56] and Roberts and Attoh-Okine [57] used the traditional regression technique to evaluate and predict road pavement degradations.

(1) *Random Forest Regression (RFR)*. Gong et al. [58] developed a random forest regression (RFR) model to predict the IRI of asphalt pavement using the training and testing sets. Pavement distresses, traffic, environmental data, and structural data were effective variables to estimate IRI. Furthermore, the previous variables and the IRI measurements were obtained from the LTPP database. The results revealed that the RFR model provided high accuracy and excellent indications on the pavement performance for the training and testing sets. The coefficients of determination  $R^2$  of the proposed model were 0.99 and 0.97 for training and testing sets, respectively. The  $R^2$  values indicate high efficiency in

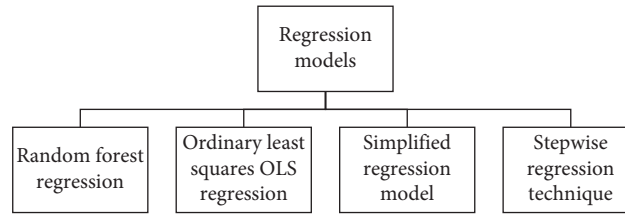


FIGURE 6: Regression models in pavement performance prediction.

implementing the RFR model. Furthermore, the results indicated that various pavement distresses and pavement age significantly influenced IRI measurements, such as alligator cracking, transverse cracking, and rutting. In contrast, others showed a limited impact on IRI measurements, including edge fracture, longitudinal cracking, and drilling.

In the same way of research, another study was conducted by Marcelino et al. [26], focused on applying a random forest algorithm for the development of pavement condition performance. A long-term pavement performance data based on the IRI values for five and ten years, as well as some other indicator factors, such as traffic volume data, environmental data, and structural data. The data were conducted for different urban roads in Canada and the U.S. (Indiana, Texas, and Saskatchewan). The main variables used in this model are annual average precipitation (AAP), annual average temperature (AAT), annual average freeze index (AAFI), pavement thickness, structural number (SN), and cumulative annual average daily truck traffic. As mentioned by Marceline and other authors, this random forest algorithm can reduce the variance of the prediction model by combining different models and performing higher accuracy results. Three categories, including quantitative, qualitative, composite of qualitative, and quantitative, were used to evaluate the prediction models [27, 59–61]. In addition to mean squared error (MSE), the standard deviation of mean squared error (SDMSE) and K-fold cross-validation were applied to estimate the number of errors in the predicted models (Table 2).

(2) *Ordinary Least Squares OLS Methods.* The development of the prediction model focused mainly on the accuracy of data sources. Arhin and Noel [62] conducted the IRI and PCI data from the Department of Transport for the selected roads in Columbia. At the same time, the ordinary least squares (OLS) regression method was performed to predict the (PCI) from IRI datasets. Additionally, Arhin and Noel [62] applied a 5% significance level to identify the significance of the proposed regression models. Subsequently, an ANOVA test was used to measure the significance of each regression model for each road classification and pavement type. The goodness of fit  $R^2$  and F-test were also tested for each regression model to estimate the validity of the proposed models. The best general regression model was formed as follows:

$$PCI = A (IRI) + K + \epsilon, \quad (15)$$

where  $A$ ,  $K$  are constants and  $\epsilon$  is an associated error.

The results showed that this prediction performance method was accurate and capable of being used in different monitoring techniques. For more illustration, based on functional classification, the results revealed that freeways were a smoother ride than arterial roads, which were smoother than collectors and local roads. Based on the pavement type, the composite pavement was smoother than asphalt and concrete pavement, respectively. The  $R^2$  values of the functional classification models ranged between 0.56 and 0.74, which was relatively low, while the goodness of fit  $R^2$  values of the pavement type models ranged between 0.72 and 0.74.

(3) *Simplified Regression Model.* Elhadidy et al. [22] focused on creating a simplified regression model to predict the relationship between pavement condition index PCI and the international roughness index IRI. The proposed model used a database from the LTPP database in America and Canada. They used variables such as traffic levels, climate conditions, pavement age, type of pavement, and pavement distresses. Moreover, Elhadidy et al. [22] evaluated the proposed model accuracy using the coefficient of determination  $R^2$  and RMSE. The study results showed that the proposed model was accurate, with a value of  $R^2$  0.99, and it could be used to predict IRI based on PCI for any pavement segment (Table 2).

(4) *Stepwise Regression Technique.* Ahmed et al. [63] focused on developing a performance prediction model based on PCI using the stepwise regression technique. The study used field observations to measure the PCI values of different types of pavement distress in urban roads in Baghdad. Furthermore, different types of pavement distress were inspected and investigated to find PCI values, including fatigue cracking, rutting, potholes, bleeding, depression, slippage cracking, longitudinal cracking, and patching, see Table 2. Ahmed et al. [63] developed a prediction model for PCI depending on the type of pavement distresses. Equation (16) illustrates the proposed model.

$$PCI = 85.336 - 0.4415(\text{Slippage}) - 2.3254(\text{Potholes}) - 37.2875(\text{Rutting}). \quad (16)$$

In the aforementioned model, only three significant types of distress were mentioned, as these distresses can provide an effective impact on PCI values. The coefficient of determination  $R^2$  for the proposed model was 0.80, indicating that the model is adequate and acceptable to transport agencies and researchers. However, there were limitations to using this model, as the model can only work with specific ranges of variables. Moreover, validation based on the mean and standard deviation of the observed and developed PCI values was applied to achieve a high-precision prediction model.  $T$ -test and mean levels were also measured at 95% to determine the accuracy of the proposed model. Equation (17) presents the relationship between expected and observed PCI.

$$PCI \text{ observed} = 0.9903 (PCI \text{ predicted}) + 3.0149. \quad (17)$$

The goodness of fit of the proposed model revealed no significant difference between the observed and predicted PCI values.

Table 2 shows the previous studies that applied shallow machine learning to predict pavement performance. The table also shows the technique used for each type of pavement indices to perform the prediction. The data sources are provided with the standard matrices used to measure each developed model's validation and accuracy. Besides, the strength and weaknesses of each model are presented and discussed.

#### 4. Discussion and Research Gaps

The main reason for the focus on the limitations of pavement prediction models is the importance and significance of these performance models in estimating the health status of pavement degradation. Once the researchers decided to develop a performance model, they must be very careful to find and select an appropriate prediction model. They also must have clear information and adequate knowledge on the model inputs, outputs, parameters, and affective variables to be used. Forming model functions and equations is a significant step in developing any prediction model. Therefore, boundary conditions must govern the equations, depending on the performance models' purpose.

In pavement prediction, performance models must have the growth of pavement degradation, distresses, and damages, or pavement performance indexes such as roughness index, serviceability index, and pavement condition rating [4]. Finding appropriate variables to be used in prediction models is considered one of the main constraints that faced many researchers. As known, the accuracy of any model is mainly related to the chosen variables. The selection of key variables depends mainly on the type of prediction model and the forecast condition. Moreover, the pavement condition variables are divided into the following main categories based on the conditions affecting pavement surfaces: traffic level, environmental condition, material quality, and paving structure.

Furthermore, many studies claimed that not all variables are available in the LTPP database or are easy to obtain. It is

considered a significant problem for the model's developer [30, 41, 64, 66]. At the end of the preparation, all elements, such as physical and mathematical boundaries, dependent and independent variables, and raw and prepared data, must be ready to be obtained and used in the developed prediction models. This review paper presents the most significant time-series prediction models, including mechanistic-empirical, empirical, regression, support vector machine, fuzzy logic, and others.

Figure 7 below presents the accuracy values depending on each prediction model's confinement of determination  $R^2$  value. MEM: mechanistic-empirical models, EM: empirical models, FL: fuzzy logic, PM: probabilistic modeling, DM: deterministic models, ANNs: artificial neural network models, MLAs: machine learning algorithms, and RM: regression models. According to Figure 7, the ANNs, MLA, and RM accuracy show a high accuracy value in predicting, classifying, and detecting pavement damage conditions. Using DM offers low accuracy to predict the actual health status of pavement, especially the NCHRP models.

Mechanistic and mechanistic-empirical performance models can estimate and extrapolate the pavement performance data. Furthermore, these models need more data to be calibrated, however, at the same time, they have simplification advantages compared with other prediction models like the empirical models [10]. However, in empirical models, George [23] claimed that selecting appropriate prediction equations is significant for developing the best performance model. To develop empirical models, the researchers should have a large dataset on pavement conditions and identify mathematical and physical boundaries of the equations to build a clear and accurate model and avoid significant errors [26].

In regression models, any equation can be used in regression analysis because of the simplicity of use. The efficiency of assumed functions or equations for the development of the regression models can be measured and evaluated using statistical measures to determine the ability of the proposed model to fit the observed data [58]. However, many researchers claimed that the coefficient of determination is a fundamental tool for assessing the adequacy of the prediction model. Still, the goodness of fit can also be evaluated using other statistical measures based on the percentage of the conducted error [58, 60, 62].

On the other hand, there are some limitations of using neural network models to predict pavement performance, including the availability of data, such as traffic level, climate condition, and other pavement condition indices in the long-term pavement performance LTPP database [42]. Furthermore, one of the most significant limitations is the need for numerical verification and statistical tests to verify the accuracy of neural network models for artificial neural networks and neuro-fuzzy models [49]. Moreover, model developers usually face difficulties in obtaining pavement condition data, especially data related to PCI, and they are unable to find suitable flexible pavement with full-service life details. Ziari et al. [49] discussed another issue in conducting the data, which is about pavement condition indices. The pavement condition index values deteriorate to the worst



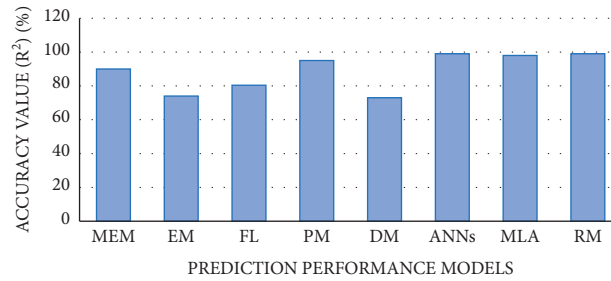


FIGURE 7: Comparing of models accuracy values.

rating class with pavement age. The pavement surface is exposed to a different climate, traffic, and other external factors that cause damages and distress. However, after applying maintenance and rehabilitation procedures, the pavement condition indices provide the best rating for pavement condition. Therefore, finding a pavement in the LTPP database with no rehabilitation or treatment process during service life is not easy [67, 68]. Besides, there is no necessity to start evaluating pavement condition from the beginning of pavement service life, where pavement age forms the last overlay.

## 5. Future Directions

Pavement condition performance is a promising area of pavement management research and is the future of monitoring and maintenance systems. Several studies have been conducted to measure the pavement condition performance and the future health states of road surfaces. Besides, the research doors are still open for more investigations and innovations to find the competitive ways that can predict future pavement performance, in addition, to providing enough information about the short and long forecasting the pavement condition performance. The main future direction should focus on selecting strength variables to efficiently develop accurate forecasting models.

Another future direction can be focusing on getting pavement condition data. It is essential to apply advanced methodologies and use highly accurate equipment to monitor the pavement condition and apply the most significant prediction model to diagnose the performance of the pavement. More clearly, using the dynamic and static pavement monitoring system to gain accurate assessment results may provide and enhance the outcomes of prediction models. For example, using accelerometer sensors with high frequency and sensitivity can provide consistent vibration data to develop prediction models. Also, advanced pavement monitoring instruments, such as probe vehicles with scanner laser and high-quality line scan cameras, are used to identify the pavement damages and use the data to validate the prediction models.

## 6. Conclusions

The increase in the number of road users results in minor and significant damage and the degradation of pavement surfaces, which mainly affects the safety and comfort of road

users. However, many researchers have conducted studies to assess the current health condition of pavement degradation and future changes in the pavement structure under the recent changes. Moreover, PMS has an essential role in developing different prediction performance models to estimate the condition of the pavement surface and the severity of pavement degradation after a specified time. This review paper sets out to the IRI- and PCI-based pavement degradation prediction model. Various prediction models have been developed to estimate pavement conditions and the level of pavement damage at various flexible pavement sites around the world.

Most previous studies have focused on developing performance prediction models based on data sets from the LTPP database and pavement state indices values. In contrast, other studies have been performed based on field observations or data collection. Many performance models have been developed using ML algorithms and ANN modeling. Most researchers agreed that both prediction methods, ML and ANN, have accurate estimation results for pavement condition, and they are beneficial in dealing with variables, such as traffic conditions, pavement age, and weather conditions. In addition, regression models showed high accuracy in detecting and classifying pavement damages. At the same time, some deterministic models showed a deficiency in predicting the actual condition of pavement surfaces.

In summary, each model has specific features, strengths, and weak points. Therefore, some prediction models are strong for multiprediction and multiclassification purposes, such as ANN, ML, and RE. In contrast, other models are significant for binary classification and detection, such as the SVM, Al Omari and Darter model, and Markov model. Hence, selecting an appropriate prediction model is the first step to a high-quality prediction performance system [69].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] A. Shtayat, S. Moridpour, B. Best, A. Shroff, and D. Raol, "Dynamic monitoring of asphalt pavement using mobile application," in *Proceedings of the 26th World Road Congress World Road Association (PIARC)*, Abu Dhabi, UAE, October 2019.

- [2] A. Shtayat, S. Moridpour, B. Best, A. Shroff, and D. Raol, "A review of monitoring systems of pavement condition in paved and unpaved roads," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 7, 2020.
- [3] A. Shtayat, S. Moridpour, and B. Best, "Using e-bikes and private cars in dynamic road pavement monitoring," *International Journal of Transportation Science and Technology*, 2021.
- [4] R. L. Lytton, "Concepts of pavement performance prediction and modelling," in *North American Conference on Managing Pavements* vol. 2, 2nd edition, 1987.
- [5] F. MoghadasNejad, R. Maskani, A. Rasouli, and R. Imaninasab, "Prediction of Pavement Condition and Maintenance and Rehabilitation Action Using Artificial Neural Network and Expert System," in *Proceedings of the 13th International Conference on Traffic and Transportation Engineering (IRAN)*, Iran, February 2014.
- [6] Z. Ghafoori, S. M. Erfani, S. Rajasegarar, J. C. Bezdek, S. Karunasekera, and C. Leckie, "Efficient unsupervised parameter estimation for one-class support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 5057–5070, 2018.
- [7] L. Premkumar and W. R. Vavrik, "Enhancing pavement performance prediction models for the Illinois Tollway System," *International Journal of Pavement Research and Technology*, vol. 9, no. 1, pp. 14–19, 2016.
- [8] F. Otto, P. Liu, Z. Zhang, D. Wang, and M. Oeser, *International Journal of Transportation Science and Technology*, 2021.
- [9] A. Abed, N. Thom, and L. Neves, "Probabilistic prediction of asphalt pavement performance," *Road Materials and Pavement Design*, vol. 20, no. 1, pp. S247–S264, 2019.
- [10] M. Ling, X. Luo, Y. Chen, F. Gu, and R. L. Lytton, "Mechanistic-empirical models for top-down cracking initiation of asphalt pavements," *International Journal of Pavement Engineering*, vol. 21, no. 4, pp. 464–473, 2020.
- [11] F. M. A. Karim, K. A. H. Rubasi, and A. A. Saleh, "The road pavement condition index (PCI) evaluation and maintenance: a case study of Yemen," *Organization, Technology & Management in Construction: An International Journal*, vol. 8, no. 1, pp. 1446–1455, 2016.
- [12] S. A. Sultan and Z. Guo, "Evaluating the performance of sustainable perpetual pavements using recycled asphalt pavement in China," *International Journal of Transportation Science and Technology*, vol. 5, no. 3, pp. 200–209, 2016.
- [13] R. Hafizyar and M. A. Mosaberpanah, "Evaluation of flexible road pavement condition index and life cycle cost analysis of pavement maintenance: a case study in Kabul Afghanistan," *International Journal of Scientific Engineering and Research*, vol. 9, no. 8, pp. 1909–1919, 2018.
- [14] A. Shtayat, M. Abu Alfoul, S. Moridpour, N. Al-Hurr, K. Magableh, and I. Harahsheh, "Waiting time of public transport passengers in Jordan: magnitude and cost," *The Open Transportation Journal*, vol. 13, no. 1, 2019.
- [15] M. S. Zafar, S. N. Raza Shah, M. J. Memon, T. Ali Rind, and M. A. Soomro, "Condition survey for evaluation of pavement condition index of a highway," *Civil Engineering Journal*, vol. 5, no. 6, pp. 1367–1383, 2019.
- [16] Z. Ye, H. Xiong, and L. Wang, "Collecting comprehensive traffic information using pavement vibration monitoring data," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 2, pp. 134–149, 2020.
- [17] T. I. Al-Suleiman, Z. M. Hamici, S. M. Bazlamit, and H. S. Ahmad, "Assessment of the effect of alligator cracking on pavement condition using WSN-image processing," in *Proceedings of the International Conference on Engineering, Project, and Product Management*, pp. 265–274, Springer, Cham, Manhattan, NY, USA, September 2017.
- [18] B. Mataei, F. Moghadas Nejad, M. Zahedi, and H. Zakeri, "Evaluation of pavement surface drainage using an automated image acquisition and processing system," *Automation in Construction*, vol. 86, pp. 240–255, 2018.
- [19] M. Radwan, A. H. Mostafa, M. Hashem, and H. Faheem, "Modeling pavement performance based on LTPP database for flexible pavements," *Teknik Dergi*, vol. 31, no. 4, Article ID 10127, 2020.
- [20] M. A. Younos, R. T. Abd El-Hakim, S. M. El-Badawy, and H. A. Afify, "Multi-input performance prediction models for flexible pavements using LTPP database," *Innovative Infrastructure Solutions*, vol. 5, no. 1, pp. 1–11, 2020.
- [21] R. Justo-Silva, A. Ferreira, and G. Flintsch, "Review on machine learning techniques for developing pavement performance prediction models," *Sustainability*, vol. 13, no. 9, p. 5248, 2021.
- [22] A. A. Elhadidy, S. M. El-Badawy, and E. E. Elbeltagi, "A simplified pavement condition index regression model for pavement evaluation," *International Journal of Pavement Engineering*, vol. 22, no. 5, pp. 643–652, 2021.
- [23] K. P. George, A. S. Rajagopal, and L. K. Lim, "Models for Predicting Pavement Deterioration," *Transportation Research Record*, no. 1215, 1989.
- [24] A. Sidess, A. Ravina, and E. Oged, "A model for predicting the deterioration of the international roughness index," *International Journal of Pavement Engineering*, pp. 1–11, 2020.
- [25] N. O. Attoh-Okine, "Predicting roughness progression in flexible pavements using artificial neural networks," *Transportation research board conference proceedings*, vol. 1, no. 1, 1994.
- [26] P. Marcelino, M. de Lurdes Antunes, E. Fortunato, and M. C. Gomes, "Machine learning approach for pavement performance prediction," *International Journal of Pavement Engineering*, vol. 22, no. 3, pp. 341–354, 2021.
- [27] S. Inkoom, J. Sobanjo, A. Barbu, and X. Niu, "Prediction of the crack condition of highway pavements using machine learning models," *Structure and Infrastructure Engineering*, vol. 15, no. 7, pp. 940–953, 2019.
- [28] S. Inkoom, J. Sobanjo, A. Barbu, and X. Niu, "Pavement crack rating using machine learning frameworks: partitioning, bootstrap forest, boosted trees, naive bayes, and K -nearest neighbors," *Journal of Transportation Engineering, Part B: Pavements*, vol. 145, no. 3, Article ID 04019031, 2019.
- [29] S. M. Piryonesi and T. E. El-Diraby, "Data analytics in asset management: cost-effective prediction of the pavement condition index," *Journal of Infrastructure Systems*, vol. 26, no. 1, Article ID 04019036, 2020.
- [30] N. D. Hoang and Q. L. Nguyen, "Automatic recognition of asphalt pavement cracks based on image processing and machine learning approaches: a comparative study on classifier performance," *Mathematical Problems in Engineering*, vol. 2018, Article ID 6290498, 16 pages, 2018.
- [31] S. Serin, N. Morova, Ş. Sargin, S. Terzi, and M. Saltan, "Modeling Marshall Stability of Lightweight Asphalt Concretes Fabricated Using Expanded clay Aggregate with Anfis," in *Proceedings of the 2nd International Balkans Conference on Challenges of Civil Engineering, BCCCE*, Tirana, Albania, July 2013.
- [32] T.-L. Nguyen, S. Kavuri, and M. Lee, "A multimodal convolutional neuro-fuzzy network for emotion understanding of movie clips," *Neural Networks*, vol. 118, pp. 208–219, 2019.

- [33] W. Li, J. Huyan, L. Xiao, S. Tighe, and L. Pei, "International roughness index prediction based on multigranularity fuzzy time series and particle swarm optimization," *Expert Systems with Applications X*, vol. 2, no. 2, Article ID 100006, 2019.
- [34] T. I. Al-Suleiman and A. M. S. Shiyab, "Prediction of pavement remaining service life using roughness data-case study in Dubai," *International Journal of Pavement Engineering*, vol. 4, no. 2, pp. 121–129, 2003.
- [35] L. Liu and N. G. Gharaibeh, "Simulation-based methodology for developing performance-related specifications for pavement preservation treatments," *Journal of Transportation Engineering*, vol. 141, no. 8, Article ID 04015011, 2015.
- [36] S. K. Suman and S. Sinha, "Pavement surface condition prediction by Markov chains," *Highway Research Journal*, vol. 9, no. 1, 2018.
- [37] C. Chen and J. Zhang, "Comparisons of IRI-Based pavement deterioration prediction models using New Mexico pavement data," in *Proceedings of the Geo-Frontiers 2011: Advances in Geotechnical Engineering*, pp. 4594–4603, March 2011.
- [38] B. Al and I. Darter, "Effect of pavement deterioration types on IRI and rehabilitation," *Transportation Research Record*, vol. 1505, p. 57, 1995.
- [39] New Mexico Department Of Transportation, "Pavement design guideline of Pavement Engineering," vol. 4, no. 2, pp. 121–129, 2008.
- [40] D. Porras-Alvarado, Z. Zhang, and L. G. L. Salazar, "Probabilistic approach to modeling pavement performance using IRI data," in *Proceedings of the 93rd Annual Meeting*, Washington, DC, USA, 2014.
- [41] S. Alimoradi, A. Golroo, and S. M. Asgharzadeh, "Development of pavement roughness master curves using Markov chain," *International Journal of Pavement Engineering*, vol. 23, no. 2, pp. 1–11, 2020.
- [42] A. M. Alsugair and A. A. Al-Qudrah, "Artificial neural network approach for pavement maintenance," *Journal of Computing in Civil Engineering*, vol. 12, no. 4, pp. 249–255, 1998.
- [43] Y. Huang and R. K. Moore, "Roughness level probability prediction using artificial neural networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1592, no. 1, pp. 89–97, 1997.
- [44] G. Tchémou, L. S. Minsili, A. M. Mokotemapa, R. M. Eko, and J. H. Manguelle, "Prediction of flexible pavement degradation: application to rutting in Cameroonian highways," *Electronic Journal of Geotechnical Engineering, EJGE*, vol. 16, pp. 1301–1319, 2011.
- [45] J. Yang, *Road Crack Condition Performance Modeling Using Recurrent Markov Chains and Artificial Neural Networks*, 2004.
- [46] H. Shahnazari, M. A. Tutunchian, M. Mashayekhi, and A. A. Amini, "Application of soft computing for prediction of pavement condition index," *Journal of Transportation Engineering*, vol. 138, no. 12, pp. 1495–1506, 2012.
- [47] M. Jalal, I. Floris, and L. Quadrifoglio, "Computer-aided prediction of pavement condition index (PCI) using ANN," in *Proceedings of the International Conference on Computers and Industrial Engineering, CIE*, Portugal, 2017.
- [48] The World Bank, *The Highway Development and Management Series*, The World Bank, Washington, DC, USA, Vol. 4, pp. 52–56, 2000.
- [49] H. Ziari, M. Maghrebi, J. Ayoubinejad, and S. T. Waller, "Prediction of pavement performance: application of support vector regression with different kernels," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2589, no. 1, pp. 135–145, 2016.
- [50] J. D. Lin, J. T. Yau, and L. H. Hsiao, "Correlation analysis between international roughness index (IRI) and pavement distress by neural network," in *Proceedings of the 82nd Annual Meeting of the Transportation Research Board*, pp. 12–16, January 2003.
- [51] N. Karballaezadeh, F. Zaremotekhas, S. Shamshirband et al., "Intelligent road inspection with advanced machine learning; hybrid prediction models for smart mobility and transportation maintenance systems," *Energies*, vol. 13, no. 7, p. 1718, 2020.
- [52] X. Wang, J. Zhao, Q. Li et al., "A hybrid model for prediction in asphalt pavement performance based on support vector machine and grey relation analysis," *Journal of Advanced Transportation*, vol. 2020, 2020.
- [53] H. Ziari, J. Sobhani, J. Ayoubinejad, and T. Hartmann, "Prediction of IRI in short and long terms for flexible pavements: ANN and GMDH methods," *International Journal of Pavement Engineering*, vol. 17, no. 9, pp. 776–788, 2016.
- [54] I. Steinwart and A. Christmann, "Support Vector Machines," *Springer Science & Business Media*, 2008.
- [55] N.-D. Hoang, "Image processing based automatic recognition of asphalt pavement patch using a metaheuristic optimized machine learning approach," *Advanced Engineering Informatics*, vol. 40, pp. 110–120, 2019.
- [56] S. Madanat and W. H. W. Ibrahim, "Poisson regression models of infrastructure transition probabilities," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 267–272, 1995.
- [57] C. A. Roberts and N. O. Attoh-Okine, "A comparative analysis of two artificial neural networks using pavement performance prediction," *Computer-Aided Civil and Infrastructure Engineering*, vol. 13, no. 5, pp. 339–348, 1998.
- [58] H. Gong, Y. Sun, W. Hu, P. A. Polaczyk, and B. Huang, "Investigating impacts of asphalt mixture properties on pavement performance using LTPP data through random forests," *Construction and Building Materials*, vol. 204, pp. 203–212, 2019.
- [59] Y. Dong, Y. Shao, X. Li et al., "Forecasting pavement performance with a feature fusion LSTM-BPNN model," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1953–1962, Beijing, China, November 2019.
- [60] X. Guo and P. Hao, "Using a random forest model to predict the location of potential damage on asphalt pavement," *Applied Sciences*, vol. 11, no. 21, Article ID 10396, 2021.
- [61] A. Ashrafian, M. J. Taheri Amiri, P. Masoumi et al., "Classification-based regression models for prediction of the mechanical properties of roller-compacted concrete pavement," *Applied Sciences*, vol. 10, no. 11, p. 3707, 2020.
- [62] S. A. Arhin and E. C. Noel, *Predicting Pavement Condition index Using International Roughness index in Washington DC (No. DDOT-RDT-14-03)*, 2014.
- [63] N. G. Ahmed, G. J. Awda, and S. E. Saleh, "Development of pavement condition index model for flexible pavement in Baghdad City," *Journal of Engineering*, vol. 14, no. 1, pp. 2120–2135, 2008.
- [64] N. Abdelaziz, R. T. Abd El-Hakim, S. M. El-Badawy, and H. A. Afify, "International Roughness Index prediction model for flexible pavements," *International Journal of Pavement Engineering*, vol. 21, no. 1, pp. 88–99, 2020.
- [65] L. Mallika, D. V. Ratnam, S. Raman, and G. Sivavaraprasad, "Performance analysis of Neural Networks with IRI-2016 and

- IRI-2012 models over Indian low-latitude GPS stations,” *Astrophysics and Space Science*, vol. 365, no. 7, pp. 1–14, 2020.
- [66] S. P. Soncim, I. C. S. de Oliveira, and F. B. Santos, “Development of fuzzy models for asphalt pavement performance,” *Acta Scientiarum. Technology*, vol. 41, Article ID e35626, 2019.
- [67] H. R. Al-Masaeid and A. Shtayat, “Performance of urban transit in Jordan,” *International Journal of Engineering Research in Africa*, vol. 6, no. 8, pp. 7–12, 2016.
- [68] M. I. Hossain, L. S. P. Gopiseti, and M. S. Miah, “International roughness index prediction of flexible pavements using neural networks,” *Journal of Transportation Engineering, Part B: Pavements*, vol. 145, no. 1, Article ID 04018058, 2019.

## Research Article

# Evaluation of the Cost of Intelligent Upgrades of Transportation Infrastructure for Intelligent Connected Vehicles

Zongwei Liu,<sup>1,2</sup> Haokun Song,<sup>1,2</sup> Hong Tan,<sup>1,2</sup> Han Hao,<sup>1,2</sup> and Fuquan Zhao <sup>1,2</sup>

<sup>1</sup>State Key Laboratory of Automobile Safety and Energy, Tsinghua University, Beijing 100084, China

<sup>2</sup>Tsinghua Automobile Strategy Research Institute, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Fuquan Zhao; zhaofuquan@tsinghua.edu.cn

Received 5 September 2021; Accepted 27 December 2021; Published 18 January 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Zongwei Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intelligent connected vehicles (ICVs) have become the focus and development direction of the automobile industry. As a flexible intelligent terminal, ICVs will become a necessary part of the intelligent transportation system. The routes of developing ICVs based on “vehicle to X” (V2X) can effectively alleviate the demands of vehicles for intelligent functions and cut related research costs, accelerating commercialization of ICVs and leading to many social benefits. At present, China has made it clear to develop ICVs based on V2X, which requires simultaneous intelligent upgrades of vehicles and transportation infrastructure. Therefore, intelligent upgrades of transportation infrastructure must match the functional requirements of ICVs. In addition, the investment in intelligent upgrades of transportation infrastructure is mainly from the government, so the costs must be controlled reasonably to find the most cost-effective upgrade route. In this paper, the types of intelligent transportation infrastructures were determined by sorting out the demands of ICVs for transportation infrastructure, and the deployment methods and upgrade routes of intelligent transportation infrastructures were designed. Then, the cost evaluation model for intelligent upgrade of transportation infrastructures was established, based on which, the cost evaluation of different intelligent upgrade routes of transportation infrastructure was carried out in closed highway and open urban road scenarios to determine the optimal route. Besides, the key elements affecting the cost of transportation infrastructure upgrades were identified, and their impact degrees on transportation infrastructure upgraded were analyzed by scenario analysis. The results show that the intelligent transportation infrastructure for advanced ICVs mainly includes communication base stations, roadside units (RSUs), vision sensors, millimeter-wave radars, laser radars (LiDARs), meteorological sensors, intelligent signal machines, edge computing servers, and cloud computing centers. The route of deploying primary intelligent transportation infrastructure at first and then directly upgrading them to advanced level can well match the functional requirements of ICVs on the basis of lower costs. The costs of RSUs, LIDARS, and edge computing servers as well as data transmission rate of 5G are key elements affecting the costs of intelligent upgrades of transportation infrastructure.

## 1. Introduction

Intelligent connected vehicles (ICVs) based on the new generation of information and communication technology have become the recognized development direction of future vehicles all over the world [1, 2]. Based on the core autonomous driving functions, ICVs can replace human beings to complete driving tasks. It can not only provide passengers with a safer, more comfortable, and intelligent driving experience but also improve travel efficiency [3–6], save energy and reduce emissions [7–9], and reduce the

traffic accident rate [10–12]. At present, the development route of autonomous driving is gradually changing from vehicle intelligence to “vehicle to X” (V2X) [13]. In the past, some automobile and automobile-related industry giants tried to develop ICVs based on vehicle intelligence, but industry practice shows that the technical difficulty of realizing autonomous driving on this route is far beyond expectation, and the costs are high. As a result, there are increasingly more countries and enterprises attaching importance to the development route of V2X [14]. Among them, China, whose government has a stronger ability of

resource coordination, is the most typical one. At present, it has made it clear that it will promote the coordinated construction of V2X with Chinese characteristics, that is, supporting and promoting the accelerated development of ICVs with intelligent transportation infrastructure.

In the mode of V2X, ICVs can share information with other connected vehicles as well as high-performance sensors and computing platforms at roadside through high-speed communication technologies such as 5G so as to realize multiple perception fusion and decision optimization. It can not only provide more comprehensive sensing information for vehicles to improve safety but also reduce performance requirements of vehicle hardware, enable redundancy, and reduce costs, which contributes to the improvement of the penetration rate of ICVs and the establishment of a new transportation system with higher efficiency and more convenience. Furthermore, it is also in line with the direction of intelligent upgrades of urban and transportation systems and will drive the upgrade of information infrastructure [15].

Therefore, the environment of intelligent transportation infrastructure is a key element supporting the future development of ICVs in China. The intelligent upgrade of transportation infrastructure should not only match the technical progress of ICVs to produce synergies but also be appropriately ahead of it and reserved to avoid repeated construction. Since cost performance evaluation is one of the significant elements to make decisions on the intelligent upgrade of transportation infrastructure, it is important to evaluate the upgrade cost of different development paths based on ICV compatibility of intelligent transportation infrastructure. At present, some of the scholars have elaborated on intelligent transportation infrastructure at the macrolevel [15–19], and others have optimized and innovated the deployment methods of them, such as communication environment [20, 21], roadside units (RSUs) [22–26], and sensors [22, 27–30]. However, there are few scholars analyzing the deployment costs of them.

Referring to the hierarchical definition of vehicle intelligence in China, intelligent transportation infrastructures were clarified by sorting out the requirements of different ICVs for them. On this basis, two typical scenarios were chosen, including open highways and closed urban roads, to evaluate the cost of intelligent transportation infrastructure through different deployment routes. In combination with the predictions of the market penetration rate and technical development level of ICVs at various levels, the optimal route for intelligent upgrade of transportation infrastructure was determined. Finally, some directional suggestions were put forward for the development of the ICV industry based on the research results.

## 2. Setting of Intelligent Levels of Transportation Infrastructure

*2.1. Classification of Intelligent Levels of Vehicles.* In the mode of V2X, the intelligent levels of vehicles should be evaluated from two aspects: automation level and connectivity level. As shown in Figure 1, referring to the classification of

autonomous driving technology by the Society of Automotive Engineers and the classification of vehicle automation and connectivity levels as set forth in the Energy-Saving and New Energy Vehicle Technology Roadmap 2.0 released by China Society of Automotive Engineers in 2020 [31], vehicle intelligence is divided into three levels: primary, intermediate, and advanced levels, each of which corresponds to a certain level of automation and connectivity.

*2.2. Requirements of ICVs for Transportation Infrastructure.* Considering the functions that need to be realized in automation and connectivity of ICVs, the functional requirements of ICVs for transportation infrastructure are divided into five aspects: communication guarantee, information transmission, safety warning, driving assistance, and extended services [16].

In terms of communication guarantee, the data transmission rate of a primary ICV in the mode of V2X is lower than 100 Kbps, and the communication delay should be controlled within 100 ms. Only the LTE communication network can meet the demand. As for an intermediate ICV, the data transmission rate is bigger than 10 Mbps, and the communication delay needs to be controlled between 20 and 100 ms. LTE communication network can no longer meet the demand, and only the 5G-NR communication network is able to guarantee the data transmission rate. As for an advanced ICV, the data transmission rate is bigger than 100 Mbps, and the communication delay must be strictly controlled within 20 ms, which can be guaranteed only by a complete 5G-NR communication network.

In terms of information delivery, the primary ICVs, on the one hand, are able to get real-time traffic and road condition information through RSUs. On the other hand, they can obtain relevant location and status information for communication with service areas. On the basis of primary ICVs, intermediate intelligent vehicles require RSUs to provide state information of nearby traffic participants to assist decision-making and update dynamic HD maps in real time. On the basis of primary and intermediate intelligent vehicles, advanced intelligent vehicles need to communicate with all urban intelligent terminals through the Internet of Things so as to realize unified vehicle scheduling and platooning.

In terms of safety warning, primary ICVs can realize real-time monitoring of illegal behaviors through RSUs and roadside sensors. Intermediate ICVs need real-time dangerous driving warning and vehicle violation warning through RSUs and roadside sensors. Advanced ICVs need real-time information sharing to avoid risks autonomously.

In terms of driving assistance, primary ICVs require intelligent transportation infrastructure to provide auxiliary perception information. In addition to auxiliary perception requirements, intermediate ICVs also need roadside edge computing servers to provide partial support of calculation and decision-making. Advanced ICVs are bound to face more complex driving tasks, requiring more efficient intelligent transportation infrastructure to assist in vehicle perception, decision-making, and control.

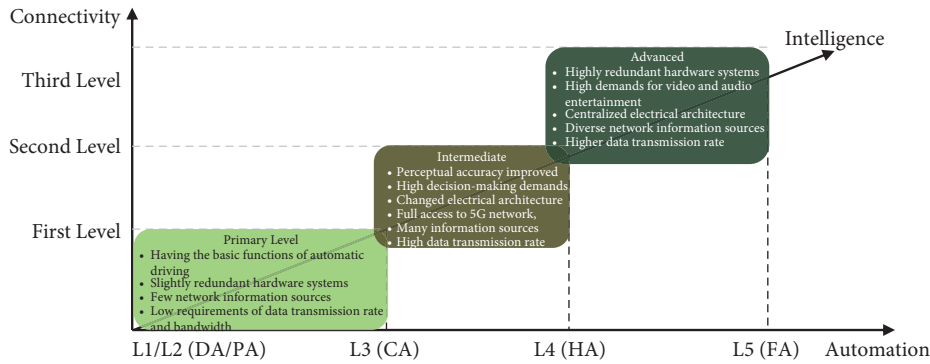


FIGURE 1: Classification of intelligent levels of vehicles.

In terms of service extension, intelligent transportation infrastructure should support sharing among intelligent vehicles, smart energy services, and so on.

**2.3. Classification of Intelligent Levels of Transportation Infrastructure.** According to the functional requirements analyzed above, the intelligent transportation infrastructure is divided into three levels, primary, intermediate, and advanced levels, as shown in Table 1.

**2.4. Architecture of Intelligent Transportation Infrastructure under Ideal Conditions.** Ideally, intelligent transportation infrastructure can be divided into four layers, as shown in Figure 2: traffic participants, sensors/actuators, edge computing servers, and cloud computing centers. In the data collection stage, data generated by people, vehicles, and roads in traffic activities are collected by sensors such as vision sensors, LiDARs, and millimeter-wave radars, transmitted to all edge computing servers for immediate processing, and finally collected to the cloud computing centers. After data analysis is completed in cloud computing centers, execution instructions are transmitted along the path of “cloud computing center-edge computing server-actuator-traffic participant,” ultimately affecting the individual terminal directly. In this process, communication base stations provide a reliable communication environment for information exchange between all levels of infrastructure so as to truly realize the real-time acquisition and transmission of data as well as update and delivery of instructions.

### 3. Cost Evaluation Model for Intelligent Upgrade of Transportation Infrastructure

**3.1. Key Assumptions of the Model.** The cost evaluation model of intelligent upgrade of transportation infrastructure is built based on the following three assumptions:

- (1) Due to the possible emergence of new intelligent transportation infrastructure in the future, the costs of intelligent upgrade will be evaluated based on the transportation infrastructure that has been mass-produced or has not been mass-produced but successfully developed; that is, the demand for

improving intelligent levels could be met by increasing the number of related known facilities deployed.

- (2) Advanced intelligent transportation infrastructure can provide ICVs with integrated services of perception, decision-making, and control. Sensors with stronger sensing ability and anti-interference ability are required. Therefore, it is assumed that the cost of sensors for primary and intermediate intelligent transportation infrastructure refers to the current market price while that of advanced intelligent transportation infrastructure would increase by 20% based on the current market price.
- (3) Intelligent transportation infrastructure is a combination of hardware and software, and the possible trend is that the importance of software will grow. However, the iteration cycle of software is short, and the cost is often lower than that of hardware and usually directly included in hardware systems. Therefore, only the cost of hardware for transportation infrastructure is evaluated. In addition, it is difficult to estimate the medium cost usually regarded as fixed expenditure, such as human input, in the process of deployment and operation management of transportation infrastructure, which will not be considered.
- (4) The change of penetration rates of ICVs at different levels exerts a direct impact on the demand for intelligent transportation infrastructure. The aim of this paper is to compare the maximum construction costs required to upgrade intelligent transportation infrastructure at different levels. Therefore, it is assumed that the penetration rate of ICVs at each level is 100% in the corresponding scenario.

**3.2. Model Construction.** In this part, the framework of the cost evaluation model is built, based on which the research work is gradually carried out. As shown in Figure 3, firstly, the required functions of road traffic infrastructure are inferred based on the intelligent levels of ICVs, based on which the corresponding levels of intelligent transportation infrastructure are set. Secondly, the plan for deploying different levels of intelligent transportation infrastructure

TABLE 1: Classification of intelligent levels of transportation infrastructure.

Intelligent levels of transportation infrastructure	Targets	Functions
Primary	Shares key information with ICVs and replaces them to complete part of the perception work	(1) Provides ICVs with information on traffic signals, real-time traffic flows, and other road conditions based on vehicle-road-cloud communication (2) Provides traffic management and law enforcement departments with road information, such as traffic accidents and violations of laws and regulations, based on road sensors (3) Provides ICVs with information of gas stations, parking lots, and other peripheral service facilities based on the network of transportation auxiliary infrastructure
Intermediate	Replaces the ICVs to complete all perception work and shares a part of the computing decision-making work through edge computing servers	(1) Provides real-time point-to-point information of surrounding roads and helps ICVs compute and make decisions based on vehicle-road-cloud-people communication and high-precision GPS positioning (2) Provides traffic participants with more peripheral service information at a faster pace with the gradual expansion of the coverage of the 5G communication environment
Advanced	Improves computing ability of ICVs and connects ICVs to intelligent urban for unified scheduling	(1) Controls traffic participants in a cooperative manner and makes all decisions for ICVs based on the communication among vehicles, roads, clouds, people, buildings, and intelligent devices (2) Smoothly provides ICVs with more diversified services, information, and more functions such as photovoltaic wireless charging

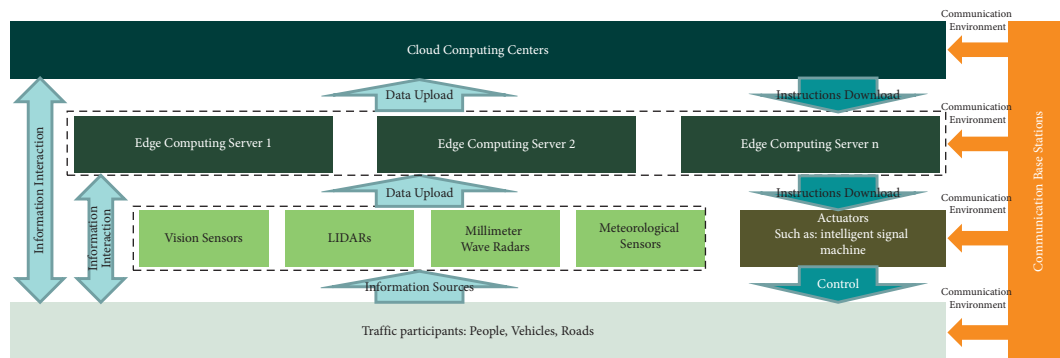


FIGURE 2: Architecture of intelligent transportation infrastructure under ideal conditions.

(namely, the types and quantities of specific facilities) under different scenarios is determined, and the corresponding one-time upgrade cost is calculated based on the coverage of various intelligent transportation infrastructure and combined with the transmission rate of road network data. Finally, the costs of four different routes to complete the deployment of advanced intelligent transportation infrastructure are evaluated, and as a result, suggestions for intelligent upgrade of transportation infrastructure are given in combination with the actual situation of the industry.

As for specific application scenarios, closed highways with large traffic flow and open roads in typical large cities in China are selected for evaluation. On the one hand,

compared with other scenarios, these two scenarios are more representative and complex, contributing to greater benefits if the intelligent upgrade of corresponding transportation infrastructure is realized. On the other hand, they are also the two scenarios that local governments in China focus on to promote the development of ICVs and intelligent transportation systems. In addition, these two scenarios correspond to the standard of Chinese road classification. Among them, Beijing-Zhangjiakou Highway is selected as a closed highway scenario, while the entire road network within the Fifth Ring Road of Beijing is selected as an open road scenario. Tables 2 and 3 show the specific attribute parameters of the two evaluation objects.



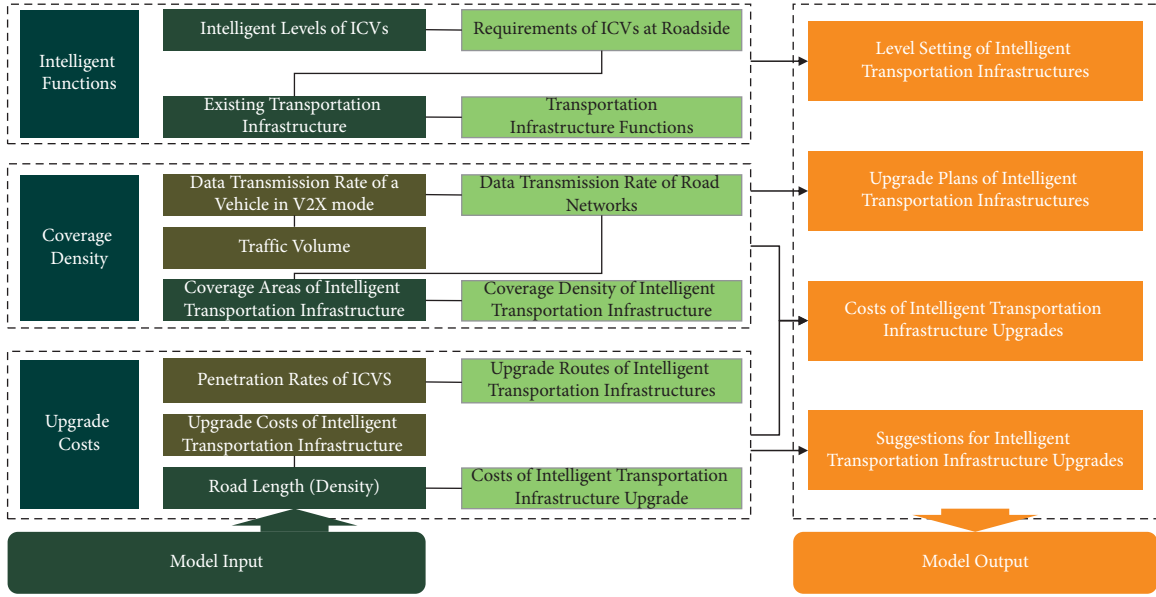


FIGURE 3: Cost evaluation model for intelligent upgrade of transportation infrastructure.

TABLE 2: Attribute parameters of the closed highway scenario.

Parameter	Value
Road length (km)	148.2
Traffic flow in one lane (number of vehicles/m)	0.02
Number of lanes (two directions)	6
The year construction started	1998
Investment	¥3,000,000,000

TABLE 3: Attribute parameters of the open urban road scenario.

Parameter	Value
Road length (km)	3,728.53
Traffic flow in one lane (number of vehicles/m)	0.1
Number of lanes (two directions)	4
Road density (km/km <sup>2</sup> )	5.59
Average density of road intersections (number of vehicles/km)	2.3

**3.3. Model Inputs.** Based on the analysis above of intelligent function requirements of traffic infrastructure, the main traffic infrastructure involved is determined to include LTE base stations, 5G base stations, RSUs, vision sensors, millimeter-wave radars, LiDARs, intelligent signals, meteorological sensors, edge computing servers, and cloud computing centers. As shown in Table 4, currently available or forthcoming transportation infrastructure with relatively good performance is selected as the model input. Its functional parameters and costs can be found in e-commerce platforms, the manufacturers' websites, product manuals, or research reports.

After the types of transportation infrastructure products are determined, the demand for various products per kilometer is calculated for different intelligent levels of transportation infrastructure.

The communication base stations are used to build communication network so as to provide communication environment for ICVs and intelligent transportation infrastructure. LTE or 5G communication base stations need to be deployed to meet both requirements of full coverage and data transmission rates.  $i$  represents a closed highway or an open urban road when it is equal to 1 or 2. On closed highways, the demand per kilometer for the communication base stations to fully cover the roads  $d_{com,1}$  can be calculated by equation (1), where  $r_{com}$  is the coverage radius of base stations.

$$d_{com,1} = \frac{1}{2r_{com}}. \quad (1)$$

On open urban roads, the deployment of communication base stations to meet requirements of full coverage is

TABLE 4: Performance parameters and costs of intelligent transportation infrastructure.

Facility	Product	Performance or cost/unit	Value
LTE base station	Zte BS8912	Coverage radius/km	1
		Transmission rate/Mbps	110
		Cost/CNY(¥)	3,000
5G base station	No specific product information, estimated according to relevant literature	Coverage radius/km	0.2
		Transmission rate/Mbps	1
		Cost/CNY(¥)	30,000
RSU	Datang Telecom DTVL3100-RSU	Coverage radius/km	1
		Cost/CNY(¥)	70,000
Vision sensor	Haikang cd7087f/DS - 2 V	Coverage radius/km	0.2
		Primary and intermediate costs/CNY(¥)	4,630
		Advanced cost/CNY(¥)	5,556
Millimeter-wave radar	Continent ARS 408-21 77 GHz	Coverage radius/km	0.25
		Primary and intermediate costs/CNY(¥)	3,500
		Advanced cost/CNY(¥)	4,200
LiDAR	Velodyne VLP-32C	Coverage/km	0.2
		Primary and intermediate costs/CNY(¥)	203,667
		Advanced cost/CNY(¥)	244,400
Intelligent signal machine	Hisense SC3101	Primary and intermediate costs/CNY(¥)	46,000
Meteorological sensor	TRM-ZS7	Deployment density on highways/ number/km	0.067
		Deployment density on urban roads/ number/km <sup>2</sup>	0.33
		Cost/CNY(¥)	75,000
Edge computing server	No specific product information, referring to research reports	Deployment density on highways/ number/km	4
		Deployment density on urban roads/ number/cross	2
		Cost/CNY(¥)	175,000
Cloud computing center	No specific product information, referring to research reports	Deployment density on highways/ number/km	0.01
		Deployment density on urban roads/ number/km	0.025
		Cost/CNY(¥)	20,000,000

shown in Figures 4 and 5, respectively. Squares surrounded by black borders in both of the figures represent urban areas of 1 km<sup>2</sup>, while circles (including full circles, semicircles, and quarter circles) represent areas covered by communication stations. In Figure 4, four gray quarter circles (1, 2, 3, and 4) and a cyan full circle (5) cover an urban area of 4 km<sup>2</sup> (2 km × 2 km). Each urban area of 1 km<sup>2</sup> needs to be covered by two-quarter circles. Similarly, in the left picture of Figure 5, four full circles (5, 6, 8, and 9), four semicircles (2, 3, 4, and 7), and a quarter circle (1) in dark green cover the square urban area of 1 km<sup>2</sup> for the first time. Then in the right picture of Figure 5, the blank left by the first coverage was filled by the four full circles (10, 11, 13, and 14), four semicircles (12, 15, 16, and 17), and a quarter circle (18) in cyan. Combined with the road density within the Fifth Ring Road of Beijing, the demand per kilometer in different scenarios for communication base stations to realize full coverage  $d_{com,2}$  can be worked out by equation (2), where  $n_{qc}$ ,  $n_{sc}$ , and  $n_{fc}$  represent the number of quarter circles, semicircles, and full circles in square areas separately and  $\rho$  represents the road density.

$$d_{com,2} = \frac{0.25n_{qc} + 0.5n_{sc} + n_{fc}}{\rho}. \quad (2)$$

Then, whether the requirements on data transmission rates are met has to be verified. If not, the number of base stations should be added on the basis of the requirements of full coverage until the requirements of data transmission rates are met. The base station increment per kilometer  $\Delta d_{com,i}$  can be calculated by equation (3), where  $C_{demand}$  represents the ideal data transmission rate and  $w_{com}$  represents the data transmission rate of a base station.

$$\Delta d_{com,i} = \frac{C_{demand} - w_{com}d_{com,i}}{w_{com}}, i = 1, 2. \quad (3)$$

The calculation above shows that, on closed highways, as for the primary intelligent transportation infrastructure, LTE base stations with full coverage can meet both of the requirements. On open urban roads, however, the number of LTE base stations should be increased. In two different scenarios above, as for the intermediate and advanced intelligent transportation infrastructure, 5G base stations

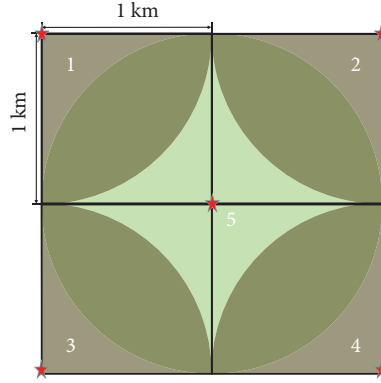


FIGURE 4: Deployment of LTE base stations on open urban roads.

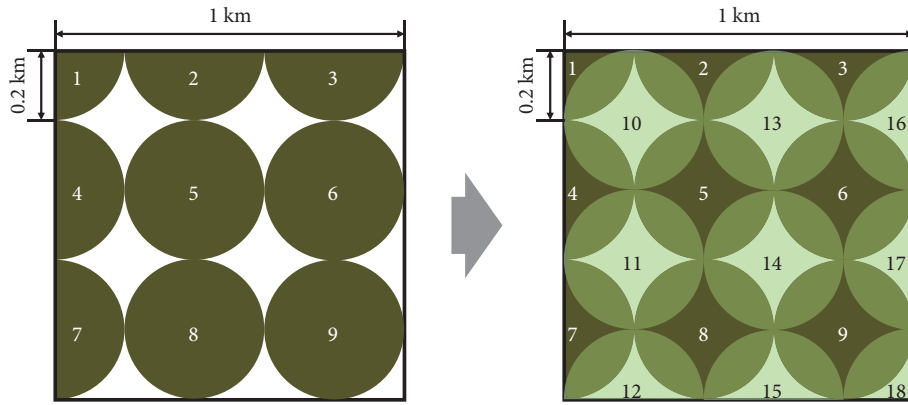


FIGURE 5: Deployment of 5G base stations on open urban roads.

instead of LTE base stations must be used to meet the requirements of data transmission rate. A possible upgrade route is clearly indicated then.

Sensors are the “eyes” of smart roads, sensing vehicle behaviors, traffic flows, emergencies, and even weather conditions in real time. The demand per kilometer for visual sensors  $d_{cam,i}$  and millimeter-wave radars  $d_{mwr,i}$  in different road scenarios can be uniformly calculated by equation (4), where  $r_{cam}$  and  $r_{mwr}$  represent the coverage radius of vision sensors and millimeter-wave radars.

$$\begin{aligned} d_{cam,i} &= 2 \cdot \frac{1}{r_{cam}}, \\ d_{mwr,i} &= 2 \cdot \frac{1}{r_{mwr}}, \quad i = 1, 2. \end{aligned} \quad (4)$$

LiDAR selected in this paper is a mechanical rotatable one, whose coverage area is a circle with the radar as the center and the coverage radius as the coverage distance. Therefore, the demand per kilometer for LiDAR in different road scenarios  $d_{li,i}$  can be calculated by equation (5), where  $r_{li}$  is the coverage radius of LiDAR.

$$d_{li,i} = 2 \cdot \frac{1}{2r_{li}}, \quad i = 1, 2. \quad (5)$$

The coverage range of meteorological sensors refers to the Chinese national standard [32]. As for closed highways, the deployment spacing should not be greater than 15 km, which means the demand for meteorological sensors is 0.067 units per kilometer. As for open urban roads, its coverage area should not be bigger than 3 km<sup>2</sup>, which means that the coverage radius is about 1 km. As a result, its deployment scheme is basically the same as that of LTE base stations in Figure 3, with a demand  $d_{wt,2}$  of 0.089 units per kilometer.

Intelligent signal machines are mainly deployed at urban intersections. Under the control of edge or central cloud control platforms, it can optimize signal timing in real time according to the optimization algorithm of traffic flows and road networks. The signal status information can also be sent to ICVs to help them pass through intersections more efficiently. In general, only one intelligent signal machine is deployed at each intersection. Combined with the road density within the Fifth Ring Road of Beijing, the demand for intelligent signal machines per kilometer of urban roads  $d_{si,2}$  is 2.3 units.

RSU is a communication gateway deployed at the roadside, which is responsible for data exchange between the ICVs, roads, and clouds. Sensors can upload data to cloud control platforms of intelligent transportation through RSUs. Therefore, in order to eliminate the perceptual blind area of roads, the number of RSUs should be the same as that of the type of sensors with the highest deployment density.

On closed highways, RSUs are mainly used to connect sensors to the network. Its demand per kilometer  $d_{RSU,1}$  can be calculated according to the following equation:

$$d_{RSU,1} = \max\{d_{cam,1}, d_{mur,1}, d_{li,1}, d_{wt,1}\}. \quad (6)$$

On open urban roads, RSUs are used to connect sensors and intelligent signal machines to the network at the same time. Its demand per kilometer  $d_{RSU,2}$  can be calculated according to the following equation:

$$d_{RSU,2} = \max\{d_{cam,2}, d_{mur,2}, d_{li,2}, d_{wt,2}\} + d_{si,2}. \quad (7)$$

Cloud control platforms of intelligent transportation are composed of basic cloud control platforms and collaborative applications. Collaborative applications mainly include control algorithms related to ICVs and intelligent transportation, while the basic cloud control platforms consist of edge computing servers and cloud computing centers, providing data, communication, and runtime environment for all kinds of collaborative applications. Therefore, the basic cloud control platforms are key components of the cyber-physical systems of intelligent transportation [16]. Due to the difficulty in obtaining pricing and functional parameters for edge computing servers and cloud computing centers, the viewpoints of several research reports are referred to [33, 34]. On closed highways, the demand for computing servers  $d_{MEC,1}$  and cloud computing centers  $d_{ctr,1}$  per kilometer is 4 units and 0.01 units, respectively. On open urban roads, the demand for computing servers  $d_{MEC,2}$  and cloud computing centers  $d_{ctr,2}$  per kilometer is 2 units and 0.025 units. As primary intelligent transportation infrastructure is only used to provide ICVs with simple information of traffic and related services, the demand for edge computing and cloud computing servers is not urgent. However, intermediate and advanced intelligent transportation infrastructure are in urgent need of edge computing servers and cloud computing centers because they are used to provide sensing and decision-making services for ICVs and even used to optimize the traffic conditions of road networks, which requires edge computing servers and cloud computing centers. Therefore, as for primary intelligent transportation infrastructure, there is probably no need to deploy cloud computing centers, and the demand for edge computing servers should be multiplied by the sum of current market penetration rates of intermediate and advanced ICVs in China [35].

Finally, by multiplying the demand per kilometer for each type of transportation infrastructure product at different intelligent levels and then adding them together, the costs required to complete the corresponding deployment at one time can be calculated.

## 4. Evaluation of the Cost of Intelligent Upgrades of Transportation Infrastructure

### 4.1. Evaluation of the Cost of Different Intelligent Upgrade Routes for Transportation Infrastructure

**4.1.1. Cost of One-Time Intelligent Upgrade of Transportation Infrastructure.** In the actual upgrade process, it is unnecessary to obey the order of intelligent upgrade of primary,

intermediate, and advanced infrastructure. Instead, a certain level of transportation infrastructure can be directly deployed ignoring some levels before it. Let  $R(l, m, n, i)$  represent different construction routes, where  $i$  represents closed highways or open urban roads when it is equal to 1 or 2, respectively. Besides,  $l, m, n \in \{1, 2, 3\}$  represents the intelligent levels of transportation infrastructure at the first, second, and third construction stages, and numbers 1, 2, and 3 refer to primary, intermediate, and advanced intelligent infrastructure. For instance,  $R(1, 3, 3, 1)$  represents the route of open urban roads in which primary transportation infrastructure is built in the first place and then directly upgraded to the advanced level.

Using the model established above, the one-time deployment costs per kilometer of primary, intermediate, and advanced intelligent transportation infrastructure under different scenarios are calculated. On closed highways, the one-time construction costs per kilometer of primary, intermediate, or advanced intelligent transportation infrastructure are ¥895,880, ¥2,772,633, and ¥3,276,160, respectively, as shown in Figure 6. The device with the highest deployment cost among primary intelligent transportation infrastructure is RSU, whose cost takes about 78% of the total. This is due to its high price and rigid and big demand for sensors. The device with the highest deployment cost among intermediate and advanced intelligent transportation infrastructure is LiDAR, whose cost takes about 36% and 37% of the total separately. This is because the intermediate and advanced ICVs have urgent demands for the roadside perception abilities and HD maps, so that LiDARs with higher perception accuracy and robustness are needed to cover the whole roads. Meanwhile, the costs of RSUs and edge computing servers are also high, taking about 25% and 21%, respectively.

On open urban roads, the one-time construction costs per kilometer of primary, intermediate, or advanced intelligent transportation infrastructure are ¥914,441, ¥4,428,008, and ¥5,913,908, respectively, as shown in Figure 7. Similar to the situation of closed highways, the device with the highest deployment cost of primary intelligent transportation infrastructure is RSU, whose cost takes approximately 83% of the total. As for intermediate and advanced intelligent transportation infrastructure, the devices with the highest, second highest, and third highest costs are LiDARs, RSUs, and edge computing servers, respectively, taking about 44% and 40%, 19% and 15%, and 18% and 14% of the total in order.

It is found that the deployment costs per kilometer of primary, intermediate, and advanced intelligent transportation infrastructure on open urban roads are ¥18,561, ¥1,655,375, and ¥2,637,748 higher than that those on closed highways, respectively. On the one hand, the geometric configuration of urban roads is network structure whose road density is much higher than that of closed highways; on the other hand, traffic volume of urban roads is much bigger than that of closed highways. Therefore, there is a higher demand for communication base stations, meteorological sensors, edge computing servers, cloud computing centers, and RSUs. In addition, intermediate and advanced

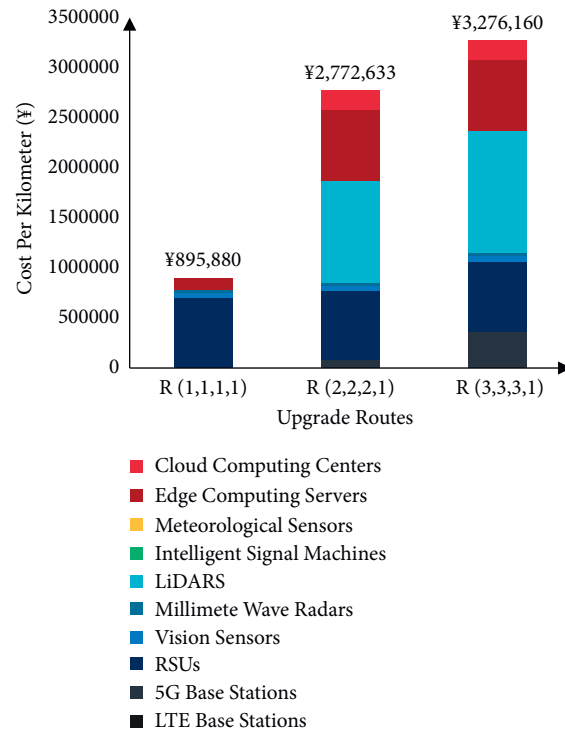


FIGURE 6: Cost of one-time upgrade of intelligent transportation infrastructure on closed highways.

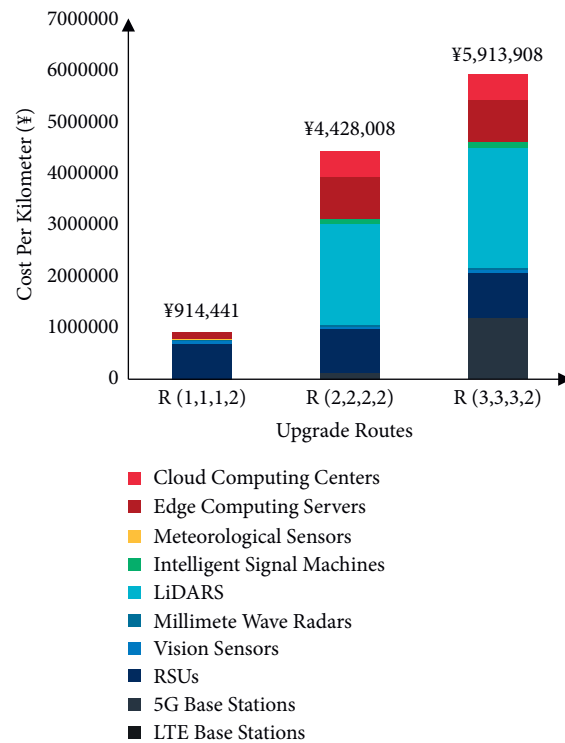


FIGURE 7: Cost of one-time upgrade of intelligent transportation infrastructure on open urban roads.

intelligent transportation infrastructure requires intelligent signal machines at intersections on open urban roads, but not on closed highways.

*4.1.2. Evaluation of the Cost of Different Intelligent Upgrade Routes for Transportation Infrastructure.* Transportation infrastructure should be upgraded to the advanced level eventually to meet the demands of advanced ICVs, so there are four possible upgrade routes, as shown in Figure 8, which need upgrades one to three times.

$R(1, 2, 3, i)$  represents the route in which infrastructure is sequentially upgraded from primary to intermediate to advanced level. When primary intelligent infrastructure is upgraded to an intermediate one, RSUs and sensors can continue to be used. But in order to satisfy intermediate ICVs on perception, decision-making, communication, and other higher requests, LTE base stations should be replaced by 5G base stations and LiDARs, and intelligent signal machines and cloud computing centers should be deployed. Besides, the number of edge computing servers should be increased. When intermediate intelligent infrastructure is upgraded to an advanced one, existing 5G base stations, RSUs, intelligent signal machines, meteorological sensors, edge computing servers, and cloud computing centers can continue to be used, but there are necessary needs to add 5G base stations and replace existing sensors with those that have higher performance. So, the costs of single-stage deployment on the step-by-step upgrade route are equal to the costs of the incremental components of the intelligent transportation infrastructure between two stages, which are less than the deployment costs of a one-time upgrade to the corresponding level. There are similar situations existing in  $R(1, 3, 3, i)$  and  $R(2, 3, 3, i)$ , which are separately considered in corresponding cost evaluations.

Figures 9 and 10 depict the total costs per kilometer of different upgrade routes on closed highway and open urban road scenarios, respectively. On closed highways, the costs of  $R(1, 2, 3, 1)$ ,  $R(1, 3, 3, 1)$ ,  $R(2, 3, 3, 1)$ , and  $R(3, 3, 3, 1)$  are ¥4370293, ¥3351960, ¥4368793, and ¥3276160. Compared with  $R(3, 3, 3, 1)$ ,  $R(1, 2, 3, 1)$ ,  $R(1, 3, 3, 1)$ , and  $R(2, 3, 3, 1)$  need additional costs of 33.4%, 2.3%, and 33.3% separately. On open urban roads, the costs of  $R(1, 2, 3, 2)$ ,  $R(1, 3, 3, 2)$ ,  $R(2, 3, 3, 2)$ , and  $R(3, 3, 3, 2)$  are ¥7,944,499, ¥5,989,299, ¥7,943,408, and ¥5,913,908. Compared with  $R(3, 3, 3, 2)$ ,  $R(1, 2, 3, 2)$ ,  $R(1, 3, 3, 2)$ , and  $R(2, 3, 3, 2)$  need additional costs of 34.3%, 1.3%, and 34.3% separately.

What is more, a comparison was made between the cost evaluation results above and the costs of ordinary roads (replaced by investment amount). As for closed highways, Beijing-Zhangjiakou Highway spanning 148 kilometers began to be built at the end of 1998, which was invested ¥3,000,000,000 then. Considering the inflation of RMB from 1999 to 2020, the total investment now is about ¥4,680,000,000, that is, ¥31,621,622 per kilometer. It is concluded that for Beijing-Zhangjiakou Highway, the intelligent upgrade cost per kilometer is only 13.8% of the

original construction cost per kilometer, even if the route with the highest costs is chosen.

As for open urban roads, it is difficult to estimate the construction costs per kilometer of urban roads due to the differences among different cities in land prices, labor costs, geographical features, and construction plans. However, because the prices of urban land are generally much higher than those of the suburban areas through which the highway passes, and the complex urban traffic conditions have high requirements for the advanced and complete degree of infrastructure, it is inferred that the cost per kilometer of urban roads is probably higher than that of highways. In addition, the deployment of intelligent transportation infrastructure on open city roads is expected to bring higher benefits since traffic volume there is far greater than that on highways.

All in all, although intelligent upgrade of transportation infrastructure leads to higher costs, it is far from being difficult to bear the burden. In other words, the V2X mode is cost-feasible at the roadside.

*4.1.3. Selection of Intelligent Upgrade Routes for Transportation Infrastructure.* As is shown above, there is no doubt that  $R(3, 3, 3, i)$  is the route with the lowest costs. But in practice, the feasibility of one-step deployment is very low. Firstly, advanced ICVs are still under research and development, and their market penetration rate is almost zero, so the payback and accumulation of advanced intelligent transportation infrastructure would be limited. Secondly, replacement is necessary when service time is beyond infrastructure life, so advanced deployment is likely to lead to serious waste. Thirdly, current advanced ICVs and transportation infrastructure are still of much uncertainty. Driven by technological progress and business model innovation, new transportation infrastructure products with higher cost performance may appear in the future. As a result, it is not appropriate to directly deploy advanced intelligent transportation infrastructure based on existing products.

Taking into account the costs of intelligent upgrading of transportation infrastructure and the expected market penetration rates of ICVs at different levels, it is concluded that  $R(1, 3, 3, i)$  is likely to be the best choice at present. First of all, it has a cost advantage. Costs of  $R(1, 3, 3, 1)$  and  $R(1, 3, 3, 2)$  are only 2.3% and 1.3% higher than  $R(3, 3, 3, 1)$  and  $R(3, 3, 3, 2)$ , respectively, lower than  $R(1, 2, 3, i)$  and  $R(2, 3, 3, i)$ . Next, at present, intermediate ICVs are likely to play only a continuing role in the technological transition stage. In fact, under specific application scenarios, there is no essential difference between L4 and L5 autonomous driving, which means that advanced ICVs will also be qualified for mass production when intermediate ICVs are fully industrialized. At last, both intermediate and advanced ICVs need a 5G communication environment, and advanced intelligent transportation infrastructure can meet the demands of intermediate and advanced ICVs. Therefore, it is reasonable to deploy intelligent infrastructure ignoring the intermediate stage.

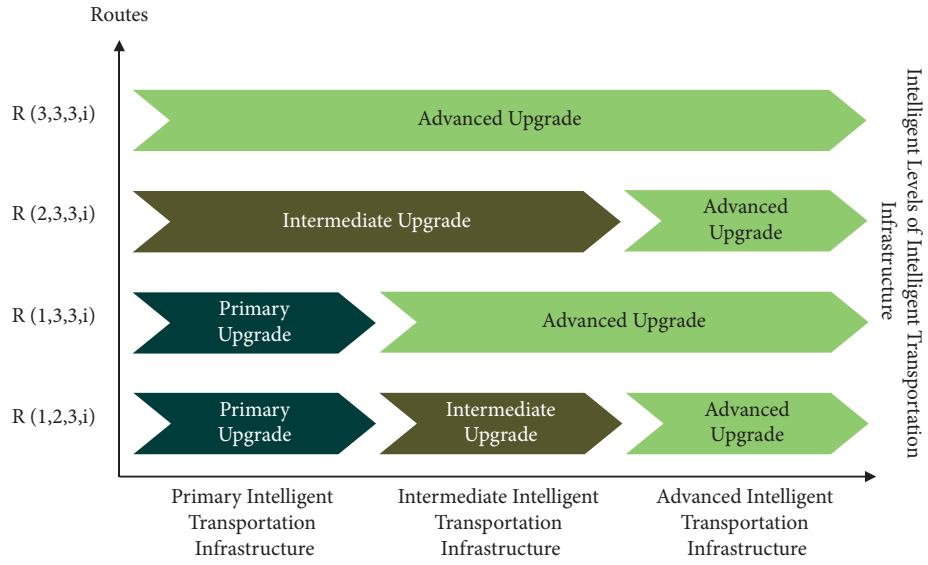


FIGURE 8: Intelligent upgrade routes of transportation infrastructure.

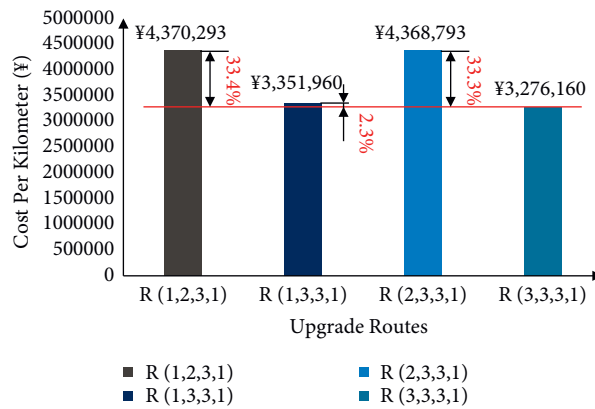


FIGURE 9: The total costs per kilometer of different upgrade routes on closed highways.

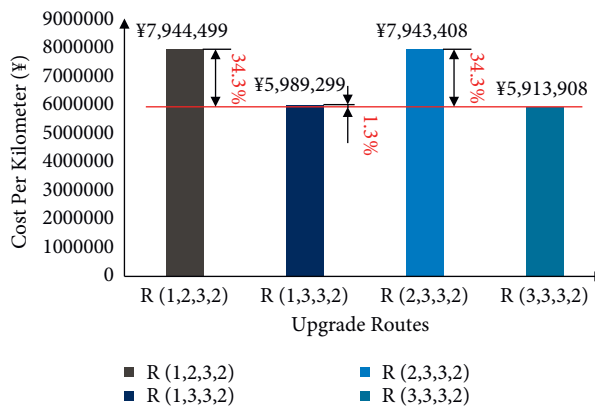


FIGURE 10: The total costs per kilometer of different upgrade routes on open urban roads.

#### 4.2. Key Elements Affecting the Cost of Intelligent Upgrade of Transportation Infrastructure

**4.2.1. Identification of Key Elements.** Figures 11 and 12 depict the share of cost components of one-time intelligent upgrade of transportation infrastructure in closed highway and open urban road scenarios. The cost components of the two scenarios are similar. In the primary stage, RSUs account for the highest cost ratio and far exceed other components due to their high costs and great demands. In the intermediate and advanced levels, LiDARs' costs are the highest, followed by RSUs' and edge computing servers', which are all due to their high costs. In addition, because the data transmission rate of 5G base stations has not yet reached the standard, the costs of 5G base stations in the advanced stage increase apparently compared to those in the intermediate stage. So, a large number of base stations have to be added to meet the demand for data transmission rate of every advanced ICV, which is approximately 100 Mbps. In summary, the data transmission rate of 5G base stations, as well as the costs of RSUs, LiDARs, and edge computing servers, are key factors affecting the total costs of intelligent transportation infrastructure.

**4.2.2. Impact of Key Elements on the Upgrade Cost.** Due to the lack of maturity and popularity of some new technologies, much transportation infrastructure is currently expensive. As the technologies gradually mature, the prices of related transportation infrastructure are expected to decline in the future [36]. Therefore, scenario analysis is adopted to analyze the sensitivity of the four key elements recognized above. There are four scenarios set for analysis:

- (1) As for an RSU, it is assumed that the mature technology will reduce its cost by 50% to ¥35,000.
- (2) As for an edge computing server, it is assumed that the mature technology will reduce its cost by 50% to ¥87,500.
- (3) For LiDAR, it is assumed that the mature technology will reduce its cost by 50% of the primary and intermediate costs, reaching ¥101,834.
- (4) As for 5G base stations, it is believed that the data transmission rate of 5G will eventually meet the standard, that is, be upgraded from 1 Gbps to 10 Gbps.

The results of the scenario analysis of the impact of key elements on the upgrade cost for both scenarios are shown in Tables 5 and 6. It is obvious that the total cost of intelligent upgrade of transportation infrastructure has changed significantly. 50% cost reduction of a LiDAR has the greatest impact on the costs of all four routes, saving 21.3% to 28.0% of the total cost on closed highways and 22.9% to 29.5% on open urban roads. 50% cost reduction of an RSU can save 8.0% to 10.7% of the total cost on closed highways and 5.4% to 7.3% on open urban roads. Similar to RSUs, 50% cost reduction of an edge computing server results in cost savings of 8.0% to 10.7% on closed highways and of 5.1% to 6.8% on open urban roads, respectively. The data transmission rate of

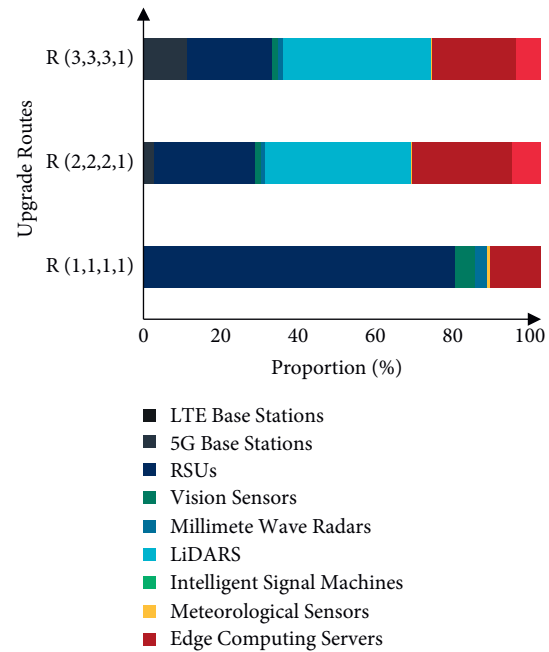


FIGURE 11: Share of cost components of one-time intelligent upgrade of transportation infrastructure in closed highway scenario.

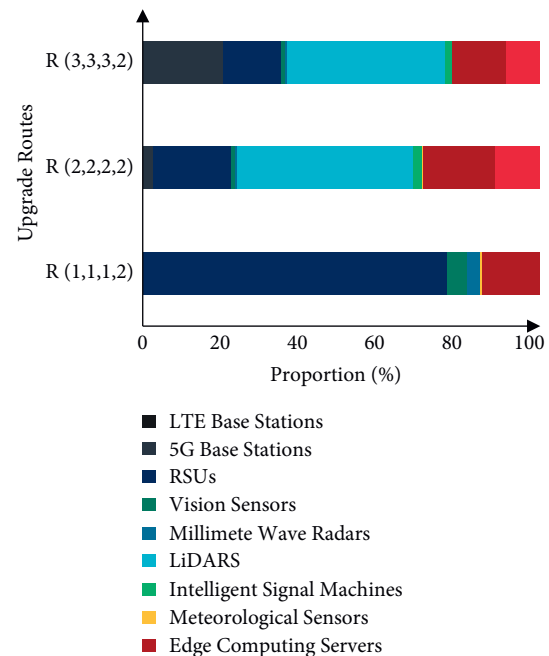


FIGURE 12: Share of cost components of one-time intelligent upgrade of transportation infrastructure in open urban road scenario.

standard-compliant 5G has a great impact on the upgrade cost on open urban roads, which can reduce cost by 13.6% to 18.3%. This is due to the higher traffic volume on open urban roads, which requires a higher data transmission rate.

As a result, faced with the upgrade demand of advanced intelligent transportation infrastructure in the future, it is of great significance to promote the development of RSUs and edge computing servers and accelerate the technological



TABLE 5: Scenario analysis: impact of key elements on the total upgrade cost per kilometer on closed highways.

Scenario	Route of closed highways			
	<i>R</i> (1, 2, 3, 1)	<i>R</i> (1, 3, 3, 1)	<i>R</i> (2, 3, 3, 1)	<i>R</i> (3, 3, 3, 1)
50% cost reduction of an RSU	-8.0%	-10.4%	-8.0%	-10.7%
50% cost reduction of an edge computing server	-8.0%	-10.4%	-8.0%	-10.7%
50% cost reduction of a LiDAR	-28.0%	-21.3%	-28.0%	-21.8%
The data transmission rate of standard-compliant 5G	-6.5%	-8.5%	-6.5%	-8.7%

TABLE 6: Scenario analysis: impact of key elements on the total upgrade cost per kilometer on open city roads.

Scenario	Route of open urban roads			
	<i>R</i> (1, 2, 3, 2)	<i>R</i> (1, 3, 3, 2)	<i>R</i> (2, 3, 3, 2)	<i>R</i> (3, 3, 3, 2)
50% cost reduction of an RSU	-5.4%	-7.2%	-5.4%	-7.3%
50% cost reduction of an edge computing server	-5.1%	-6.7%	-5.1%	-6.8%
50% cost reduction of a LiDAR	-29.5%	-22.9%	-29.5%	-23.1%
The data transmission rate of standard-compliant 5G	-13.6%	-18.0%	-13.6%	-18.3%

improvement and deployment process of 5G, which is expected to effectively reduce the cost of intelligent upgrade of intelligent transportation infrastructure.

## 5. Conclusion and Discussion

In this study, intelligent levels and corresponding functions of transportation infrastructure were defined based on ICVs with different intelligent levels and relevant demands of ICV intelligent upgrade. Based on the existing intelligent transportation infrastructure products, the possible upgrade routes were analyzed, and the types and quantities of the required intelligent transportation infrastructure products were determined. Then the cost evaluation model of intelligent upgrade of transportation infrastructure was established, based on which the costs of four intelligent upgrade routes of transportation infrastructure were evaluated under closed highway and open urban way scenarios. Furthermore, combined with actual industry situation, the best route was recommended. At last, key elements that affect the upgrade costs were identified, and their impacts were evaluated through scenario analysis.

Several conclusions could be drawn from the study. First of all, on closed highways and open urban roads, the costs of completed intelligent upgrade of transportation infrastructure are ¥3,276,160 to ¥4,370,293 and ¥5,913,908 to ¥7,944,499, respectively. Next, considering the practical feasibility, the optimal route is to first build primary intelligent transportation infrastructure and then directly upgrade them to the advanced level, ignoring the intermediate level. Then, the costs of intelligent transportation infrastructure upgrade are less than one-seventh of the construction costs of ordinary road, which are not hard to afford, and it is possible to promote ICV development in V2X mode. At last, the costs of RSUs, edge computing servers, LiDARs, and data transmission rate of 5G transmission rate are four key elements exerting apparent impacts

on the upgrade costs. Therefore, in the future, relevant technology maturation should be accelerated to reduce the costs of intelligent transportation infrastructure upgrades.

According to the evaluation results, it is feasible to accelerate the development of ICVs based on V2X mode so as to sufficiently obtain multiple benefits such as traffic efficiency, driving safety, energy conservation, and environmental protection. So, automobile, transportation, and information industries should pay more attention to V2X, especially the construction of future transportation systems, and match the intelligent functional requirements of ICVs on their own. In fact, the future human society is expected to present a new situation in which ICVs, smart transportation, and smart city develop and work in a coordinated and integrated way. As the only flexible mobile tool, vehicles are much likely to become the link and key terminal of smart transportation and smart city with the support of the Internet of Things. Hence, intelligent upgrades of transportation infrastructure based on ICV demands can generate important value to improve the overall operational efficiency and governance of a city and are expected to share the cost of V2X construction.

There are still some points to be optimized in this study. On the one hand, the cost evaluation model of intelligent upgrades of transportation infrastructure did not consider the costs of software algorithms, device operation and maintenance, manual labor, device transportation, and so on. On the other hand, only two scenarios were considered in this study, and it is necessary to analyze them in detailed road classification according to Chinese standards so as to get more comprehensive and accurate results. In addition, limited by the reliability and availability of data, the new devices which are still in the conceptual or research stage were not considered. As a matter of fact, the current pace of technological progress in the field of ICVs and smart transportation is very fast, so it is necessary to carry out the research on a rolling basis to make conclusions fit the latest situation of the industries at any time.

## Data Availability

The data used to support the findings of this study were collected from research reports and appliers, which are available from the corresponding author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (U1764265).









## References

- [1] Z. W. Liu, *Zhao Fuquan's Insights on the Automotive Industry*, China Machine Press, Beijing, China, 2020.
- [2] K. Q. Li, *Annual Report on the Development of China's Intelligent Connected Vehicle Industry*, Social Sciences Academic Press, Beijing, China, 2019.
- [3] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, 2006.
- [4] C. Stogios, D. Kasraian, M. J. Roorda, and M. Hatzopoulou, "Simulating impacts of automated driving behavior and traffic conditions on vehicle emissions," *Transportation Research Part D: Transport and Environment*, vol. 76, pp. 176–192, 2019.
- [5] Y. Lu, X. Xu, C. Ding, and G. Lu, "A speed control method at successive signalized intersections under connected vehicles environment," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 3, pp. 117–128, 2019.
- [6] A. Olia, S. Razavi, B. Abdulhai, and H. Abdelgawad, "Traffic capacity implications of automated vehicles mixed with regular vehicles," *Journal of Intelligent Transportation Systems*, vol. 22, no. 3, pp. 244–262, 2018.
- [7] D. Milakis, B. van Arem, and B. van Wee, "Policy and society related implications of automated driving: a review of literature and directions for future research," *Journal of Intelligent Transportation Systems*, vol. 21, no. 4, pp. 324–348, 2017.
- [8] A. Olia, H. Abdelgawad, B. Abdulhai, and S. N. Razavi, "Assessing the potential impacts of connected vehicles: mobility, environmental, and safety perspectives," *Journal of Intelligent Transportation Systems*, vol. 20, no. 3, pp. 229–243, 2016.
- [9] L. Qiu, L. Qian, Z. Abdollahi, Z. Kong, and P. Pisu, "Engine-map-based predictive fuel-efficient control strategies for a group of connected vehicles," *Automotive Innovation*, vol. 1, no. 4, pp. 311–319, 2018.
- [10] T. Kuehbeck, G. Hakobyan, A. Sikora, C. C. Chibelushi, and M. Moniri, "Evaluation of performance enhancement for crash constellation prediction via car-to-car communication," in *Proceedings of the International Workshop on Communication Technologies for Vehicles*, pp. 57–68, Midtown Manhattan, NY, USA, 2014.
- [11] H. Tan, F. Zhao, H. Hao, and Z. Liu, "Estimate of safety impact of lane keeping assistant system on fatalities and injuries reduction for China: scenarios through 2030," *Traffic Injury Prevention*, vol. 21, no. 2, pp. 156–162, 2020.
- [12] Y. Li, Z. Li, H. Wang, W. Wang, and L. Xing, "Evaluating the safety impact of adaptive cruise control in traffic oscillations on freeways," *Accident Analysis & Prevention*, vol. 104, pp. 137–145, 2017.
- [13] F. Q. Zhao, Z. W. Liu, and K. Q. Yang, *Automotive Technology Innovation*, China Machine Press, Beijing, China, 2019.
- [14] K. Q. Li, Y. F. Dai, and S. B. Li, "State-of-the-Art and technical trends of intelligent and connected vehicles," *Journal of Automotive Safety and Energy*, vol. 8, no. 1, pp. 1–14, 2017.
- [15] Y. P. Wang and X. P. Yan, *Introduction to Intelligent Transportation Technology*, Tsinghua University Press, Beijing, China, 2020.
- [16] X. Y. Chang, *Intelligent and Connected Vehicles' Cloud Control System and its Control Technology*, Tsinghua University (Doctor's thesis), Beijing, China, 2020.
- [17] H. Wieker, J. Vogt, and M. Fuenfrocken, "Intelligent transportation system infrastructure and software challenges," in *Automotive Systems and Software Engineering*, pp. 295–319, Springer, Cham, Midtown Manhattan, New York City, 2019.
- [18] J. M. León-Coca, D. G. Reina, S. L. Toral, and F. Barrero, "Intelligent transportation systems and wireless access in vehicular environment technology for developing smart cities," *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 285–313, Springer, Cham, Midtown Manhattan, NY, USA, 2014.
- [19] M. Mirboland and K. Smarsly, "BIM-based description of intelligent transportation systems for roads," *Infrastructure*, vol. 6, no. 4, p. 51, 2021.
- [20] J. Xu, L. Wu, L. Shi, Y. Shi, and W. Zhou, "Research on 5G Internet of vehicles facilities based on coherent beamforming," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*, pp. 68–77, Midtown Manhattan, NY, USA, 2020.
- [21] X. Zhang, J. Jiang, L. Qi, and X. Xi, "An Edge Server Placement Method with Cyber-Physical-Social Systems in 5G," in *Proceedings of the 12th EAI International Conference in Simulation Tools and Techniques SIMUtools 2020, August 28-29, 2020, Proceedings, Part II 12*, pp. 127–139, Guiyang, China, August 2020.
- [22] W. Zhang, *Research on Layout Optimization of Hybrid VANET-Sensor Network for Autonomous Driving*, Master's thesis, Dalian Maritime University, Dalian, China, 2019.
- [23] H. Zhu, *Research of RSU Deployment Algorithms for VANET*, Master's thesis, Dalian Maritime University, Dalian, China, 2016.
- [24] F. G. Abdulkadhim, Z. Yi, A. N. Onaizah, F. Rabee, and A. M. A. Al-Muqarm, "Optimizing the roadside unit deployment mechanism in VANET with efficient protocol to prevent data loss," *Wireless Personal Communications*, pp. 1–29, 2021.
- [25] M. Al Shareeda, A. Khalil, and W. Fahs, "Towards the Optimization of Roadside Unit Placement Using Genetic Algorithm," in *Proceedings of the 2018 International Arab Conference on Information Technology (ACIT)*, pp. 1–5, Islamic University of Lebanon, Lebanon, November 2018.
- [26] L. Sun, Y. Wu, J. Xu, and Y. Xu, "An RSU-Assisted Localization Method in Non-GPS Highway Traffic with Dead Reckoning and V2R Communications," in *Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 149–152, Yichang, China, April 2012.
- [27] H. J. Li, H. H. Dong, D. W. Xu, and L. Jia, "Urban road traffic oriented sensor networking optimization," *Journal of Central South University*, vol. 43, no. 3, pp. 1192–1197, 2012.
- [28] X. Xu, H. K. Lo, A. Chen, and E. Castillo, "Robust network sensor location for complete link flow observability under

- uncertainty,” *Transportation Research Part B: Methodological*, vol. 88, pp. 1–20, 2016.
- [29] F. Zhan, X. Wan, Y. Cheng, and B. Ran, “Methods for multi-type sensor allocations along a freeway corridor,” *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 134–149, 2018.
- [30] A. Danczyk, X. Di, and H. X. Liu, “A probabilistic optimization model for allocating freeway sensors,” *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 378–398, 2016.
- [31] Strategic Advisory Committee of Energy-Saving and New Energy Vehicle Technology Roadmap, “China Society of Automotive Engineering,” *Energy-saving and New Energy Vehicle Technology Roadmap 2.0*, China Machine Press, Beijing, China., 2020.
- [32] China Association for Road Traffic Safety, *Roadside Facilities Setting Guide for Internet of Vehicles*, Standards Press of China, Beijing, China, 2020.
- [33] Sinolink Securities, *Research on Communication Industry*, Sinolink Securities, Shanghai, China, 2020.
- [34] Z. Y. Yang, W. S. Liu, and S. Zhang, *V2X Internet of Vehicles: The Leader of 5G New Infrastructure*, Citic Securities, Beijing, China, 2020.
- [35] X. Kuang, *Research on the Benefits and Business Models of Intelligent Connected Vehicles*, Doctor’s thesis, Tsinghua University, Beijing, China, 2019.
- [36] M. A. Schilling, W. Xie, Y. Wang, and P. X. Li, *Strategic Management of Technology Innovation*, Tsinghua University Press, Beijing, China, 2005.

## Research Article

# Crack Detection Method of Sleeper Based on Cascade Convolutional Neural Network

Liming Li <sup>1,2,3</sup> Shubin Zheng <sup>1,3</sup> Chenxi Wang <sup>1</sup> Shuguang Zhao <sup>2</sup>  
Xiaodong Chai <sup>1,3</sup> Lele Peng <sup>1,3</sup> Qianqian Tong <sup>1</sup> and Ji Wang <sup>1</sup>

<sup>1</sup>School of Urban Railway Transportation, Shanghai University of Engineering Science, Shanghai 201620, China

<sup>2</sup>School of Information Science and Technology, Donghua University, Shanghai 201620, China

<sup>3</sup>Shanghai Engineering Research Center of Vibration and Noise Control Technologies for Rail Transit, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Shubin Zheng; shubin.zheng@sues.edu.cn

Received 30 July 2021; Accepted 16 December 2021; Published 11 January 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Liming Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a new method for sleeper crack identification based on cascade convolutional neural network (CNN) to address the problem of low efficiency and poor accuracy in the traditional detection method of sleeper crack identification. The proposed algorithm mainly includes improved You Only Look Once version 3 (YOLOv3) and the crack recognition network, where the crack recognition network includes two modules, the crack encoder-decoder network (CEDNet) and the crack residual refinement network (CRRNet). The improved YOLOv3 network is used to identify and locate cracks on sleepers and segment them after the sleeper on the ballast bed is extracted by using the gray projection method. The sleeper is inputted into CEDNet for crack feature extraction to predict the coarse crack saliency map. The prediction graph is inputted into CRRNet to improve its edge information and local region to achieve optimization. The accuracy of the crack identification model is improved by using a mixed loss function of binary cross-entropy (BCE), structural similarity index measure (SSIM), and intersection over union (IOU). Results show that this method can accurately detect the sleeper crack image. During object detection, the proposed method is compared with YOLOv3 in terms of directly locating sleeper cracks. It has an accuracy of 96.3%, a recall rate of 91.2%, a mean average precision (mAP) of 91.5%, and frames per second (FPS) of 76.6/s. In the crack extraction part, the F-weighted is 0.831, mean absolute error (MAE) is 0.0157, and area under the curve (AUC) is 0.9453. The proposed method has better recognition, higher efficiency, and robustness compared with the other network models.

## 1. Introduction

China's total railroad mileage is expected to exceed 128,000 km by the end of 2020, prompting researchers to improve maintenance techniques for railroad infrastructure [1]. In Figure 1, the sleeper is used to support the rail and transfer the huge impact brought by the train to the roadbed. Accordingly, the sleeper needs to have a certain degree of flexibility and can be slightly deformed to cushion the pressure. However, the cracks and other damage generated within it will undermine the integrity of the sleeper and diminish the support force provided by the sleeper to the train above when the load bending moment is greater than the cracking strength. This situation poses a safety hazard to

trains passing at a high speed. In recent years, nondestructive testing techniques, such as those in the literature [2], have been widely used in the maintenance of track facilities. This method of sleeper cracking can be quick and efficient in preventing accidents.

At present, the main method of sleeper crack detection has shifted from manual identification to a series of physical detection means, such as ultrasonic, eddy current detection, and ray detection. Although this method has been developed, it still has the limitations of the use methods and the common problem of poor crack detection. The efficiency and accuracy of crack detection have been enhanced with the development of the computer vision technology. The main methods applied to this field are as follows: image

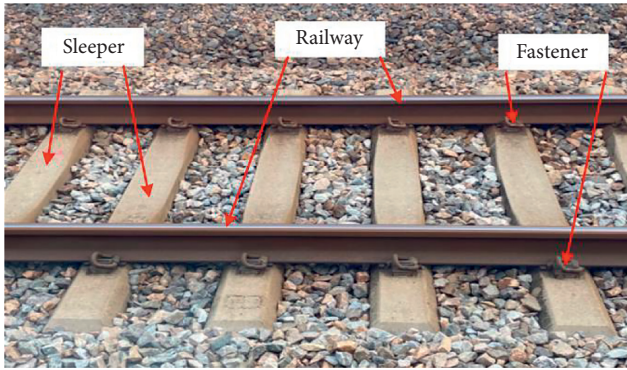


FIGURE 1: Railway track line.

processing-based methods [3], machine learning-based methods [4], and deep convolutional neural network (DCNN)-based methods [5]. The methods represented by DCNNs are subdivided into methods based on image classification [5], object detection [6], and pixel-level segmentation [7], depending on the way the crack detection problem is handled. The network used to detect cracks in sleepers in this cascade is based on the latter two types of methods.

The main crack detection methods based on object detection include Faster R-CNN [8], single-shot multibox detector (SSD) [9], and You Only Look Once (YOLO) [10] to determine the location of cracks in the input image and localize them with bounding boxes. Cha et al. [11] proposed a concrete crack detection method based on Faster R-CNN. The network is improved to quickly detect and locate multiple types of cracks in real time, allowing for more accurate detection results. Mandal et al. [12] proposed an automated detection method based on DCNNs for road concrete cracks. However, the achieved detection accuracy is low. Li et al. [13] proposed an improved YOLO network to improve the detection accuracy of track plate cracks. However, the method is less versatile due to the single background information of the track plate. Bao et al. [14] proposed a triplet graph reasoning network for the problem of insufficient samples of metal surface defects.

Crack detection methods based on pixel-level segmentation mainly include fully convolutional networks (FCNs) [15], U-Net [16], and Seg-Net [17]. Labels can be assigned to crack pixel points to determine the presence of cracks and to obtain important features, such as the location, size, and shape of cracks. Cheng et al. [18] proposed an automatic U-Net-based road crack detection method and tested it in a crack dataset to obtain a high pixel-level segmentation accuracy. Islam and Kim [19] proposed a full CNN-based concrete crack detection method. This network consisting of encoder and decoder patterns is tested and exhibits good detection results on publicly available crack datasets. Dung [20] designed a full CNN with Visual Geometry Group-16 (VGG-16) based on a codec framework. This network further improves the accuracy of crack detection. Literature [21] compared three U-Net algorithms of different depths for automatic pavement crack detection systems. The objective is to verify whether a model architecture with greater

depth necessarily results in better detection accuracy. Experiments prove that choosing a network architecture with the right depth can guarantee the detection accuracy and improve the detection speed.

Although great progress has been made in the field of crack detection based on DCNNs, how to obtain more detailed crack features still needs to be explored. For the sleeper crack detection, the crack is small, similar to the background of the sleeper, the boundary is unclear, and the regional information is incomplete. This paper proposes a new cascade network for crack detection. YOLOv3 is used as one of the mainstream frameworks for object detection. The YOLO series is improved on the basis of YOLOv3. Given that YOLOv3 uses a residual network in the feature extraction part, three feature layers of different depths are simultaneously extracted, and a stacked stitching approach is used to obtain the prediction results [22]. The aforementioned method can be used to detect cracks of different sizes. However, the crack detection effect is unsatisfactory for the complex background of the rail sleeper. Accordingly, we add the squeeze and excitation (SE) module at the end of the YOLOv3 backbone network to improve the crack region extraction accuracy. Further quantitative parameter detection of cracks is needed to complete high-precision crack identification and provide more scientific detection data. Crack encoder-decoder network (CEDNet) and crack residual refinement network (CRRNet) are used to extract and optimize the features of rail sleeper cracks. The shallow information of the crack image can be passed to the corresponding decoding process after the feature extraction of the input rail cracks by the coding part of CEDNet. Consequently, the low-level detail features are fused with the high-level complex semantics to improve the network feature extraction performance. CRRNet is added because the coarse saliency map obtained in the previous step has deficiencies, such as blurred crack boundaries and missing important regions. CRRNet can be optimized by learning the residuals between the coarse saliency map and the ground truth.

The main contributions of this paper are summarized as follows:

- (1) A two-level cascade network based on DCNN is proposed. This network fuses CEDNet and CRRNet, which can play the role of crack feature extraction and optimization in one step. Its F-weighted is 0.831, mean absolute error (MAE) is 0.0157, and area under the curve (AUC) is 0.9453.
- (2) An improved YOLOv3 network is proposed to localize the cracks, and the attention mechanism, SE module, is added at the end of the backbone network. The mean average precision (mAP) is improved by 6.9% compared with YOLOv3.
- (3) The optimization effects of loss functions binary cross-entropy (BCE), intersection over union (IOU), and structural similarity index measure (SSIM) on crack recognition are superimposed to propose a new hybrid loss function for the crack recognition. Particularly, our method improves  $F_{\text{weighted}}$  by

68.4%, 74.8%, 84.1%, and 99.0% on  $l_{bce} + l_{iou}$ ,  $l_{bce}$ ,  $l_{iou}$ , and  $l_{ssim}$ , respectively.

The rest of this paper is organized as follows: Section 2 introduces the method overview, including the overall steps and the specific theory for each step. Section 3 shows some experimental results of our method and compares them with other methods. Section 4 gives the conclusion and outlook.

## 2. Method Overview

In the acquired image of rail sleeper cracks, the edge of ballast can interfere with the recognition of rail sleeper cracks because the imaging of ballast and concrete rail sleeper is similar. Given that the edge of the rail sleeper has obvious features, a strict size regulation, and differs from the grayscale of the ballast, the rail sleeper area can be first segmented. The cracks on the rail sleeper can then be located and identified by using the network. The proposed crack detection algorithm is divided into two parts: crack localization and crack identification. The crack recognition part incorporates a feature extraction network and a boundary refinement network. The overall methodological flow is shown in Figure 2. In the first step, we choose the gray projection method to extract the sleeper area first because the large amount of ballast in the background of the sleeper affects the crack detection. In the second step, a modified YOLOv3 is used to locate and segment the cracks on the basis of the extraction of the rail sleeper area. In the third step, further quantitative parameter detection of cracks is needed to complete high-precision crack identification and provide more scientific detection data; hence, CEDNet is used for feature extraction. A boundary refinement network is designed for further optimization because the extracted cracks have partial boundary and region information incompleteness:

- (1) The location of the sleeper is extracted by using the gray projection method [23] combined with the empirical value of the sleeper pixels, and then, SE [24] and spatial pyramid pooling (SPP) [25] are added at the end of the YOLOv3 backbone network to locate the sleeper cracks
- (2) CEDNet, a crack coarse saliency feature extraction network, is used to obtain more detailed saliency information by fusing low-level features and high-level features of crack images through the network structure of codec patterns
- (3) CRRNet, a crack boundary refinement network, is used to learn the residuals between the original and ground truth maps of the crack for optimization purposes by fusing the outputs of the network feature layers

**2.1. Crack Location Module.** The dimensions are strictly defined, and they differ from the ballast grayscale because the sleeper edge features are obvious. The gray projection method combined with the empirical values of the sleeper

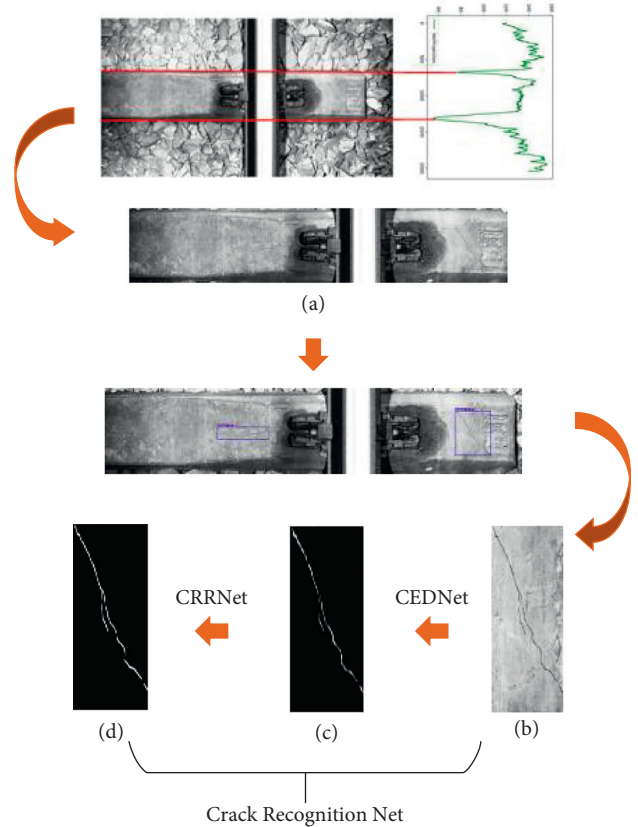


FIGURE 2: Process of the proposed method. (a) Gray projection. (b) Improved YOLOv3. (c) Feature extraction. (d) Edge refinement.

pixels can be used to locate the position of the sleeper. The gray projection method has better results for object edge detection with complex backgrounds, relying mainly on the peaks and valleys in the gray projection curve to determine the coordinates of the object edge position. Assuming that the image is represented as  $f(x, y)$ , the gray projection function in the  $x$ -direction is  $f_x(x)$ , the coordinates of the pixel points in the image are  $(x, y)$ , and the value of the gray projection function in the horizontal direction is

$$f_x(x) = \sum_y f(x, y). \quad (1)$$

The edge coordinates of the horizontal direction of the sleeper can be obtained in accordance with the gray projection method. The pixel width of the edge of the sleeper is relatively fixed in the captured roadbed images. Figure 3(a) shows the original drawing of the ballasted roadbed. The valley of the horizontal projection in Figure 3(b) depicts the contact edge between the sleeper and the ballast. Figure 3(c) presents the segmentation results.

The prediction results are obtained by stacking and splicing after simultaneously extracting three feature layers with different depths because YOLOv3 uses a residual network in the feature extraction part. Therefore, this network can be used to detect cracks of different sizes. However, in the complex background of the sleeper, the crack detection effect is poor. Inspired by the literature [24–26], the SE module

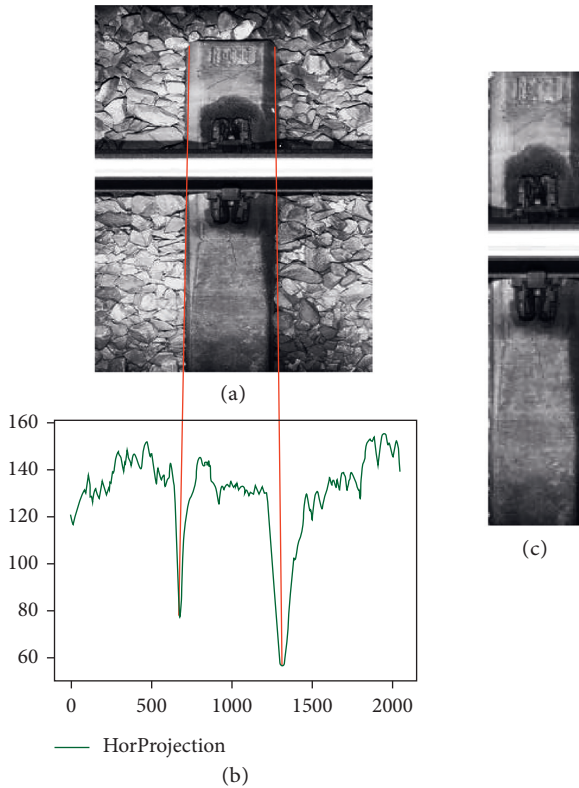


FIGURE 3: Gray projection experimental results. (a) Original image. (b) Horizontal projection. (c) Segmentation results.

suppresses the interference of background and other noises, and the SPP module can improve the operation efficiency by relieving the network of the size requirement for input images while ensuring that the images are not distorted. The end-to-end semisupervised object detection method, the object detection head with unified awareness from the attention perspective, and Composite Backbone Network Version2 (CBNetV2), which eliminates the pretraining process, can avoid the more complex multistage training approach in the literature [27–29]. However, the algorithms in the above documents still have some shortcomings, such as slow detection speed, large consumption of network resources, low accuracy and recall rate, and poor detection accuracy. Therefore, we choose to add SE and SPP modules at the end of the backbone network to make the model simpler in the training process and to improve the accuracy of crack region extraction while minimizing additional overhead. An improved algorithm based on YOLOv3 is designed in this paper, and its overall structure is shown in Figure 4.

The SE module belongs to one of the more classical algorithms of the attention mechanism. The accuracy of crack detection can be significantly improved by designing special parameters capable of removing the invalid information extracted by the YOLOv3 network [25]. This module compresses the sleeper crack image to a size of  $1 \times 1 \times 1024$  after a global averaging pooling layer. The activation is performed by two modules in fully connected layers and activation functions. The crack feature channels are weighted uniformly. The designed residual module ensures

effective training so that the network extracts more accurate information about crack features and suppresses interference from other noises in the sleeper images.

When performing prediction of the a priori frame on three scales of the crack image, YOLOv3 requires consistent size of the crack feature maps outputted by the backbone feature extraction network. The cropping or shape change of the image tends to cause partial loss of information, resulting in biased crack detection results. Accordingly, the SPP module is added after the SE module to remove the limitation of the fixed size of the input image [26]. The sleeper crack images outputted from the backbone network of this module are simultaneously pooled at three scales after one convolution operation. The output crack features are fused and inputted to the fully connected layer. We can obtain a fixed size crack image output without losing the original information for any size and scale of the crack image input.

**2.2. Crack Recognition Module.** After locating and segmenting the cracked area of the rail sleeper, this paper proposes a crack identification module to obtain more detailed crack characteristics. The module uses a crack boundary refinement network to optimize the predicted saliency map because the extracted crack information is incomplete. The final crack saliency map is obtained by fusing the crack boundary refinement network with the feature extraction network, and the general block diagram of this module is shown in Figure 5.

**2.2.1. Feature Extraction Module.** The backbone network used for feature extraction is the crack coarse saliency feature extraction network CEDNet, which is a codec network focusing on crack regions and boundaries. The network is built on the basis of ResNet-34 (Residual Network with 34 parameter layers) [30] using a codec form. After feature extraction of the input sleeper cracks in the encoding part, the resulting image features are further optimized and processed by the decoding part. The shallow information of the cracked image is passed to the corresponding decoding process, which enables the fusion of low-level detailed features with high-level complex semantics as a method to improve the network feature extraction performance. The structure is shown in Figure 6.

The specific structure and operational steps of the network are as follows:

- (1) The coding part consists of an input convolutional layer and six stages consisting of basic residual blocks, with a modified ResNet-34 structure for the input convolutional layer and the first four convolutional stages. The improvements mainly include the use of a  $3 \times 3$  convolution filter and a convolution kernel with a stride of 1. The pooling operation is removed after the input convolutional layer to guarantee that the feature map in the first stage has the same spatial resolution as the input image. By contrast, the first feature map in the original ResNet has only one-quarter of the resolution of the input map. This change allows the network to obtain higher resolution feature maps in

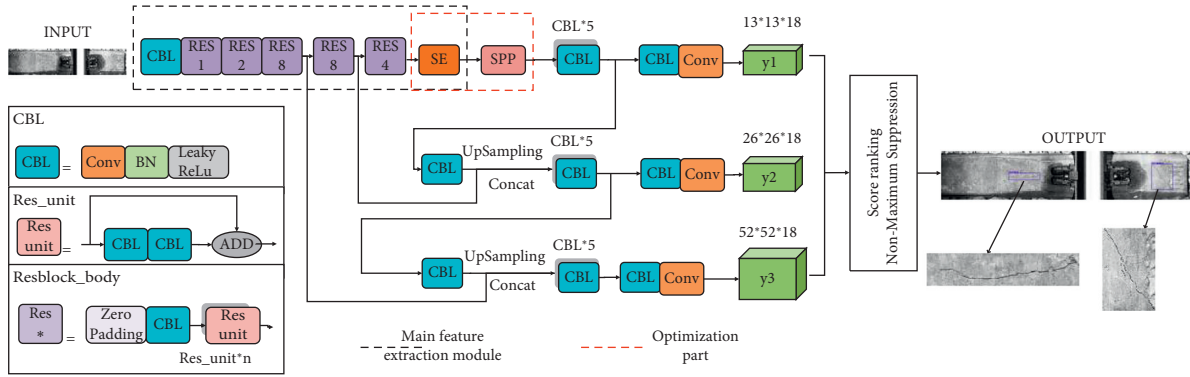


FIGURE 4: Improved YOLOV3 structure process diagram.

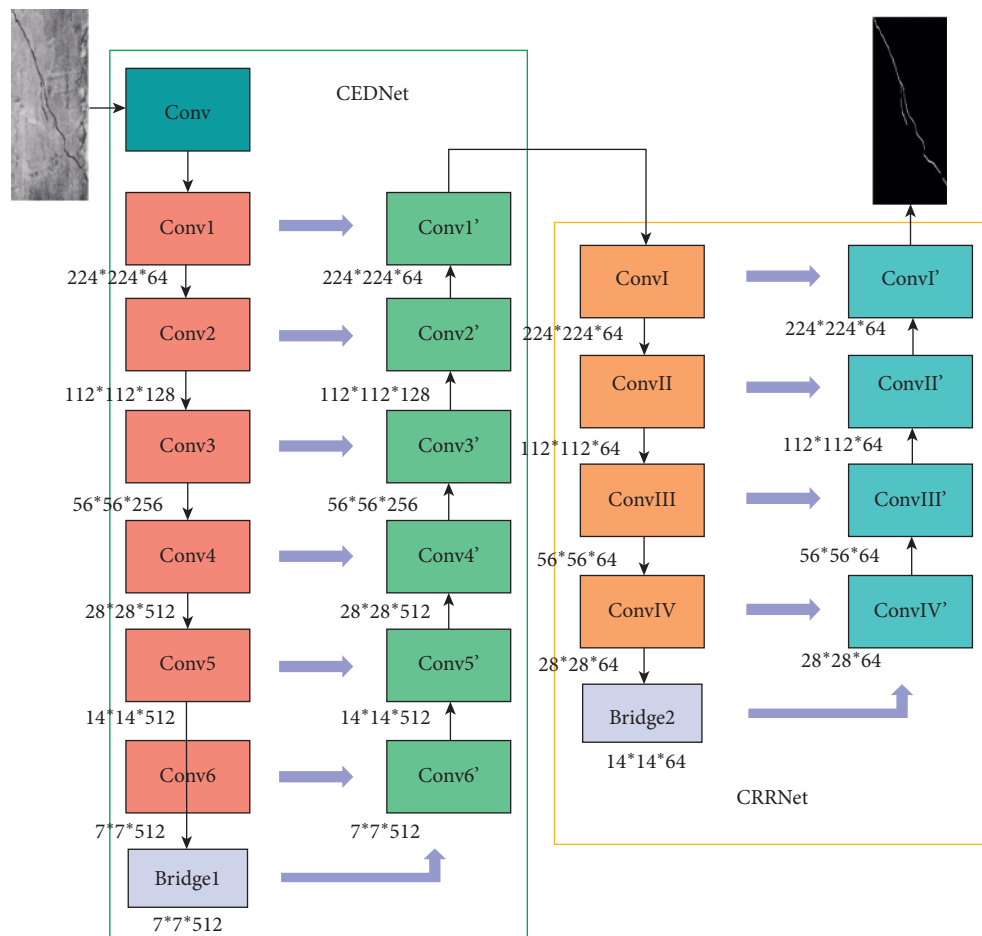


FIGURE 5: Crack recognition module.

previous layers although reducing the overall receptive field. Consequently, Conv5 and Conv6, which are two convolutional stages consisting of 512 filters and three basic residual blocks, are added to obtain a greater extent of the object detection region on the original map and achieve the same receptive field as the original ResNet.

- (2) A bridge connection structure is used to further obtain the global information of cracks. The bridge connection structure contains three modules

consisting of a Conv layer, a batch normalization (BN) layer [31], and a rectified linear unit (ReLU) activation function [32], where each convolutional layer consists of 512  $3 \times 3$  dilated convolutions [33].

- (3) The input of each level of the decoding section is cascaded from the previous level and the pooled output of the corresponding level in the encoding section. A sigmoid function is added to each layer after using bilinear up-sampling for mapping the predicted values to  $[0, 1]$ . Seven saliency mappings are generated in this



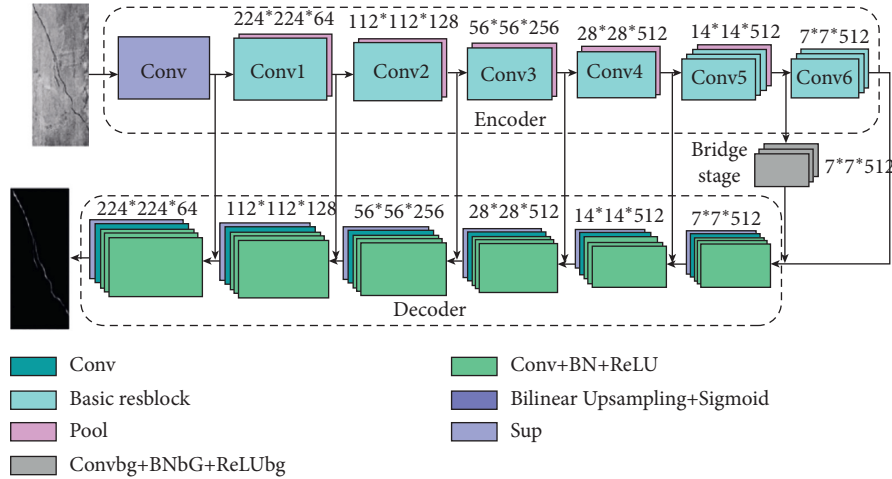


FIGURE 6: CEDNet.

module, containing six postcascade feature mappings and the final output feature mapping. However, only the last feature map with the highest accuracy can be inputted into the CRRNet. The supervision of the ground truth map is supervised at the last layer of each decoding stage to reduce overfitting, as in holistically nested edge detection [34].

**2.2.2. Edge Refinement Module.** After the object detection and feature extraction, the predicted crack coarse saliency map can be obtained for the sleeper cracks. Figure 7 shows the original map of cracks, the ground truth map, and the coarse saliency map after the CEDNet extraction.

In the coarse saliency map, the crack boundary is blurred, some salient regions are missing, and the background is incorrectly marked as the object and inaccurately located. Therefore, the boundary information and local details of the extracted crack feature map are incomplete. Therefore, the extracted feature map is fed into CRRNet for further optimization.

The network is built in codec form and achieves optimization by learning the residuals between the original and the ground truth maps, using two 1D filters (i.e.,  $3 \times 1$  and  $1 \times 3$  convolutional layers) rather than of  $3 \times 3$  in size, which can improve the network optimization performance while avoiding a large computational effort [35]. Coarse feature maps of the input and stacked outputs are fused by using residual module propagation with identity mapping branches to facilitate training, and iterations are conducted to optimize coarse saliency map accuracy. The boundary refinement map under the sigmoid function mapping is used as the final output of the network, as shown in Figure 8.

The network structure consists of three parts: encoder, decoder, and bridge connection.

The coding section consists of four stages with two 1D filters and a maximum pooling layer for down-sampling and reduced computational effort. The order of the built convolutional layers is  $3 \times 1$  in front and  $1 \times 3$  convolution in the back. Only one ReLU layer is added after the former, and a BN layer and a ReLU layer are placed after the convolutional layer

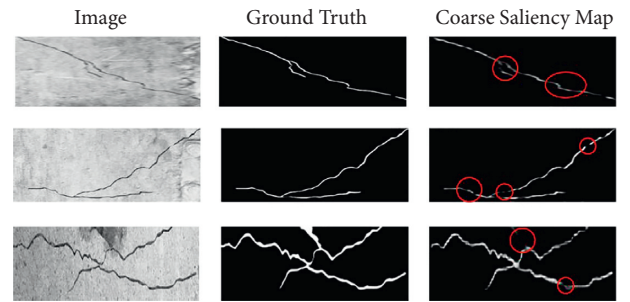


FIGURE 7: Coarse saliency map of crack.

of the latter [36]. This design allows the network to be built to a deeper level with less degradation in performance and mitigates to a certain extent the effect of gradient diffusion on network training, balancing network optimization performance and computational efficiency.

The decoding part is composed of a bilinear interpolation unit for up-sampling to match the feature dimensions and two 1D filters identical to the encoding part. The 1D filter is built in the reverse order of the coding part. This part also consists of four stages, and the codec pattern is reflected in the decoding part, where the  $1 \times 3$  convolution in each stage is cascaded with the  $3 \times 1$  convolution in the corresponding stage of the coding part.

The bridge connection part contains a Conv layer, a BN layer, and a ReLU layer. The convolutional layer in the structure has 64 filters and a convolutional size of  $3 \times 3$ .

**2.3. Hybrid Loss Function.** The training loss function in this paper is defined as the sum of the outputs of all saliency feature mappings:

$$L = \sum_{k=1}^k \alpha_k I^{(k)}, \quad (2)$$

where  $I^{(k)}$  is the loss of the  $k$ th lateral output and  $\alpha_k$  is the weight of each loss.  $k$  is taken as 8, indicating the presence of 8 outputs of the supervised sleeper crack detection network, 7 of which are from CEDNet and the rest from CRRNet. A

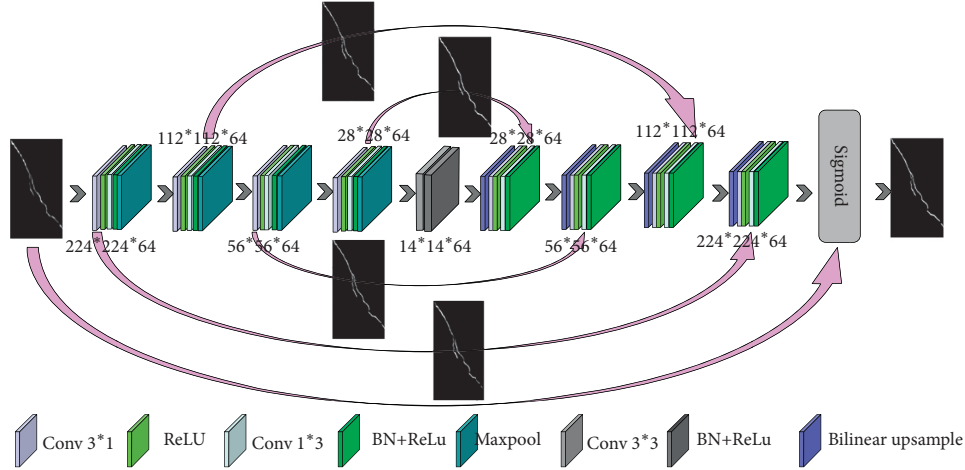


FIGURE 8: CRRNet.

hybrid loss function  $l^{(k)}$  that mixes three losses of BCE, SSIM, and IOU is used to obtain a high-quality detection object with complete information:

$$l^{(k)} = l_{\text{bce}}^{(k)} + l_{\text{ssim}}^{(k)} + l_{\text{iou}}^{(k)}, \quad (3)$$

where  $l_{\text{bce}}^{(k)}$ ,  $l_{\text{ssim}}^{(k)}$ , and  $l_{\text{iou}}^{(k)}$  denote the BCE [37], SSIM [38], and IOU losses [39], respectively.

BCE is used as a loss function in this network to supervise the training accuracy of object detection from the pixel level, which can be performed pixel by pixel. The pixel points of foreground and background pixel points are considered equally important and ignore the labeling of the neighboring regions. Accordingly, all pixel points can be converged. BCE is mainly applied to binary classification and segmentation tasks. The definitions are as follows:

$$l_{\text{bce}} = - \sum_{(r,c)} [G(r,c) \log(S(r,c)) + (1 - G(r,c)) \cdot \log(1 - S(r,c))], \quad (4)$$

where  $G(r,c) \in \{0, 1\}$  is the ground truth label of the pixel  $(r,c)$  and  $S(r,c)$  is the predicted probability of the saliency object.

SSIM is used as a loss function for supervised object detection from the local domain level to evaluate the image quality. This loss function assigns a higher weight to the boundary making the loss near the boundary higher, that is, focusing on the attention to the foreground and background boundaries. Progressively more important background losses come into play as the prediction of background pixel points approaches the ground truth, making the boundaries of cracks in the background prediction clearer. SSIM captures structural information in the image; therefore, it is integrated into the blend function to learn the structural information of the saliency object. The definition is as follows:

$$l_{\text{ssim}} = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (5)$$

where  $x \in \{x_j; j = 1, \dots, N^2\}$  and  $y \in \{y_j; j = 1, \dots, N^2\}$  are the pixel values of two corresponding patches cropped

from the predicted probability map  $S$  and the binary ground truth mask  $G$ , respectively,  $\mu_x, \mu_y$  and  $\sigma_x, \sigma_y$  are the mean and standard deviations of  $x$  and  $y$ , respectively, and  $\sigma_{xy}$  is their covariance.  $C_1 = 0.01^2$  and  $C_2 = 0.03^2$  to avoid dividing by zero.

IOU is originally used to calculate the similarity between two sets and extended to a standard method for evaluating the effectiveness of object detection and segmentation. After the foreground loss is reduced to zero combined with the three loss functions, the BCE can be used to maintain all pixel point gradients and make the IOU focus more on the foreground as the prediction confidence of the foreground network gradually increases. At the feature map level, the following formula is used to oversee the training of object detection and ensure its differentiability in the training loss function.

$$l_{\text{iou}} = 1 - \frac{\sum_{r=1}^H \sum_{c=1}^W S(r,c)G(r,c)}{\sum_{r=1}^H \sum_{c=1}^W [S(r,c) + G(r,c) + S(r,c)G(r,c)]}, \quad (6)$$

where  $G(r,c) \in \{0, 1\}$  is the ground truth label of the pixel  $(r,c)$  and  $S(r,c)$  is the predicted probability of the saliency object.

### 3. Experiment and Results

**3.1. Dataset.** The image acquisition device used in the paper is mainly composed of industrial high-speed line matrix camera and camera lens used in accordance with the field design requirements. As shown in Figure 9, the image acquisition system consists of an industrial computer and the LQ-H3X module, where the LQ-H3X module mainly consists of a laser light source and a line array camera. The main parameters of the LQ-H3X module are shown in Table 1.

**3.2. Experimental Setup.** The model in this paper runs under a Win10 operating system, with dual CPU Intel Xeon Silver 4214 2.2 GHz and NVIDIA RTX 2080Ti 11 GB graphics card. The three networks of object localization, coarse saliency feature extraction, and boundary refinement are built and run under the integrated development environment of PyTorch framework and PyCharm.



FIGURE 9: Image acquisition. (a) Special rail inspection vehicle. (b) Picture of image acquisition in high-speed railway line.

TABLE 1: LQ-H3X module parameters.

Characteristics	Parameters
Camera resolution	2048/4096 pixels
Scanning frequency	2000 kHz (CL)
Laser power	15 W/25 W
Laser center wavelength	808/915 nm

3.3. *Hyperparameter Configuration.* For the saliency detection part, several parameters with deeper influence, such as initial learning rate, batch size, and epochs, are adjusted during model training. The initial learning rate is closely related to the update of the weight parameters. If it is extremely large, the loss value increases, and the network model is infinitely divergent. If it is extremely small, the loss value decreases extremely slowly, and the parameters are updated extremely slowly. Choosing minibatch stochastic gradient descent and appropriate epochs can improve the running speed of neural network, and let the model converge properly. The actual situation with different combinations of important parameters is compared through several experiments to improve the model training speed, and the results are shown in Table 2.

Initially, with the batch size and epochs unchanged, the loss value decreases faster and faster with the downward adjustment of  $lr$ . On the basis of determining the  $lr$  of 0.001, the batch size of 4 is selected first in accordance with the performance of the device graphics card and GPU memory size. The epochs are chosen to be adjusted downward from 300 to 100 for the case that the rail crack dataset does not have data diversity. The parameter combination of the lowest loss of 0.046 is established. In consideration of improving the running speed of the neural network, the epochs are increased from 200 to 300 to achieve the same accuracy when the batch size was adjusted to 5. The loss value does not drop as fast as the former in the whole process.

In summary, the optimal combination of parameters selected for the crack recognition module in this paper is as

TABLE 2: Hyperparameter configuration.

Initial learning rate	Batch size	Epochs	Loss
0.001	4	200	0.046
0.001	4	100	0.054
0.001	4	300	0.050
0.001	5	200	0.052
0.001	5	300	0.049
0.002	4	200	0.053
0.002	5	300	0.051
0.005	4	200	0.057
0.005	5	300	0.055

TABLE 3: Combination of parameters.

Parameters	Value
Input size	$224 \times 224$
Initial learning rate	0.001
Batch size	4
Epochs	200

follows: initial learning rate, batch size, and epochs are set to 0.001, 4, and 200, respectively, and the results are shown in Table 3.

3.4. *Evaluation Metric.* The selected evaluation metrics include  $F$ -measure, mAP,  $F$ -weighted [40], MAE [41], and AUC [42]. The  $F$ -measure is a comprehensive index for the evaluation of the final obtained crack detection results. mAP is used as the average accuracy rate to measure the recognition accuracy, with larger values indicating higher accuracy rates.  $F$ -weighted is calculated from the corresponding  $PR$  value. The weight of the  $PR$  value is the percentage of samples in the total number of samples. The larger the value, the stronger the network performance. MAE is used to measure the error of the test results. The AUC value indicates the high or low performance of the network in classifying the crack and rail background. The closer to 1, the better the network classification.

Its calculation formula is as follows:

$$F_\lambda = \frac{(1 + \lambda^2)P * R}{\lambda^2 * P + R}, \quad (7)$$

$$P = \frac{T_{P_{-1}}}{T_{P_{-1}} + F_{P_{-1}}} * \omega_{-1} + \frac{T_{P_0}}{T_{P_0} + F_{P_0}} * \omega_0 + \frac{T_{P_1}}{T_{P_1} + F_{P_1}} * \omega_1, \quad (8)$$

$$MAE = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |\bar{S}(x, y) - \bar{G}(x, y)|,$$

where  $P$  denotes the precision,  $R$  denotes the recall, and  $\lambda^2$  is 0.3, similar to those in reference [40];  $\omega_{-1}$ ,  $\omega_0$ , and  $\omega_1$  are the weight ratios of each precision. After the recall is calculated, the  $F$ -weighted is obtained from Equation (7).  $W$  and  $H$  are used to represent the length and width of the input sleeper crack image to be processed.

**3.5. Hybrid Loss Function.** This work compares and verifies the performance of the proposed hybrid loss function  $l$  with single and multiple forms of loss function combined with the network model. As shown in Figure 10, the saliency map predicted by the proposed algorithm is the closest to the ground truth. The integrity of the cracked part of the region with the clarity of the boundary is shown to be the best situation compared with the others.

The quantitative analysis is shown in Table 1. After the comparison experiments for individual loss functions, the more effective  $l_{bce}$  and  $l_{iou}$  are then selected for the combined analysis. Table 4 shows that the network performance can be optimized only when all three loss functions are simultaneously used. Particularly, our method improves  $F_{weighted}$  by 68.4%, 74.8%, 84.1%, and 99.0% on  $l_{bce} + l_{iou}$ ,  $l_{bce}$ ,  $l_{iou}$ , and  $l_{ssim}$ , respectively.

**3.6. Object Detection.** In this experiment, for the comparison of YOLOv3, YOLOv4, and YOLOv5, we conduct the corresponding experiments. The settings of our experimental parameters are shown in Table 5. The initial parameter values for input size, initial learning rate, class, batch size, and epochs for the training of rail crack images are provided.

On the basis of this experimental condition, tests are performed for Tiny YOLOv3, YOLOv3, YOLOv4, and YOLOv5. The model accuracy is verified in terms of the three metrics: precision, recall, and MAP, and the model speed is verified in terms of frames per second (FPS), as shown in Table 6.

YOLOv3 has a higher recognition accuracy than Tiny YOLOv3 and a faster recognition speed than YOLOv4 and YOLOv5. The recognition accuracy can be optimized with the help of SE module and SPP module. In accordance with the experimental results, YOLOv3 can reach the same or even exceed the level of YOLOv4 and YOLOv5.

Therefore, a preliminary conclusion is that YOLOv3 is a more ideal target for optimization. This conclusion can be verified in the final optimized test results.

The prediction frame when the network locates cracks is more accurate compared with the original YOLOv3 by using the improved YOLOv3 network to complete the detection of cracks in the sleeper due to the added attention mechanism to improve the ability to capture the location of cracks. The detection effect is shown in Figure 11.

YOLOv3 and the proposed algorithm are used to detect cracks of the overall roadbed image and the segmented sleeper image by using gray projection method. The comparison of experimental results is shown in Table 7. The comparison of the two inputs of the overall roadbed and sleeper areas shows that the mAP of crack detection is improved by 35.4% and 38.8% on YOLOv3 and improved YOLOv3 after rail sleeper area extraction, respectively, proving the necessity of sleeper area extraction for crack detection. The data entered in the sleeper region column show that the improved YOLOv3 improves the mAP by 6.9% compared with the original network, proving the significant superiority of the present algorithm for sleeper crack detection.

**3.7. Feature Extraction.** With regard to the sleeper crack dataset constructed in this work, the results of sleeper crack saliency detection obtained using the method of this work are compared with those of several other network models. The models include BAS [43], R2Net [44], SOD100k [45], EDR [46], PFA [47], HED [34], and POOLNet [48]. Figure 12 shows that the proposed algorithm has a good detection of cracks in a variety of situations, including low contrast (1st, 4th, and 6th columns), small target (4th and 6th columns), and complex background (2nd, 3rd, 5th, and 7th columns).

The above evaluation metrics are applied to make a quantitative analysis of all network performance, as shown in Figures 13 and 14. In terms of AUC, the proposed algorithm improves by 6.0%, 0.2%, 1.2%, 2.8%, 3.8%, 10.4%, 15.5%, and 50.9% compared with CEDNet, EDR, BAS, POOLNet, R2Net, PFA, SOD 100 k, and HED, respectively. This result indicates that the proposed algorithm has better classification prediction performance. The MAE value of this work is 0.015, verifying that the algorithm has a small error and high accuracy rate compared with the other networks. The closer the curve composed of precision and recall to the upper-right corner, the better the network classification, and the larger the area enclosed by the  $F$  curve and the horizontal axis, the stronger the performance of the network.

The proposed algorithm has better crack integrity and clarity than other algorithms and depends on the form of cascade network used herein. A more complete crack feature can be obtained after cascading the residual networks of codec modes (i.e., CEDNet and CRRNet). In comparison with EDR, the pooling operation after the input convolutional layer is removed in the feature extraction stage to improve the image resolution in this work, and Conv5 and Conv6 are designed to restore the network receptive field. The crack information obtained in this stage is more detailed. By contrast with BAS, a 1D filter is used in

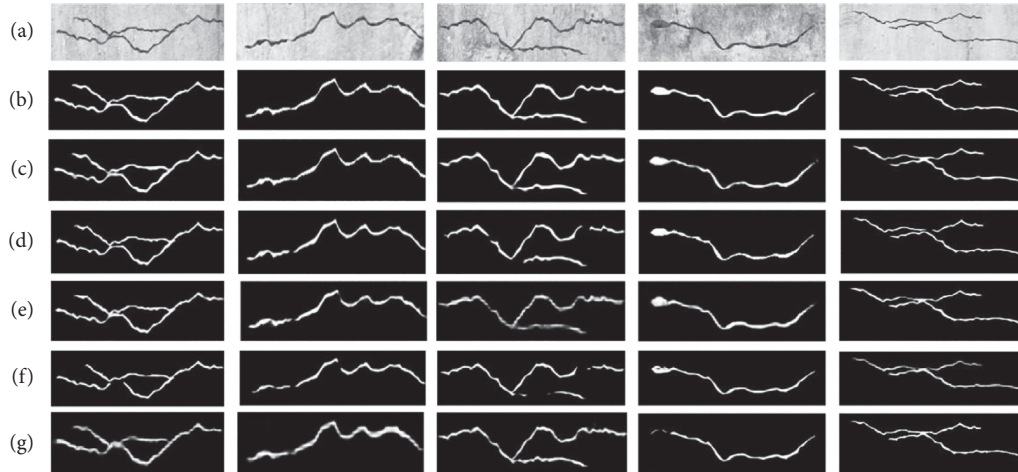


FIGURE 10: Saliency maps under different loss functions. (a) Image. (b) Ground truth. (c)  $l$ . (d)  $l_{bce} + l_{iou}$ . (e)  $l_{bce}$ . (f)  $l_{iou}$ . (g)  $l_{ssim}$ .

TABLE 4: Performance comparison of different loss functions.

Evaluation metrics	F-weighted $\uparrow$	MAE $\downarrow$
CEDNet + CRRNet + $l$	<b>0.805</b>	<b>0.015</b>
CEDNet + CRRNet + $l_{bce} + l_{iou}$	0.254	0.038
CEDNet + CRRNet + $l_{bce}$	0.203	0.039
CEDNet + CRRNet + $l_{iou}$	0.128	0.040
CEDNet + CRRNet + $l_{ssim}$	0.008	0.043

TABLE 5: Setting of initial parameter values.

Parameters	Value
Input size	$128 \times 608$
Initial learning rate	0.1
Class	1
Batch size	6
Epochs	200

TABLE 6: Comparisons of experimental results.

Models	Precision	Recall	MAP	FPS
Tiny YOLOv3	0.361	0.452	0.392	146.35
YOLOv3	0.794	0.877	0.856	81.7
YOLOv4	0.866	0.924	0.884	26.23
YOLOv5x	0.932	0.905	0.911	32.52
<b>Ours</b>	<b>0.963</b>	<b>0.912</b>	<b>0.915</b>	<b>76.6</b>

the optimization part to balance the refinement performance and computational efficiency. In FPN-based U-Net structures, such as POOLNet and R2Net, the high-level semantic features are continuously diluted because of their

structural limitations when fusing with low-level image features, and the different receptive fields in each layer of the network lead to the loss of local information in the crack saliency map.

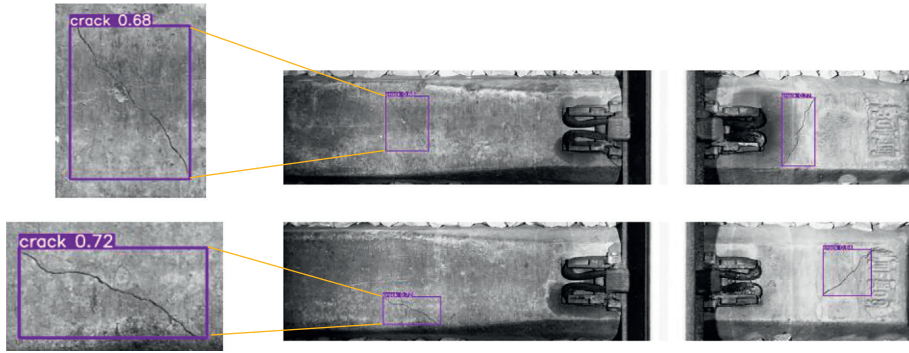


FIGURE 11: Sleeper crack location.

TABLE 7: Comparison of crack detection results.

Algorithm model	Enter the overall roadbed area			Input sleeper area		
	Precision	Recall	mAP	Precision	Recall	mAP
YOLOv3	0.444	0.736	0.632	0.794	0.877	0.856
Ours	<b>0.469</b>	<b>0.792</b>	<b>0.659</b>	<b>0.963</b>	<b>0.912</b>	<b>0.915</b>

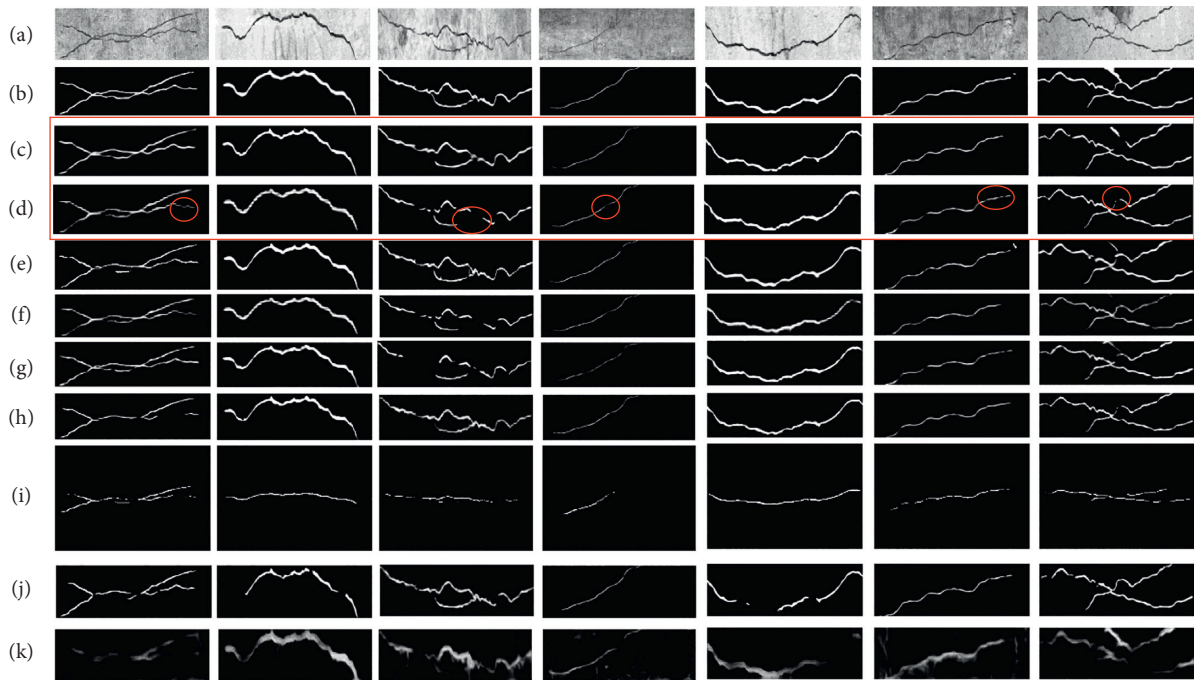


FIGURE 12: Comparison of saliency maps. (a) Image. (b) Ground truth. (c) Ours. (d) CEDNet. (e) EDR. (f) BAS. (g) POOLNet. (h) R2Net. (i) PFA. (j) SOD 100k. (k) HED.

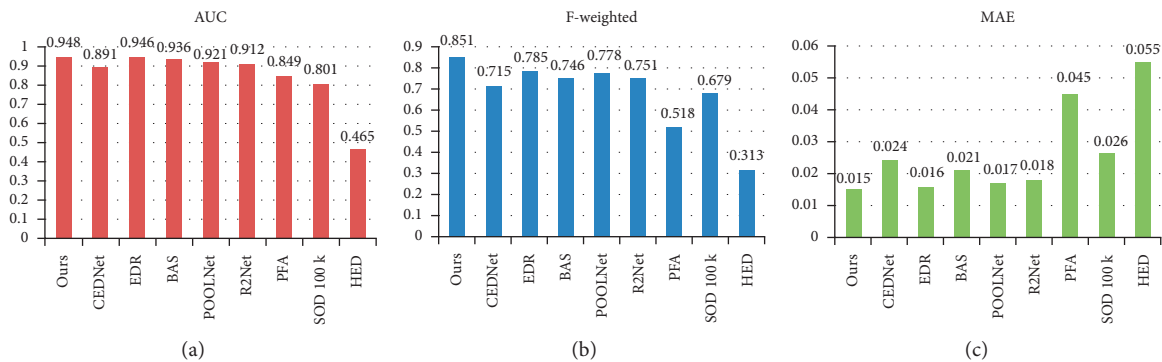


FIGURE 13: Performance comparison of each algorithm. (a) AUC. (b)  $F$ -weighted. (c) MAE.

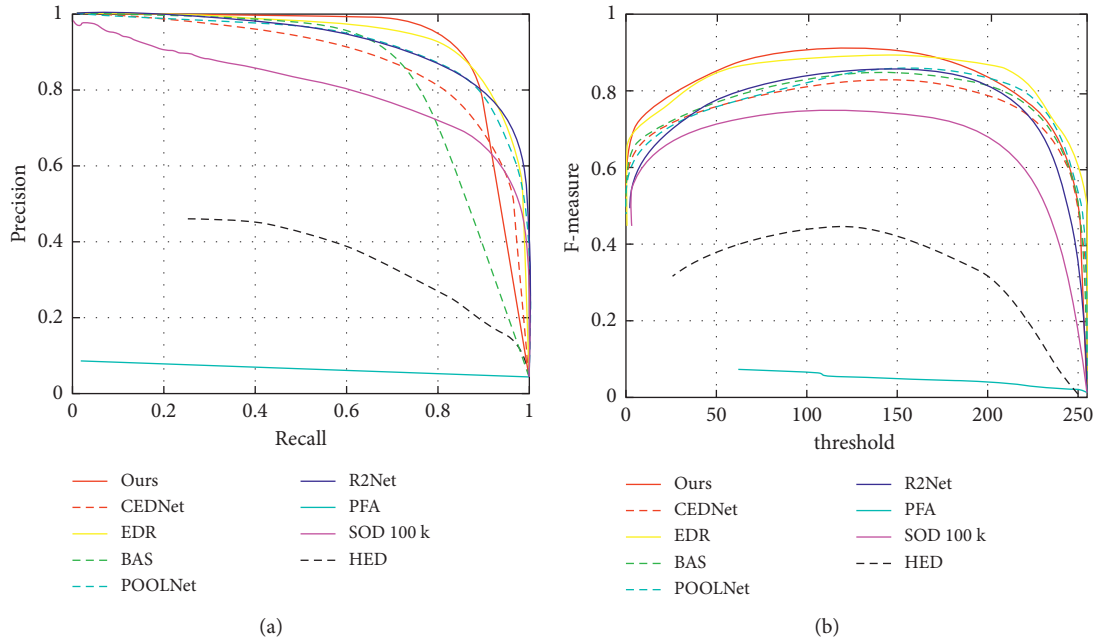


FIGURE 14: Performance comparison diagram of each algorithm. (a) Precision-recall curves. (b)  $F$ -measure curves.

#### 4. Conclusion and Expectations

We propose a method for detecting cracks in rail sleepers based on DCNN to address the lack of accuracy in crack detection in crack recognition. The CNN used consists of a modified YOLOv3 network for localization and CEDNet and CRRNet for extracting and optimizing the rail sleeper crack features, respectively. In locating the rail sleeper crack region, the crack on the concrete rail sleeper has some similarity with the ballast edge in the captured images due to the lighting and other causes. However, a grayscale difference can be observed between the rail sleeper and the ballast. Hence, the rail sleeper area is first segmented for the next step. The attention module SE is added at the end of the original YOLOv3 network to extract the cracked areas, thereby improving the accuracy of the rail sleeper crack detection while preserving the network computation speed. CEDNet is constructed to extract more crack information by fusing the high- and low-level features of crack images. The crack boundary refinement network CRRNet is added to optimize the cracks, and the stacked output of the crack coarse saliency feature map and the network can be optimized by learning the residuals from the ground truth. A cascade approach is adopted for the above two networks to obtain a crack saliency map with more complete boundary and region information. The conclusions of this work are as follows:

- (1) A new crack detection method is designed. A cascade network combining CEDNet and CRRNet is used to improve the integrity of crack detection. Its  $F$ -weighted is 0.831, MAE is 0.0157, and AUC is 0.9453.
- (2) An improved YOLOv3 network is proposed to localize the cracks, and the attention mechanism SE

module is added at the end of the backbone network. The mAP is improved by 6.9% compared with that of YOLOv3.

- (3) The optimization effects of loss functions BCE, IOU, and SSIM on crack recognition are superimposed to propose a new hybrid loss function for the crack recognition. Particularly, our method improves  $F_{\text{weighted}}$  by 68.4%, 74.8%, 84.1%, and 99.0% on  $l_{\text{bce}} + l_{\text{iou}}$ ,  $l_{\text{bce}}$ ,  $l_{\text{iou}}$ , and  $l_{\text{ssim}}$ , respectively.
- (4) A comprehensive evaluation of the proposed methodology is conducted. Our method has strong robustness and high level of crack detection efficiency compared with the seven state-of-the-art methods.

The proposed crack recognition module consists of two parts. In the optimization stage, we perform the crack boundary refinement process directly on the basis of the first output. Compared with end-to-end learning, this approach requiring secondary adjustment of model parameters increases the time cost and requires more manual processing. Therefore, if the optimization part can be encapsulated into a plug-and-play module, it will greatly improve the efficiency of model operation, which is the next optimization intention of this paper. This paper effectively improves the accuracy of the identification of cracks in the rail sleeper but does not measure the geometric parameters. How to calculate the actual size of the cracks on the basis of existing data is a direction for our future efforts, which is extremely helpful for practical engineering applications.

#### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 51975347 and 51907117).

## References

- [1] W. Zhu, G. Fan, X. Meng et al., "Ultrasound SAFT imaging for HSR ballastless track using the multilayer sound velocity model," *Insight*, vol. 63, no. 4, pp. 199–208, 2021.
- [2] L. Peng, S. Zheng, P. Li, Y. Wang, and Q. Zhong, "A comprehensive detection system for track Geometry using fused vision and inertia," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.
- [3] A. Mohan and S. Poobal, "Crack detection using image processing: a critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018.
- [4] P. Kannadaguli and V. Bhat, "Microwave imaging based automatic crack detection system using machine learning for columns," in *Proceedings of the IEEE 9th International Conference on Communication Systems and network technologies (CSNT)*, vol. 5–8, April 2020.
- [5] Y. A. Hsieh and Y. J. Tsai, "Machine learning for crack detection review and model performance comparison," *Journal of Computing in Civil Engineering*, vol. 34, no. 5, 2020.
- [6] M. Nie and C. Wang, "Pavement crack detection based on yolo v3," in *Proceedings of the 2nd International Conference on Safety Produce Informatization (IICSPI)*, pp. 327–330, Chongqing, China, November 2019.
- [7] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deepcrack: learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2018.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [9] W. Liu, A. Dragomir, and E. Dumitru, "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, vol. v 9905 LNCS, pp. 21–37, Amsterdam, The Netherlands, September 2016.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only Look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–786, IEEE, Las Vegas, NV, USA, June 2016.
- [11] Y. Cha, W. Choi, and O. Buyukozturk, "Deep learning based crack damage detection using CNNs," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 316–378, 2017.
- [12] V. Mandal, L. Uong, and Y. Adu-Gysmfi, "Automated road crack detection using deep convolutional neural networks," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, pp. 5212–5215, IEEE, Seattle, USA, December 2018.
- [13] W. Li, Z. Shen, and P. Li, "Crack detection of track plate based on YOLO," in *Proceedings of the 12th International Symposium on Computational Intelligence and Design*, pp. 15–18, Hangzhou, China, December 2019.
- [14] Y. Bao, K. Song, J. Liu et al., "Triplet-graph reasoning network for few-shot metal generic surface defect segmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, Article ID 5011111, 2021.
- [15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.
- [16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Cham, October 2015.
- [17] V. B. Adrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder–decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [18] J. Cheng, W. Xiong, W. Chen, and Y. Gu, "Pixel-level crack detection using," in *Proceedings of the TENCON 2018-2018 IEEE Region10 Conference*, pp. 462–466, Jeju Island, Korea, October 2018.
- [19] M. M. M. Islam and J.-M. Kim, "Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder-decoder network," *Sensors*, vol. 19, no. 19, p. 4251, 2019.
- [20] C. V. Dung, "Autonomous concrete crack detection using deep fully CNN," *Automation in Construction*, vol. 99, p. 5258, 2019.
- [21] U. Escalona, F. Arce, E. Zamora, and H. Sossa, "Fully convolutional networks for automatic pavement crack segmentation," *Computación Y Sistemas*, vol. 23, no. 2, pp. 451–460, 2019.
- [22] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>.
- [23] S. Tian, "Grayscale projection image stabilization algorithm based on gray bit-plane for moving object," in *Proceedings of the Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pp. 610–613, China, July 2018.
- [24] J. Hu, S. Li, and G. Sun, "Squeeze-and-Excitation networks," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Jeju Island, South Korea, June 2018.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [26] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks," *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018*, vol. 11070, pp. 421–429, 2018.
- [27] M. Xu, Z. Zhang, H. Hu et al., "End-to-End semi-supervised object detection with soft teacher," 2021, <https://arxiv.org/pdf/2106.09018.pdf>.
- [28] X. Dai, Y. Chen, B. Xiao, and D. Chen, "Dynamic head: unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, June 2021.
- [29] T. Liang, X. Chu, Y. Wang, Z. Tang, and W. Chu, "CBNetV2: a composite backbone network architecture for object detection," 2021, <https://arxiv.org/abs/2107.00420>.
- [30] 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*, pp. 770–778, Honolulu, HI, USA, June 2016.
- [31] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Lille, France, July 2015.
- [32] R. H. R. Hahnloser, H. S. Seung, and J. J. Slotine, “Permitted and forbidden sets in symmetric threshold-linear networks,” *Advances in Neural Information Processing Systems*, vol. 15, pp. 217–223, 2001.
- [33] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” 2015, <https://arxiv.org/abs/1511.07122>.
- [34] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1395–1403, Santiago, Chile, December 2015.
- [35] C. Peng, X. Zhang, and G. Yu, *Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network*, pp. 4353–4361, CVPR, Honolulu, HI, USA, 2017.
- [36] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” *In AISTATS*, pp. 315–323, 2011.
- [37] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.
- [38] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *Proceedings of the IEEE Asilomar Conference on Signals*, vol. 2, pp. 1398–1402, Pacific Grove, CA, USA, December 2003.
- [39] G. Mattyus, W. Luo, and R. Urtasun, “DeepRoadMapper: extracting road topology from aerial images,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3458–3466, Venice, Italy, October 2017.
- [40] P. Tao, H. Yi, C. Wei, L. Ge, and L. Xu, “A method based on weighted F-score and SVM for feature selection,” in *Proceedings of the 25th Chinese Control and Decision Conference (CCDC)*, May 2013.
- [41] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, “Salient object detection: a benchmark,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.
- [42] F. Liu, T. Shen, and S. Lou, “Deep network saliency detection based on global model and local optimization,” *Acta Optica Sinica*, vol. 37, no. 12, Article ID 1215005, 2017.
- [43] X. Qin, Z. Zhang, C. Huang, and C. Gao, “BASNet: boundary-aware salient object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7471–7481, Long Beach, CA, USA, June 2019.
- [44] M. Z. Alom, M. Hasan, C. Yakopcic, and T. M. Taha, “Nuclei Segmentation with Recurrent Residual CNNs Based U-Net (R2U-Net),” in *Proceedings of the NAECON 2018 - IEEE National Aerospace and Electronics Conference*, pp. 228–233, Dayton, OH, USA, 2018.
- [45] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: an in-depth survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, <https://arxiv.org/pdf/1904.09146>, 2021.
- [46] G. Song, K. Song, and Y. Yan, “EDRNet: encoder-decoder residual network for salient object detection of strip steel surface defects,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9709–9719, 2020.
- [47] T. Zhao and X. Wu, “Pyramid feature attention network for saliency detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3080–3089, Long Beach, CA, USA, June 2019.
- [48] J. Liu, Q. Hou, M. Cheng, and J. Feng, “A simple pooling-based design for real-time salient object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3912–3921, Long Beach, CA, USA, April 2019.