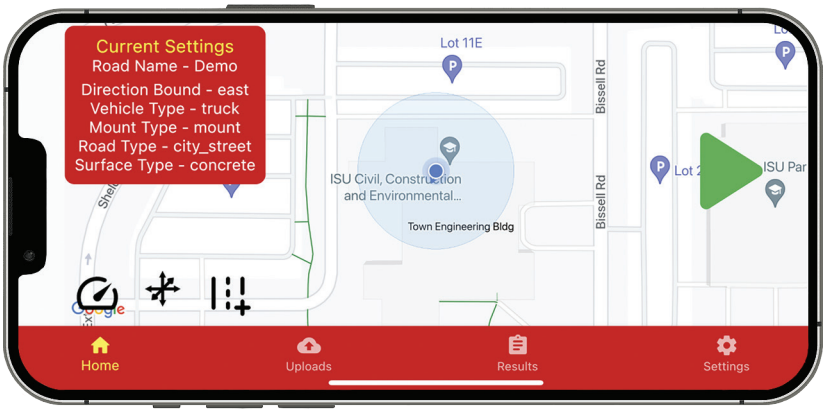
# Development of a Smartphone-Based Road Performance Data Collection Tool

## Final Report
## December 2023





IOWA STATE UNIVERSITY
Institute for Transportation



**Sponsored by**
Iowa Highway Research Board
(IHRB Project TR-777)
Iowa Department of Transportation
(InTrans Project 19-705)

## About the Program for Sustainable Pavement Engineering and Research

The overall goal of the Program for Sustainable Pavement Engineering and Research (PROSPER) is to advance research, education, and technology transfer in the area of sustainable highway and airport pavement infrastructure systems.

## About the Institute for Transportation

The mission of the Institute for Transportation (InTrans) at Iowa State University is to save lives and improve economic vitality through discovery, research innovation, outreach, and the implementation of bold ideas.

## Iowa State University Nondiscrimination Statement

Iowa State University does not discriminate on the basis of race, color, age, ethnicity, religion, national origin, pregnancy, sexual orientation, gender identity, genetic information, sex, marital status, disability, or status as a US veteran. Inquiries regarding nondiscrimination policies may be directed to the Office of Equal Opportunity, 3410 Beardshear Hall, 515 Morrill Road, Ames, Iowa 50011, telephone: 515-294-7612, hotline: 515-294-1222, email: eooffice@iastate.edu.

## Disclaimer Notice

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. The opinions, findings and conclusions expressed in this publication are those of the authors and not necessarily those of the sponsors.

The sponsors assume no liability for the contents or use of the information contained in this document. This report does not constitute a standard, specification, or regulation.

The sponsors do not endorse products or manufacturers. Trademarks or manufacturers' names appear in this report only because they are considered essential to the objective of the document.

## Iowa DOT Statements

Federal and state laws prohibit employment and/or public accommodation discrimination on the basis of age, color, creed, disability, gender identity, national origin, pregnancy, race, religion, sex, sexual orientation or veteran's status. If you believe you have been discriminated against, please contact the Iowa Civil Rights Commission at 800-457-4416 or the Iowa Department of Transportation affirmative action officer. If you need accommodations because of a disability to access the Iowa Department of Transportation's services, contact the agency's affirmative action officer at 800-262-0003.

The preparation of this report was financed in part through funds provided by the Iowa Department of Transportation through its "Second Revised Agreement for the Management of Research Conducted by Iowa State University for the Iowa Department of Transportation" and its amendments.

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the Iowa Department of Transportation.

## Technical Report Documentation Page

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| IHRB Project TR-777 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| Development of a Smartphone-Based Road Performance Data Collection Tool | December 2023 |
| | **6. Performing Organization Code** |

| 7. Author(s) | 8. Performing Organization Report No. |
|---|---|
| Bo Yang (orcid.org/0000-0002-7774-5233), Yueyi Jiao (orcid.org/0000-0002-8264-5771), Hao Jiang (orcid.org/0000-0001-5936-996X), Venkata Ashish Gorthy (orcid.org/0000-0002-0151-3356), Md Abdullah All Sourav (0000-0003-3387-740X), Halil Ceylan (orcid.org/0000-0003-1133-0366), Wensheng Zhang (orcid.org/0000-0003-2155-5139), Sunghwan Kim (orcid.org/0000-0002-1239-2350), Chichang Lin (orcid.org/0009-0003-0315-0442), Shengwei Mao (orcid.org/0009-0004-3205-8570), Danny Waid (orcid.org/0000-0002-1652-2028), and Brian P. Moore (orcid.org/0000-0003-1746-009X) | InTrans Project 19-705 |

| 9. Performing Organization Name and Address | 10. Work Unit No. (TRAIS) |
|---|---|
| Institute for Transportation<br>Iowa State University<br>2711 South Loop Drive, Suite 4700<br>Ames, IA 50010-8664 | **11. Contract or Grant No.** |

| 12. Sponsoring Organization Name and Address | 13. Type of Report and Period Covered |
|---|---|
| Iowa Highway Research Board<br>Iowa Department of Transportation<br>800 Lincoln Way<br>Ames, IA 50010 | Final Report |
| | **14. Sponsoring Agency Code** |

**15. Supplementary Notes**

Visit https://intrans.iastate.edu for color pdfs of this and other research reports.


**16. Abstract**

Pavement roughness serves as a critical indicator of the overall ride quality of road surfaces. Elevated levels of pavement roughness can give rise to concerns regarding driving safety, fuel consumption, and exhaust gas emissions. The International Roughness Index (IRI) stands as the globally accepted standard for quantifying road surface roughness. Although one of the primary responsibilities of local public agencies (LPAs) involves monitoring and maintaining appropriate IRI levels for the local road system, existing techniques used to collect IRI data, such as using high-speed and walking profilometers, typically entail high annual costs for network-level inspection. Consequently, LPAs are in need of a cost-effective IRI data collection system that enables them to gather pavement performance data annually. Smartphones come equipped with an array of sensors, including multi-axis accelerometers and global positioning systems (GPS), that present an efficient and economical approach for collecting vehicle suspension data, specifically vertical acceleration. These data can then be harnessed to estimate pavement profiles and roughness. This study endeavored to develop a low-cost, smartphone-based, nonproprietary data collection system designed for use by LPAs to gather pavement roughness data on an annual basis. This adaptable system can be implemented on Android phones, iPhones, or custom-developed smart boxes that incorporate accelerometers and GPS units. All data gathered in this way can be seamlessly transferred to a cloud-based server and subsequently processed using a Python-based algorithm to compute the IRI. The accuracy of such a system was assessed using four distinct smartphones, one custom-developed smart box, and a Class 1 high-speed profilometer that was used to establish reference IRI values. The field data collection program encompassed 24 Story County sites selected to validate the accuracy of IRI measurements. The results demonstrate that the smartphone-based and smart box systems developed offer a dependable, low-cost, and user-friendly solution for LPAs to use in assessing local road roughness. A prototype smartphone application for detecting road surface distress through captured videos and images was also developed and evaluated.


| 17. Key Words | 18. Distribution Statement |
|---|---|
| distress detection—international roughness index—pavements—smart box—smartphone | No restrictions. |

| 19. Security Classification (of this report) | 20. Security Classification (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified. | Unclassified. | 98 | NA |

**Form DOT F 1700.7 (8-72)** **Reproduction of completed page authorized**

# DEVELOPMENT OF A SMARTPHONE-BASED ROAD PERFORMANCE DATA COLLECTION TOOL

**Final Report**
**December 2023**


**Principal Investigator**
Halil Ceylan, Director
Program for Sustainable Pavement Engineering and Research
Institute for Transportation, Iowa State University

**Co-Principal Investigators**
Wenshen Zhang, Associate Professor
Department of Computer Science, Iowa State University

Sunghwan Kim, Research Scientist
Institute for Transportation, Iowa State University

**Research Associates**
Bo Yang, Yueyi Jiao, Hao Jiang, Venkata Ashish Gorthy, and Md Abdullah All Sourav

**Authors**
Bo Yang, Yueyi Jiao, Hao Jiang, Venkata Ashish Gorthy, Md Abdullah All Sourav,
Halil Ceylan, Wensheng Zhang, Sunghwan Kim, Chichang Lin,Shengwei Mao,
Danny Waid, and Brian P. Moore



Sponsored by
Iowa Highway Research Board and
Iowa Department of Transportation
(IHRB Project TR-777)

Preparation of this report was financed in part
through funds provided by the Iowa Department of Transportation
through its Research Management Agreement with the
Institute for Transportation (InTrans Project 19-705)



A report from
**Institute for Transportation**
**Iowa State University**
2711 South Loop Drive, Suite 4700
Ames, IA 50010-8664
Phone: 515-294-8103 / Fax: 515-294-0467
https://intrans.iastate.edu

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## ACKNOWLEDGMENTS

The authors would like to thank the Iowa Department of Transportation (DOT), the Iowa Highway Research Board (IHRB), and the Iowa County Engineers Association Service Bureau (ICEASB) for sponsoring this research. The Iowa county engineers who served on the project's technical advisory committee (TAC), including Karen Albert (Montgomery County), Lee Bjerke (ICEASB), Zach Gunsolley (Iowa DOT), Todd Kinney (Clinton County), Mark Nahra (Woodbury County), John Riherd (Butler County), Brad Skinner (Appanoose County), and Jacob Thorius (Washington County), are also gratefully acknowledged for their guidance, support, and direction throughout the research. Special thanks are due to Steve De Vries (ICEASB) and Danny Waid (ICEASB), who developed the original concept of this study, and to Brian P. Moore (ICEASB) and other ICEASB staff for their guidance, support, and direction throughout the research.

The authors wish to extend their appreciation to the diverse group of individuals who assisted with this project. Special thanks go to Lars Forslöf, the creator of Roadroid, as well as the dedicated developers and contributors behind the various smartphone applications noted in this study. Additionally, the invaluable support from Dr. Kristen Cetin, Dr. Joseph Zambreno, and colleagues and students in Iowa State University's Program for Sustainable Pavement Engineering and Research (PROSPER) at the Institute for Transportation (InTrans); the Department of Civil, Construction, and Environmental Engineering; the Department of Computer Science; and the Department of Electrical and Computer Engineering has been instrumental throughout the duration of this project.

## EXECUTIVE SUMMARY

Pavement roughness serves as a critical indicator of overall road quality, with far-reaching implications with respect to fuel consumption and emissions. The International Roughness Index (IRI), the global standard for assessing such roughness, measures the vertical displacement of a standard vehicle over a specific distance. Local public agencies (LPAs) commonly collect IRI data to inform decisions on pavement upkeep. Since traditional methods of roughness measurement are either inefficient or expensive, LPAs increasingly seek affordable, efficient alternatives for annual data collection.

Modern smartphones, replete with advanced sensors such as multi-axis accelerometers and global positioning systems (GPS), offer a cost-effective means to collect vehicle suspension data, specifically vertical acceleration. Similarly, vehicle smart boxes integrated with accelerometers and GPS modules provide another economical avenue for data collection. Advanced algorithms can convert these raw data into estimates of pavement roughness.

This study aimed to develop an affordable, smartphone-based, open-source data collection system for use by LPAs. A prototype smartphone application (app), CyRoads, was developed for use with both Android and iOS phones, and a comparable system was developed for use with a smart box to collect vertical acceleration and GPS data. These tools were integrated with a Python-based IRI computation algorithm hosted on an Amazon Web Services (AWS) cloud server that can communicate with numerous devices to compute and transmit IRI values while interfacing with operational tools from the Iowa County Engineers Association Service Bureau (ICEASB).

To validate these prototypes, a comprehensive field study was conducted across 24 diverse road sites in Story County, Iowa. With a Class 1 high-speed profilometer providing reference IRI data, the study evaluated the systems' efficacy across multiple variables, including device type, vehicle category, and device placement. Speed corrections were also implemented to improve measurement reliability. The results confirm that both the CyRoads smartphone app and the smart box reliably measure IRI, with an $R^2$ value exceeding 0.77 and an average percent difference varying by less than 13% from the reference IRI.

This study also explored the potential for using a smartphone app to detect pavement cracks. A prototype app was developed that incorporates a deep-learning model to analyze real-time video and imagery captured by the smartphone. The app demonstrated a high precision rate of up to 17.04 frames per second and rapid image processing times of just 40 milliseconds.

While this study establishes the utility of smartphone-based and smart box systems in measuring road roughness, further research is required to maximize this technology's potential for comprehensive road infrastructure evaluation and maintenance.

# CHAPTER 1: INTRODUCTION

## Background and Motivation

Over the past few decades, the United States has witnessed the construction of millions of miles of pavement catering to the demands of society and the economy. Pavements, a vital component of transportation infrastructure, are expected to offer a smooth surface with satisfactory skid resistance throughout their entire lifespan. Pavement roughness, encompassing irregularities on a pavement surface, adversely affects not only ride quality but also fuel consumption. The International Roughness Index (IRI) has been established as the prevalent metric employed to quantify road roughness. As a pivotal indicator of roadway performance, the determination of pavement IRI necessitates annual collection by local public agencies (LPAs). IRI is incorporated into local pavement management information systems (PMIS) for evaluation, and this panoramic view of pavement conditions facilitates the formulation of an effective maintenance strategy.

Within this process, dependable and efficient data collection for assessing pavement performance plays a foundational role. In the United States, several methodologies and tools exist for assessing road conditions. For example, many LPAs conduct traditional windshield surveys and utilize portable devices like bump integrator trailers, rods, and beams (Sayers et al. 1986, Yang et al. 2017). In recent years, significant technological strides have resulted in an increasing number of LPAs gravitating toward automated data collection methods (McGhee 2004, Cafiso et al. 2017, Yang et al. 2017, Yang et al. 2022). For example, the high-speed profilometer (HSP), an automated road roughness surveying instrument, has been extensively employed for years to gauge pavement roughness (Tian et al. 2021, Choubane and McNamara 2001). Despite the practicality of using automated instruments in road roughness assessment, their cost remains a concern for LPAs operating on constrained budgets (McGhee 2004), prompting LPAs to actively seek an economical and convenient avenue for surveying their pavement networks.

In the present era, mobile phones have become seamlessly integrated into daily life. Modern smartphones, equipped with a plethora of built-in sensors such as three-axis accelerometers, high-resolution digital image sensors, global positioning systems (GPS), temperature sensors, gyroscopes, light intensity sensors, and magnetic field sensors, offer the potential for the cost-effective collection of vertical acceleration and horizontal displacement data, two data types crucial for estimating pavement roughness. Consequently, a smartphone-based data collection system could emerge as a cost-efficient and effective means for use by LPAs in gathering road performance data.

In recent years, significant efforts have been directed toward exploring the viability of smartphones for road performance evaluation, including several successful endeavors that have resulted in user-centric smartphone applications (apps) such as TotalPave, Carbin, Roadroid, RoadLab, RoadBounce, and RoadBump. Although these commercial apps represent an important development, they are often subject to limitations, inconveniences, and challenges that hinder users from obtaining the desired data. Additionally, these commercial apps might not align with the specific requirements of the Iowa County Engineers Association Service Bureau (ICEASB) in terms of project-level data collection. Independent design and development of a smartphone-

based data collection system therefore emerges as a feasible solution. The data collected by such an independent system could potentially be wirelessly uploaded to a cloud server and seamlessly integrated with other pavement management tools under development by the ICEASB.

## Research Objective and Scope

This research, conducted by Iowa State University (ISU) in collaboration with the ICEASB, aimed to develop a smartphone-based data collection tool. The research objectives were as follows:

1. Develop a smartphone-based pavement roughness measurement system (mobile application) for collecting roughness data at a frequency required for effective pavement management and maintenance planning.
2. Assess the potential of such a smartphone-based tool to detect and measure various types of road surface distress, including cracking, rutting, and faulting.
3. Create a standardized, nonproprietary data collection tool capable of gathering the roughness data needed for pavement management.
4. Test and calibrate the standardized, nonproprietary data collection tool for compatibility with various popular smartphone brands and types. Such a calibration process involves comparing the tool's measurements with those of a Class 1 profilometer that boasts high accuracy in measuring IRI and GPS values.

## Report Organization

This report is structured into nine chapters (Figure 1). Chapter 1 outlines the background, motivation, and objectives of the study. Chapter 2 provides a literature review of pavement roughness calculation using smartphones and smart boxes. Chapter 3 details the development of a roughness calculation algorithm for use with both the smartphone app and smart box developed in this study. Chapters 4 and 5 provide an in-depth description of the development of the smartphone app (called CyRoads) and smart box, respectively, focusing on IRI measurement. Chapter 6 encompasses the evaluation, calibration, and validation of the smartphone app and smart box through the collection of field data. Chapter 7 demonstrates the use of a smartphone app for pavement distress detection. Finally, Chapter 8 offers concluding remarks, while Chapter 9 proposes recommendations for future research. Appendices at the end of the report include operation manuals for both the CyRoads smartphone app and the smart box.

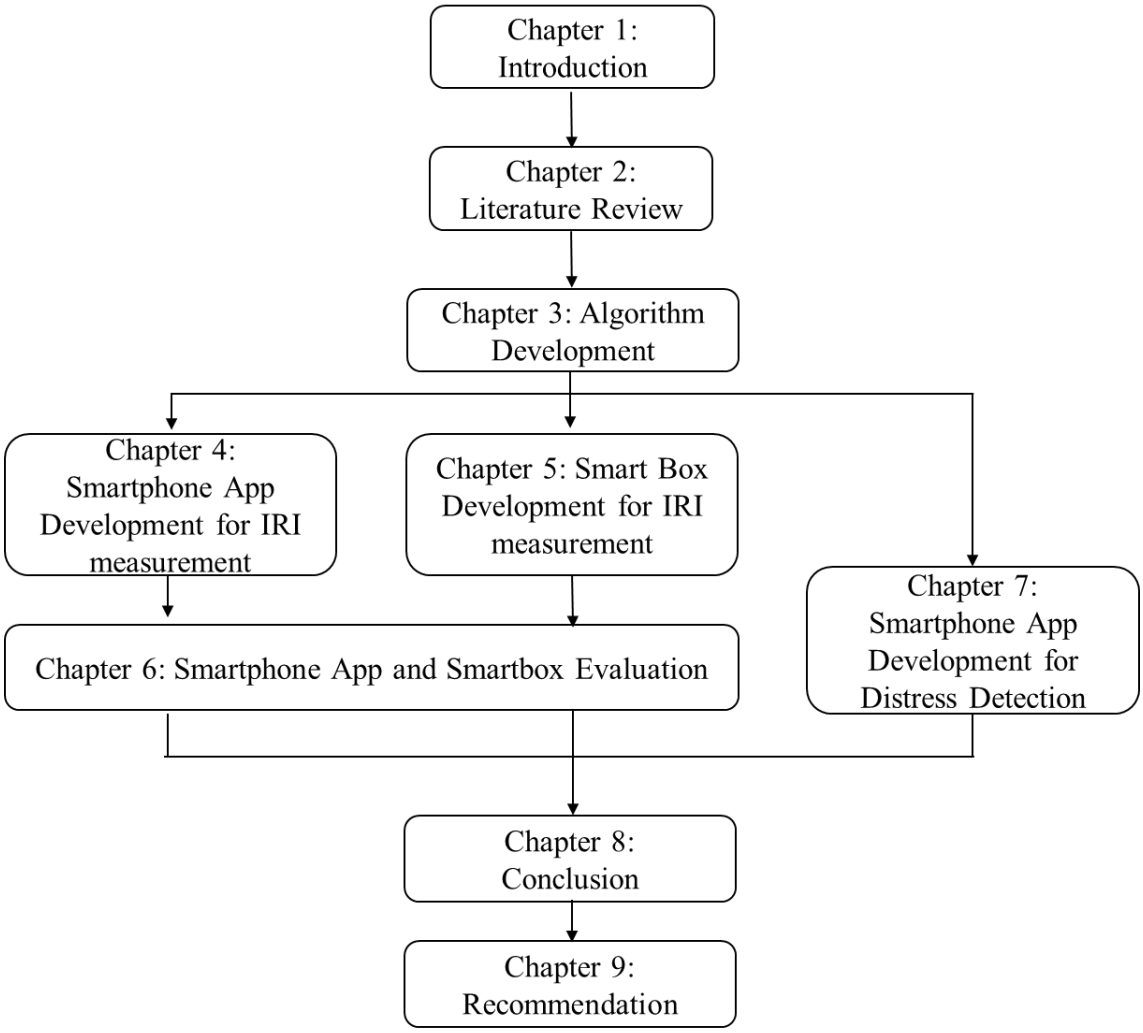



**Figure 1. Report organization flow chart**

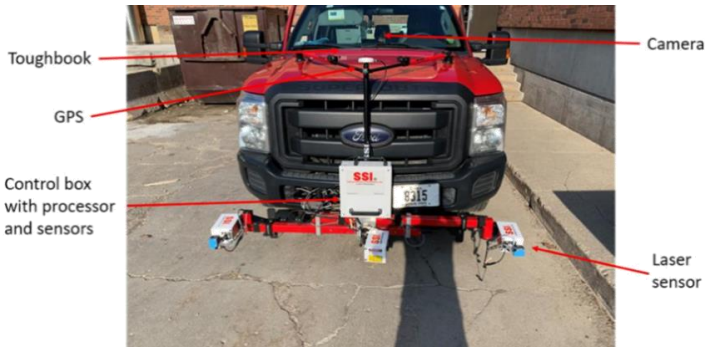
## CHAPTER 2: REVIEW OF PAVEMENT ROUGHNESS

### Overview of Pavement Roughness

According to ASTM E867, pavement roughness is characterized as the deviation of a road surface from a true planar geometry, and it is often cited as a critical indicator of driver satisfaction. A smoother pavement surface not only enhances driver comfort and safety but also contributes to reduced fuel consumption and lower emission levels (Brickman et al. 1972, Gillespie and Sayers 1981, Smith et al. 1997). To monitor and manage pavement roughness in terms of maintenance and rehabilitation objectives, many LPAs have formulated relevant guidelines and deployed specialized measurement instruments and techniques.

Over the past several decades, a variety of instruments have been employed by LPAs to gauge pavement roughness. In earlier years, simple straightedges were commonly used to detect surface deviations. Comprised of a steel frame and a central profiling wheel, along with a mechanical data recording system, these straightedges were limited to operations at slow speeds, typically under 5 mph, making them ill-suited for large-scale data collection (Smith and Ram 2016). By the 1990s, more advanced profilograph models emerged, such as the California and Rainhart profilographs. These devices featured multiple profiling wheels and systems capable of continuous data collection, but they had their own limitations, such as introducing bias in measurements due to wavelength adjustments (Smith and Ram 2016).

Inertial profilers were later developed to address some of the drawbacks associated with traditional profiling equipment and to provide high-quality road surface data. One of the most widely used inertial profilers is the HSP, depicted in Figure 2. Equipped with multiple accelerometers and lasers, the HSP measures the vertical acceleration and displacement of a vehicle's system. It also incorporates a distance measuring instrument (DMI) to record the driving distance. The device can be mounted on either the front or rear of a vehicle and is capable of collecting high-resolution data at speeds ranging from 20 to 50 mph. According to ASTM E950, the HSP qualifies as a Class 1 instrument due to its high-resolution capabilities, and it can also capture other pertinent data, such as GPS coordinates, road surface imagery, and environmental factors like temperature and moisture levels.




Tian et al. 2021

**Figure 2. Class 1 high-speed profilometer for IRI measurement**

Various metrics exist for describing pavement roughness, each with its own nuances and applications. The pavement serviceability rating (PSR) is a scale ranging from 0 to 5 that represents the ride quality of a road surface. In this system, a score of 0 denotes extremely poor conditions, while a score of 5 signifies excellent conditions. Another prevalent metric is the profile index (PI), calculated using profile traces and significant bumps recorded by profilograph devices.

To establish a universally accepted standard for quantifying pavement roughness, in the 1980s the World Bank introduced the concept of the IRI (Sayers et al. 1986). Over the years, a variety of models have been deployed to compute IRI values. By definition, the IRI is derived from the longitudinal profile of a road's traveled wheel path using the quarter-car (QC) model. It offers a ratio that quantifies the accumulated vertical suspension movement of a vehicle in relation to its covered distance.

Today, many LPAs consider the IRI as the go-to metric for representing pavement roughness, and inertial profilers are commonly used to collect annual IRI data that serve as a comprehensive rating for assessing the condition of pavement systems. Table 1 outlines three typical IRI thresholds: good (less than 95 in./mile), fair (between 95 and 170 in./mile), and poor (greater than 170 in./mile). In addition to the IRI, some LPAs employ other indices, such as the mean ride index (MRI) and half-car ride index (HRI), to gauge road smoothness. These latter metrics use the profiles of both the left and right wheel paths to provide a more holistic estimation of road roughness.

**Table 1. IRI thresholds for in-service pavements**

| | Good | Fair | Poor |
|---|---|---|---|
| IRI (in./mile) | < 95 | 95–170 | > 170 |

Source: Baladi et al. 2017

## Review of IRI Calculation Models Developed

### *Quarter-Car Suspension Model*

The QC suspension model, also known as the golden car (GC) model, is a mathematical framework designed to simulate a vehicle's suspension behavior using a single-tire system (Buttlar and Islam 2014). One of the primary benefits of this model is its focus on the response from just one tire, thereby simplifying dynamic effects and providing an easier means of understanding how road surface irregularities trigger vehicle vibrations (Sayers 1989).

This model's utility has been validated through its successful application in a variety of pavement surface measurement projects, and it has a proven track record of accurately correlating road surface conditions with vehicle dynamics (Sayers and Karamihas 1998, Xu et al. 2014, Misaghi et al. 2021).

The mathematical equations governing the QC model's behavior are presented in equations (1) and (2). These equations serve as the backbone for capturing complex interactions between the road surface and a vehicle's suspension system, enabling accurate roughness assessments.

$$m_1 x'' = k_s(y - x) + b(y' - x') - k_w(x - r) \quad (1)$$

$$m_2 y'' = -k_s(y - x) + b(y' - x') \quad (2)$$

As illustrated in Figure 3, the tire system in the QC model incorporates five key parameters: the sprung mass ($M_2$), the suspension damping rate (b), the suspension spring rate ($K_s$), the tire spring rate ($K_w$), and the unsprung mass ($M_1$). These parameters work in concert to capture the complex dynamics of a vehicle's interaction with the road surface.

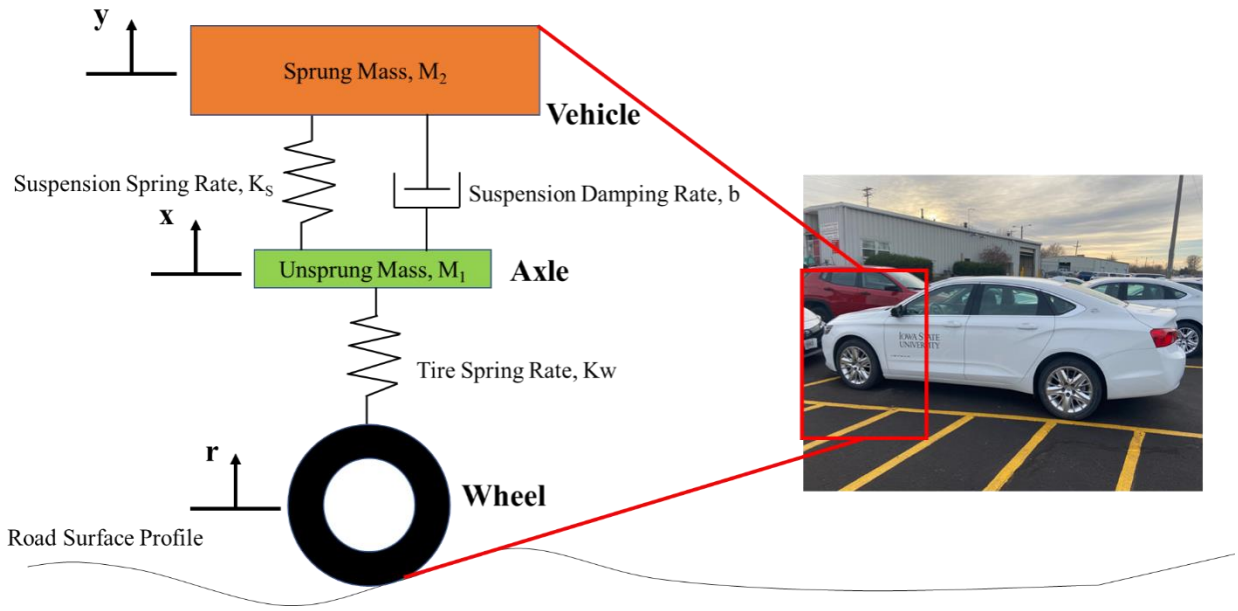



**Figure 3. Schematic illustration of the quarter-car model**

With these parameters specified, the model can be employed to calculate the sum of the simulated vertical suspension movements. This sum is then divided by the simulated travel length to yield the IRI per equation (3). The IRI is commonly expressed in inches per mile (in./mile), serving as a standardized measure for evaluating pavement roughness. Such a calculated IRI offers a quantitative means of assessing road quality, providing valuable data for road maintenance and improvement efforts.

$$IRI = \frac{1}{l}\int_0^{\frac{l}{v}} |z_y' - z_x'|\, dt \quad (3)$$

where l is the surveyed road length (miles), v is vehicle speed (mph), t is time (hours), and $z_x$ and $z_y$ are the vertical displacements (in.) of $m_1$ and $m_2$, respectively (Islam 2015).

*Double-Integration and Trapezoidal Rule*

The process for computing the IRI based on the QC model can be broadly divided into two primary stages: (1) acquiring the vertical displacement, also known as the profile, and (2) computing the IRI itself. Given that a smartphone or a smart box can be considered a component of the vehicle's sprung mass ($M_2$), the accelerations measured by these devices usually represent the vehicle body's vertical suspension movements.

To translate these acceleration readings into vertical displacement, a double-integration method is commonly employed. This involves taking raw acceleration data ($a_t$) and subjecting it to a series of mathematical transformations, as delineated in equations (4) and (5). Through double-integration, these raw acceleration measurements are first converted into estimates of vertical velocity ($v_t$) and subsequently into the pavement profile ($z_t$). Once these profiles are obtained, they can be used to calculate the IRI values, providing a quantitative measure of the road's roughness condition. This methodology leverages the capabilities of inertial sensors in smartphones or smart boxes, making it a practical and cost-effective approach for road quality assessment.

$$v_t = \int_0^t a_t dt \tag{4}$$

$$z_t = \int_0^t v_t dt \tag{5}$$

The trapezoidal rule is a numerical method used for approximating the definite integral of a function. Unlike methods that use rectangles to estimate the area under the curve, the trapezoidal rule divides the total area into smaller trapezoids to achieve a more accurate approximation. This approach is particularly useful when dealing with a data set that consists of discrete points, as is often the case with acceleration input.

In this context, the trapezoidal rule can be employed to perform the integration work necessary for deriving vertical displacement from acceleration data. This method approximates the area under the curve of the acceleration function by breaking it down into smaller trapezoidal segments, as illustrated in Figure 4. The sum of these trapezoidal areas gives an estimate of the integral, which in turn can be used to calculate vertical velocity and displacement.

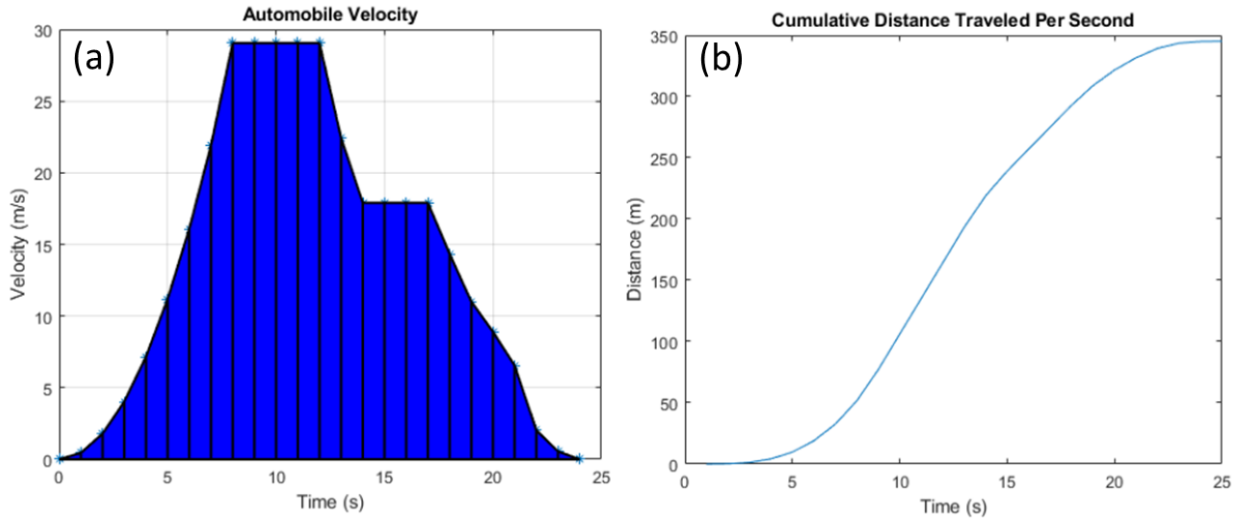



**Figure 4. Example of using the trapezoidal rule to obtain travel distance**

By leveraging the trapezoidal rule for integration, one can achieve a more accurate and reliable calculation of IRI values, thereby enhancing the quality of road condition assessments.

Once the profile data have been obtained, the QC model can be applied to calculate the IRI. ASTM E1926 provides a standardized Fortran-based program for computing IRI using these profile data as the input. As technology has advanced, a multitude of programming languages have emerged, including Java, C++, SQL, JavaScript, MATLAB, Python, and HTML, among others. Depending on the specific requirements of the calculation, the algorithms defined in ASTM E1926 can be translated into any of these modern programming languages to facilitate IRI computation.

To ensure uniformity and minimize bias in the expression of road roughness, ASTM E1926 also outlines standardized guidelines for IRI computation. These guidelines serve as a framework to ensure that the calculated IRI values are both reliable and comparable, regardless of the programming language or technology used. This standardization is crucial for maintaining the integrity and consistency of road quality assessments across different platforms and methodologies.

*Vehicle Suspension Parameters*

As previously noted, vehicle suspension parameters are critical elements in the QC model. However, these parameters can be significantly influenced by the type, make, and model of the vehicle in question. Initially, a set of generalized parameters known as golden car parameters (GCPs) was established in the 1970s to address this variability (Sayers 1995), while advancements in automotive technology in recent decades have led to marked changes in vehicle suspension responses. Distinct differences in the suspension responses of various trucks and cars also exist.

To refine the accuracy of IRI computations, many studies have adopted adjusted car parameters (ACPs) tailored based on factors such as vehicle type, make, and model. Because direct measurement of these parameters under both laboratory and field conditions can be challenging, ACPs are often derived through backcalculations. Table 2 compiles suspension parameters obtained from various studies, underscoring the wide range of variability dependent on vehicle type, make, and model.

**Table 2. Representative vehicle parameters from various studies**

| | Golden Car (Martinez 2014) | Ford Fiesta (Rath et al. 2014) | Honda Accord (Roy and Liu 2008) | Dodge Avenger (Islam 2015) | Honda CR-V (Islam 2015) |
|---|---|---|---|---|---|
| $M_1/M_2$ | 0.15 | 0.13 | 0.10 | 0.08 | 0.10 |
| $Ks/M_2$, $s^{-2}$ | 63 | 100 | 217 | 24 | 26 |
| $b/M_2$, $s^{-1}$ | 6 | 6 | 22 | 3 | 3 |
| $K_w/M_2$, $s^{-2}$ | 653 | 849 | 966 | 405 | 471 |

In the present study, ACPs were backcalculated for both a truck and a mid-size car used for field data collection. The calibration site for this purpose had produced well-documented historical IRI data provided by the Iowa Department of Transportation (DOT). Because the use of ACPs ensures that the computed IRI values are close to the reference values, implementing these ACPs helps level the playing field among various vehicle types, enhancing the consistency and reliability of IRI measurements. The use of ACPs also simplifies the process of making subsequent IRI corrections compared to the use of GCPs for that purpose. This approach thereby streamlines the IRI measurement process while maintaining high accuracy.

## Review of Application of Smartphone to Measure Pavement Roughness

### *Technical Features of Smartphones*

Conventional methods of determining pavement roughness rely on direct measurement using rod and level survey techniques that are very time-consuming. In recent decades, automated survey methods have been developed that use vehicles equipped with sensors capable of evaluating the pavement profile to more accurately calculate the roughness. This method is costly and not usually affordable by road agencies with limited budgets. Given such limitations, extensive research has been performed on calculating pavement roughness with improved accuracy matching the Class 1 information quality level (IQL), and discovering budget-friendly techniques is of widespread interest (Paterson and Scullion 1990).

In this context, the ubiquity of smartphones offers a compelling solution. These devices are equipped with a variety of high-precision sensors, including accelerometers and GPS, that can be used for civil engineering applications like determining pavement roughness, identifying road distress, and assessing traffic volume. Given that approximately 82.9% of the US population uses smartphones, this presents an opportunity for large-scale data collection at a fraction of the cost of more traditional methods.

Although smartphones are powerful devices, their computational and storage resources are not limitless, especially when shared by their operating system and other apps. This is where cloud computing comes in, which can extend the capacity of mobile devices by offloading computation and providing additional storage space. Such a cloud-based architecture would allow for crowdsourcing of data collection, thereby covering a vast road network area with minimal individual effort.

With network connectivity technologies like WiFi, 4G, and 5G, collected data can be quickly uploaded to a server and processed, and the results can be shared with contributors or relevant authorities. This not only reduces the cost and time involved in pavement assessment but also makes it possible to achieve ongoing, real-time monitoring, which is essential for timely and effective maintenance actions.

In summary, leveraged smartphone technology coupled with cloud computing can revolutionize the way we think about infrastructure maintenance. Not only could this approach make the process more cost-effective, but it could also make it far more timely and accurate. Providing a platform for crowdsourcing to active participants or volunteers could lead to continuous large-scale pavement assessment, thereby facilitating more effective and proactive maintenance strategies.

*Existing Smartphone Applications for IRI Measurement*

**Roadroid** is an Android-based smartphone app that translates accelerometer sensor information into IRI values used to assess pavement texture. It is considered a Class 3 IQL, and compared to a Class 1 IQL, Roadroid is economical and convenient to use in places where costly equipment is challenging to use. The app provides two roughness values: estimated IRI (eIRI) and calculated IRI (cIRI); the former is a relative measure obtained by correlating the peak and root mean square (RMS) analysis of the data to laser measurements, while the latter is a roughness index calculated using the standard quarter-car model. The initial phases of data collection and analysis have provided valuable insight in terms of determining IRI values by computing the influence of speed on the IRI data collected and introducing a calibration mechanism for addressing differences in the IRI data collected from various devices. It was found that the best results were obtained when driving at a consistent speed between 43 and 50 mph, and various settings can also be adjusted before the data collection process based on user preferences, as explained in detail in the user guide (Forslöf 2012). Roadroid also allows data to be uploaded to a server where the results can be calculated and presented to users as layers on a map through the Road Data Management System (RDMS), which also has tools for analyzing and viewing data in a geographic information system (GIS) environment. The results were approximately 81% accurate compared to laser measurements (Forslöf 2012, Jones and Forslöf 2015).

**RoadLab Pro** is a smartphone app developed by World Bank in association with Beldor Center, Soft Teco, and Progress Analytics LLC for collecting data used to calculate IRI values from gyroscope and accelerometer data (Wang and Guo 2016). It is available for both Android phones and iPhones. Before data collection, the initial setup involves linking the app with Dropbox or

Google Drive for storing the recorded data in the cloud, choosing the type of suspension based on the vehicle type, and mounting the phone either on the windshield or the dashboard.

RoadLab Pro provides two tabs for visualizing the data collection process. The first tab is called the Info tab, whose significant components are a visualization graph for vertical acceleration, a roughness indicator for the past 328 ft (100 m), a bump/pothole detection feature, and a button to start/stop data collection. The second tab is the Map tab, which displays the data collection location both textually and on a map, the bumps detected, and tags related to the current measurement.

The user can create, read, update, and delete data collected at four different levels: project level, road level, measurement level, and road interval level. The project level is at the top, while the road level includes the portion of the surveyed road on which data have been collected within a project. The measurement level consists of a list of 328 ft (100 m) road intervals, and the road interval level allows the user to explore IRI and GPS information related to the start and end points of data collection.

RoadLab Pro is considered to provide an economical way of assessing road conditions and is used in many developing countries as a comparable low-cost alternative.

**RoadBump** is an Android-based app that monitors road conditions using GPS and accelerometer data to calculate IRI values on different road types such as concrete, asphalt, dirt, and gravel (Materson 2015). The free version of RoadBump allows the user to record data and analyze the resulting maps and graphs, while the commercial version has additional features such as the ability to manage recorded data, share data to a cloud service like Google Drive or Dropbox, and split the data recording segments into shorter intervals.

The RoadBump interface consists mainly of three screens. The home screen allows the user to interact using a start/stop button to initiate or cease data collection. Before data collection begins, user preferences can be selected by accessing the settings icon to choose the units of measure, the minimum speed at which data collection should start, and a vehicle factor (0.8 for trucks, 1.0 for sport utility vehicles [SUVs], and 1.2 for mid-size cars). After data are recorded, the recording can be viewed on the home screen, and further details can be accessed through the folder button, where the desired recording can be chosen for viewing/analysis, renaming, deletion, sharing via email, or exporting as a CSV file (containing interpolated, segment-level, accelerometer, and GPS data). The map screen can be used to see a visualization of the recording. The recorded path is displayed as a Google Map (street map or satellite map) on the top half of the screen, with the bottom half displaying one of various graphs such as an IRI moving average graph, an IRI step graph, a PSR moving average graph, and an accelerometer graph. A summary of the recording, consisting of the IRI average and other technical information, can also be displayed.

Other highlights of the app include the ability to export geolocation data capable of importation to Esri's ArcGIS, collect data when a network connection is not available, choose fixed segment

intervals for which IRI is to be calculated, and view the recorded path as a color-coded polyline, with colors for depicting roughness (smooth, normal, or rough) configurable by the user.

**Carbin** is a crowdsourced app developed by Massachusetts Institute of Technology Concrete Sustainability Hub (MIT CSHub) researchers in association with the American University of Beirut and Birzeit University to assess road quality and analyze the effects of poor-quality roads on fuel consumption (MIT CSHub 2019). The goal of Carbin is to provide commuters with information about the greenest and cheapest routes that also offer comfortable ride quality. The initial motivation for the app arose from the research team's study that found that driving on poor-quality roads increases carbon emissions by 15% and increases cost-per-mile by 10% when fuel consumption and depreciation of vehicle performance are considered.

Carbin utilizes a third-party map provider, Mapbox, to display the vehicle's current location and the path on which accelerometer data have been collected to calculate the roughness index. Data are collected for a minimum of two minutes, and data collection is controlled through a start/stop button. Once recording is completed, the results can be viewed in the MyTrips tab, where data from all previous trips are available. Since this is a crowdsourced app, the results of each data collection event are sent to a server where they are further analyzed and then pushed to FixMyRoad.us, where a color-coded map is available for viewing worldwide.

The developers note that the app is capable of displaying the $CO_2$ savings of the user's vehicle and can infer the quality of the vehicle based on detected vibrations. The future scope of Carbin includes the capability to take advantage of the crowdsourced data and provide users with good-quality road options. As of today, Carbin has assessed approximately 600,000 miles of road worldwide.

**Roadbounce** is a data acquisition tool that calculates IRI values, recognizes potholes, differentiates the ride quality among different vehicles at various speeds, and generates alerts for poor-quality roads. It is available in three different versions (Azizan and Taher 2021):

- The free version allows users to collect data and export ride quality maps.
- The pro version includes additional features for custom calibration to various vehicles, speeds, and phones; conversion to other scales when assessing road roughness; and engineering of the collected data. This version is suitable for those seeking to perform quality checks on particular road sections.
- The enterprise version is designed specifically for use by government or private entities responsible for pavement management. It provides various features such as a color-coded map with GPS coordinates, the ability to compare results on a chosen road, storage of IRI data from different roads to visualize the road network, and a camera view.

Roughness estimation is initiated by mounting the smartphone in a fixed position and then starting the data collection process by tapping the start button. When the desired destination location is reached, the stop button is clicked to end the data collection. The reports generated can be viewed in the Saved Maps section, which provides information about overall ride quality.

**TotalPave** is a Canadian startup led by Coady Cameron that aims to help small cities or towns economically investigate road conditions, including by monitoring IRI and pavement condition index (PCI). The company has developed two products, TotalPave IRI and TotalPave PCI, that can be used to evaluate and ultimately improve road conditions. A comparative study between a Class 1 IQL profiler and TotalPave IRI showed that the roughness index measured by TotalPave IRI is 10% less than the value obtained from a laser profiler (TotalPave 2021).

The data collection process is similar to that of other smartphone apps based on the use of accelerometers and GPS sensors. The home screen of TotalPave IRI has a start button that is used to initiate data collection, and once the desired endpoint is reached the user ends data collection by tapping the stop button. This data collection process does not require a stable network connection and therefore can be carried out in places without network availability.

All of the logs saved on the smartphone can be uploaded to TotalPave's web portal, which can be accessed through a TotalPave account. The data are analyzed and mapped onto the community's roads as color-coded segments depicting each segment's roughness and to display any road distresses detected along a particular road segment. The portal also provides functionality for using stacked bar charts to view IRI progression over the years, allowing the user to assess the current condition of a road in its lifecycle before the cost of rehabilitation unduly increases.

In summary, commercial smartphone apps offer essential functionalities for users seeking to gather IRI values. While these apps are primarily based on QC models for calculations, they also offer diverse features depending on their design orientation, with some even estimating $CO_2$ emissions and PCI values. However, because they are proprietary, the specific methodologies they use for tasks such as filtering raw signal data, estimating road profiles, adjusting vehicle suspension parameters, and calibrating IRI calculations are usually not publicly disclosed.

Table 3 provides an overview of the various functions included in the commercial apps summarized above. As mentioned, each app is tailored to meet unique needs of different user groups. Interestingly, four of these apps are exclusively available on Android devices. While free versions are available for four of these apps, their capabilities are notably restricted. To access a broader set of features such as pavement data management, users have the option to purchase a license.

**Table 3. Summary of commercial smartphone apps in terms of pavement roughness measurements**

| | TotalPave | Carbin | Roadroid | RoadLab | RoadBump | RoadBounce |
|---|---|---|---|---|---|---|
| Available for iPhone | x | | | x | | |
| Active camera recording | | | x | | | x |
| IRI threshold mapping | x | | | x | x | x |
| Real-time data graphing | x | | | x | | |
| Calibration | | | x | | | x |
| Access to raw data | | | | x | x | |
| Vehicle settings | | x | x | x | x | x |
| Road settings | | | | x | x | |
| Segment roughness | x | | x | x | x | x |
| Free to the public | | x | | x | x | x |

Despite the convenience offered by these commercial apps, LPAs are often hesitant to fully adopt them, with reservations stemming from budget constraints, limited functionalities, and compatibility issues with existing local pavement management systems. Therefore, many LPAs continue to explore alternative solutions for pavement roughness data collection.

# CHAPTER 3: DEVELOPMENT OF IRI CALCULATION MODEL AND DATA PROCESSING ALGORITHM FOR SMARTPHONE APP AND SMART BOX

## Data Processing Algorithm Development

*Overview of Algorithm*

As discussed in earlier chapters, the standard method for calculating IRI is outlined in ASTM E1926 and commonly implemented using the Fortran programming language. However, in recent years the Python programming language has gained immense popularity due to its simplicity, accessibility, high degree of compatibility, and extensive library support. To capitalize on these advantages, this study developed a Python-based program that translates the original Fortran-based script specified in ASTM E1926 and incorporates double-integration of acceleration data. This newly developed algorithm can consequently directly process raw acceleration data to generate IRI values for evaluated roadways.

The algorithm's general workflow is illustrated in Figure 5. It leverages the standard double-integration method as well as the QC model for calculating road surface roughness. Notable differences between this study and others primarily involve the use of noise filtering, ACPs, and IRI corrections, all of which will be discussed in subsequent sections.

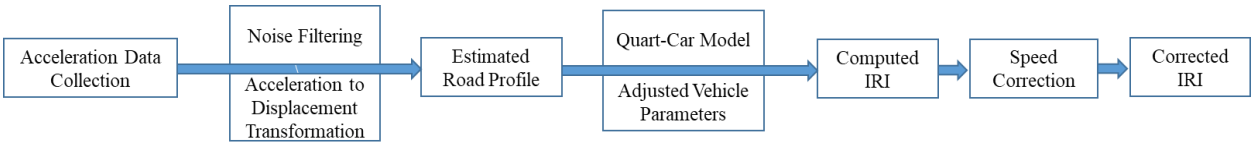



**Figure 5. Flow chart of IRI computation algorithm**

This approach provides a contemporary tool that offers more flexibility and compatibility than its Fortran-based predecessor, potentially making it easier for researchers and practitioners to more effectively assess road conditions.

*Acceleration Data Filtering and Processing*

The physical forces imparted to microelectromechanical system (MEMS) accelerometers embedded inside of the data collector device are converted into electrical signals that can be read by a smartphone or a Raspberry Pi (RPi) integrated circuit.

Multiple flaws can be passively introduced into this system, impairing the performance of the accelerometer. Critical errors, also referred to as misalignment, include bias offset, scale error, and deterministic error (El-Diasty and Pagiatakis 2008). Bias offset is a systematic error associated with an offset of the sensor's measurement from its true value. It is highly dependent on the particular MEMS sensor model used, and each such sensor could present significantly varying values. Since bias offset is a constant value in the time domain, signal processing that

converts from the time domain into the frequency domain will concentrate this type of error in the low-frequency band. Scale error is generated when the sensor's input-to-output ratio is inaccurate. In Figure 6(a), the slope of the predicted signal (dotted violet line) is a scale factor that can easily be calibrated by examining the output signal and input signal in the laboratory (Wang 2019). Misalignment of the sensor to the test vehicle causes deterministic error, also known as non-orthogonality of axes. The act of mounting the sensor to the test vehicle can compound any misalignment error and contribute to the total non-orthogonality of axes.

In addition to the error introduced by the sensor itself, noise is always present throughout vehicle experiments. For example, unanticipated vibrations and fluctuating temperatures both contribute to noise. Noise is primarily located in the high-frequency region after time-to-frequency domain transformation (i.e., Fourier transformation), as shown in Figure 6(b).

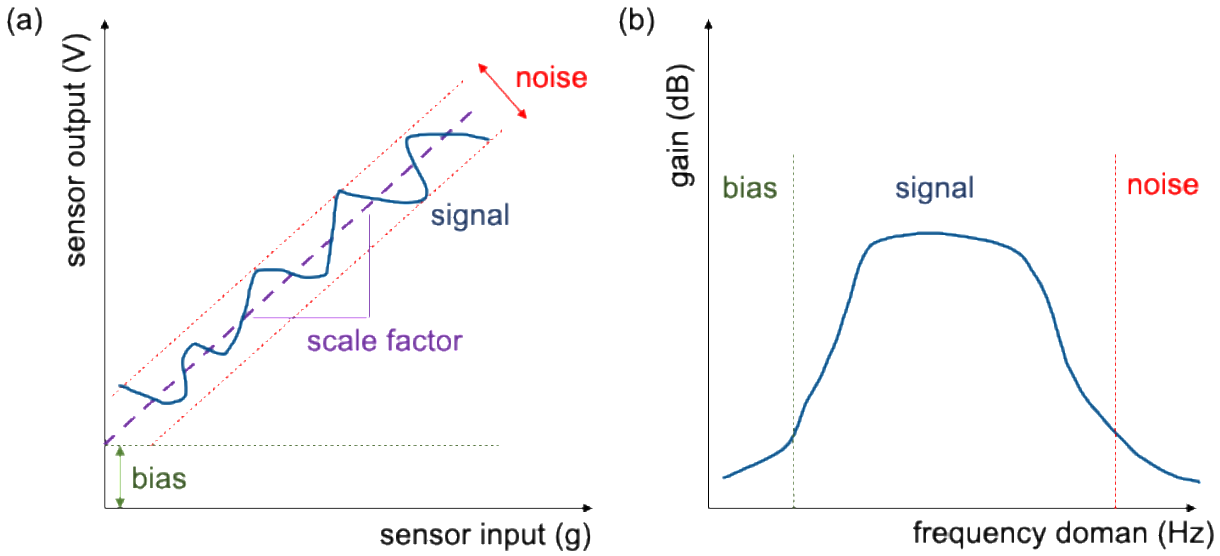



**Figure 6 (a) Errors from MEMS accelerometers (Wang 2019) and (b) acceleration errors in the frequency domain**

The accelerometer signal reported to the integrated circuit system contains a combination of the true acceleration and errors, and the contaminated portions of the signal may accumulate and contribute to a net vertical acceleration error over a long vehicle travel distance. Since the acceleration signal can be numerically double-integrated to determine velocity and displacement, resulting in the presence of a net acceleration that will generate a parabolic rise in the measurement, it is critical to calibrate the sensor and preprocess the data before feeding the raw signal into the model. Unlike built-in smartphone sensors, the MEMS sensors purchased for the smart box require precalibration to improve the accuracy of the scale factor ratio, and voltage readings relative to orientation were therefore carefully evaluated.

Since misalignment of smartphones and smart boxes could result in a misalignment error, thereby reducing data quality, it is necessary to minimize the angular difference between the direction of gravity and the vertical axis of the motion sensor. Karamihas (2021) proposed a

mathematical method to remove alignment error by collecting x-, y-, and z-axis acceleration in a stationary state after mounting the sensor; this method was utilized in this study. To remove low-frequency bias offset and high-frequency noise, a sixth-order bandpass Butterworth filter with a low-pass filter set at 5 Hz, a high-pass filter set at 200 Hz, and a moving average filter were applied to preprocess the test data before they enter the QC model.

This approach might deliver a more accurate result for smart boxes than for smartphones with various sensor models. According to previous research, since the acceleration recorded by different models of MEMS sensors could significantly vary from one unit to another (Kos et al. 2016), and because of such discrepancies among hardware models, a single version of the algorithm may not be able to provide consistent results. Consequently, it can be foreseen that smart boxes with consistent MEMS sensors could offer more accurate IRI measurements because all data preprocessing is based on the same accelerometer model. After precalibration and filtering, the raw acceleration can be integrated into velocity and then be integrated again to produce vertical displacement. To obtain uncorrected IRI values, the obtained displacement, referred to as the profile, can then be processed using the developed Python-based algorithm following ASTM E1926.

The developed algorithm utilized ACPs instead of GCPs because two types of vehicles, a truck and a mid-size car, were selected to collect IRI in order to enable a comparison of results. As mentioned above, determining the dynamic parameters for a single-tire system is very challenging. The applicable method is backcalculation using data from a site with known IRI values. In this study, the ACPs for trucks and mid-size cars were backcalculated from a selected portland cement concrete (PCC) calibration site in Boone, Iowa. The HSP was utilized to collect the latest reference IRI at the calibration site.

The final step is to correct the IRI values through use of speed correction functions. Although ACPs were implemented for these IRI data collection systems, the predicted IRI (or uncorrected IRI) from smartphones and smart boxes still differ from the reference IRI collected from the Class 1 HSP. Based on field observation, variation in driving speed could influence the tire response and result in vertical acceleration changes, leading to variations in IRI. Therefore, generalized speed correction functions were introduced and incorporated into the algorithm to improve the reliability and accuracy of IRI measurement.

## CHAPTER 4: DEVELOPMENT OF THE SMARTPHONE APP AND THE ASSOCIATED CLOUD SERVER

### Introduction

In this study, the ISU team developed a smartphone-based prototype application called CyRoads that uses smartphone sensors, i.e., accelerometers and GPS, to collect vertical acceleration and GPS data and display the IRI for a road section. CyRoads is available for both Android- and iOS-based smartphones, so most smartphone users can install this application and start making IRI measurements.

### System Design

The overall system design of CyRoads is shown in Figure 7. The vital components of this system are a smartphone device, a Flutter application (CyRoads), a Node.js-based cloud server, an IRI calculation algorithm, and a database. Before a data collection activity, the user must first fix the smartphone in the vehicle and set up CyRoads. Once the car has arrived at the target road, the user can begin data collection immediately while driving. After data collection, CyRoads allows users to upload raw data to the server and receive calculated IRI results.

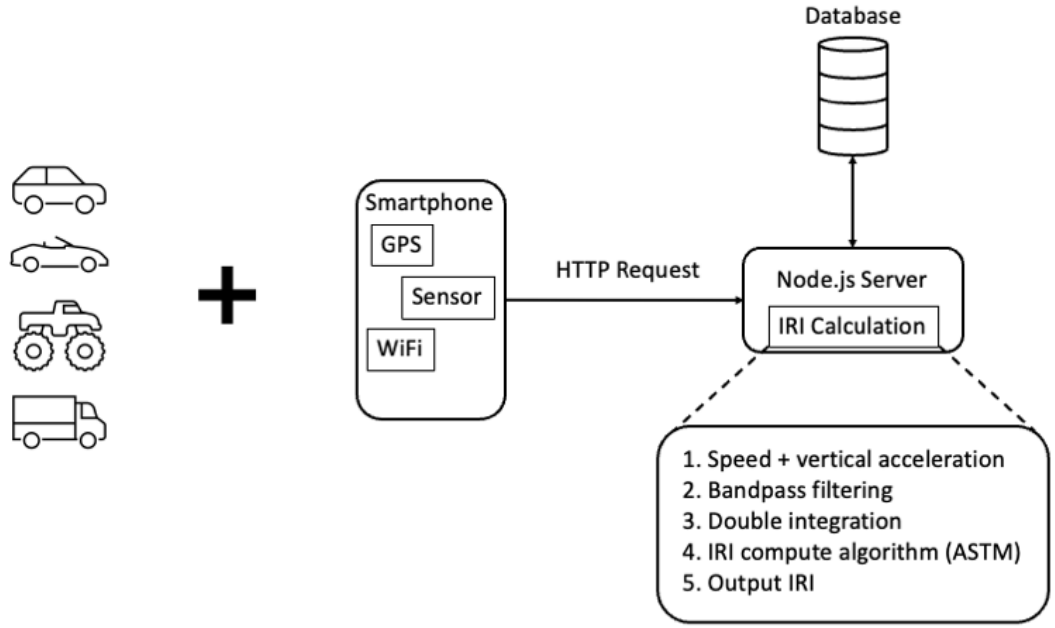



**Figure 7. CyRoads system design**

### CyRoads Smartphone Application

The CyRoads app is implemented using Flutter (version 3.13), which enables the app to run on both iOS and Android operating systems and provides comprehensive settings options, such as

vehicle type, phone alignment, and road category, for users to meet the various data collection circumstances and needs. As shown in Figure 8, the raw acceleration sensor readings are collected through the Flutter library, which is available for use by the built-in smartphone accelerometer. The raw data are periodically written to the raw data file at a data collection frequency of 100 Hz during data collection activities, with the selection of 100 Hz based on frequency testing of both Android and iPhone accelerometers. Each collected raw data file contains timestamps, acceleration data, GPS coordinates, and moving speed data, and the files are organized by timestamp corresponding to when the data collection started. Users can then choose to upload the files to a backend server running in the cloud.

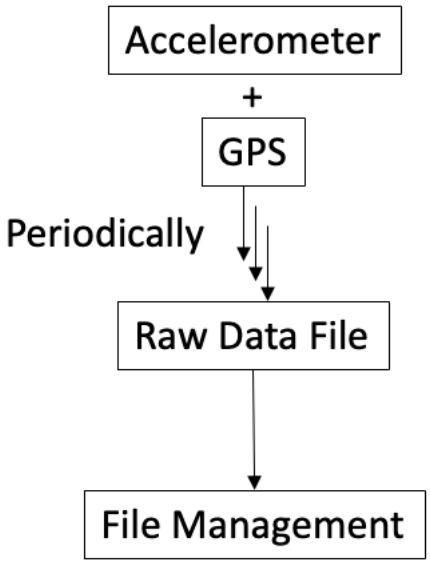



**Figure 8. CyRoads data collection diagram**

**Backend Server**

A node.js-based backend server and PostgreSQL database are deployed to the Amazon Web Services (AWS) cloud. An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the AWS infrastructure. One EC2 instance is used to host the backend server, which can call the python-based IRI calculation algorithm using the raw data file, vehicle type, and other items as input variables.

Figure 9 demonstrates the dataflow process initiated when the user uploads a raw data file from the smartphone application to the cloud server. First, the user selects the raw data file to upload and then taps the upload button in the CyRoads app. Second, the raw data file and other trip information are posted to the backend server by a Hypertext Transfer Protocol (HTTP) request. Third, the backend server calls the IRI calculation algorithm to calculate the IRI values corresponding to the raw data file. Fourth, the IRI values for this trip and other trip information, including vehicle type, sampling frequency, device, road name, road type, surface type, and date, are saved to the PostgreSQL database in the measurements and trips table. Finally, the overall IRI value and segment IRI values are returned to the CyRoads app and displayed to the user.

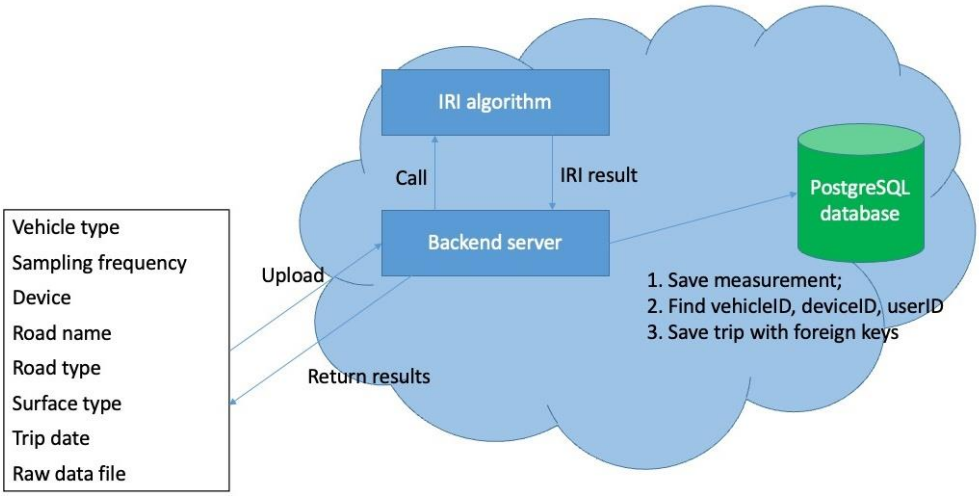



**Figure 9. Dataflow when uploading from the smartphone application to the cloud server**

Figure 10 shows the user login process from the smartphone application. Similar to the upload process, the user sends a username and hashed password to the backend server via an HTTP request. The backend server then verifies whether the password is correct by comparing it with the stored credentials in the PostgreSQL database, and if the credential is found to be correct, the backend server returns a token; otherwise, it returns an error message.

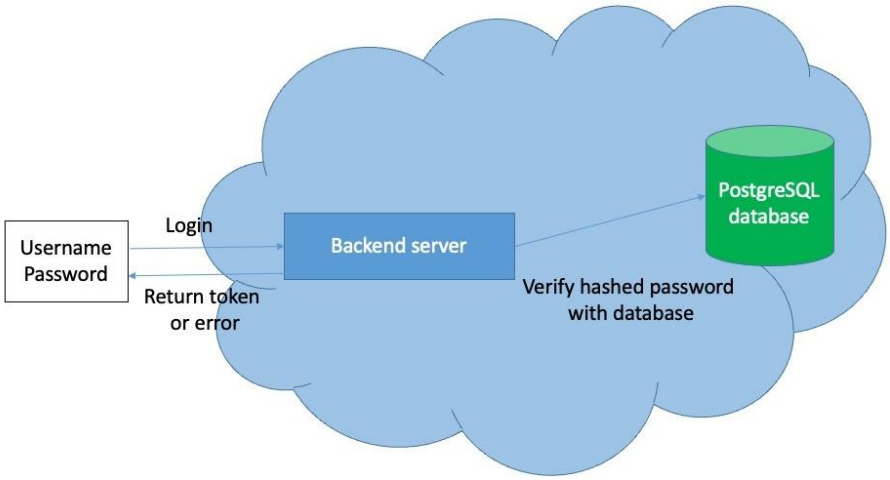



**Figure 10. Dataflow of user login from the smartphone**

Figure 11 shows the database table organization, where the trip table is the center of the database. All collected trips are saved into the trip table, which contains measurement, vehicle, device, and user information, details of which are stored in other related tables.

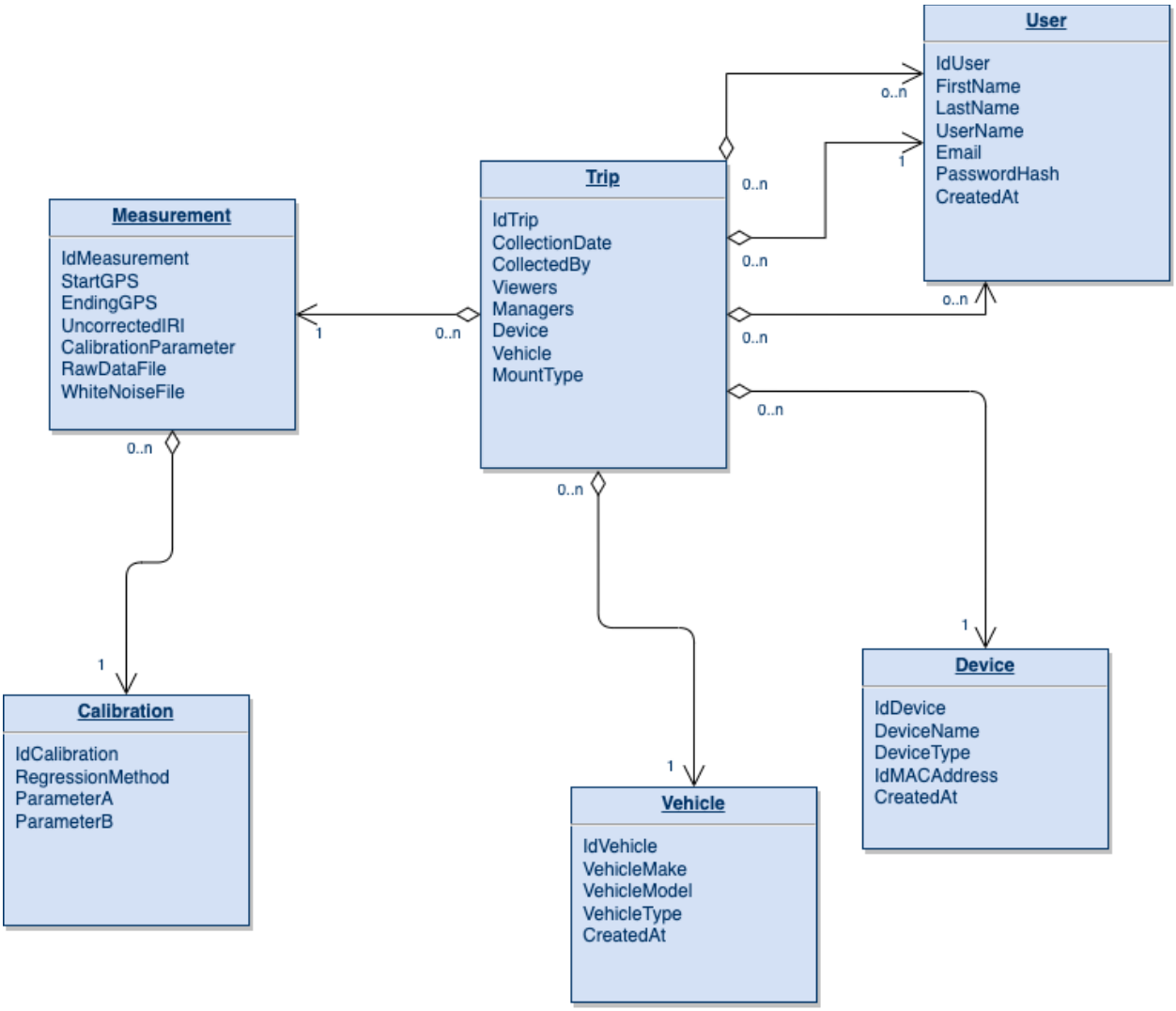



**Figure 11. Database design**

## Summary of CyRoads App

In summary, the CyRoads app provides a low-cost alternative for determining pavement roughness. As a smartphone app, it is compatible with both Android- and iOS-based mobile devices and provides full functionality for managing the raw data collected and communicating with the AWS cloud server. The IRI computation proceeds using the cloud server, and trip information, including IRI results, is stored in the database and sent back to the CyRoads app. A user manual for CyRoads is provided in Appendix A.

# CHAPTER 5: DEVELOPMENT OF THE SMART BOX

As introduced in the previous chapter, a prototype smartphone app named CyRoads was developed to measure pavement conditions in terms of IRI. The operating principle of this system is to utilize the built-in accelerometer and GPS sensors of smartphones to collect raw acceleration data and convert them into IRI values using a specific algorithm stored in a cloud server. However, as a high-tech product, the modern smartphone has many unnecessary integrated sensors that are not used for IRI measurement and are relatively expensive, which increases its utilization cost. Moreover, the different built-in accelerometers in various smartphone brands and models could potentially result in variations in IRI measurements. To overcome such challenges, a smart box integrated with accelerometer and GPS sensors was proposed.

This study aimed to develop an easy-to-operate, portable, and integrated smart box capable of monitoring inertia in a moving vehicle and providing the user with an IRI report. The developed smart box employs MEMS inertial sensors to monitor acceleration and a GPS module to report location references; the system thus uses the same type of IRI detection mechanism used in existing inertial-based IRI measuring instruments. A specifically designed smartphone app was developed to communicate with the smart box for managing data collection, uploading raw data files to the cloud server, and receiving IRI results. However, compared to smartphone-based devices, the prototype smart box has no expensive extraneous components, lowering the overall cost of the device. The use of the smart box could also bring other benefits that a smartphone app cannot provide; for example, smart boxes can use a common set of standard measurement devices, resulting in little variance in the collected data. Another advantage of the smart box is the relative ease of performing maintenance because no system updates are required.

## Packaging

As shown in Figure 12a, the prototype smart box assembly has three main parts: an aluminum metal enclosure with a microcontroller board and critical sensor components, a protective metal lid with an adjustable strap, and an external GPS antenna. The enclosed smart box measures 3.6 in. long, 2.7 in. wide, and 2.3 in. high, and its electronic components are protected from liquid spills by a waterproof case. A hole in the side of the box allows the user to observe programmed light-emitting diodes (LEDs) that indicate the status of the hardware during an IRI measurement. The box lid, attached by removable screws, provides good protection for the internal components and allows for easy repair or customization of the box for future improvements.

There are two open hardware interface windows at the sidewall of the box enclosure. Figure 12b shows the front side, which provides four universal serial bus (USB) ports and one ethernet switch. The sidewall of the enclosure shown in Figure 12c provides one USB type-C, one audio/video (A/V), and two micro HDMI ports. Such ports offer multiple functions, allowing the user to connect to external devices (display monitor, mouse, keyboard, headphones, etc.) without disassembling the box enclosure. The external connections can help the user easily debug the system, especially during data collection or troubleshooting.

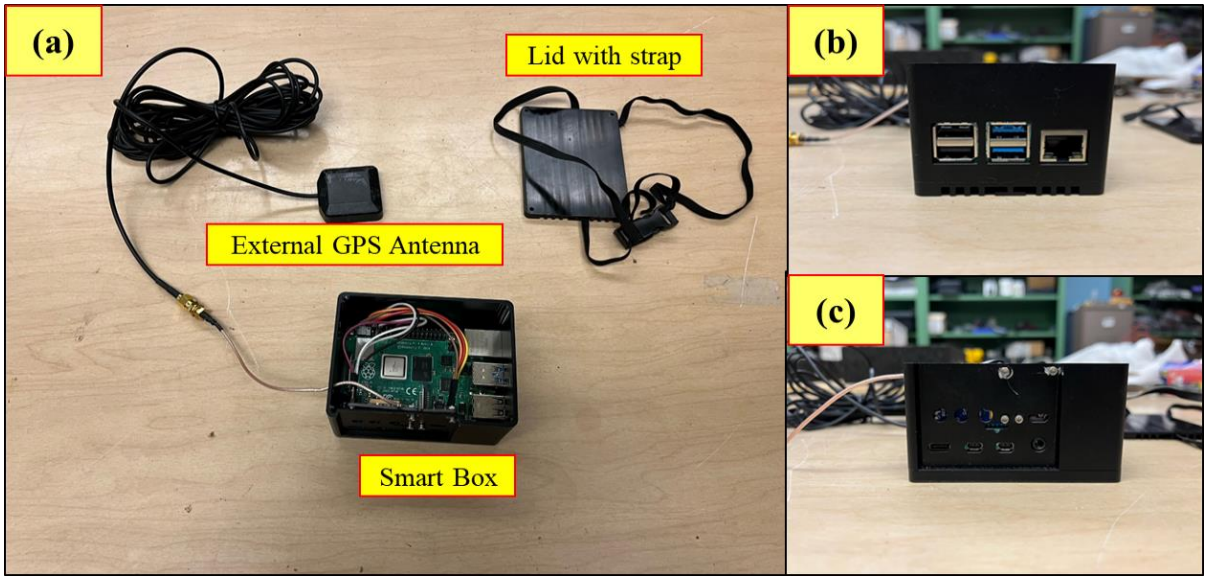



**Figure 12. Smart box features: (a) smart box package with a removable lid and an external antenna, (b) smart box enclosure front view, and (c) smart box enclosure side view**

### Electronic Components

The smart box system for IRI measurement shown in Figure 13 consists of three basic electronic components: a single-board microprocessor, RPi 4 Model B; an inertial measurement unit (IMU), MPU-9250; and a GPS module, MTK3339. Commercially available breakout boards from SparkFun (IMU Breakout - MPU-9250) and Adafruit (Ultimate GPS Breakout - 66 channel w/10 Hz updates - Version 3) are used to collect digital data from the sensors for analysis by the microcontroller. The writing/pause action for IRI data collection is controlled either remotely by a specially developed smartphone app or manually through an external display monitor, mouse, and keyboard. Other smart box components include an external GPS antenna, a car charger, and a microSD card (SanDisk).

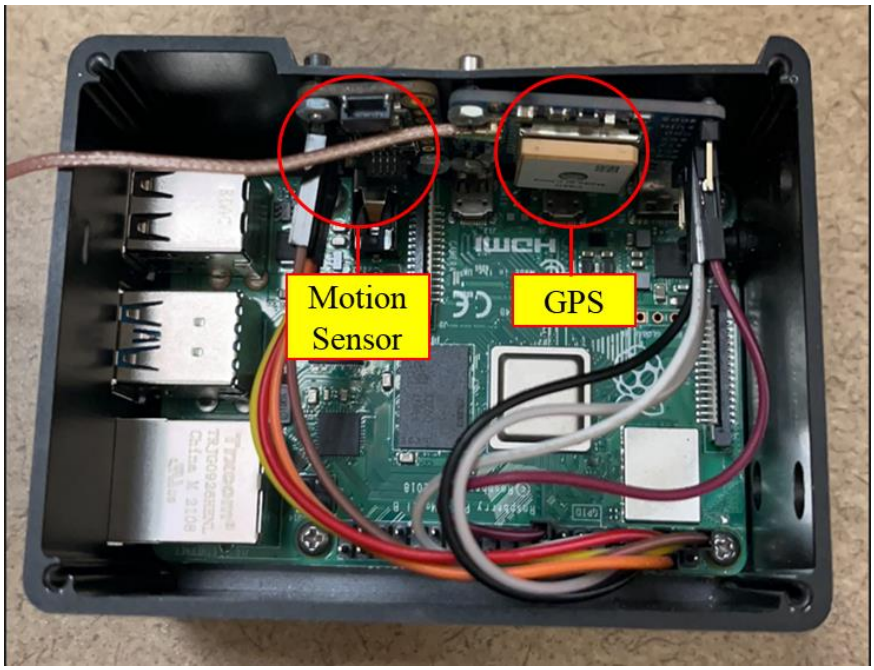



**Figure 13. Inside view of the smart box**

The RPi 4 Model B is a low-cost single-board microcomputer developed by the Raspberry Pi Foundation. It is comprised of a 1 GB or greater SDRAM, a 64-bit quad-core processor that runs at 1.5 GHz, and a dual-band 2.4 GHz and 5 GHz wireless local area network (LAN). The extended 40-pin GPIO connection provides connectivity with hardware such as the motion sensor and the GPS module. The Linux-based operating system RPi OS (Debian 10) is installed on this microcomputer for hardware logic control and data collection and processing. The TDK InvenSense MPU-9250 is a 9-degree-of-freedom (DoF) MEMS-based IMU that can detect linear acceleration as well as angular velocities. The analog voltage values from the inertial sensor are digitized using an on-chip 16-bit resolution analog-to-digital converter (ADC). The inertial sensor data are transferred to the RPi through an inter-integrated circuit (I2C) interface, with a sampling frequency set to 100 Hz, the same as the setting of CyRoads. The MTK3339 GPS module has a sensitivity of -165 dB and provides a positional accuracy of less than 10 ft. With its built-in antenna, this GPS device can track up to 22 satellites on 66 channels. Based on field testing, the GPS signals were found to be weak and unstable due to obstructions such as the car body, so an external GPS antenna was added to gain an additional 28 dB of sensitivity. The GPS operates at 1 Hz and uses a universal asynchronous receiver-transmitter (UART) serial interface to communicate with the RPi.

Because of its higher power requirements, the RPi 4 Model B requires a 3.0 A USB type-C power supply. Figure 14 shows the RPi GPIO pins connected to the inertial sensor through an I2C interface and to the GPS module via serial connections. The yellow LED on the RPi indicates that a file transfer to the microSD card is in progress, and the red LED on the GPS module indicates GPS signal acquisition. The electronic connections are either tin-soldered or connected by pins to jumper wires affixed with electrical tape. Because the electronic components are screwed to the sidewall of the box enclosure, the acceleration and location of the

motion sensor accurately reflect the moving status of the smart box attached to the vehicle. When the box is held flat (Figure 12a), the y-axis of the accelerometer is precisely aligned to face downward toward the earth (in the direction of gravity).

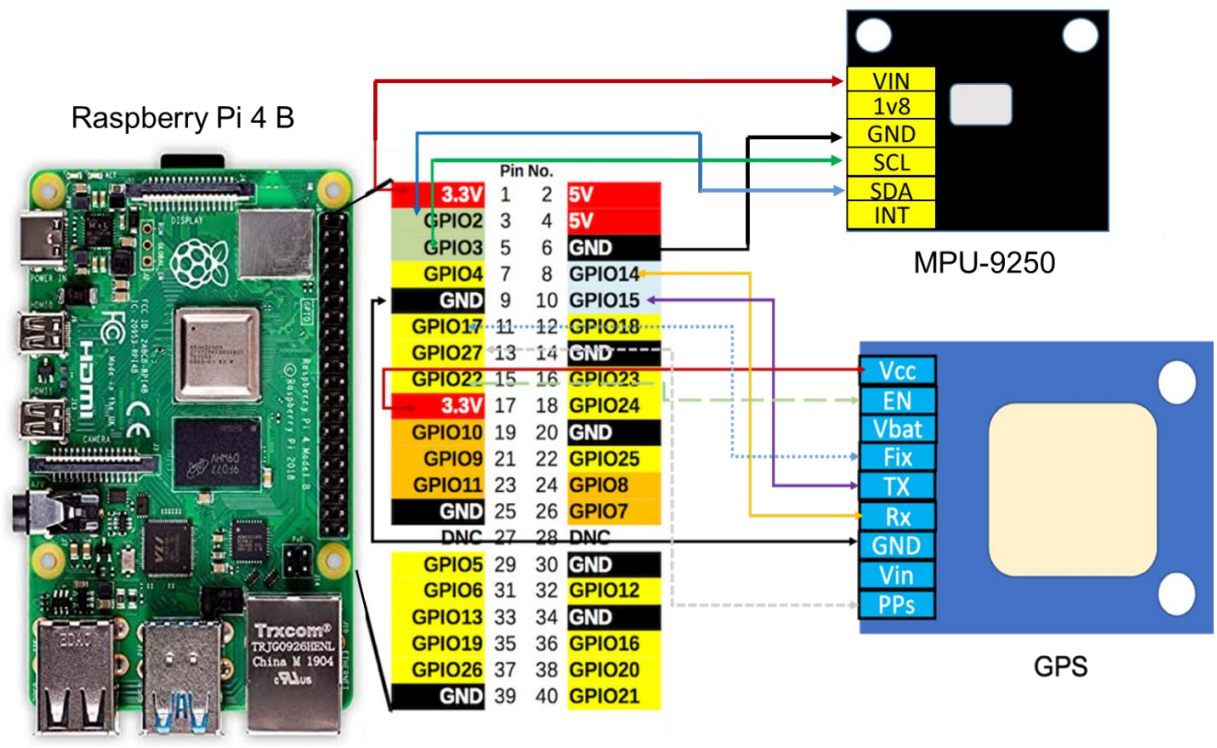



**Figure 14. Connections between the major components**

The prototype smart box can be manufactured much more cheaply than existing smartphones capable of Class 3 IRI measurement because it has less integrated hardware (Hanson et al. 2014, Aleadelat et al. 2018, Wang et al. 2020, Zhang et al. 2021). While the IRI measurement process involves only a few smartphone components (i.e., chips, inertial sensors, and GPS units), a typical smartphone contains additional hardware that is unneeded for this application. For example, camera lenses, touch screens, power management integrated circuits, radio-frequency transceivers, audio systems, and Bluetooth connectivity add unnecessary expense. The prototype smart box utilizes fewer electronic components to obtain IRI values, resulting in a much lower cost, as shown in Table 4.

**Table 4. Component prices of the prototype smart box (before-tax prices as of December 13, 2022)**

| Component | Functionality | Cost |
|---|---|---|
| Raspberry 4 Model B (2 GB) | Microcontroller | $45.00 |
| Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates - Version 3 | GPS module | $39.95 |
| Adafruit TDK InvenSense ICM-20948 9-DoF IMU (MPU-9250 Upgrade) | Motion sensor | $14.95 |
| Female-female jumper wires | Connection | $0.80 |
| Box enclosure with screws and nuts | Protection case | $27.50 |
| GPS Antenna - External Active Antenna - 3-5V 28dB 5 meter SMA | User input | $0.45 |
| SMA to uFL/u.FL/IPX/IPEX RF adapter cable | Connection cable to GPS antenna | $3.95 |
| Adjustable strap | Fix the smart box in the vehicle | $5.00 |
| Car charger and USB type-C cable | Power supply | $15.00 |
| Sum | | $152.60 |

Different brands and models of smartphones also use different IMUs, and because the acceleration output highly influences the IRI results, variance in IRI measurements between different smartphone brands and models is inescapable. In contrast, the prototype smart box eliminates the distinction between developer and consumer devices, resulting in users benefitting from an optimally calibrated system that produces a consistent and reliable IRI report.

**Smart Box Application Development**

Although the prototype smart box was successfully developed and could collect data in the vehicle, an external display monitor, a keyboard, and a mouse were still required to operate the device, which meant that two people were required in the vehicle to use the system: one driving the vehicle and the other operating the smart box. A custom smartphone app was therefore developed to communicate with the smart box and cloud server and replace the external display monitor, keyboard, and mouse.

The smart box runs a Linux-based operating system (RPi OS), as noted above, and a Python/Flask server is used to support HTTP requests from the smartphone app. Adafruit libraries and the serial port are used to receive accelerometer and GPS data. When the smart box is powered on, a new WiFi hotspot for the ISU smart box appears, and the user can use the smartphone app to connect to this access point to communicate with the smart box.

When a user logs in to the smart box connection app, he or she can enter the IP address for the RPi server or cloud-based node.js backend server. Figure 15a shows the settings tab for this smart box connection app, through which the user can enter the road name of the current trip and select the road category, vehicle type, and surface pavement type. Once the app and smart box are connected, the user can trigger data collection, and when data collection is terminated, the user will receive the data transmitted from the smart box through the app. The user can then

upload the raw data file to the cloud server using the upload function in the smart box connection app, as shown in Figure 15b.

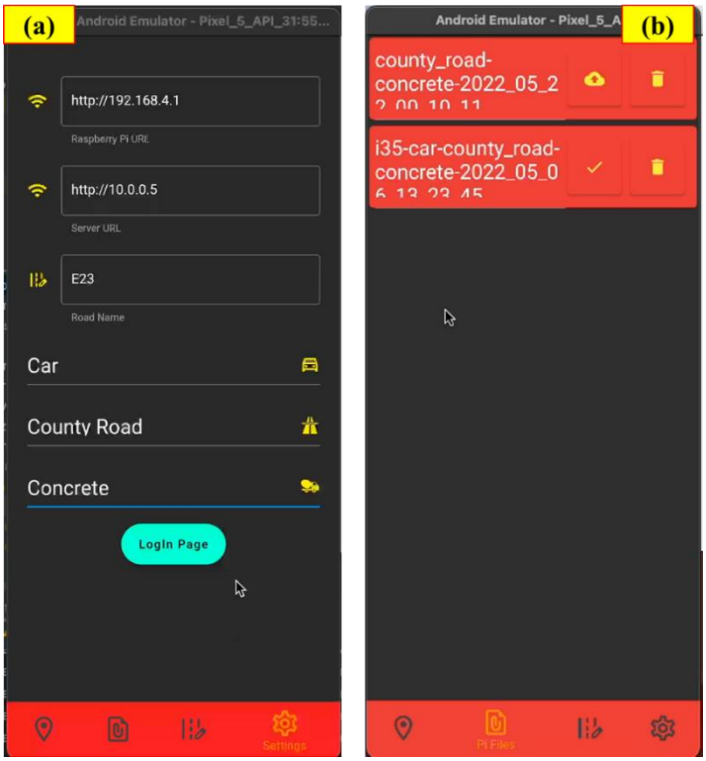



**Figure 15. Smart box connection app: (a) settings tab and (b) file management tab**

### Summary of Smart Box

The design of the entire smart box system is shown in Figure 16, where it can be seen that the smart box system consists of a smart box enclosure integrated with electronic components, a smart box connection app, and a cloud server. This system can successfully achieve functions similar to those of CyRoads at a much lower cost, approximately $150. Another advantage of this smart box system is that it uses consistent electronic components, avoiding the variations in IRI measurements that can arise from the use of different sensor types and models. Moreover, the smart box could be extended to provide more specific functions based on the user's needs. A detailed operation manual for the smart box is provided in Appendix B.

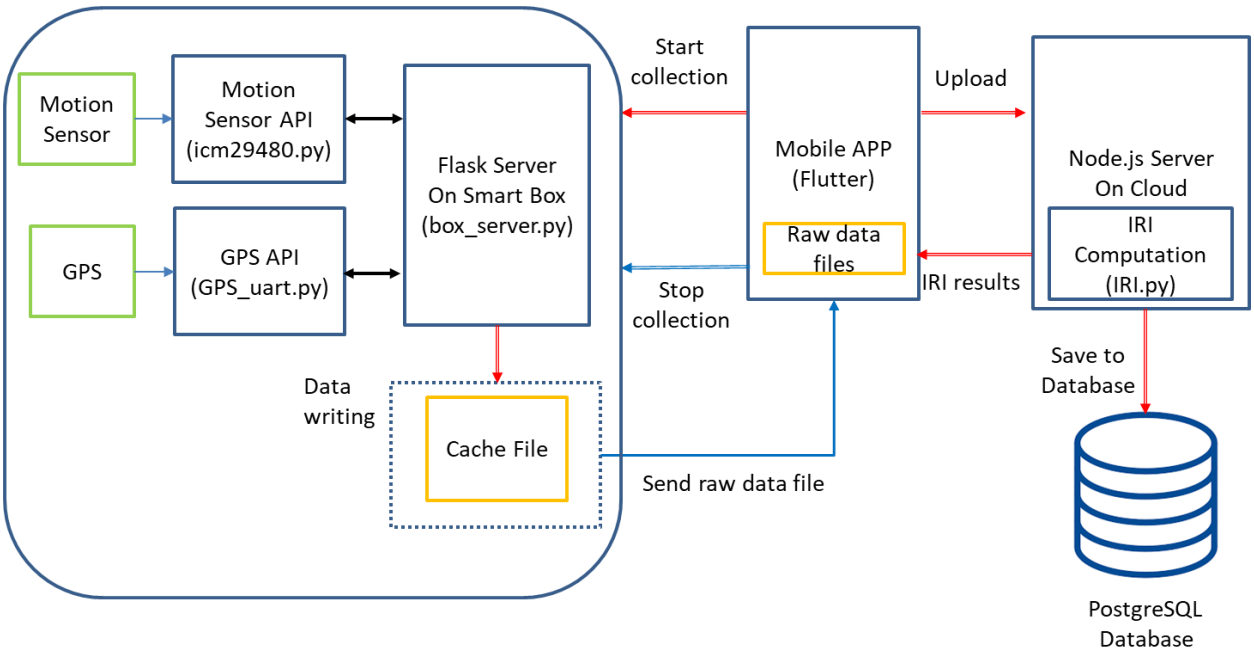



**Figure 16. System design of the smart box**

# CHAPTER 6: EVALUATION, CALIBRATION, AND VALIDATION OF SMARTPHONE APP AND SMART BOX

This study introduced two innovative systems for collecting IRI data on road pavements: the CyRoads smartphone app and a corresponding smart box. Both systems were integrated with a shared cloud server to support seamless data processing and trip information management. To assess these tools, the ISU team conducted a comprehensive field data collection protocol aimed at evaluating, calibrating, and validating their reliability, stability, and measurement accuracy.

## Field Data Collection Protocol

### *Validation Sites*

To evaluate road roughness, data were collected from two types of sites. The first was a calibration site used for tuning vehicle suspension parameters and developing an IRI correction function. The second was comprised of a group of 24 validation sites located near Ames, Iowa, that were selected to assess the performance of the CyRoads smartphone app and the smart box. These sites included 12 highways, 10 county roads, and 2 city streets, all paved with either PCC or hot-mix asphalt (HMA). Each surveyed pavement section was less than 2 miles long.

Devices used for data collection at these validation sites included a Class 1 HSP, four different smartphones, and a smart box. Table 5 provides detailed information for all 24 sites. Notably, Sites 13, 14, 23, and 24 were evaluated at varying speeds starting from 30 mph to gauge the accuracy of both CyRoads and the smart box at low to medium speeds. The maximum testing speed was capped at 60 mph because Class 1 HSP measurements are most reliable within a 20 to 60 mph speed range. Each section was also surveyed with a Class 1 HSP to establish reference IRI values.

**Table 5. Information on field data collection sites near Ames, Iowa**

| Site No. | Group | Route | Direction | Begin. Milepost | End. Milepost | Surface Type | Speed (mph) | Length (ft) | HSP (in./mile) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Highway | US 30 (1) | EB | 152 | 153 | PCC | 60 | 5,200 | 95 |
| 2 | Highway | US 30 (1) | WB | 152 | 153 | PCC | 60 | 5,200 | 96 |
| 3 | Highway | US 30 (2) | EB | 144 | 146 | PCC | 60 | 7,500 | 71 |
| 4 | Highway | US 30 (2) | WB | 144 | 146 | PCC | 60 | 7,500 | 79 |
| 5 | Highway | US 30 (3) | EB | 142 | 144 | PCC | 60 | 7,500 | 68 |
| 6 | Highway | US 30 (3) | WB | 142 | 144 | PCC | 60 | 7,500 | 76 |
| 7 | Highway | I-35 | NB | 111 | 113 | HMA | 60 | 5,800 | 61 |
| 8 | Highway | I-35 | SB | 111 | 113 | PCC | 60 | 5,800 | 98 |
| 9 | Highway | US 69 | NB | 190th St. | 180th St. | HMA | 55 | 7,800 | 86 |
| 10 | Highway | US 69 | SB | 180th St. | 190th St. | HMA | 55 | 7,800 | 91 |
| 11 | Highway | IA 17 | NB | 205th St. | 195th St. | HMA | 50 | 5,300 | 65 |
| 12 | Highway | IA 17 | SB | 195th St. | 205th St. | HMA | 50 | 5,300 | 71 |
| 13 | County Rd. | R38 (1) | NB | Lincoln Way | 215th St. | PCC | 30, 40, 50 | 6,000 | 139 |
| 14 | County Rd. | R38 (1) | SB | 215th St. | Lincoln Way | PCC | 30, 40, 50 | 6,000 | 119 |
| 15 | County Rd. | R38 (2) | NB | 212th St. | Cameron Rd | PCC | 50 | 2,600 | 143 |
| 16 | County Rd. | R38 (2) | SB | Cameron Rd. | 212th St. | PCC | 50 | 2,600 | 118 |
| 17 | County Rd. | E26 | EB | X Ave. | Y Ave. | PCC | 50 | 4,900 | 100 |
| 18 | County Rd. | E26 | WB | Y Ave. | X Ave. | PCC | 50 | 4,900 | 92 |
| 19 | County Rd. | E23 | EB | 510th Ave. | R50 | HMA | 50 | 4,400 | 53 |
| 20 | County Rd. | E23 | WB | R50 | 510th Ave. | HMA | 50 | 4,400 | 60 |
| 21 | County Rd. | R50 | NB | Cameron Rd. | 180th St. | HMA | 50 | 7,300 | 149 |
| 22 | County Rd. | R50 | SB | 180th St. | Cameron Rd. | HMA | 50 | 7,300 | 143 |
| 23 | Street | Airport Rd. | EB | S Loop Dr. | S Duff Ave. | PCC | 30, 35, 40, 45 | 7,000 | 174 |
| 24 | Street | Airport Rd. | WB | S Duff Ave. | S Loop Dr. | PCC | 30, 35, 40, 45 | 7,000 | 178 |

Note: EB = eastbound, WB = westbound, NB = northbound, and SB = southbound. All data were collected in the right lane of each site.

*Vehicle Types*

Figure 17 depicts the two vehicles used for data collection: a 2012 Ford F250 pickup truck and a 2018 Chevrolet Impala mid-size car. The study highlighted that vehicle type significantly impacts IRI measurements because of differences in vehicles' suspension response systems. These differences are quantified in the QC model through specific parameters.


**Figure 17. Vehicles used for data collection: (a) Ford F250 equipped with a high-speed profilometer and (b) 2018 Chevrolet Impala**

In accordance with other commercial apps such as Carbin, RoadLab, and Roadroid, the CyRoads app and smart box offer "Truck" and "Car" as vehicle type options. Representative parameters for these vehicle types were obtained through tests conducted at a calibration site in Boone County, Iowa. These parameters, detailed in Table 6, are based on the 2012 Ford F250 for trucks and the 2018 Chevrolet Impala for cars. The CyRoads app and smart box use these parameters to apply different algorithms and speed correction functions to the collected data, resulting in more accurate IRI computation.

**Table 6. Actual vehicle parameters for truck and mid-size car**

| | Unsprung Mass | Tire Spring Rate | Suspension Spring Rate | Suspension Damping Rate |
|---|---|---|---|---|
| Symbol, unit | $M_{us}/M_s$ | $K_t/M_s$ | $K_s/M_s$ | $C_s/M_s$ |
| Golden car parameter | 0.15 | 653 | 63 | 6 |
| Adjusted car parameters - Truck | 0.25 | 653 | 63 | 14 |
| Adjusted car parameters - Car | 0.175 | 750 | 63 | 14 |

Note: 2012 Ford F250 and 2018 Chevrolet Impala parameters represent truck and car parameters, respectively.

*Smartphone Types*

Table 7 gives the specifications of the four different smartphones used for IRI data collection in this study: iPhone 13, Samsung Galaxy S8, Google Pixel 4 XL, and Motorola G7 Power. The table highlights differences in phone sizes, weights, and processors, factors that could potentially

influence IRI measurement accuracy. While specific information about the accelerometer and GPS sensors in these smartphones is not available, the study sought to account for device diversity by selecting these four phones as representative models for a broader range of data collection.

**Table 7. Specifications of four smartphones used for IRI data collection**

| | Dimensions (in.) | Weight (g) | Processor | Release Year |
|---|---|---|---|---|
| iPhone 13 | 5.8 | 174 | A15 Bionic | 2021 |
| Google Pixel 4 XL | 6.3 | 193 | Snapdragon 835 | 2019 |
| Motorola G7 Power | 6.4 | 193 | Snapdragon 665 | 2019 |
| Samsung Galaxy S8 | 5.8 | 155 | Snapdragon 855 | 2017 |

*Alignment of Smartphone and Smart Box in the Vehicle*

Smartphone and smart box alignment methods could influence the processing algorithm because different methods can change the vertical gravity axis of the motion sensors. In this study, two alignment methods for smartphones and smart boxes were tested to evaluate their impact on IRI data collection: mounting devices on the windshield (mount mode) and banding devices to the center console (band mode). Figure 18a and Figure 18b depict the use of a rigid magnetic mount for windshield alignment. Flexible mounts were avoided due to their potential for inflating IRI measurements. For center console alignment, Figure 18c and Figure 18d show how devices were securely strapped to the console. The choice of alignment method impacts the algorithm's input: for the mount mode, the x-axis accelerometer data are used, while for the band mode, the z-axis data are used. Though windshield mounts may introduce some inaccuracies in IRI measurements, they offer advantages such as enabling the use of the device's camera or providing a real-time map view.


**Figure 18. Alignment methods: (a) magnetic windshield mount, (b) smartphone mounted to the windshield, (c) smartphone banded to the center console, and (d) smart box banded to the center console**

## Calibration

This study employed a calibration site operated by the Iowa DOT for various calibration activities, including calibration of the vehicles' ACPs and speed correction parameters. Figure 19 shows that this site is situated on 217th Road in Boone, Iowa, in close proximity to the Central Iowa Expo. The site features a PCC road that the Iowa DOT regularly uses to evaluate routine maintenance activities and calibrate devices, particularly its Class 1 HSP.


**Figure 19. PCC calibration site: (a) location and (b) appearance**

Historical roughness data collected by the Iowa DOT at this site has ranged from 90 to 95 in./mile. The most recent IRI measurement, conducted for this study in 2022 utilizing an HSP, indicated a value of 100 in./mile in the eastbound direction. These data not only demonstrated reliability but also served as the reference IRI value for the calibration site in this study.

As described above, the calibration site initially served to determine ACPs for both the truck and the mid-size car, specifically for use in the QC model. Both vehicles were driven at a speed of 50 mph, and IRI data were collected using smartphones mounted with specialized banding mechanisms.

A subsequent test was conducted to identify the actual suspension parameters that yielded the smallest differences in IRI measurements among the five devices used in this study. These results are detailed in Table 6.

The subsequent phase of the study involved identifying speed correction functions tailored to the various vehicle types, devices, and data collection modes (e.g., banded or mounted devices). To develop these speed correction functions, five driving speeds ranging from 30 to 70 mph were selected for analysis. With appropriate traffic control measures in place, field data were collected at the calibration site in 2022. All five devices—one smart box and four smartphones—were utilized for this purpose.

Figure 20 illustrates the impact of speed on IRI measurements in the absence of any speed correction.

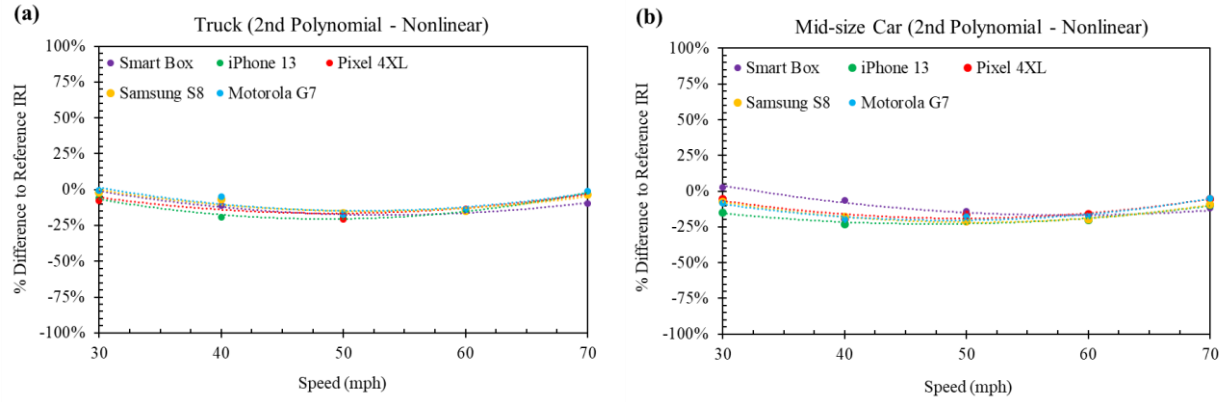



**Figure 20. Differences in IRI values between the smart devices and the HSP at the calibration site at various speeds**

These IRI values were calculated using the previously identified ACPs. The data reveal that the performance of all devices is influenced by driving speed. Furthermore, the observed trendline closely approximates a second-degree polynomial curve, as described in equation (6):

$$y = av^2 + bv + c \tag{6}$$

where y is the percent difference to the reference IRI, v is the vehicle speed (mph), and a, b, and c are speed correction parameters.

Based on the second-degree polynomial equation, the IRI values after speed correction could be expressed as equation (7):

$$IRI_c = \frac{IRI_{uc}}{av^2+bv+c+1} \tag{7}$$

where $IRI_c$ is the IRI after speed correction, $IRI_{uc}$ is the uncorrected IRI, v is the average driving speed while using CyRoads or the smart box, and a, b, and c are speed correction parameters.

Figure 21 shows IRI outcomes at the calibration site both before and after speed correction. The charts clearly demonstrate that the IRI values are markedly influenced by speed variations, owing to the dynamic interaction between vehicle tires and the road surface. Upon implementing speed correction, this speed-related variance was substantially mitigated. The corrected IRI values closely approximated the benchmark IRI value of 100 in./mile, suggesting that the accuracy of the IRI computation algorithm can be enhanced through the incorporation of speed correction techniques.

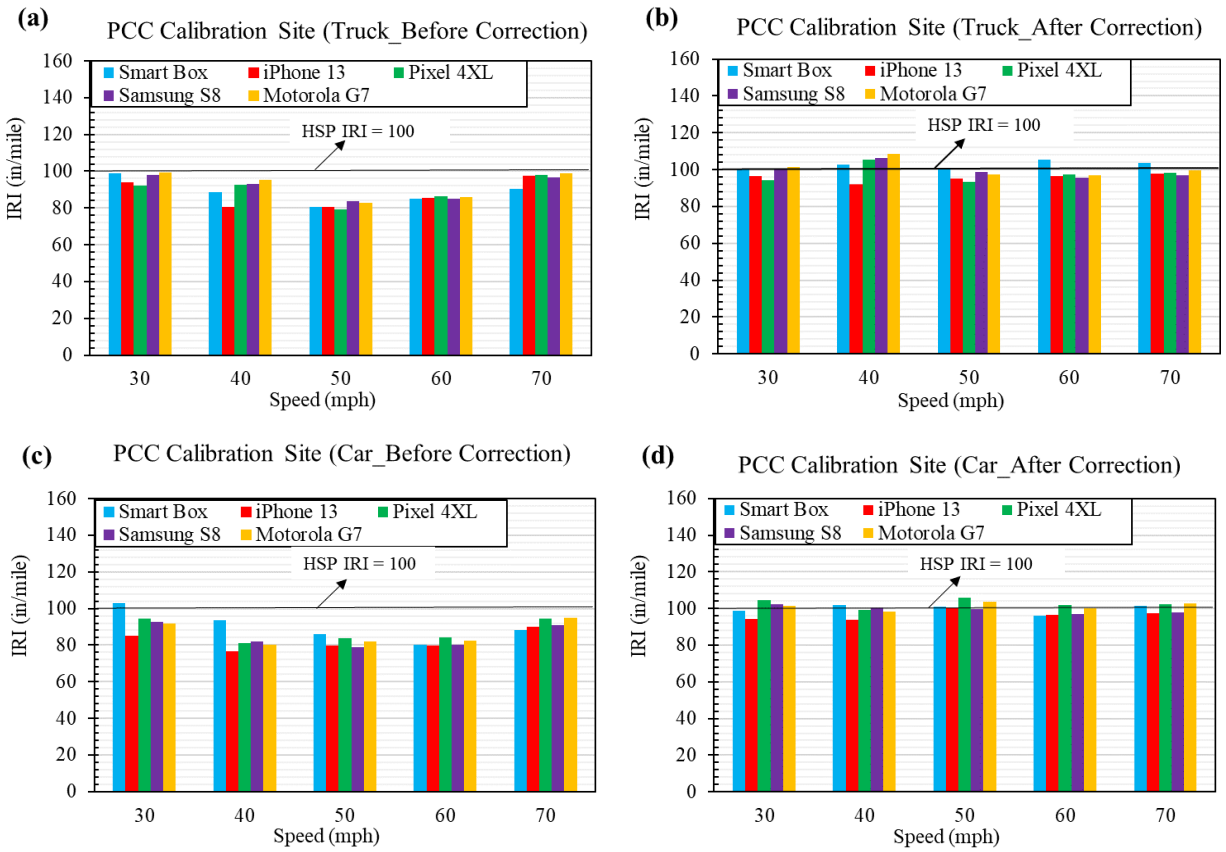



**Figure 21. Effects of speed corrections at the calibration site**

The speed correction parameters utilized in this study are detailed in Table 8. It is important to note that the parameters for smartphones were generalized based on a select sample of four devices.

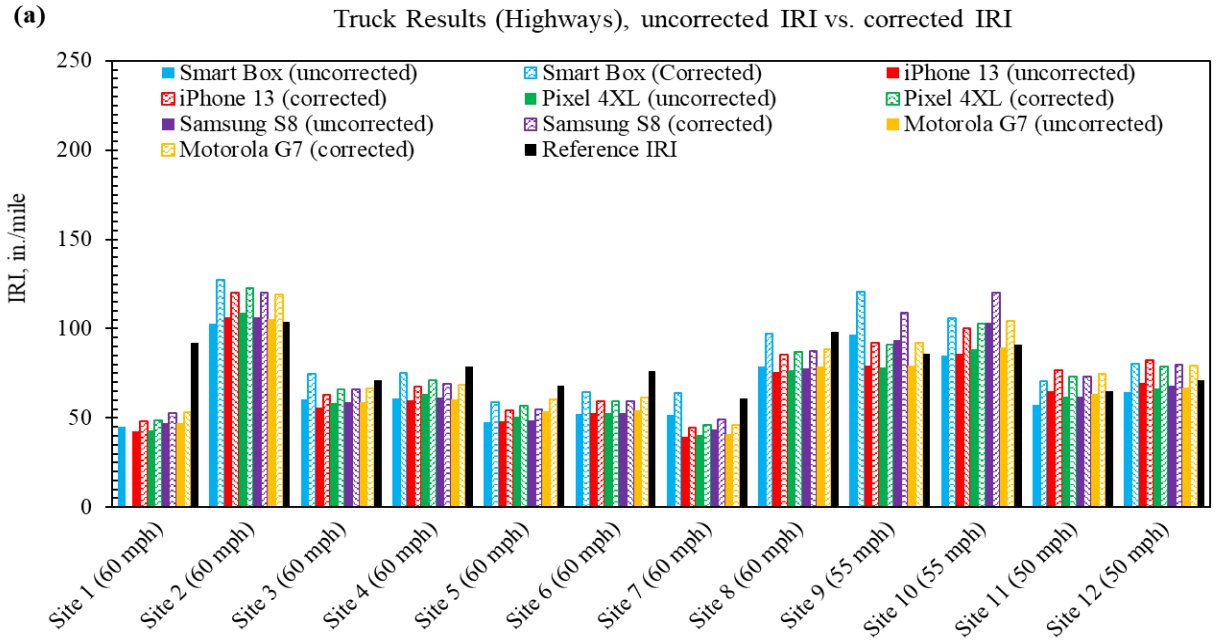
**Table 8. Speed correction parameters**

| | Speed Correction Parameters | Smart Box | iPhone 13 | Pixel 4XL | Samsung S8 | Motorola G7 | Generalized Smartphone Parameters |
|---|---|---|---|---|---|---|---|
| Truck | a | 0.000307 | 0.000399 | 0.000307 | 0.000312 | 0.000358 | 0.000358 |
| | b | -0.032783 | -0.038764 | -0.030086 | -0.032374 | -0.036762 | -0.036762 |
| | c | 0.697110 | 0.736953 | 0.572365 | 0.688894 | 0.795502 | 0.795502 |
| Car | a | 0.000258 | 0.000247 | 0.000326 | 0.000330 | 0.000332 | 0.000309 |
| | b | -0.030096 | -0.023381 | -0.032255 | -0.033532 | -0.032245 | -0.030353 |
| | c | 0.711145 | 0.324924 | 0.608377 | 0.634059 | 0.579017 | 0.536594 |

## Field Data Validation

The protocol for gathering field data was extensively outlined in the opening section of this chapter. To summarize, 24 sites—including highways, county roads, and city streets—were chosen for the purposes of validation. The study tested two types of vehicles, five different devices, and a range of driving speeds as variables. Reference IRI values for these 24 sites were collected using the Class 1 HSP.

Figure 22 and Figure 23 display the uncorrected and corrected IRI results as measured for trucks and cars, respectively. As observed in these figures, the uncorrected IRI values generally fall below the reference IRI values. While implementing speed correction adjustments elevated these IRI values, thereby enhancing the precision of the measurements, it is noteworthy that at certain locations (e.g., Site 24), the IRI values were overcorrected at driving speeds of 40 and 45 mph. This overcorrection is because the second-order polynomial correction function is more aggressive in its adjustments between speeds of 40 and 60 mph but exhibits less correction at speeds below 40 mph and above 60 mph.



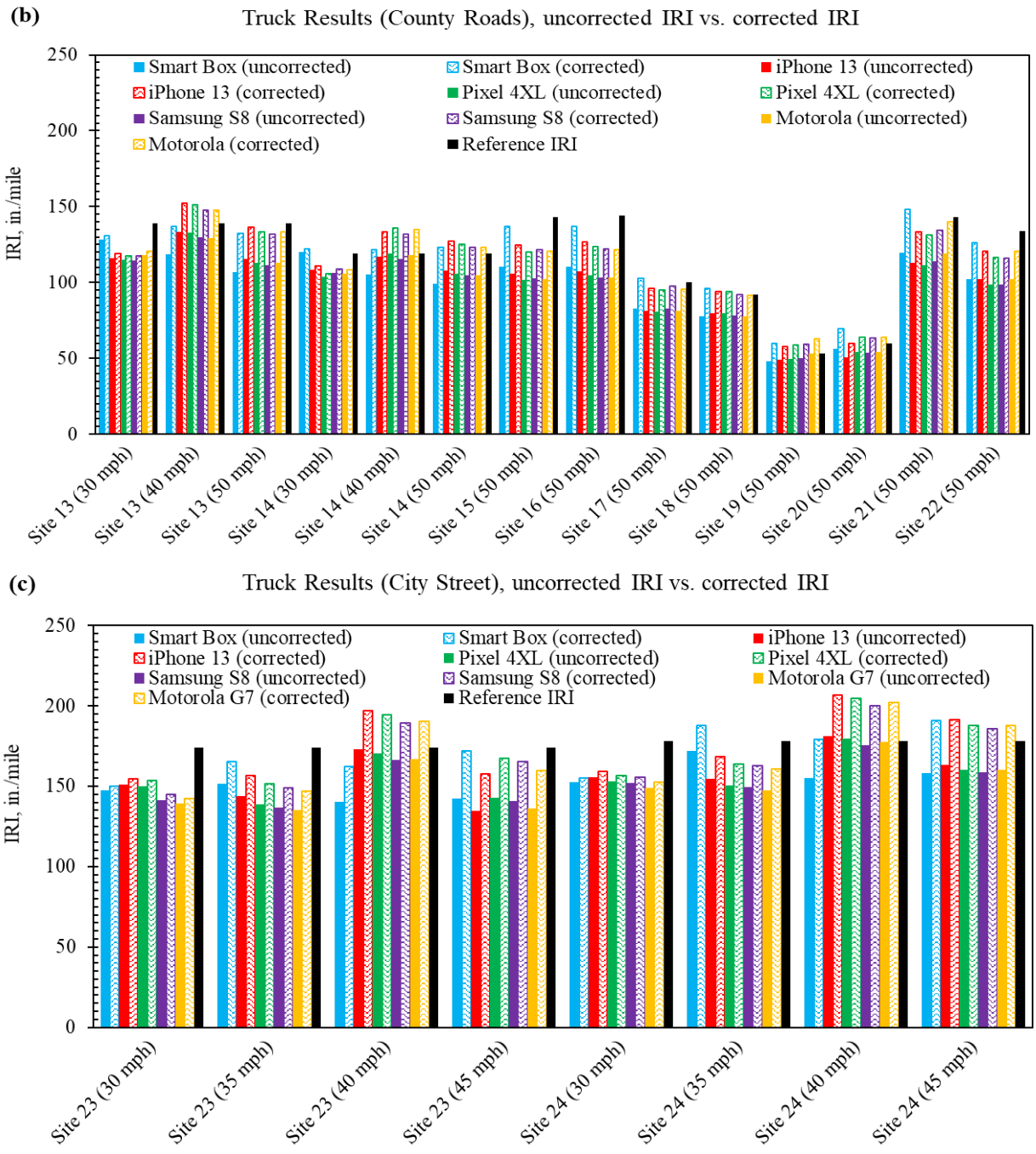
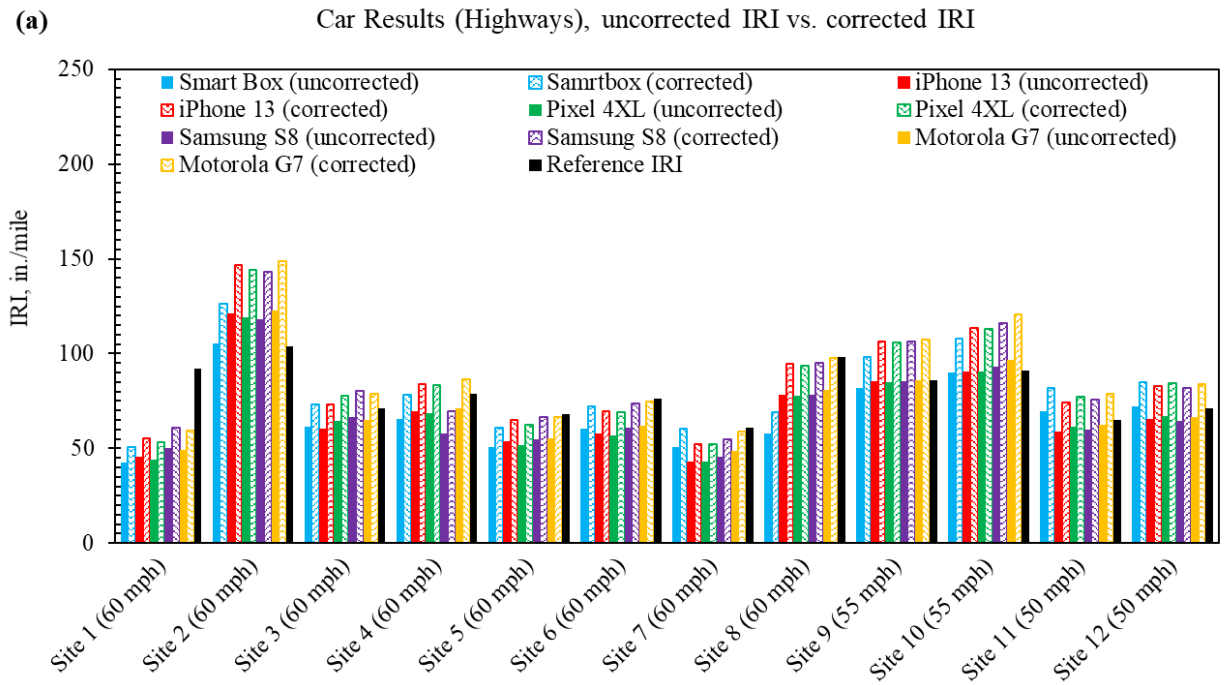
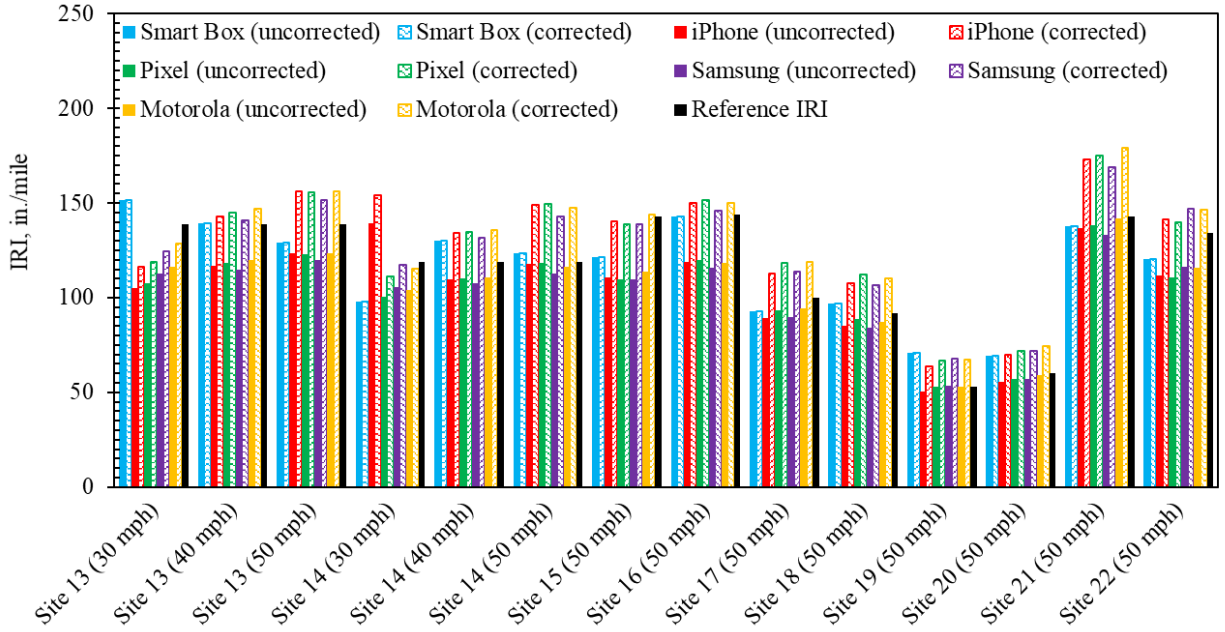



**Figure 22. Uncorrected and corrected IRI results measured by truck on (a) highways, (b) county roads, and (c) city streets**

**(a)**


Car Results (Highways), uncorrected IRI vs. corrected IRI
Smart Box (uncorrected)
Samrtbox (corrected)
iPhone 13 (uncorrected)
iPhone 13 (corrected)
Pixel 4XL (uncorrected)
Pixel 4XL (corrected)
Samsung S8 (uncorrected)
Samsung S8 (corrected)
Motorola G7 (uncorrected)
Motorola G7 (corrected)
Reference IRI
IRI, in./mile
250
200
150
100
50
0
Site 1 (60 mph)
Site 2 (60 mph)
Site 3 (60 mph)
Site 4 (60 mph)
Site 5 (60 mph)
Site 6 (60 mph)
Site 7 (60 mph)
Site 8 (60 mph)
Site 9 (55 mph)
Site 10 (55 mph)
Site 11 (50 mph)
Site 12 (50 mph)

**(b)**


Car Results (County Roads), uncorrected IRI vs. corrected IRI
Smart Box (uncorrected)
Smart Box (corrected)
iPhone (uncorrected)
iPhone (corrected)
Pixel (uncorrected)
Pixel (corrected)
Samsung (uncorrected)
Samsung (corrected)
Motorola (uncorrected)
Motorola (corrected)
Reference IRI
IRI, in./mile
250
200
150
100
50
0
Site 13 (30 mph)
Site 13 (40 mph)
Site 13 (50 mph)
Site 14 (30 mph)
Site 14 (40 mph)
Site 14 (50 mph)
Site 15 (50 mph)
Site 16 (50 mph)
Site 17 (50 mph)
Site 18 (50 mph)
Site 19 (50 mph)
Site 20 (50 mph)
Site 21 (50 mph)
Site 22 (50 mph)

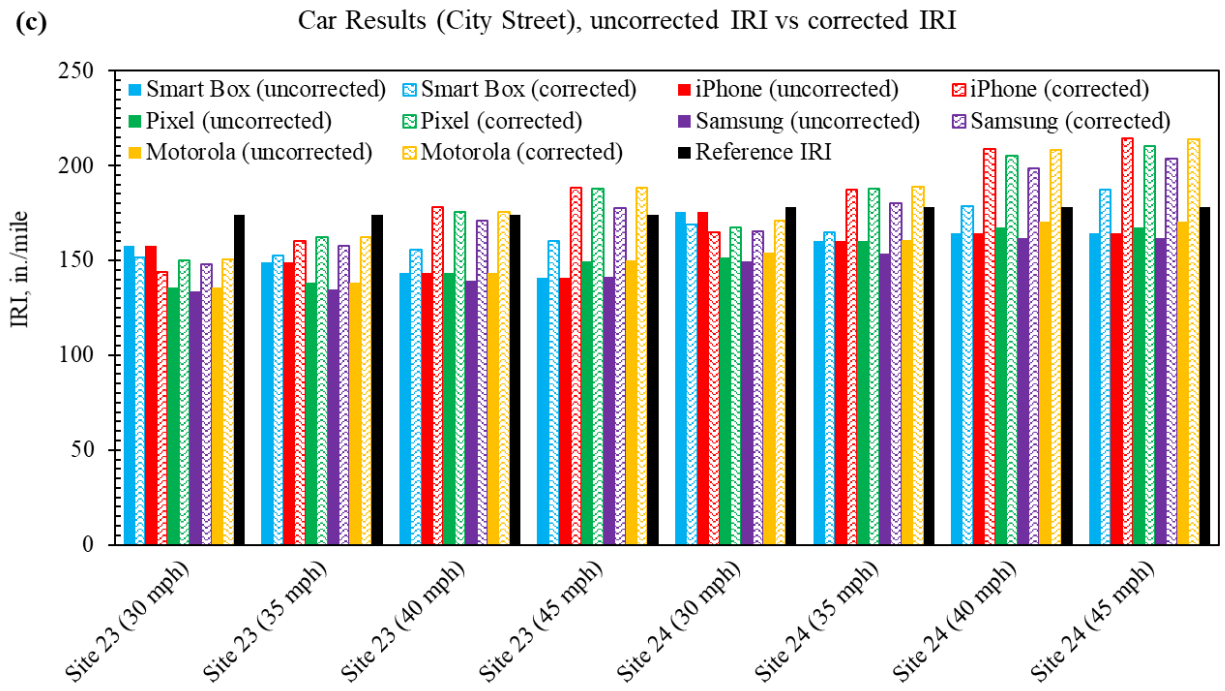



**Figure 23. Uncorrected and corrected IRI results measured by mid-size car on (a) highways, (b) county roads, and (c) city streets**

Figure 24 and Figure 25 illustrate the impact of speed correction adjustments on the measurement accuracy of each device. Both uncorrected and corrected IRI values are plotted along a 45-degree line of equality (LOE) to evaluate their respective $R^2$ values; a higher $R^2$ value is indicative of greater measurement accuracy. As evident in the data plots in Figure 24 and Figure 25, the corrected IRI data points are more closely aligned with the line of equality, indicating that the corrected IRI data set yields a higher $R^2$ value.

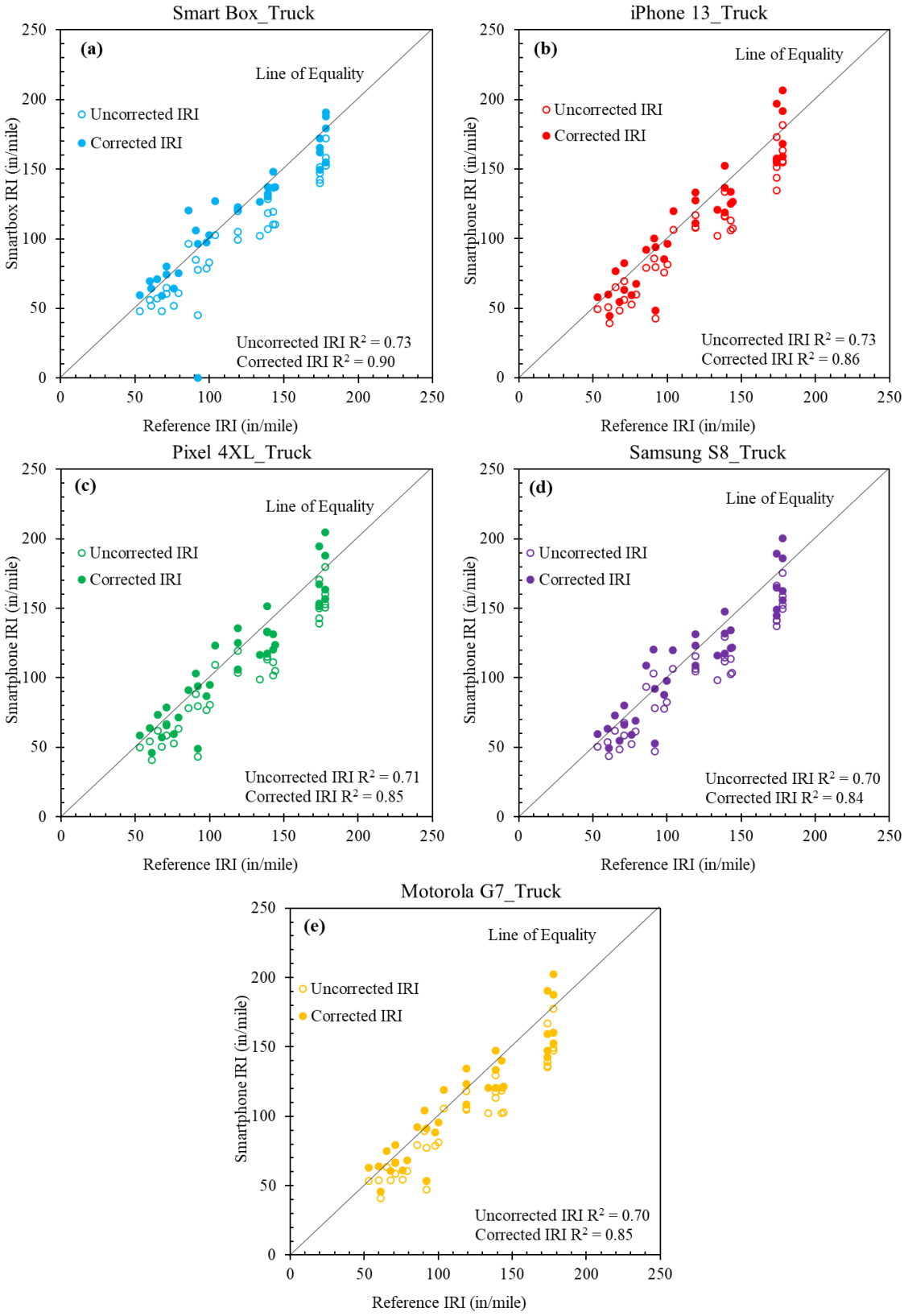



**Figure 24. Uncorrected and corrected IRI results measured by (a) smart box, (b) iPhone 13, (c) Pixel 4XL, (d) Samsung S8, and (e) Motorola G7 in a truck**

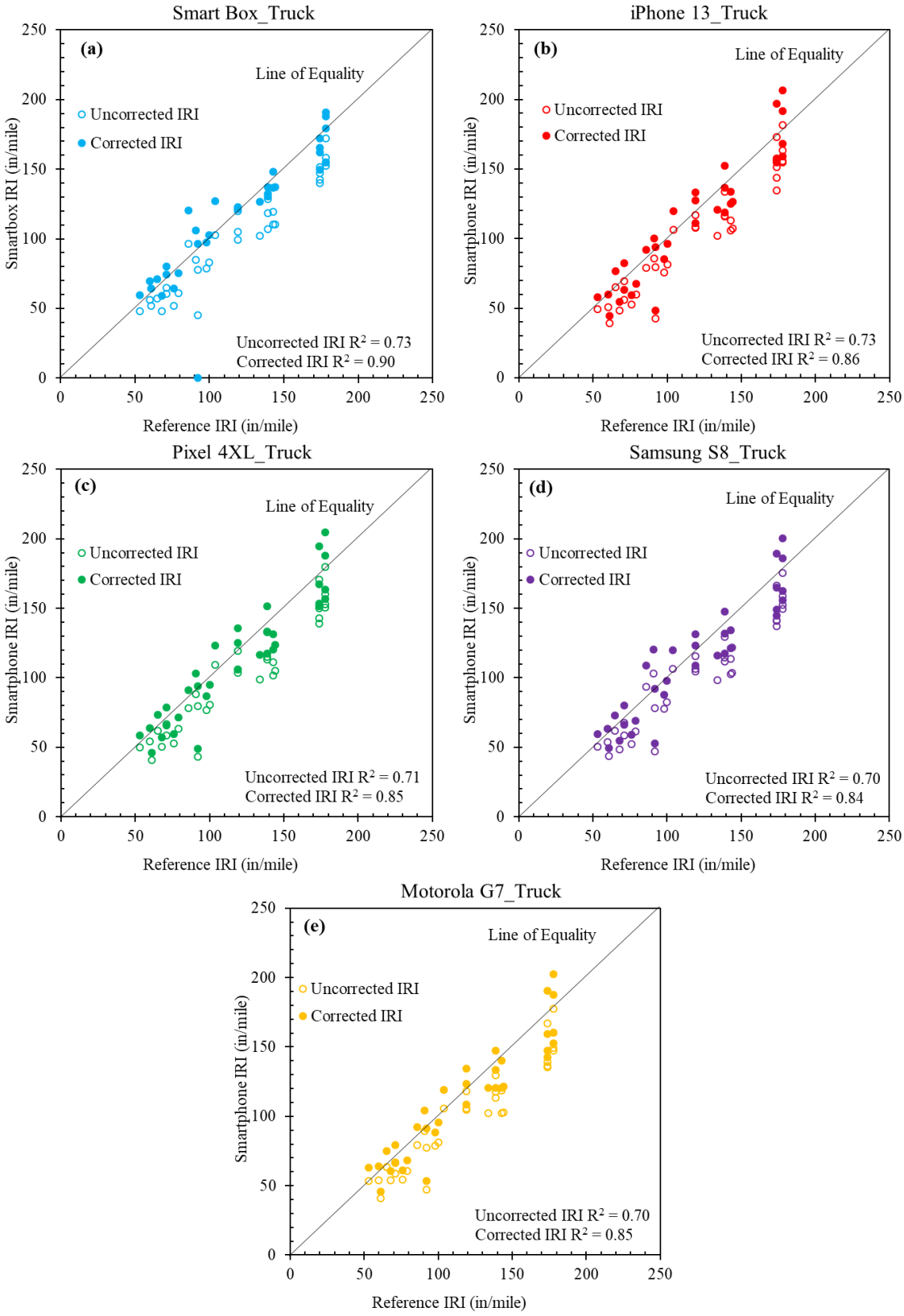



**Figure 25. Uncorrected and corrected IRI results measured by (a) smart box, (b) iPhone 13, (c) Pixel 4XL, (d) Samsung S8, and (e) Motorola G7 in a mid-size car**

Table 9 collates the $R^2$ values for different types of vehicles and devices. Notably, the iPhone 13 mounted in the truck and the Samsung S8 mounted in the mid-size car registered the highest $R^2$ values. Table 10 presents the average absolute percentage differences between the reference IRI values and those measured by the five smart devices. The results reveal that the smart box has the lowest percentage difference when used in trucks, while the Pixel 4XL has the lowest percentage difference when used in mid-size cars.

**Table 9. Line of equality $R^2$ values before and after correction**

| Vehicle | Correction | Smart Box | iPhone 13 | Pixel 4XL | Samsung S8 | Motorola G7 |
|---|---|---|---|---|---|---|
| Truck | Uncorrected | 0.73 | 0.73 | 0.71 | 0.70 | 0.70 |
| | Corrected | 0.78 | 0.86 | 0.85 | 0.84 | 0.85 |
| Car | Uncorrected | 0.81 | 0.75 | 0.76 | 0.74 | 0.79 |
| | Corrected | 0.82 | 0.77 | 0.80 | 0.85 | 0.80 |

**Table 10. Absolute percentage difference to reference IRI before and after correction**

| Vehicle | Correction | Smart Box | iPhone 13 | Pixel 4XL | Samsung S8 | Motorola G7 |
|---|---|---|---|---|---|---|
| Truck | Uncorrected | 16% | 16% | 16% | 16% | 16% |
| | Corrected | 11% | 12% | 12% | 13% | 12% |
| Car | Uncorrected | 15% | 14% | 14% | 14% | 14% |
| | Corrected | 13% | 13% | 11% | 12% | 13% |

In summary, all devices exhibited good accuracy in their IRI measurements, independent of vehicle type. The comprehensive field testing successfully validated that both the developed CyRoads smartphone app and the smart box are capable of collecting reliable pavement roughness data. Compared to traditional surveying techniques, mobile devices, and smart boxes are more cost-effective and allow for simpler data collection procedures. With the advantages that smartphones and smart boxes offer, LPAs could survey the roughness conditions of local roadways on an annual basis and develop more effective strategies for preserving pavement systems.

# CHAPTER 7: DEVELOPMENT OF A SMARTPHONE-BASED APP AND ALGORITHMS FOR DETECTING AND MEASURING ROAD SURFACE DISTRESS USING SMARTPHONE-COLLECTED IMAGES

## Introduction

Computer vision is a field of artificial intelligence that trains computers to understand and make sense of visual data, mainly images and videos. Videos are comprised of several images per second that are themselves comprised of sets of pixels representing different luminance values. In the last few decades, deep-learning algorithms, an area of study under computer vision, have been widely used in detecting and locating objects inside images and videos. The foundation of deep-learning object detection algorithms lies in the use of convolutional neural networks (CNNs), a methodology that has revolutionized computer vision. In a typical object detection model, like the one shown in Figure 26, the object detection model detects each object in an image, draws a bounding box around it, and provides it with a label. In this study, a system was designed to utilize a trained object detection model for detecting pavement distress and saving GPS-tagged images.

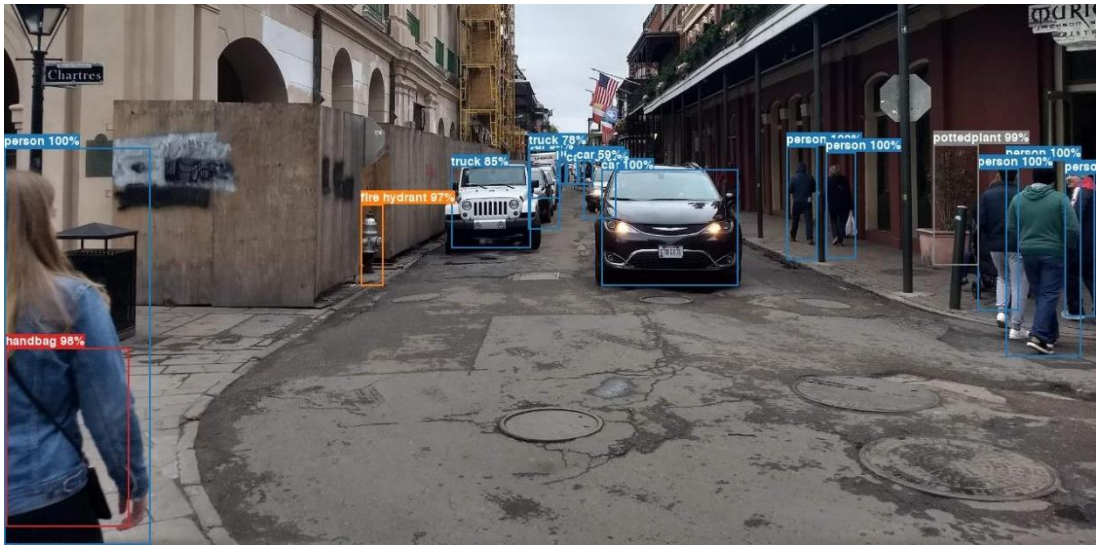



**Figure 26. Example of object detection using a deep-learning model**

## Methodology

As shown in Figure 27, the system developed in this study operates by training a model on a server and using a mobile application to run the trained model. The server's primary function is to train an object detection model and process it to make it compatible with mobile devices. Once a model is trained in this way, the object detection model is exported as a TFLite file compatible with the mobile application. The TFLite file can be downloaded onto the mobile device to replace the previously existing file and initiate distress detection. Each frame from the camera feed is passed to the model for inference and creation of bounding boxes, and the images are saved to the mobile device's internal storage.

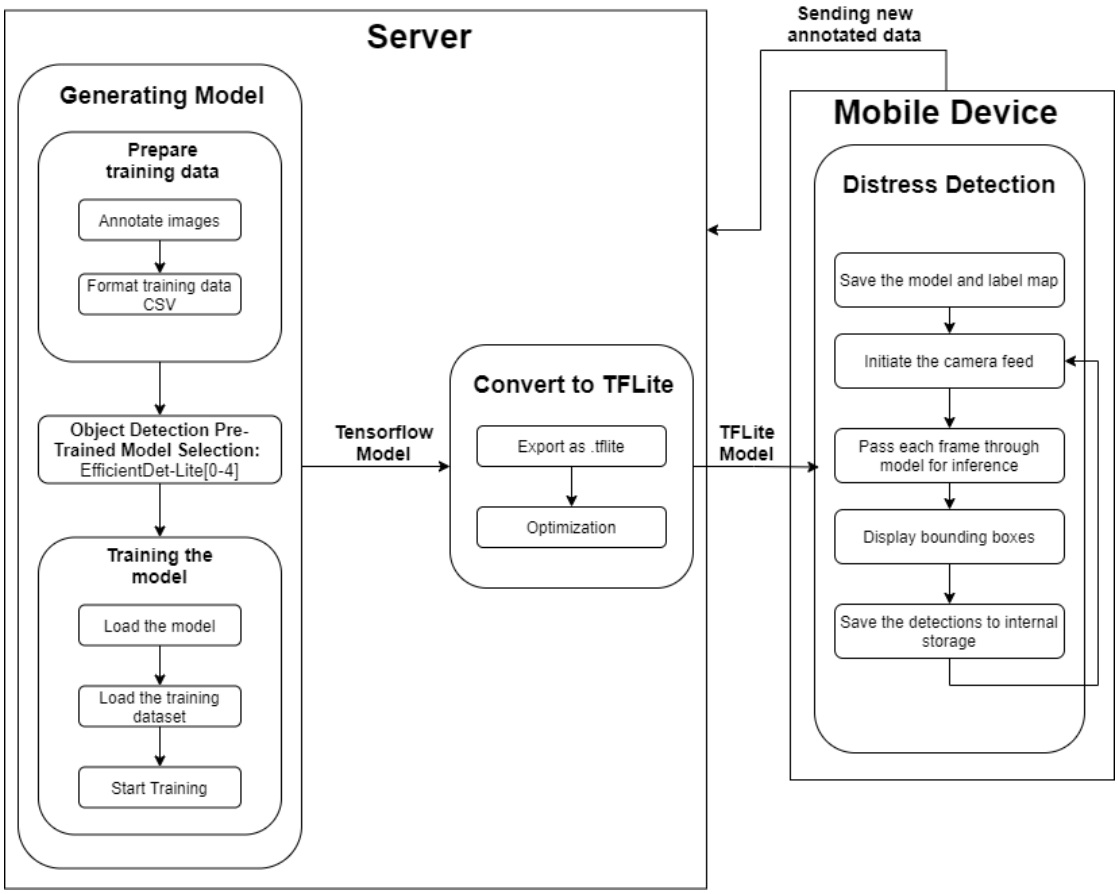



**Figure 27. Workflow of pavement distress detection using a smartphone**

*Training Data Set*

The Global Road Damage Detection Challenge (GRDDC) introduced a data set, shown in Figure 28, that consists of a total of 26,620 pavement distress images from three countries: Japan, the Czech Republic, and India (Arya et al. 2020). The data set is organized into three groups: training, test1, and test2, whose distributions are shown in Figure 28. This data set was created as part of a deep learning model development challenge where contenders trained their models using the training data set and then deployed their models to detect pavement distress in the test data sets. The training data set images were used in this research because only the images in that category had associated annotation files. The data set is comprised of four categories of pavement distress:

- D00: Longitudinal Crack
- D10: Transverse Crack
- D20: Alligator Crack
- D40: Pothole

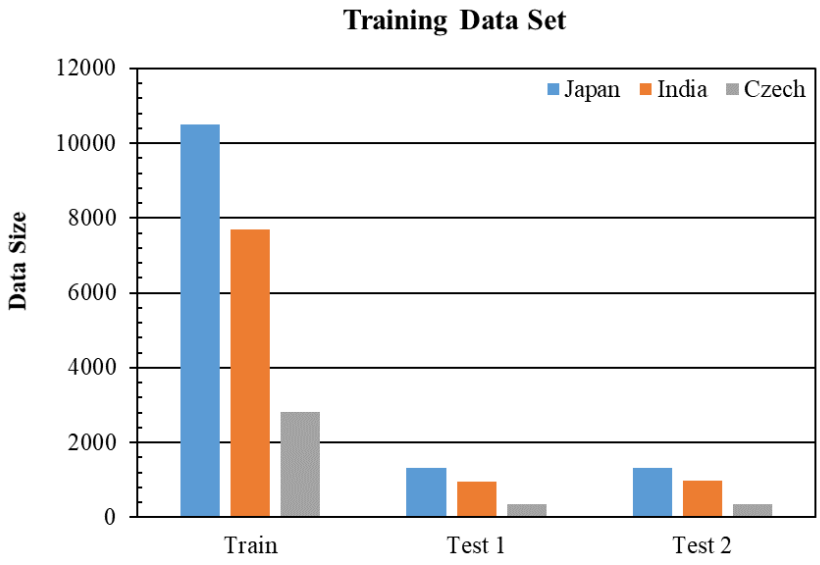



**Figure 28. Training data set details**

The data set was subdivided again into a new training data set and a new test data set for training the deep learning models in this research. Each annotation file contains a distress code and the location information for the distress. Figure 29 shows an example of the data annotation process, with LabelImg used for the development of this data set for the GRDDC. In this example, the position of the rectangular box enclosing the crack was noted using the box's top-left and bottom-right corner locations. A "D10" distress code, which represents transverse cracks, was also provided. The information was saved in a text file with the same name as the image.

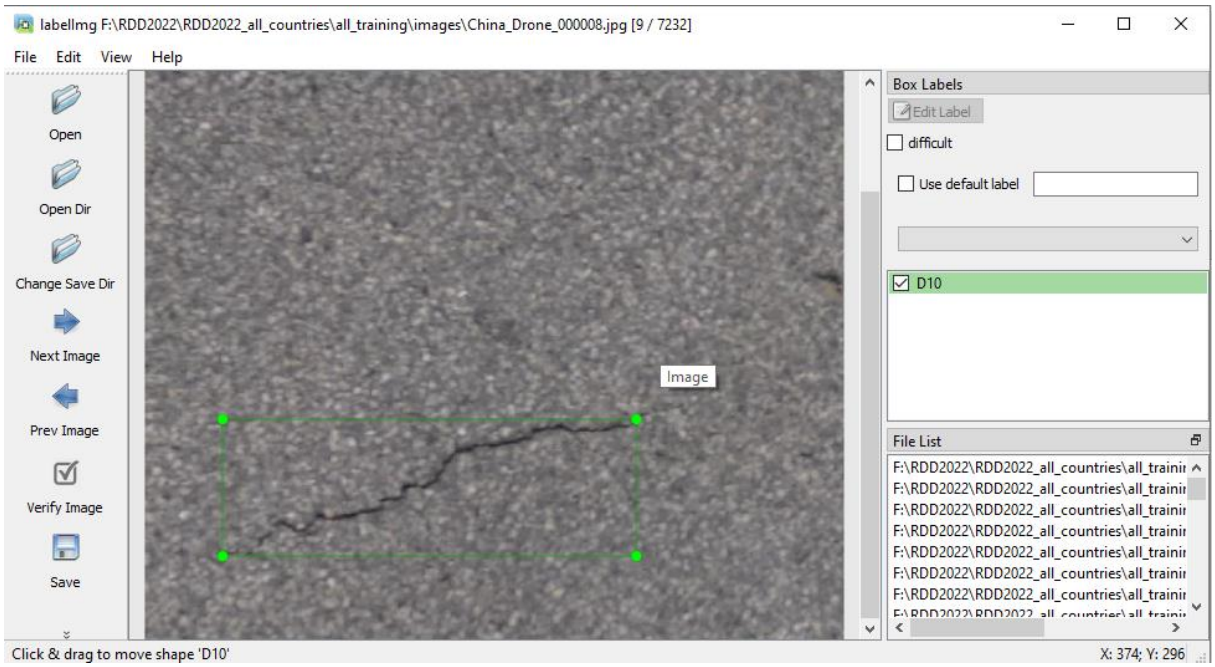



**Figure 29. Data annotation using LabelImg software**

*Object Detection Model Training*

Because the goal of this study was to develop a model that can run smoothly on a smartphone, this study opted for a lightweight object detection model. Some of the models available for training are listed in Table 11. EfficientDet 320 was determined to be a viable option based on its processing time and accuracy. YOLOv4 Tiny and two variations of SSD MobileNet V2 (MobileNet 320 and SSD MobileNet 640) were also considered (Bochkovskiy et al. 2020, Yu et al. 2021).

**Table 11. Speed and accuracy of different object detection models**

| Model name | Required time to process per image (ms) | Mean average precision |
|---|---|---|
| CenterNet HourGlass104 512x512 | 70 | 41.9 |
| CenterNet Resnet50 V2 512x512 | 27 | 29.5 |
| EfficientDet D0 512x512 | 39 | 33.6 |
| EfficientDet D1 640x640 | 54 | 38.4 |
| Faster R-CNN Inception ResNet V2 640x640 | 206 | 37.7 |
| Faster R-CNN ResNet101 V1 640x640 | 55 | 31.8 |
| SSD MobileNet V1 FPN 640x640 | 48 | 29.1 |
| SSD MobileNet V2 FPNLite 640x640 | 39 | 28.2 |
| SSD ResNet101 V1 FPN 640x640 (RetinaNet101) | 57 | 35.6 |

Source: Yu et al. 2021

Training an object detector is a computationally intensive and time-consuming task. Due to hardware constraints, Google Colab Pro, which provides access both to a GPU and a high amount of RAM, was used to train the chosen object detection model. The data set for training was stored in Google Drive and accessed through a Colab library. After the object detectors were trained, the models were saved directly to a folder in Google Drive. The objective was to train a deep-learning model and create a mobile application that detects and locates road cracks.

Training for the two SSD MobileNet V2 model variants began by cloning the publicly available folder containing the TensorFlow models in Git (Yu et al. 2020). Image and annotation files were segregated into training and testing folders and were used to generate a TensorFlow record file that the models would use for training. Pretrained weights were downloaded for each model via an associated configuration file. After necessary changes were made to the configuration file, each model was trained using the data set. Model training adjusted the bias and weight of each layer of the neural network. The performance of each model in detecting road defects was also evaluated during this training process. This performance is known as total loss, which is higher at the onset and lessens as training progresses. Once this total loss had reached a plateau, the training was stopped, and the weight and bias files were used to generate a ready-to-use trained model. A sample of the training process is given in Figure 30. Once the model was trained properly, a TFLite model was generated using the output of the training process.

```
INFO:tensorflow:Step 100 per-step time 0.999s loss=1.369
I1015 11:37:32.545120 140096613611392 model_lib_v2.py:683] Step 100 per-step time 0.999s loss=1.369
INFO:tensorflow:Step 200 per-step time 0.399s loss=1.129
I1015 11:38:12.395044 140096613611392 model_lib_v2.py:683] Step 200 per-step time 0.399s loss=1.129
INFO:tensorflow:Step 300 per-step time 0.399s loss=1.063
I1015 11:38:52.302342 140096613611392 model_lib_v2.py:683] Step 300 per-step time 0.399s loss=1.063
INFO:tensorflow:Step 400 per-step time 0.398s loss=1.164
I1015 11:39:32.114210 140096613611392 model_lib_v2.py:683] Step 400 per-step time 0.398s loss=1.164
INFO:tensorflow:Step 500 per-step time 0.397s loss=1.243
I1015 11:40:11.780810 140096613611392 model_lib_v2.py:683] Step 500 per-step time 0.397s loss=1.243
INFO:tensorflow:Step 600 per-step time 0.398s loss=0.982
I1015 11:40:51.602434 140096613611392 model_lib_v2.py:683] Step 600 per-step time 0.398s loss=0.982
INFO:tensorflow:Step 700 per-step time 0.399s loss=1.026
I1015 11:41:31.461519 140096613611392 model_lib_v2.py:683] Step 700 per-step time 0.399s loss=1.026
INFO:tensorflow:Step 800 per-step time 0.397s loss=1.112
I1015 11:42:11.148232 140096613611392 model_lib_v2.py:683] Step 800 per-step time 0.397s loss=1.112
INFO:tensorflow:Step 900 per-step time 0.398s loss=1.002
I1015 11:42:50.992996 140096613611392 model_lib_v2.py:683] Step 900 per-step time 0.398s loss=1.002
INFO:tensorflow:Step 1000 per-step time 0.396s loss=1.124
I1015 11:43:30.605527 140096613611392 model_lib_v2.py:683] Step 1000 per-step time 0.396s loss=1.124
INFO:tensorflow:Step 1100 per-step time 0.404s loss=1.031
```

**Figure 30. Training progress in command prompt/terminal**

### Mobile Application

The mobile application was developed using a cross-platform framework, Flutter, that can be installed on both Android and iOS mobile devices. However, the application was only developed for and tested on an Android device, as described in the following section. An overview of the application's architecture is shown in Figure 31. The interface of the application consists of three buttons in the body, "Start Detection," "Images," and "Capture Images," and a "Model" button in the application bar. The buttons have the following functions:

- **Start Detection**. This button navigates to a screen containing the camera feed that passes each frame through the trained model to obtain the detections. Whenever a detection occurs, its bounding boxes are displayed, and this information is saved along with the GPS-tagged image corresponding to the detection.
- **Images**. This button navigates to a screen that consists of two tabs: Detections and Captured Images. The Detections tab, as the name suggests, corresponds to all of the images that have been saved when performing road distress detection. The Captured Images tab corresponds to images saved by clicking on the "Capture Images" button. These images can be annotated to serve as new training data for generating a new model.
- **Capture Images**. This button is used to save images continuously based on GPS location, i.e., only one image is captured per unique GPS location to avoid redundant images.
- **Model**. This button routes the user to a page that consists of "Train" and "Download" buttons. The "Train" button triggers a Train API in the server that initiates training of the model with the recently uploaded images along with annotations. Once the training is complete, a new model can be downloaded from the server by clicking on the "Download" button. If multiple models are available, a list will be shown from which the user can select.

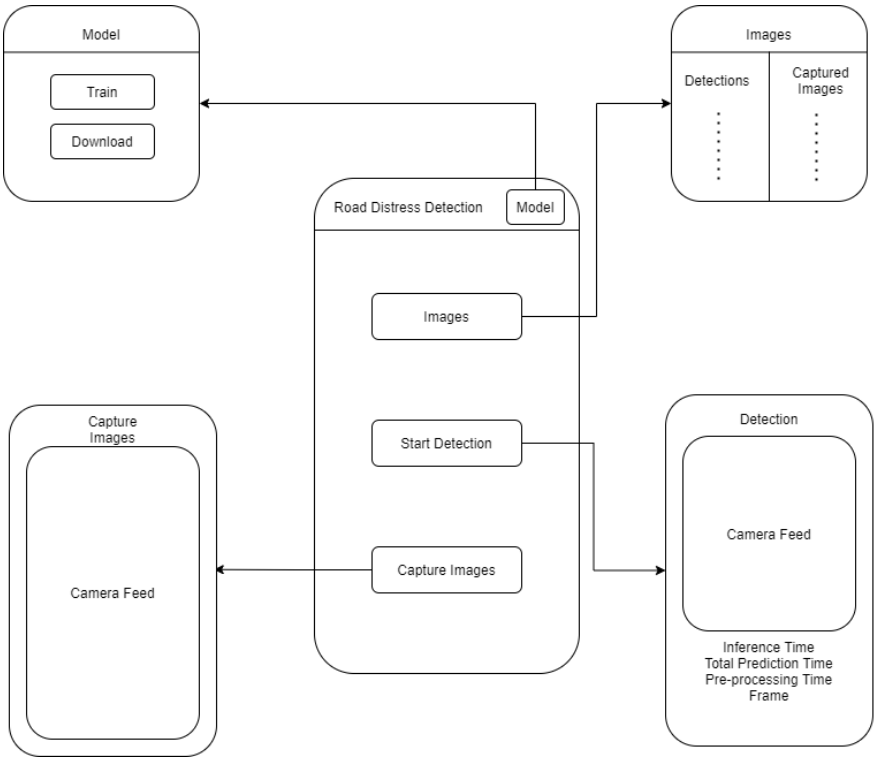



**Figure 31. Application interface**

## Evaluation

During evaluation, the metrics average precision and inference time (i.e., time spent by the model to infer distress from an image) were utilized to compare the different object detection models. The study also used speed versus throughput (i.e., the number of images that can be processed per second as the speed varies) as another metric to evaluate the performance of the developed tool. The definitions of the evaluation metrics are provided in Table 12.

**Table 12. Object detection model evaluation metrics**

| Metric | Description |
|---|---|
| True positive (tp) ↑ | A defect is detected and is present in the image |
| False positive (fp) ↓ | A defect is detected but is not present in the image |
| False negative (fn) ↓ | A defect is not detected but is present in the image |
| Recall ↑ (tp/tp+fn) | Percentage of correctly matched detections to ground-truth detections |
| Precision ↑ (tp/tp+fp) | Percentage of correctly matched detections to total detections |

Equation (8) shows the calculation of precision. Average precision, the percentage of the model's predictions that are correct, is an important metric for measuring the accuracy of an object detection model.

$$\text{Precision} = \frac{true\ positive}{true\ positive + false\ positive} \quad (8)$$

Table 13 compares the three models (SSD MobileNet 320, SSD MobileNet 640, and EfficientDet 320) in terms of accuracy. As can be seen in Table 13, the EfficientDet model delivers higher accuracy than the other two. In practice, it is more important to know of the existence and the rough locations of pavement distress than to accurately pinpoint a problematic spot. Considering the tradeoffs between accuracy, completeness, and timeliness in detecting distresses, the EfficientDet 320 model was found to be the most feasible among the three.

**Table 13. Map and inference comparisons**

| Model | Map | Inference Time (ms) |
|---|---|---|
| SSD MobileNet 320 | 11.97 | 100-130 |
| SSD MobileNet 640 | 15.50 | 250-300 |
| EfficientDet 320 | 17.04 | 40-70 |

Evaluations were conducted using a Samsung Galaxy S20 FE 5G with 6GB RAM, a Qualcomm SM8250 Snapdragon 865 5G octa-core processor, an Adreno 650 GPU, and the Android 10 Operating System. Table 14 shows an evaluation of the throughput of the developed tool (based on the EfficientDet 320 model) at varying vehicle speeds, showing that the number of images that are captured and processed decreases as the vehicle speed increases. Also, provided that the vehicle speed is not too high (i.e., 55 mph or lower), all or most of the captured images can be processed in real-time. Note that, in practice, a vehicle surveying road conditions should not run at a very high speed; otherwise, it may be unable to capture enough images for processing.

**Table 14. Speed versus throughput comparison**

| Speed (mph) | Captured Images Count | Detection Count |
|---|---|---|
| 25 | 16 | 16 |
| 30 | 15 | 13 |
| 35 | 12 | 12 |
| 40 | 10 | 10 |
| 45 | 9 | 9 |
| 50 | 7 | 7 |
| 55 | 7 | 6 |

## Results and Discussion

In this section, the processing capabilities of different object detection models are demonstrated, and their levels of performance in predicting different types of pavement distress are compared. The test images were passed through each model, followed by a visual comparison with respect to performance. Ground truth annotations for these test images were also available. Although multiple images were used to evaluate the model, only a few are presented for convenient explanation and comprehension.

Each image below is divided into two halves; the left half corresponds to detection by the model, and the right half is the ground truth. Overall, SSD 640 and EfficientDet detected longitudinal cracking, alligator cracking, and potholes. Longitudinal cracking was successfully detected by all three models in most of the test images, while alligator cracking and potholes were detected by SSD 640 and EfficientDet but were not satisfactorily detected by SSD 320. While transverse cracking was usually missed when compared to the ground truth, the detection was enhanced in images where the transverse crack was clearly visible.

The SSD 640 model provided highly accurate detections relatively close to the ground truth. In Figure 32, the detections made by the model were true positives that appeared in the ground truth data. In Figure 33, the model correctly detected longitudinal cracks, but a pothole detected by the model was not present in the ground truth data. In Figure 34, the model detected a pothole correctly but also detected a false positive alligator crack.

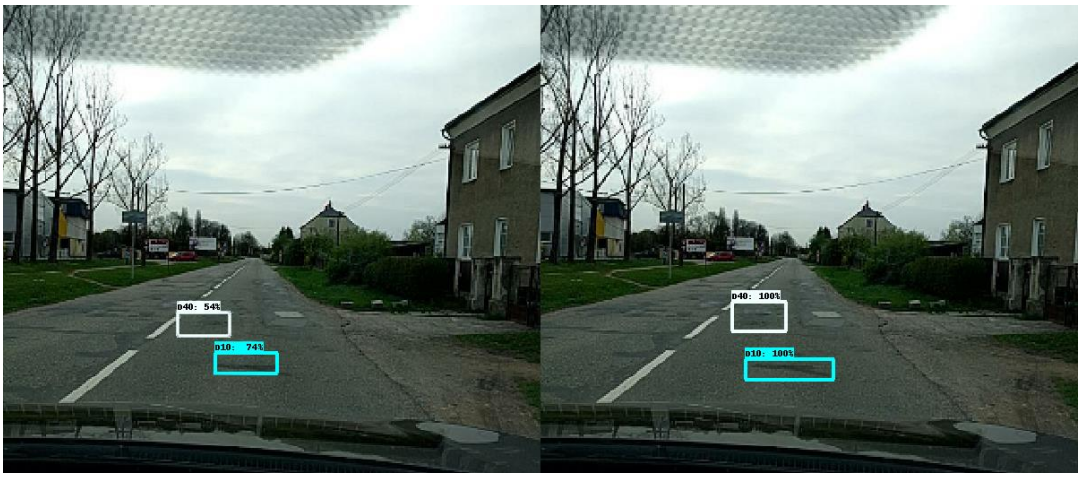



Image source: Arya et al. 2020

**Figure 32. Pothole and transverse crack distress detection by SSD 640**

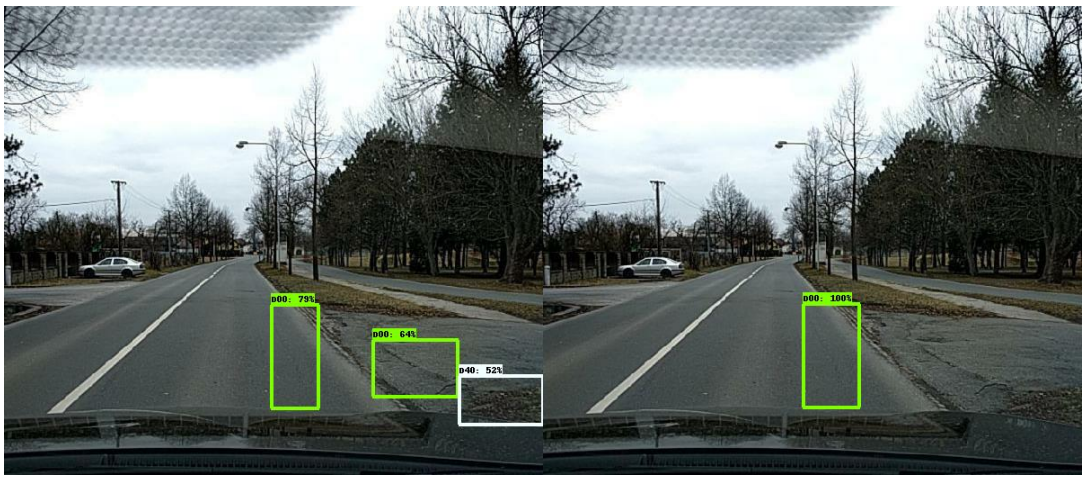



Image source: Arya et al. 2020

**Figure 33. Longitudinal crack and pothole distress detection by SSD 640**


Image source: Arya et al. 2020

**Figure 34. Pothole and alligator crack distress detection by SSD 640**

The outputs from the SSD 320 model were not as accurate as those of the other two models. SSD 320 was able to detect a longitudinal crack (Figure 35) and a transverse crack (Figure 36) correctly but was not able to detect other distresses.

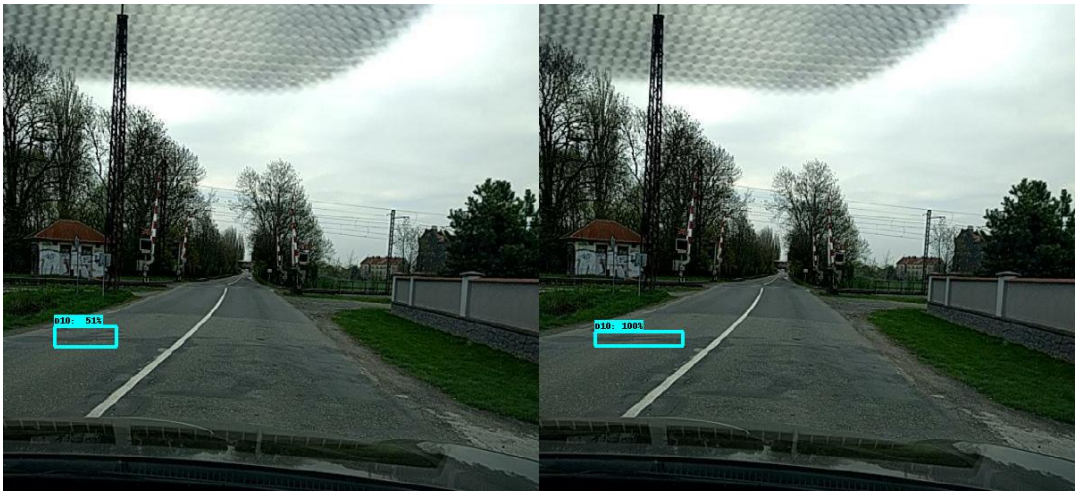



Image source: Arya et al. 2020

**Figure 35. Transverse crack distress detection by SSD 320**


Image source: Arya et al. 2020

**Figure 36. Longitudinal crack distress detection by SSD 320**

While the EfficientDet model succeeded in detecting all four types of distress (Figure 37), in some of the images, the EfficientDet model was not able to match the ground truth data.

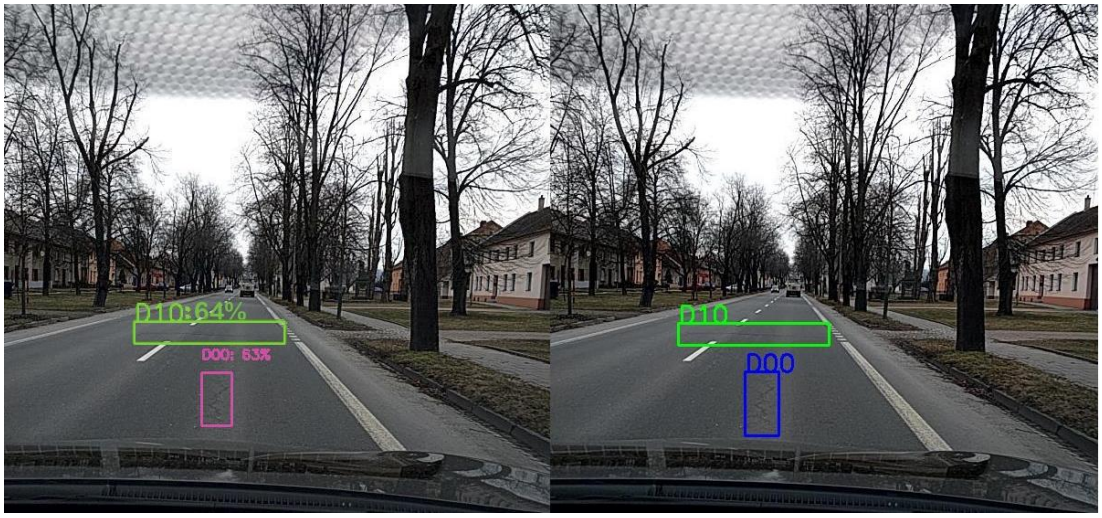



Image source: Arya et al. 2020

**Figure 37. Longitudinal crack and transverse crack distress detection by EfficientDet**

Since the mobile application is packaged with an existing trained model and data set, users can immediately perform distress detection, and engineers can further revise the resulting images for analysis and revisit the actual site for a survey. Another advantage of this tool is that it speeds up the survey process by reducing the time ordinarily used to individually assess distress. The tool also has the functionality to capture new images and train a new model based on the annotated images, allowing engineers to include new distresses in their data set. Efforts were also made to compare different object detection models based on inference speed and accuracy.

## Summary

This study developed an Android-based mobile application to demonstrate the capabilities of deep-learning object detection models in detecting road distress. The application was developed to detect road cracking and potholes in real-time video and capture images of the distress. The crowdsensing-based GRDDC data set was used to train three off-the-shelf object detection models: SSD MobileNet 320, SSD MobileNet 640, and EfficientDet 320. Because the computational power of mobile devices is limited, these models were selected from a wide range of off-the-shelf object detection models because of their low latency and high mean average precision.

The results showed that adequately trained deep-learning models could detect road cracking and potholes with a mean average precision of up to 17.04 and an image processing time as low as 40 milliseconds. The object detection models also demonstrated their efficacy in detecting road distress from images collected from different geographic locations. Since this research was mainly performed to introduce a prototype Android application, the major effort involved building a bare-bones mobile application for detecting road distress in real-time, capturing and annotating images using the application, and detecting distress in previously captured images. Further improvements to be considered would include optimizing the algorithm to remove latency in road distress detection when the vehicle's speed is 55 mph or more, ensuring that a certain number of images from the video feed are processed during real-time detection, training the model to detect new distresses, generating interactive GIS maps of the road defects using GPS data, and other improvements.

## Limitations and Future Scope

This mobile application for distress detection has some limitations that represent opportunities for future work. These can be summarized as follows:

- Since the object detection model tries to process each video frame captured through the mobile phone in real-time, only a small number of images from fast-moving vehicles are captured and processed. Therefore, there is a desire to systematically collect a specific number of images based on the vehicle speed and process them on the mobile phone.
- The model provides no quantification of the cracks detected. Calculating the number and length of cracks would be desirable to help get a clear idea of pavement condition and any required maintenance. Developing the capability to collect detailed information regarding crack type and severity could be a valuable area for further research.
- The model begins to lag above a speed of 55 mph. A distributed approach could be used wherein alternate frames are processed on a different mobile device.
- The model can be improved by introducing new types of distress as well as providing more images of the types of distress already included in the model.
- For better visualization, the geolocation data of the distresses stored in the server could be used to generate interactive GIS maps.
- An interface for accessing the backend server would facilitate efficient maintenance of the stored images and their associated data.

# CHAPTER 8: CONCLUSIONS AND RECOMMENDATIONS

## General Summary

Pavement roughness serves as a vital indicator of pavement performance and is crucial in shaping maintenance and rehabilitation strategies. Since the late 1980s, IRI has been globally recognized as the definitive measure of pavement roughness, but traditional methods for measuring IRI have been costly, somewhat unreliable, and inaccurate, failing to meet the growing demands of LPAs. Consequently, the use of smartphones equipped with arrays of sensors has garnered attention for the potential of these devices to offer cost-effective, reliable, and accurate IRI measurements.

In this study, the ISU research team developed a smartphone-based application, CyRoads, as well as a smart box designed to collect vertical suspension data from moving vehicles. The application is supported by a Python-based computational algorithm for calculating IRI hosted on a cloud server. The algorithm both allows for customized vehicle suspension parameters based on the type of vehicle and incorporates a speed correction function to enhance measurement accuracy. Once raw acceleration data have been collected, they are transmitted to the cloud server for IRI calculation, and the results are reported back to the smartphone or smart box. Comprehensive field testing involving 24 roadway sites validated the accuracy and reliability of the CyRoads app and the smart box, achieving $R^2$ values above 0.77 in all cases.

The study also successfully employed smartphone technology to detect potholes and surface cracks. Images and videos captured by smartphone cameras were processed through a deep-learning model, achieving a mean average precision of up to 17.04. Further development is required to fully realize the application's potential.

In summary, this study led to the successful development of an intelligent system for measuring road roughness that consists of (1) either the CyRoads smartphone app or a smart box and (2) a cloud server. Extensive field testing validated these tools. Iowa counties could substantially benefit from adopting this system for cost-effective pavement assessment.

## Key Findings Regarding the Use of a Smartphone App for Measuring Pavement Roughness

The developed CyRoads smartphone app has been fully tested and validated, and the key findings regarding this app can be summarized as follows:

- The CyRoads app is compatible with both Android and iOS platforms, making it accessible on most smartphones.
- The app's white noise detection feature filters out background vibrations from vehicle engines, thereby enhancing measurement accuracy.
- Users can easily customize project details via the settings, including parameters such as road name, direction, vehicle and mount types, and road and surface conditions.

- Operational tests show that CyRoads quickly processes data; it typically takes just seconds to upload a data file and receive virtually instantaneous results from the server.
- The app provides a comprehensive report that includes metrics such as overall IRI, segment IRI, average speed, measurement duration, distance, and a condition rating based on road type.
- A map view feature allows users to swiftly identify 0.1-mile-long road segments that require attention because they exhibit high surface roughness.
- For users employing different vehicles or smartphones, the app offers a speed correction tool to further refine IRI measurements.
- While field tests revealed that variables such as vehicle type, smartphone model, and speed can influence the IRI values captured by CyRoads, the app's speed correction feature mitigates these variations.
- The speed correction feature performs effectively across all smartphone models, enhancing measurement accuracy.
- The reliability of CyRoads in providing accurate IRI measurements has been substantiated by field tests, with an LOE $R^2$ ranging from 0.77 to 0.85 and an average percent difference of less than 13% compared to reference values from a Class 1 high-speed profilometer.

**Key Findings Regarding the Use of a Smart Box for Measuring Pavement Roughness**

The developed smart box and its specific control app have been fully tested and validated, and the key findings regarding the smart box and its app can be summarized as follows:

- The smart box control app is cross-platform and compatible with both Android and iOS operating systems.
- With a total component cost of approximately $150, the smart box is a more economical option than a smartphone.
- Designed to mirror the system and functionality of CyRoads, the smart box control app interfaces seamlessly with the same cloud server used by CyRoads for raw data processing.
- The speed correction feature used by CyRoads is also effective when used with the smart box, enhancing the accuracy of IRI measurements.
- Field tests have shown that the smart box offers an IRI measurement accuracy comparable to that of CyRoads, boasting an LOE $R^2$ ranging from 0.78 to 0.82. The average percentage difference was found to be less than 13% when compared to reference values from a Class 1 high-speed profilometer.
- The smart box delivers more consistent data measurements than smartphone apps, owing to uniform electronic components that are generally unaffected by system updates such as those to the Raspberry Pi operating system.

**Key Findings Regarding the Use of a Smartphone App for Detecting and Measuring Pavement Distress**

A prototype smartphone app for detecting road surface distress through captured videos and images was developed and evaluated, and the key findings can be summarized as follows:

- While at this stage the distress detection app has been designed specifically for Android phones using the Flutter platform, Flutter is capable of being used to develop an iOS version of the app.
- Field tests confirm the app's ability to employ deep-learning models for real-time identification of potholes and cracks via the processing of video and imagery captured by smartphone cameras.
- The selected deep-learning models were chosen for their low latency and high mean average precision. Tests indicate that well-trained models can identify road imperfections with a mean average precision of up to 17.04 while maintaining a rapid image processing time of just 40 milliseconds.
- The object detection models have proven effective in identifying road distress across images sourced from various geographic locations.

# CHAPTER 9: RECOMMENDATIONS FOR IMPLEMENTATION AND DIRECTIONS FOR FUTURE RESEARCH

Based on the field evaluations of the CyRoads smartphone app, smart box, and distress detection smartphone app developed in this research, several recommendations regarding implementation and future research needs can be advanced.

## Implementation Recommendations for CyRoads Smartphone App

Recommendations regarding the implementation of the CyRoads smartphone app can be summarized as follows:

- **White Noise and Engine Preparation**. Before initiating data collection with CyRoads, users are advised to activate the "White Noise" feature while keeping the vehicle engine running. This should be performed in a safe, level area.
- **Accuracy Testing**. To evaluate the app's accuracy, users should compare its IRI measurements with those obtained from a Class 1 profilometer on a known road section. If the percent difference exceeds 20% or is above 20 in./mile, the speed correction function should be activated to establish new correction parameters.
- **Operating System Updates**. Users must exercise caution during operating system updates. It is recommended that the sampling frequency of raw data be verified after each update to ensure that data collection remains unaffected.
- **Calibration Site Selection**. Choose a road with an IRI range between 80 and 120 in./mile for calibration. When applying the speed correction function, maintain a speed range of 30 to 70 mph to accurately identify new correction parameters.
- **Data Uploading**. For optimal results, upload raw data via a stable and strong WiFi connection.

## Implementation Recommendations for Smart Box

Recommendations regarding the implementation of the smart box can be summarized as follows:

- **Initial Connection and Trial Run**. Prior to data collection, it is strongly recommended that the connection between the control app and the smart box be tested. A trial road test is also advised to confirm that the smart box system is functioning properly and can collect data without issues.
- **Accuracy Testing**. Users should validate the accuracy of the smart box by comparing its IRI measurements on a specific road section against those taken by a Class 1 profilometer. If the percent difference exceeds 15% or is above 15 in./mile, the speed correction function should be activated to establish new correction parameters.
- **Calibration Site and Speed Range**. For optimal speed correction, select a calibration site with an IRI range of 80 to 120 in./mile. Maintain a speed range between 30 and 70 mph while implementing speed correction to accurately identify new correction parameters.

- **Data Uploading**. To ensure successful data transmission, it is recommended that raw data be uploaded using a stable and robust WiFi connection.

**Future Research Recommendations for CyRoads Smartphone App and Smart Box**

Future research recommendations for the CyRoads smartphone app can be summarized as follows:

- **Speed Correction Methods**. Investigate alternative techniques for speed correction. Utilizing machine learning algorithms may enhance correction accuracy.
- **Large-Scale Testing**. Collect more field data using a variety of smartphones, vehicles, and speeds to rigorously assess the app's reliability, stability, and accuracy.
- **Web Management System**. Develop a robust web management system to handle data uploaded from both smartphones and smart boxes. The system should categorize data and authenticate users.
- **Integration with PMIS**. Integrate the IRI data measured by the smartphone app and smart box into existing PMIS.
- **Smart Mapping System**. Create an integrated, color-coded mapping system that displays all measured routes and their corresponding IRI data along with other pavement condition metrics.
- **Functionality Upgrades for CyRoads**. Explore additional features for the CyRoads app, such as distress detection through image analysis and road bump identification.
- **Smart Box Enhancements**. Consider expanding the smart box's capabilities by integrating additional sensors to meet various user requirements.
- **Granular Road Testing**. Examine the app and smart box's effectiveness on granular roads and develop specialized correction methods distinct from those for paved roads.
- **Additional Pavement Metrics**. Investigate the smartphone's capability to measure other pavement performance indexes, such as the PCI.
- **Vehicle Type Expansion**. Extend the app's capabilities to include more types of vehicles, such as vans and SUVs, to validate their compatibility with the system.

**Future Research Recommendations for Using the Smartphone-Based App and Algorithms in Detecting and Measuring Road Surface Distress**

Future research recommendations for the smartphone-based distress detection app can be summarized as follows:

- **Complete App Development**. Since the current app is a prototype focused on demonstrating road distress detection capabilities, further development is recommended to create a fully functional application.
- **Latency Optimization**. Refine the algorithm to minimize latency in road distress detection, especially at vehicle speeds of 55 mph or higher.

- **Image Processing Capacity**. Given the limitations inherent in processing images captured in real-time from smartphones, research should explore systematic testing of the number of images that can be processed and analyzed at various speeds.
- **New Distress Detection**. Investigate training models to detect new types of road distress, enhancing the app's versatility.
- **GIS Mapping**. Evaluate the feasibility and applicability of generating interactive GIS maps of road defects using GPS data. The location of each distress can be mapped in bounding boxes and plotted on a shapefile to be reused by engineers in the office. This approach could be used in network-level mapping of pavement distresses in counties.
- **Backend Server Interface**. Consider developing an interface that enables easy access to the backend server, facilitating efficient maintenance of stored images and their corresponding data.
- **Quantifying Road Defects**. The existing model lacks the capability to quantify the extent and severity of road cracks. Future research should focus on devising a methodology for calculating the amount, extent, and severity of distress. These data could be invaluable for LPAs in formulating effective road maintenance strategies.
- **Additional Road Pavement Distress**. Only four types of road distress were classified in this task. The model could be trained to detect a wider array of distress types that may be present on local and state highways by using additional distress images and data sets in the training process.

## REFERENCES


Arya, D., H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto. 2020. Global road damage detection: State-of-the-art solutions. 2020 IEEE International Conference on Big Data (Big Data), December 10–13, Virtual Meeting. pp. 5533–5539.

Aleadelat, W., K. Ksaibati, C. H. Wright, and P. Saha. 2018. Evaluation of pavement roughness using an Android-based smartphone. *Journal of Transportation Engineering, Part B: Pavements*, Vol. 144, No. 3.

Azizan, M. S., and M. N. M. Taher. 2021. Evaluation of pavement ride quality on road networks using smartphone application. *Recent Trends in Civil Engineering and Built Environment*, Vol. 2, No. 1, pp.323–329.

Baladi, G. Y., T. Dawson, G. Musunuru, M. Prohaska, and K. Thomas. 2017. *Pavement Performance Measures and Forecasting and the Effects of Maintenance and Rehabilitation Strategy on Treatment Effectiveness – Chapter 3: Pavement Condition Classification*. No. FHWA-HRT-17-095. Turner-Fairbank Highway Research Center, McLean, VA.

Bochkovskiy, A., C. Y. Wang, and H. Y. M. Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*.

Brickman, A. D., W. H. Park, and J. C. Wambold. 1972. *Road Roughness Effects on Vehicle Performance*. No. TTSC-2707. Pennsylvania Department of Transportation, Harrisburg, PA.

Buttlar, W. G., and M. Islam. 2014. *Integration of Smart-Phone-Based Pavement Roughness Data Collection Tool with Asset Management System*. No. 098IY04. NEXTRANS Center, West Lafayette, Indiana.

MIT CSHub. 2019. *Crowdsourcing Pavement Data with Carbin*. Massachusetts Institute of Technology Concrete Sustainability Hub. https://cshub.mit.edu/sites/default/files/images/092320_Public%20Carbin%20Summary.pdf.

Cafiso, S., C. D. Agostino, E. Delfino, and A. Montella. 2017. From manual to automatic pavement distress detection and classification. 2017 IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), June 26–28, Napoli, Italy. pp. 433–438.

Choubane, B., and R. McNamara. 2001. *Precision of High-Speed Profilers for Measurement of Asphalt Pavement Smoothness*. No. FL-DOT-SMO-01-451. Florida Department of Transportation. Tallahassee, FL.

El-Diasty, M., and S. Pagiatakis. 2008. Calibration and stochastic modelling of inertial navigation sensor errors. *Journal of Global Positioning Systems*, Vol. 7, No. 2, pp.170–182.

Forslöf, L. 2012. Roadroid–Smartphone Road Quality Monitoring. *Proceedings of the 19th Intelligent Transport Systems (ITS) World Congress, October 22–16, Vienna, Austria*.

Gillespie, T. D., and M. Sayers. 1981. The role of road roughness in vehicle ride. *Transportation Research Record: Journal of the Transportation Research Board*, No. 836, pp. 15–20.

Hanson, T., C. Cameron. and E. Hildebrand. 2014. Evaluation of low-cost consumer-level mobile phone technology for measuring International Roughness Index (IRI) values. *Canadian Journal of Civil Engineering*, Vol. 41, No. 9, pp. 819–827.

Islam, M. S. 2015. Development of a Smartphone Application to Measure Pavement Roughness and to Identify Surface Irregularities. Ph.D. dissertation. University of Illinois at Urbana-Champaign, Champaign, IL.

Jones, H., and L. Forslöf. 2015. Roadroid: Continuous road condition monitoring with smart phones. *Journal of Civil Engineering and Architecture*, Vol. 9, No. 4, pp. 485–496.

Karamihas, S. 2021. Improvement of Inertial Profiler Measurements of Urban and Low-Speed Roadways. Ph.D. dissertation. University of Michigan, Ann Arbor, Michigan.

Kos, A., S. Tomažič, and A. Umek. 2016. Evaluation of smartphone inertial sensor performance for cross-platform mobile applications. *Sensors*, Vol. 16, No. 4, pp. 477-491.

Martinez, M. 2014. *The Rideability of a Deflected Bridge Approach Slab Phase II*. No. FHWA/LA.14/531. Springfield, VA.

Materson, A. 2015. RoadBump app cost-effectively provides IRI for road maintenance professionals. TotalPave. https://totalpave.com/blog/laser-profiler-vs-smartphone-precise-vs-accurate-iri-data/.

McGhee, K. H. 2004. *NCHRP Synthesis 334: Automated Pavement Distress Collection Techniques*. National Cooperative Highway Research Program, Washington, DC.

Misaghi, S., C. Tirado, S. Nazarian, and C. Carrasco. 2021. Impact of pavement roughness and suspension systems on vehicle dynamic loads on flexible pavements. *Transportation Engineering*, Vol. 3, pp. 100045.

Paterson, W. D., and T. Scullion. 1990. *Information Systems for Road Management: Draft Guidelines on System Design and Data Issues*. No. INU-77. World Bank, Washington, DC.

Rath, J. J., K. C. Veluvolu, and M. Defoort. 2014. Estimation of road profile for suspension systems using adaptive super-twisting observer. 13th European Control Conference (ECC), June 24–27, Strasbourg, France. pp.1675–1680.

Roy, S., and Z. Liu. 2008. Road vehicle suspension and performance evaluation using a two-dimensional vehicle model. *International Journal of Vehicle Systems Modelling and Testing*, Vol. 3, No. 1-2, pp. 68–93.

Sayers, M.W. 1989. Two quarter-car models for defining road roughness: IRI and HRI. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1215, pp. 165–172.

Sayers, M. W. 1995. On the calculation of International Roughness Index from longitudinal road profile. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1501, pp. 1–12.

Sayers, M. W., T. D. Gillespie. and C. A. V. Queiroz. 1986. *The International Road Roughness Experiment: Establishing Correlation and a Calibration Standard for Measurements*. No. WT-45. World Bank, Washington, DC.

Sayers, M.W., and S. M. Karamihas. 1998. *The Little Book of Profiling*. Transportation Research Institute at University of Michigan, Ann Arbor, MI.

Smith, K., and P. Ram. 2016. *Measuring and Specifying Pavement Smoothness*. No. FHWA-HIF-16-032. Federal Highway Administration, Washington, DC.

Smith, K. L., K. D. Smith, L. D. Evans, T. E. Hoerner, M. I. Darter, and J. H. Woodstrom. 1997. *Smoothness Specifications for Pavements*. NCHRP Project 1-31. National Cooperative Highway Research Program, Washington, DC.

Tian, K., D. King, B. Yang, H. Ceylan, and S. Kim. 2021. Characterization of curling and warping influence on smoothness of jointed plain concrete pavements. 2021 ASCE Airfield and Highway Pavement Conference, June 8–10, Virtual Meeting. pp. 110–119.
TotalPave. 2021. Laser profiler vs smartphone – Precise vs accurate IRI data. TotalPave. https://totalpave.com/blog/laser-profiler-vs-smartphone-precise-vs-accurate-iri-data/.
Wang, G. 2019. An Investigation of the Suitability of Smartphone Devices for Road Condition Assessment. Ph.D. dissertation. University of Birmingham, Birmingham, UK.
Wang, G., M. Burrow. and G. Ghataora. 2020. Study of the factors affecting road roughness measurement using smartphones. *Journal of Infrastructure Systems*, Vol. 26, No. 3, pp. 04020020.
Wang, W., and F. Guo. 2016. RoadLab: Revamping road condition and road safety monitoring by crowdsourcing with smartphone app. Proceedings of the 95th Transportation Research Board Annual Meeting, January 8–12, Washington, DC.
Xu, M.C., R. Roebuck. and D. Cebon. 2014. Modelling the performance of jointed plain concrete pavements to dynamic vehicle loads. Proceedings of the 13th International Symposium on Heavy Vehicle Transport Technology, October 27–30, San Luis, Argentina.
Yang, B., H. Ceylan, O. Smadi, K. Gopalakrishnan, S. Kim, Y. Turkan, A. Alhasan, and O. Adarkwa. 2017. Review of pavement cracking data collection practices. *Bearing Capacity of Roads, Railways, and Airfields*. CRC Press, Boca Raton, FL. pp. 861–868.
Yang, S., A. Alhasan, H. Ceylan, S. Kim, and B. Yang. 2022. Accuracy assessment of light detection and ranging system measurements for jointed concrete pavement surface geometry. *Road Materials and Pavement Design*, Vol. 24, No. 7, pp. 1695–1711.
Yu, H., C. Chen, X. Du, Y. Li, A. Rashwan, L. Hou, P. Jin, F. Yang, F. Liu, J. Kim, and L. Jing. 2020. TensorFlow model garden. https://github.com/tensorflow/models.
Yu, H., C. Chen, X. Du, Y. Li, A. Rashwan, L. Hou, P. Jin, F. Yang, F. Liu, J. Kim, and L. Jing. 2021. TensorFlow 2 Detection Model Zoo. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md.
Zhang, Z., H. Zhang, S. Xu, and W. Lv. 2021. Pavement roughness evaluation method based on the theoretical relationship between acceleration measured by smartphone and IRI. *International Journal of Pavement Engineering*, Vol. 23, No. 9, pp. 3082–3098.

## APPENDIX A: CYROADS USER MANUAL

### Introduction

CyRoads is a smartphone-based application that uses sensors from the smartphone, specifically accelerometers and GPS, to collect data and display the International Roughness Index (IRI) for a road section. IRI is an international standard that measures pavement roughness and is essential for road maintenance and surveys. CyRoads is available for both Android- and iOS-based smartphones.

### Terminology

- RAW data file – A file containing accelerometer, GPS, and speed information.
- White noise file – A file containing background noise caused by vibrations from the vehicle's engine and by variations in the smartphone's alignment position when a windshield mount is used. It is created as a part of a new trip and used as a reference for calculating an offset value that is used in computing the IRI.
- Calibration – Adjustments to the position/orientation of the smartphone.

### Prerequisites

- Smartphone (Android or iOS)
    - Android
        - Version 9 or later of the Android operating system.
        - Unknown app installations are enabled through automatic prompting or by going to Settings > Apps and Notifications > Special App Access > Install Unknown Apps.
    - iOS
        - Access to the app via TestFlight (contact Dr. Bo Yang to get the iOS app).
- Vehicle – Car or truck.
- Smartphone position – Mounted on the windshield or banded to the center console.

### Application Overview

- Navigation Bar (Figure A1) – This can be used to access the different functionalities of the application, namely the following:

    1. Home – Contains map activity.

2. Uploads – Contains recorded data and is used to upload files to the server.
3. Results – Displays results from the server.
4. Settings – Allows the user to set preferences.


**Figure A1. Navigation bar of the CyRoads app**

## Application Setup

- Once you have installed the latest version of the CyRoads app on your smartphone, enable location permissions. CyRoads will then request access to the device's location. Location permission is mandatory to use the app because it is essential for embedding the GPS information into the final data file. You will see a pop-up alert requesting you to allow the app to access the device's location, as shown in Figure A2. "Precise" and "While using the app" are the recommended settings. It should be noted that this process is only necessary during first-time installation.

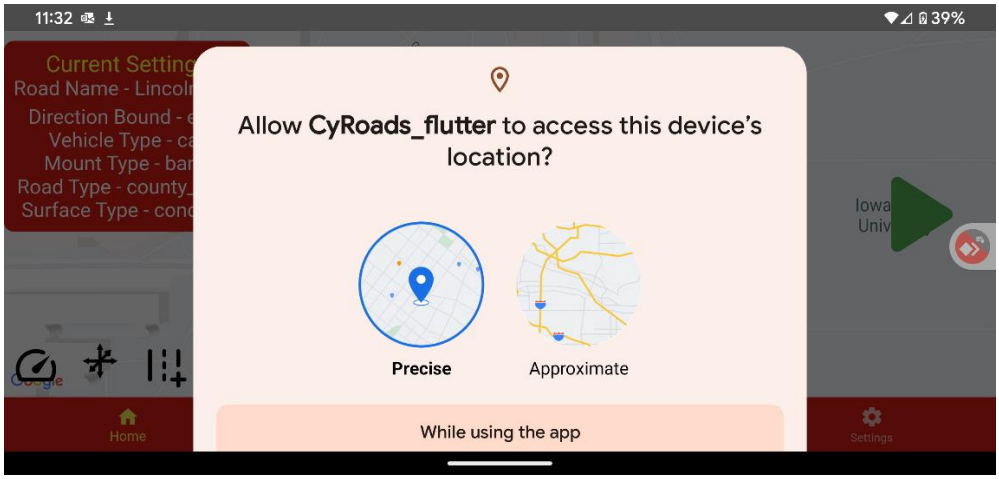



**Figure A2. Request for location permissions**

- Set up the cloud server IP address. In the CyRoads app, select the Settings tab and move to the bottom of the screen, where you will see the IP address box (Figure A3). The correct IP address is necessary to upload the raw data file to the cloud server for IRI computation. The available IP address is currently http://50.16.44.134:3001/. If the app does not display this address, enter the correct address and click "Set."

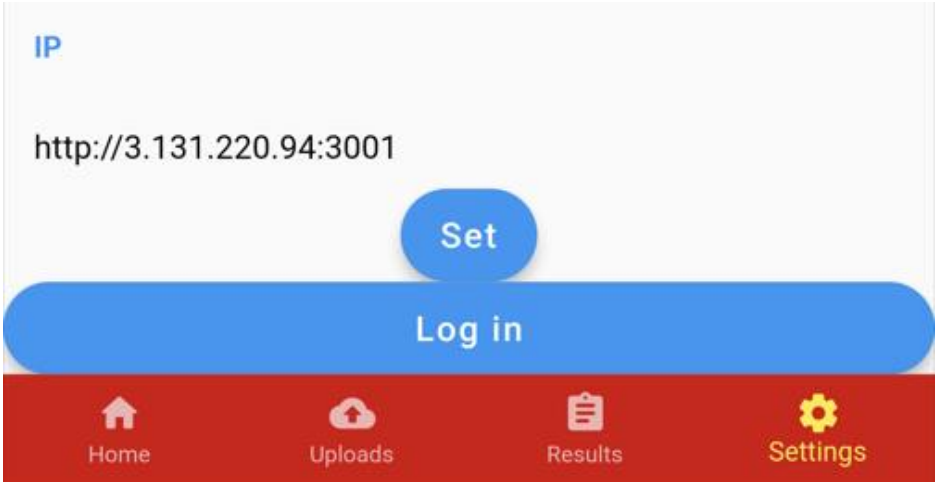



**Figure A3. Cloud server IP address setup**

- Log in to your registered account as shown in Figure A3 and Figure A4. (Contact the research team to sign up for an account.)

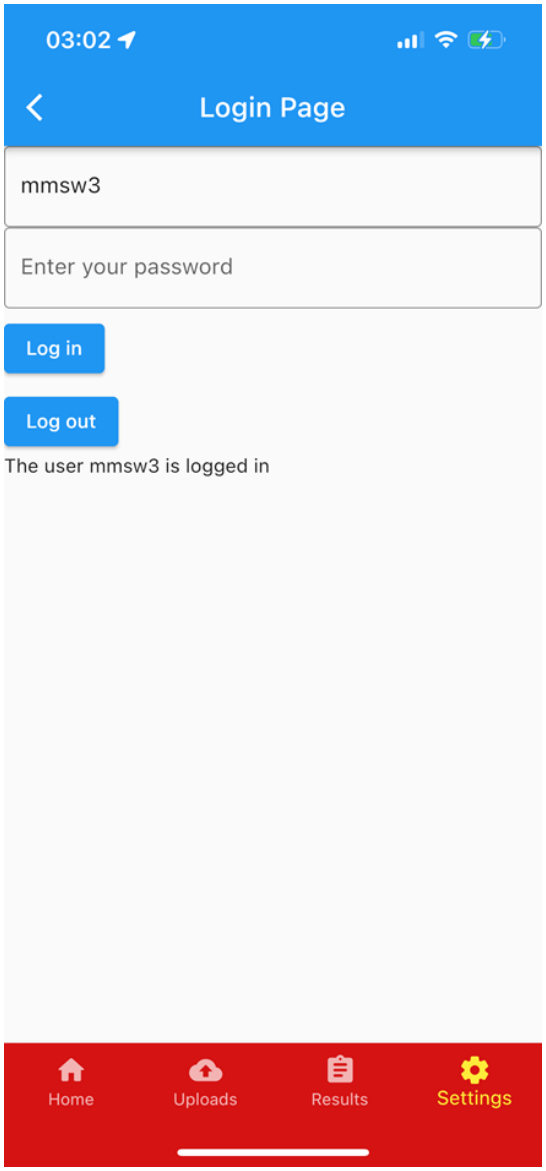



**Figure A4. Account login**

- Once you have enabled the correct IP address and successfully logged in to your account, you can set up trip information and start collecting data by following the steps below:

    - At the top of the Settings tab, name your road and specify the direction, vehicle, and mount type (Figure A5). It should be noted that when you input the road name, you should avoid using a hyphen because it will cause a data uploading issue; a correct example is provided in Figure A5. Also, note that the selected vehicle and mount type influence the speed correction parameters and ultimately influence the final IRI results. If your vehicle is an SUV, select “Car” if you are driving an SUV with five passenger seats (small-size or mid-size SUV), and select “Truck” if you are driving a full-size SUV with seven passenger seats.

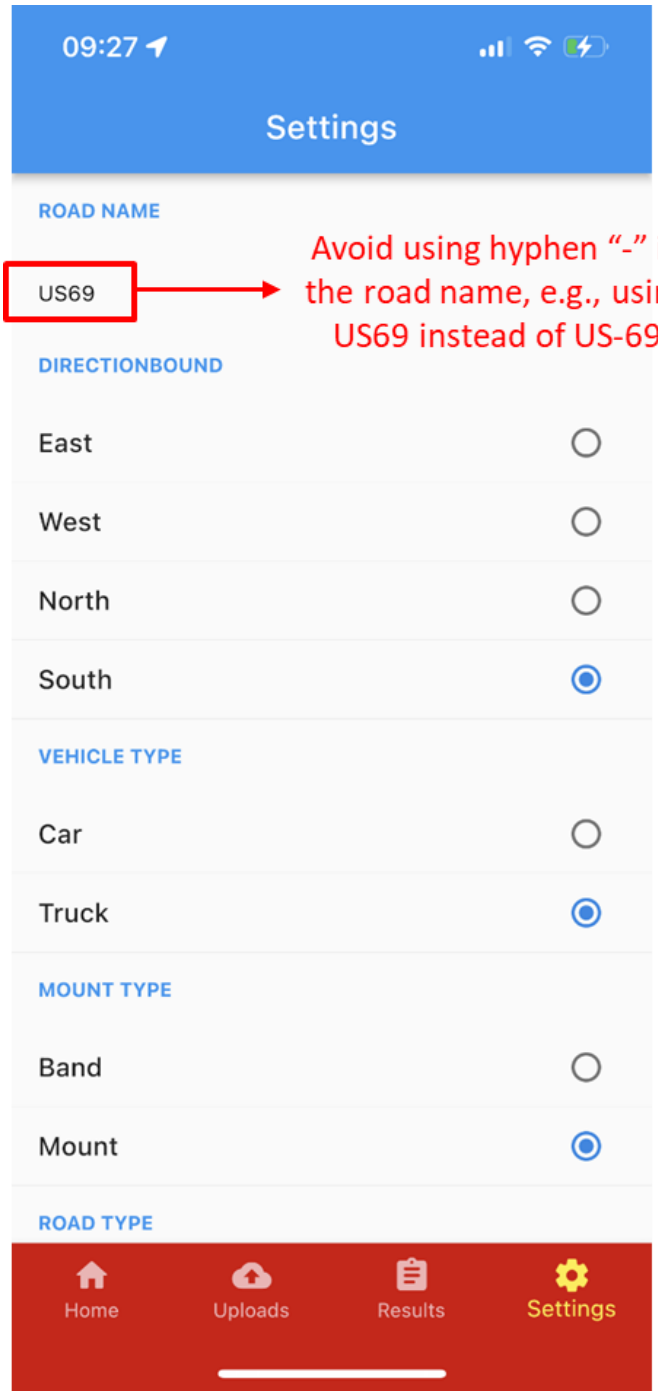



**Figure A5. Road name, direction, vehicle, and mount type settings**

- If you are using mount mode, attach the mount to the car or truck's windshield and mount the smartphone horizontally, as shown in Figure A6. A magnetic windshield mount without a flexible arm is recommended.

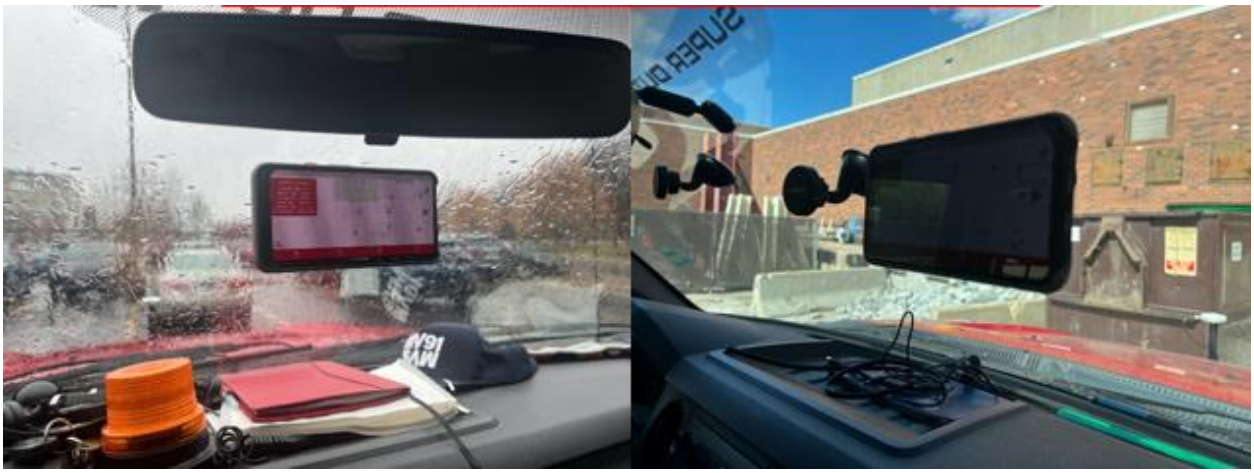


**Figure A6. Mount mode using a ChargeWorx Dash and Windshield Magnetic Mount**

- If you are using band mode, set up the band on the center console of the car or truck, as shown in Figure A7. Straps are typically available in local stores such as Walmart. If an appropriate strap is unavailable, you can use alternative options such as tape.

**Figure A7. Band mode using an adjustable strap (available at Walmart)**

- Continuing through the remaining settings options, you will see "Road Type" and "Surface Type," as shown in Figure A8. "Road Type" influences the roughness rating based on Table A1. "Surface Type" is a user memo that does not affect IRI results.

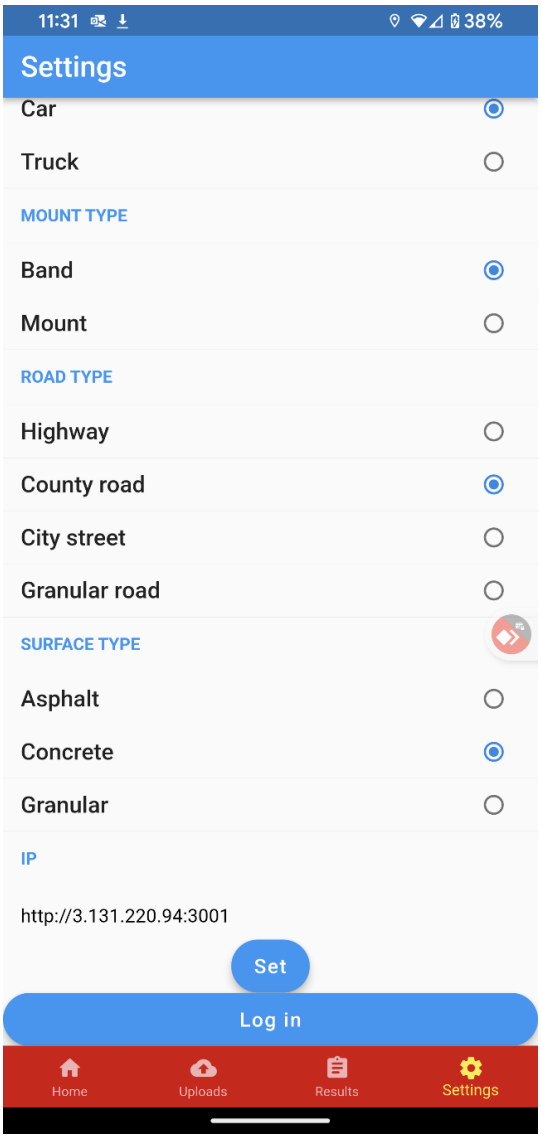



**Figure A8. Road type and surface type settings**

**Table A1. IRI thresholds for different road types**

| ***IRI Threshold (in./mi)*** | **Road Category** | | | |
|---|---|---|---|---|
| | **Highway** | **County Road** | **City Street** | **Granular Road** |
| *Very Good* | < 65 | < 85 | < 105 | < 125 |
| *Good* | 65 - 85 | 85 - 105 | 105 - 125 | 125 - 170 |
| *Fair* | 85 - 105 | 105 - 125 | 125 - 145 | 170 - 210 |
| *Bad* | 105 - 125 | 125 - 145 | 145 - 170 | 210- 250 |
| *Very Bad* | > 125 | > 145 | > 170 | > 250 |

- Once all settings information has been entered, you can select the Home tab and start data collection, as detailed below:

1. The Home tab presents a map that displays the user's current location and provides access to other functions (Figure A9). Note that the speed correction tool is currently disabled.

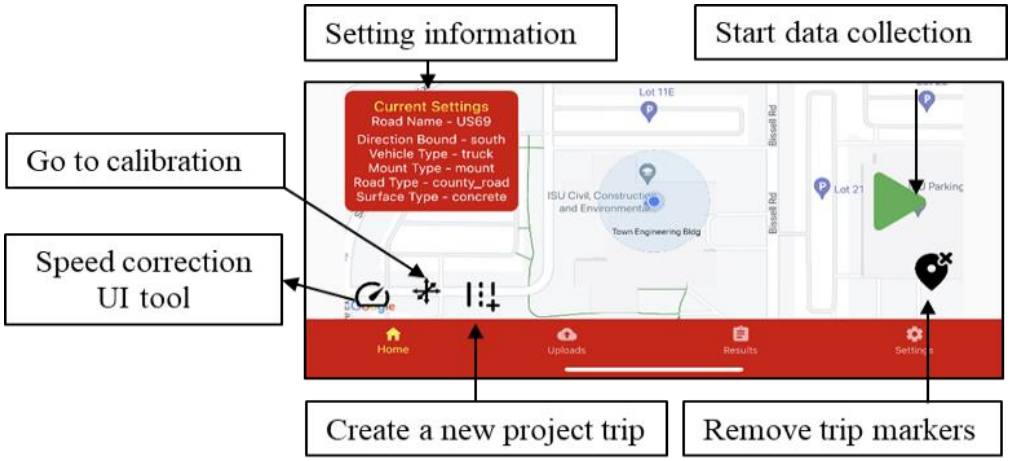



**Figure A9. Home tab**

2. The "Go to calibration" icon helps you calibrate the smartphone's position to mitigate alignment variations. The calibration view is shown in Figure A10. Adjust your phone's position to keep the green circle centered.


**Figure A10. Alignment position calibration: red (incorrect) and green (correct)**

3. After calibration, you can go back to the Home tab to create a trip for new data collection. Clicking on the "Create a new project trip" icon will open a dialog box that features three buttons (Figure A11): "OK" (to collect a new white noise file), "Previous" (to use a previously created white noise file, if available), and "Cancel" (to cancel trip creation). If you have already affixed your phone to the windshield mount or center console and have completed the calibration, you can click "OK" to immediately start a new white noise collection. Collection must be stopped after 10 to 15 seconds. Turn on the vehicle engine during white noise collection. For continuous data collection without moving your smartphone between trips, you can opt to use a previous white noise file by selecting "Previous," and the system will automatically load your previous trip's white noise file.

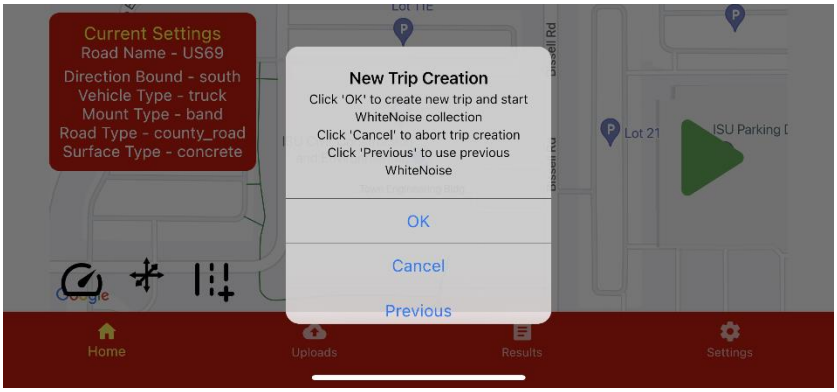



**Figure A11. Creating a new trip and selecting a white noise collection option**

4. After completing the white noise collection process, you can start collecting data. Drive at a stable speed on the road for which roughness data are to be collected, and then click on the "Start" icon (the green triangle in Figure A9) to initiate data collection. After the icon changes to "Stop" (the red square in Figure A12), you can click "Stop" to end data collection. The data file will be stored in the most recently created trip folder (named by date and time and other settings information), and markers will be added to the map to show the starting and ending positions. If you click "Remove trip markers," these markers will disappear.

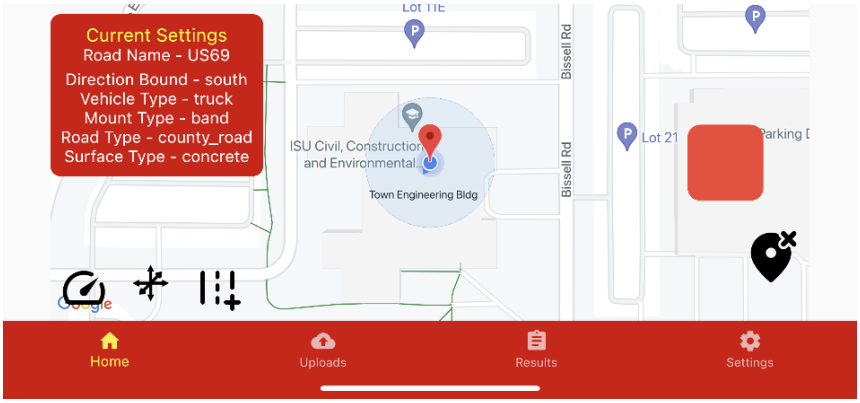



**Figure A12. Data collection**

- Use the Uploads tab to upload data files to the cloud server (Figure A13):

1. Access the folder directory where the data to be uploaded are stored. A folder is first categorized by date and then by subfolders named by date and time; each subfolder contains several raw data CSV files.
2. Each raw data file is named with the date and time and with the settings information to help the user remember which site it corresponds to.
3. Once raw data files are selected, use the "Upload" icon to upload the data to the server for the computation of IRI. Make sure your smartphone has a good network signal or WiFi connection. Once the data upload is complete, you will receive a notification from the app. Note that you do not need to select and upload white noise files.
4. Use the "Share" icon to share the folder or raw data file through email, cloud drive, text message, or other method. If you want to delete the data file, select the file and then click the "Delete" icon.
5. Tapping on a folder will show its contents. Each folder corresponds to a site; it consists of one white noise file and any number of raw data files related to that site.

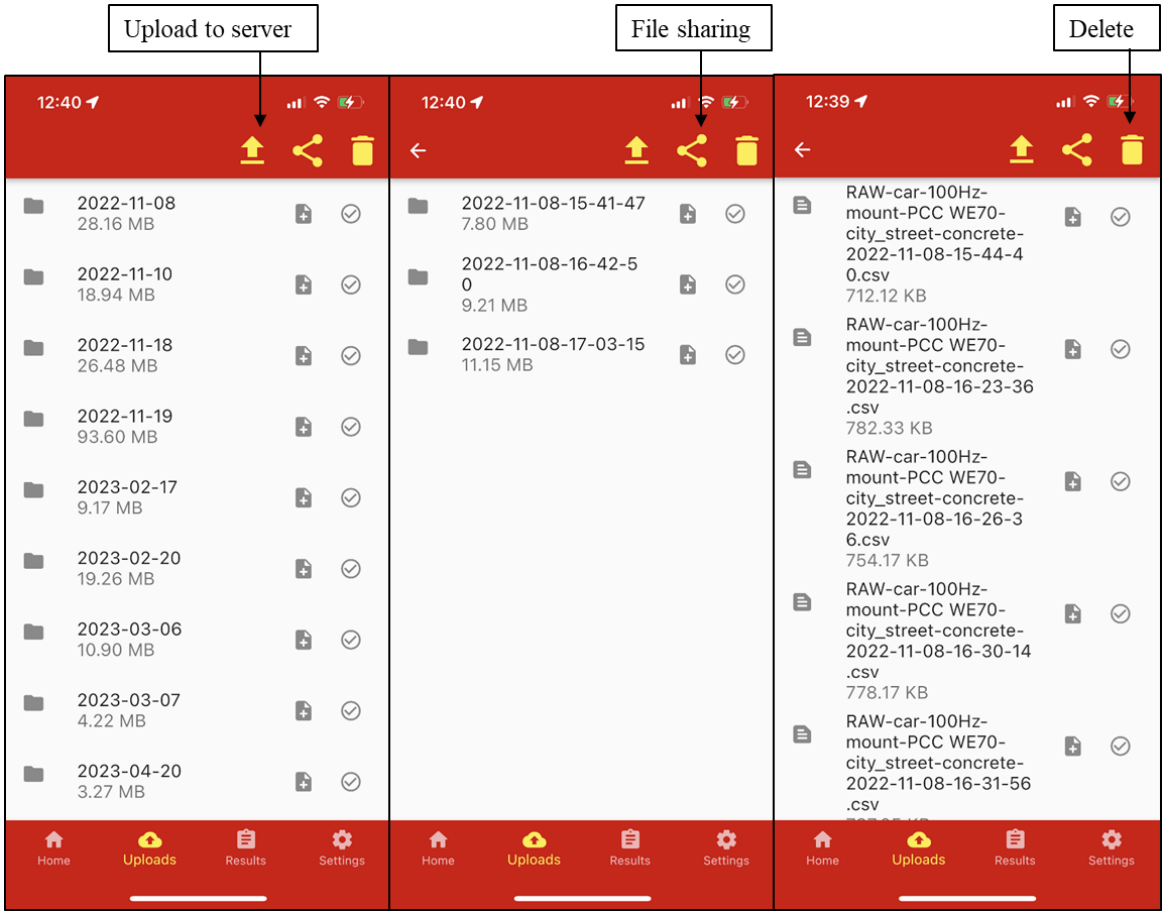



**Figure A13. Uploads tab**

- Use the Results tab to view the IRI results in CyRoads:

1. The IRI results from the server are transmitted in a .txt file.
2. The folder structure for the Results tab is the same as that of the Uploads tab and includes "Share" and "Delete" functionalities.
3. Each folder contains computed IRI summary files corresponding to their respective raw data files. Tapping an IRI summary file will open a new screen that displays the IRI results, as shown in Figure A14. This screen displays a map with markers showing the start and end of the surveyed road segment, the overall IRI, speed, distance, time, and a color-coded route, with different colors representing different roughness levels based on the road type.
4. The results also provide a segment IRI function. Click "Segment IRI" to see each segment's IRI values, GPS coordinates, and roughness levels (Figure A15). Note that each segment is 0.1 miles long.

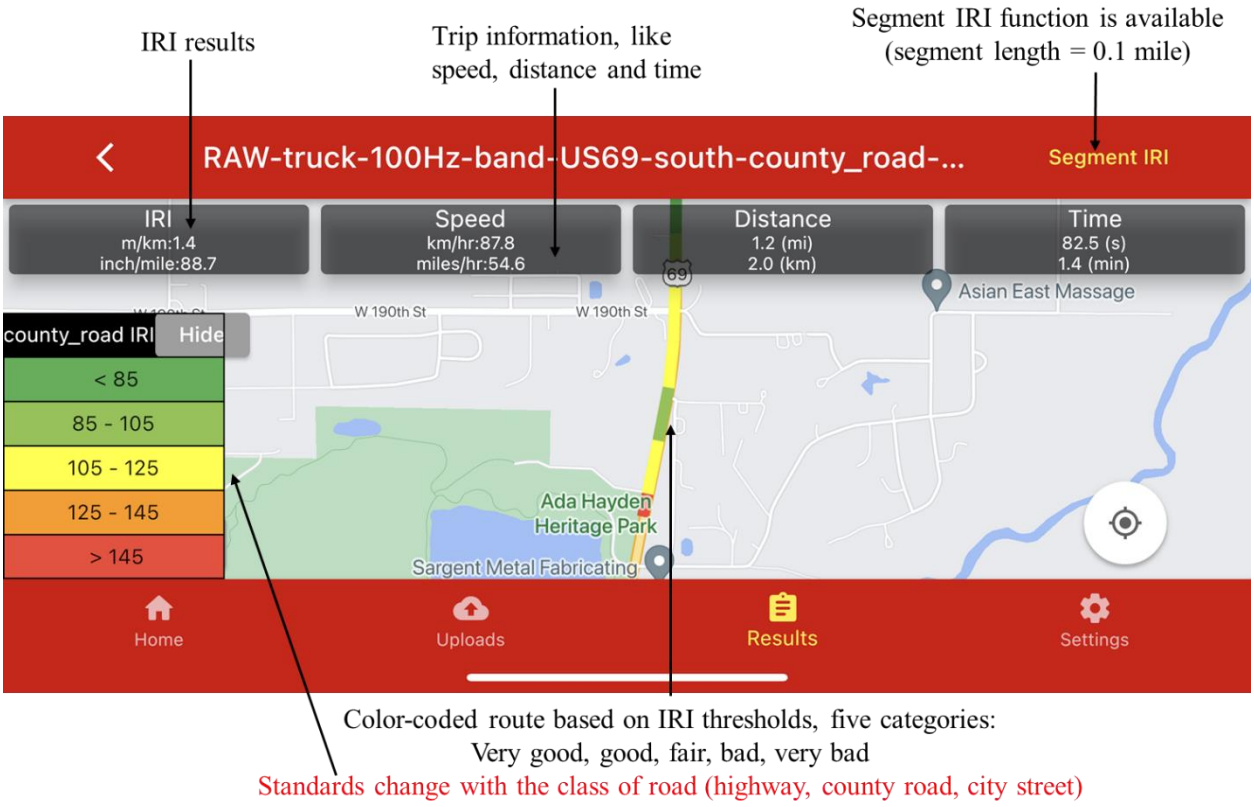



**Figure A14. IRI results – map view**

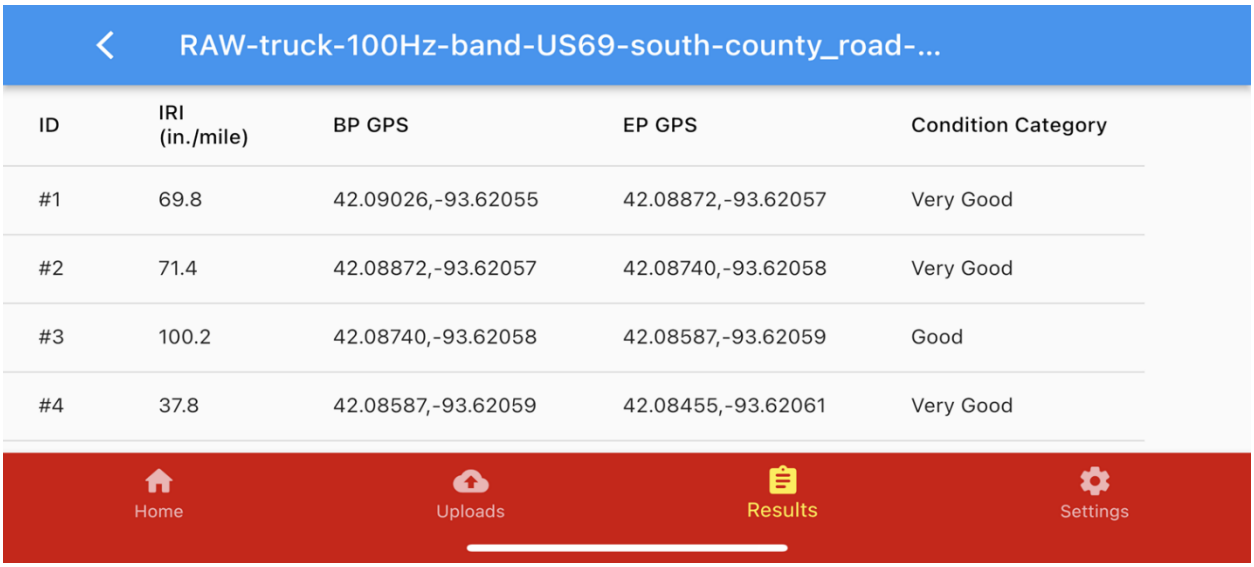



**Figure A15. Segment IRI results**

## APPENDIX B: SMART BOX USER MANUAL

### Introduction

The smart box is comprised of sensor-based equipment and a smartphone-based application that uses multiple sensors connected to a Raspberry Pi chip to collect data and display the International Roughness Index (IRI) for a road section. IRI is an international standard that measures pavement roughness and is essential for road maintenance and surveys. Raspberry Pi is a very common and low-cost chip that can connect to and process data from numerous types of sensors, including accelerometers and GPS modules. A smart box control application is available for both Android- and iOS-based smartphones.

### Terminology

- RAW data file – A file containing accelerometer and GPS data.

### Prerequisites

- Smartphone – Android or iOS.
- Vehicle – Car or truck.
- Smart box – The device requires charging via a USB car charger using a USB type-C cable. This ensures that the Raspberry Pi operates effectively. Position the smart box on the vehicle's central console to enable optimal data collection.

### Application Overview

- Navigation Bar (Figure B1) – This can be used to access the different functionalities of the application, namely the following:

    1. Map – This shows a map of the current location.
    2. Pi Files – This shows the data collected and transferred from the smart box.
    3. IRI Data – This shows the calculated IRI data returned from the remote cloud server.
    4. Settings – This allows the user to set preferences.

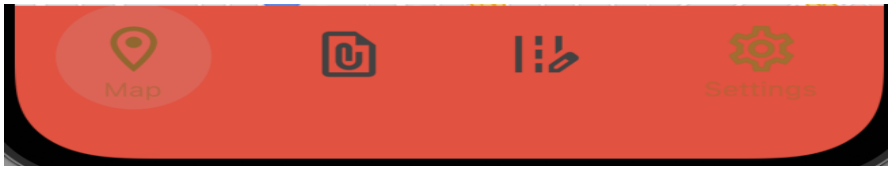



**Figure B1. Navigation bar**

**Smart Box Setup**

- It is imperative to charge the smart box before use to ensure power availability. Use a car charger along with a USB type-C cable to charge the device, as shown in Figure B2.
- Securely position the smart box on the vehicle's central console using the adjustable strap, as depicted in Figure B3.

**Figure B2. Power interface on the smart box, with the bottom left for USB type-C**

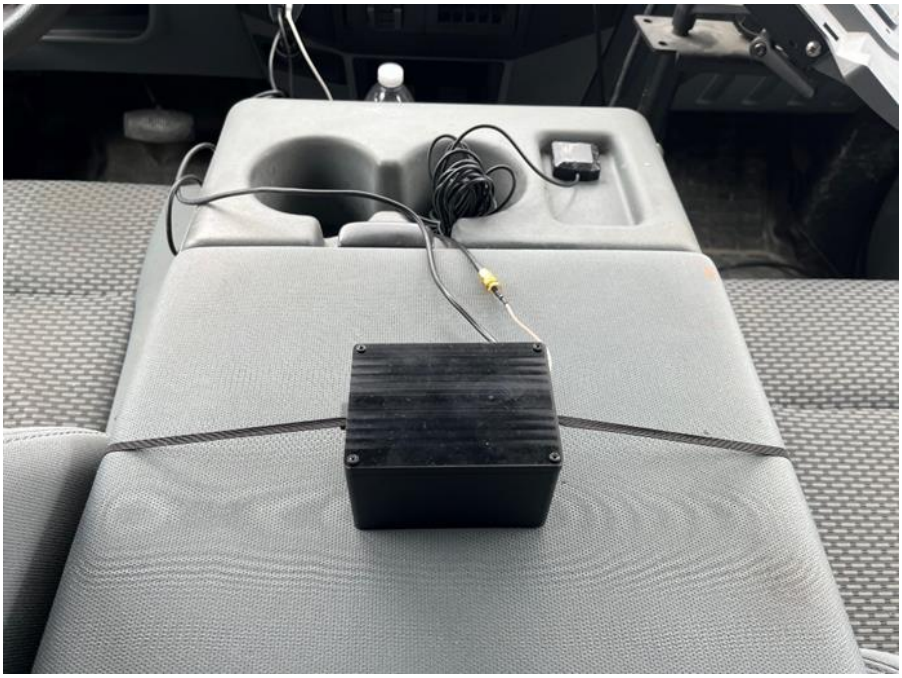


**Figure B3. Smart box strapped to the vehicle's center console**

- Make sure the antenna for the GPS module (Figure B4) is securely connected to the smart box. Position the antenna on the dashboard close to the vehicle's windshield.

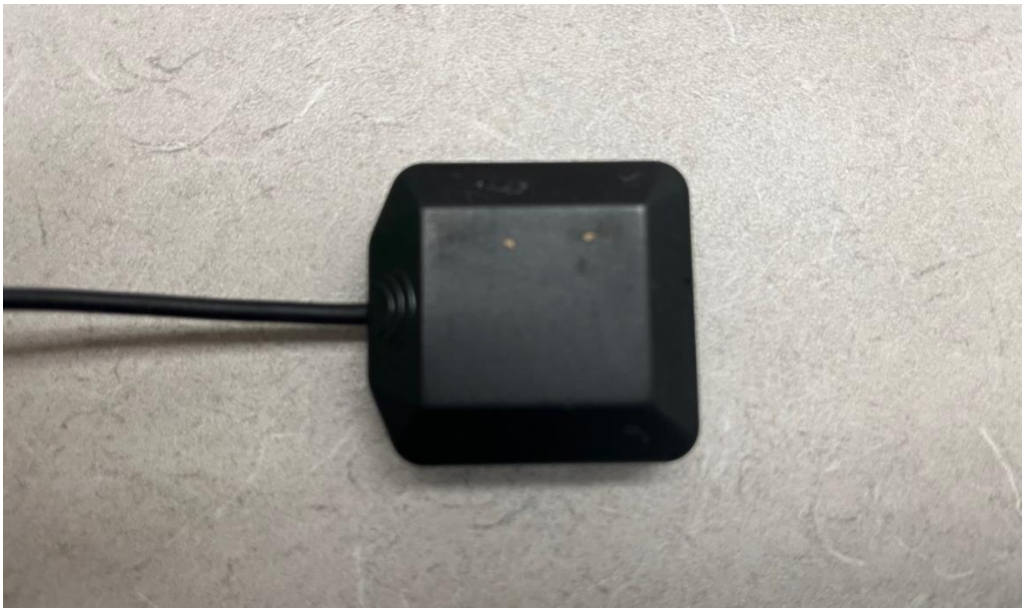


**Figure B4. Antenna for GPS module**

- Verify the status of the power light. Once the smart box is sufficiently charged, the power light should be illuminated.
- The smart box should initiate automatically when connected to the server. If the smart box fails to do so, the user might need to manually boot the Raspberry Pi operating system. In such situations, an external monitor is necessary to interface with the smart box.
- Open the app and verify that the app and smart box are connected. Click on the settings page (Figure B5) and enter the following:

1. The Raspberry Pi URL is the IP address set up in the Raspberry Pi; the default IP address is 192.168.4.1.
2. The Server URL is the IP address assigned to the cloud server.
3. Road Name is the name of the road that the user plans to measure.
4. Vehicle Type is the type of vehicle used for data collection.
5. Road Type is the type of road being surveyed. Options include Highway, County Road, City Street, or Granular Road.
6. Surface Type can be set as Asphalt, Concrete, or Granular.
7. After filling in the settings, press the "Login Page" button to navigate to the login page.

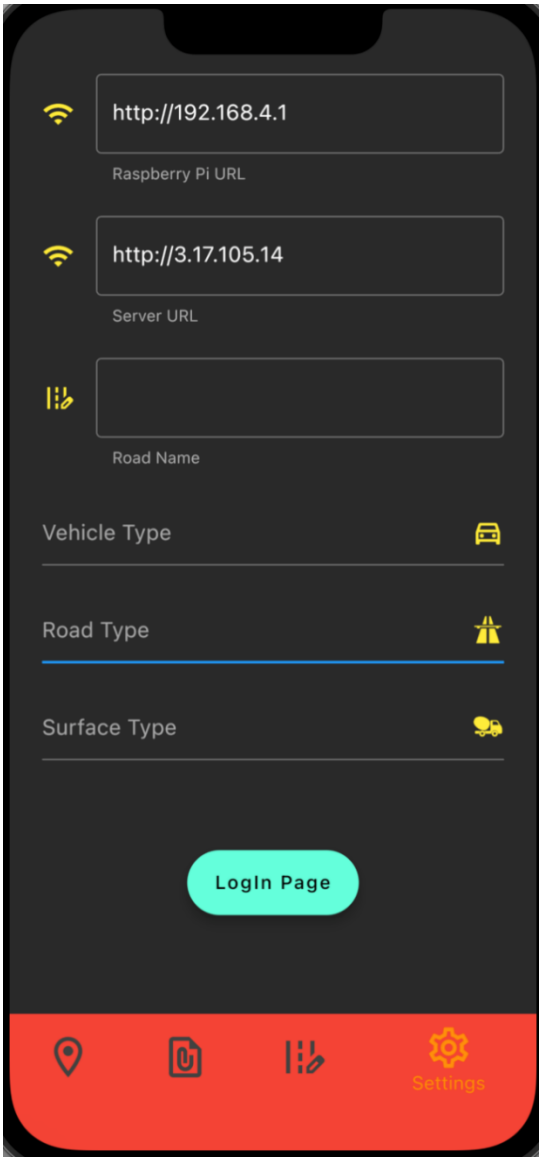



**Figure B5. Settings page of the app**

- To log in to the cloud server, enter the username and password for the preregistered user account (Figure B6).

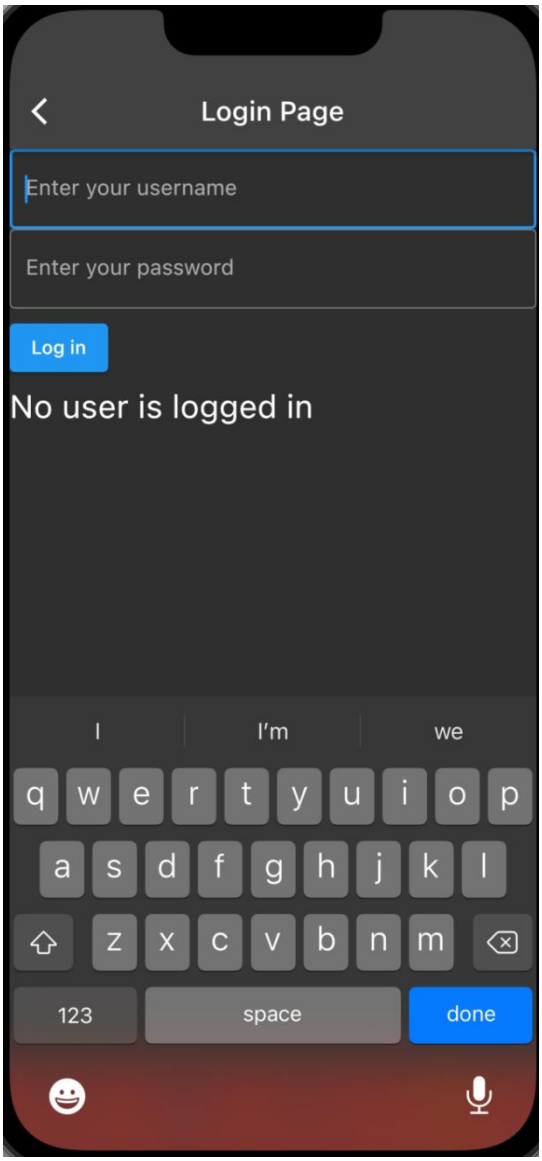



**Figure B6. Login page**

**Smart Box Measurement**

- To initiate the smart box operation, switch to the Map page (Figure B7). Click the "Start" button located in the upper left corner of the screen to begin data collection. Once sufficient data have been gathered, click the "Stop" button to cease data collection. The resulting data file will then automatically download from the smart box to the smartphone app.

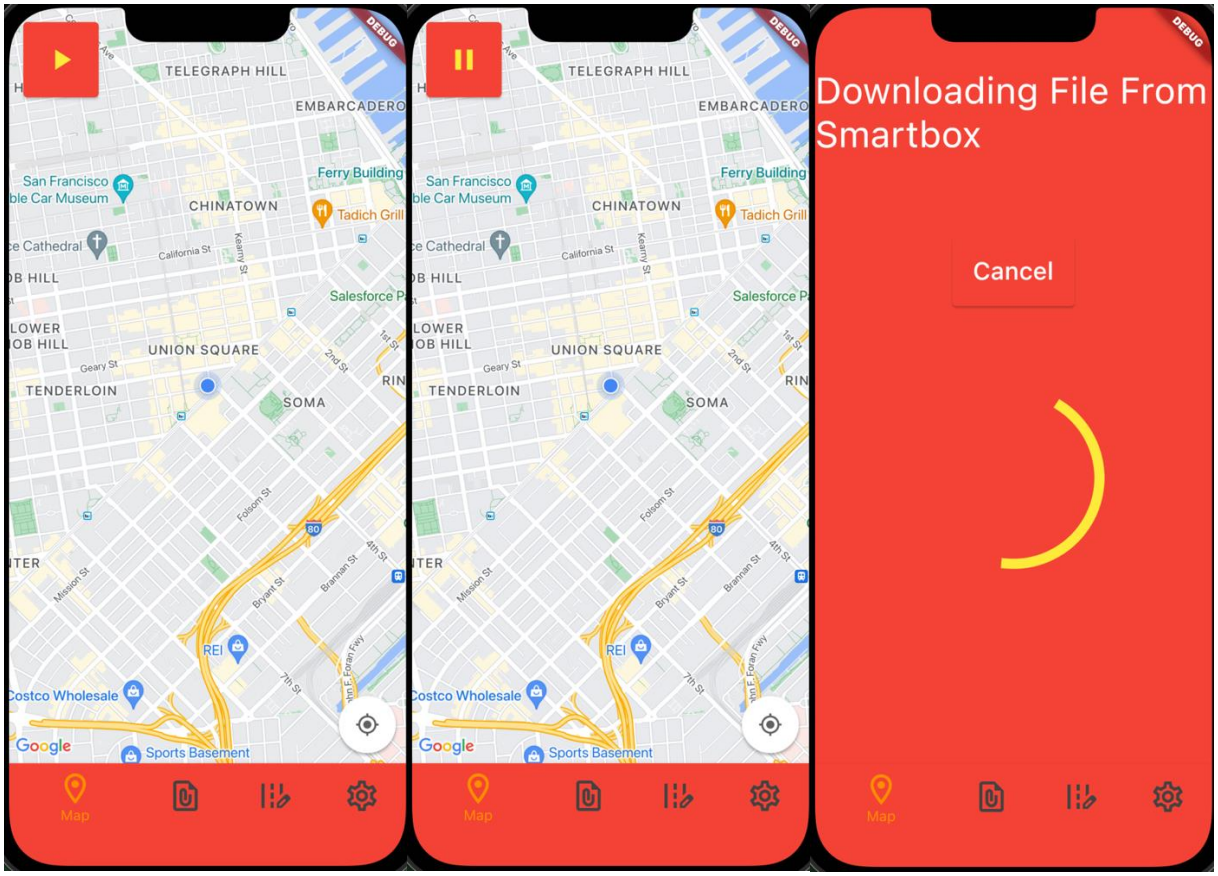



**Figure B7. Map page during measurement**

- After downloading the data file from the smart box, navigate to the Pi Files page (Figure B8). Here, users can find the data collected during a given period. Simply select the desired data file and then upload it to the cloud server for subsequent IRI calculations.

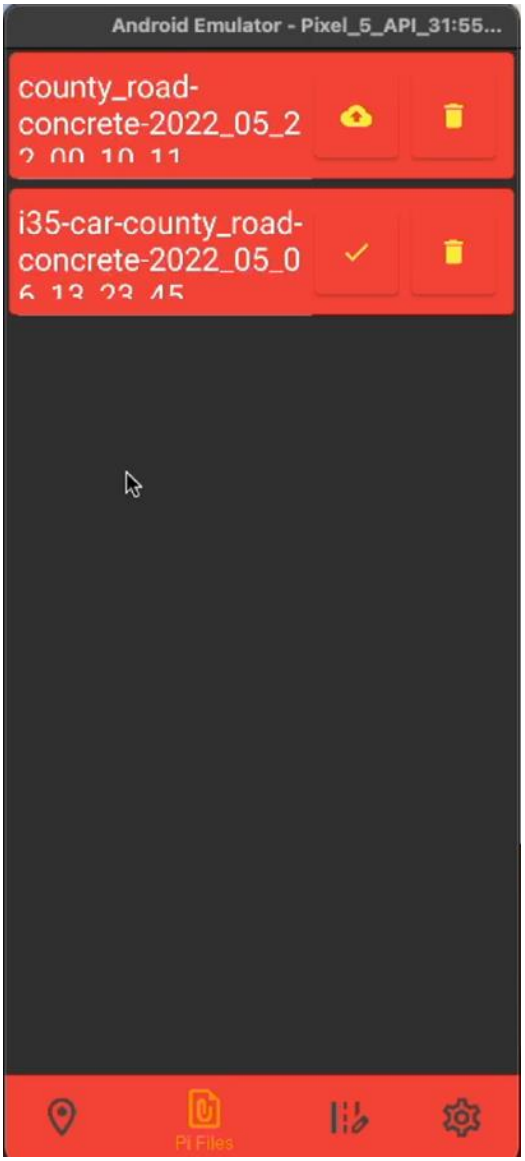



**Figure B8. Pi Files page**

- Once the data have been uploaded to the cloud server, the IRI results are automatically downloaded to the smartphone app. By navigating to the IRI Data page (Figure B9), the user can access the specific IRI data needed.

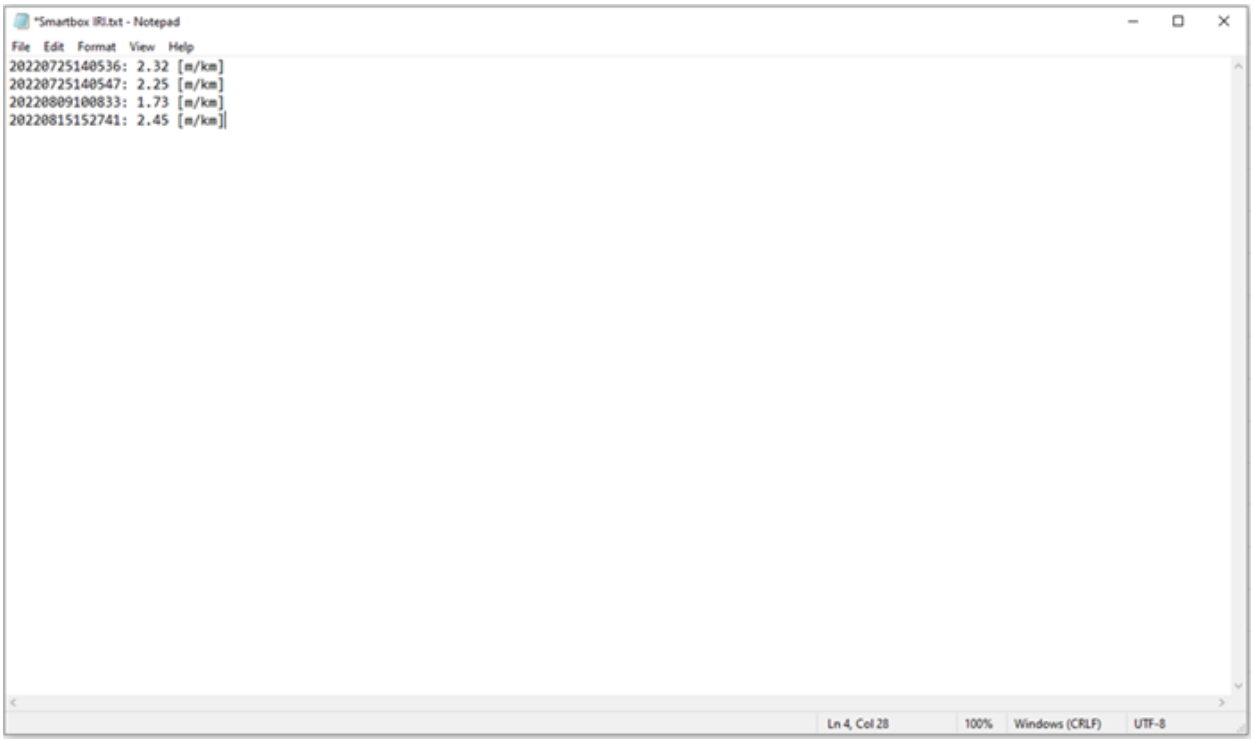



**Figure B9. IRI Data page**

**THE INSTITUTE FOR TRANSPORTATION IS THE FOCAL POINT FOR TRANSPORTATION AT IOWA STATE UNIVERSITY.**

**InTrans** centers and programs perform transportation research and provide technology transfer services for government agencies and private companies;

**InTrans** contributes to Iowa State University and the College of Engineering's educational programs for transportation students and provides K–12 outreach; and

**InTrans** conducts local, regional, and national transportation services and continuing education programs.


IOWA STATE UNIVERSITY

Visit **InTrans.iastate.edu** for color pdfs of this and other research reports.