Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Machine Learning Model for Road Asphalt Monitoring System: Vibration-Based Approach

A thesis presented by

Carlos Alonzo López Castañeda

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Monterrey, Nuevo León, Dec, 2022

Dedication

For all the people who accompanied me on this incredible journey.

Acknowledgements

I would like to thank Dr. Madhi Zareei for several reasons. First of all, as a guide and principal motivator in this thesis. He also gave me his invaluable knowledge, experience, and time. Thirdly, he helped me define the topic of my thesis, as well as allowed me to integrate it into his project.

Also thanks to the Tecnológico de Monterrey support on tuition and CONACyT with the support for living.

I would like to thank my family. For supporting me and loving me no matter the situation.

I would like to thank all the people who helped me. Thank you for the motivation and inspiration that allowed me to continue and achieve this result.

Machine Learning Model for Road Asphalt Monitoring System: Vibration-Based Approach

by

Carlos Alonzo López Castañeda

Abstract

To achieve safe and correct driving, it is necessary to have a surveillance plan and the maintenance of highways and roads, in order to maintain a good infrastructure. Mexico has a paved and unpaved network of 780, 511 km, of which is paved 174, 779 km. According to statistics from the INEGI, in 2019, there were 9,318 accidents due to poor road conditions. There are several types of breakdowns on any paved surface, and they may differ depending on the country. For example, potholes, cracks, and patches are some road surface damages essential to assess in Mexico. In 2020, INEGI presents that 96.8% of the population identified the issue of potholes in streets and avenues, as the problem with the highest frequency nationwide, above crime. Thus, the conditions of our roads are of deep concern for the population.

Different forms of road condition monitoring are proposed in the last years by specially designed instruments, using cameras, lasers, which require time and money and can only cover a limited proportion of the road network. Analogous to a video feed visually inspecting the asphalt's surface, a vibration-based system measures the ground conditions based on mechanical feedback from a vehicle. Different road anomalies, including potholes, cracks and ruts in the surface, create forces on the car, the frequency and magnitude of the forces will depend a lot on the type of anomaly. After we investigated different related works, this thesis is going to build on some of their aspects, and make a mix of others. The idea of dividing into three different categories for the classification of the roads, and the usage of supervised learning for road surface quality and anomaly detection. Regarding data collection, it was done through a phone with an Android system and an application created specifically for this job.

This thesis proposes a pothole detection model using a vibration base method, using built-in vibration sensors in smartphones. We collected road condition data in Mexico City using a dedicated vehicle and smartphones with a purpose-built mobile application designed for this study, splitting the data into: bump, bump, normal. A processing method was applied to the collected data, and features were extracted, then classified with a neural network. The results indicated that using only the subset of two of the three selected event types, together with their six characteristics, they outperformed other subsets in identifying potholes. Our neural network classifier showed classification performance, with an accuracy of 98%.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Roads play an essential role in modern societies that allow a comfortable, fast, and inexpensive way to travel from one place to another. Roads facilitate the mobility of people by connecting several different places not only within cities, shortening their distance, but also between cities and towns, as well as between different countries. Roads are a fundamental part of the infrastructure of any place since it deliberates many activities that cause the growth of everything that surrounds them, such as tourism, distribution, marketing, and construction. The networks of roads and highways have constant use and exploitation since they are currently in use at all hours of the day. This exploitation causes a constant and significant degradation of the different types of asphalt surfaces. If a valid maintenance policy is not applied, the quality of the asphalt road surface deteriorates, compromising road safety. In Mexico, the National Road Network (RNC) [12] integrates the entire paved network and most of the unpaved roads in Mexico; it has a total length of $780,511$ km, of which are paved $174,779$ km.

There are several types of breakdowns on any paved surface, and they may differ depending on the country. For example, potholes, cracks, and patches are some road surface damages essential to assess in Mexico. However, the most common forms of damage to the asphalt surface are potholes, the portion of the pavement surface that has been destroyed and removed by different factors being the most prominent dimension of 20 cm and minimum depth greater than the thickness of the folder [16]. Pothole repair is necessary when potholes compromise safety and asphalt rideability.

Detecting and estimating potholes is essential for adequately planning the repair and rehabilitation of asphalt. These types of plans are essential in Mexico since potholes are part of the main concerns of Mexican citizens. According to statistics from the INEGI on land traffic accidents in urban and suburban areas, in 2019, there were 9,318 accidents due to poor road conditions [22]. To repair a pothole there are several factors that must be taken into account: such as the level of traffic, repair time, number of personnel, mechanical equipment and materials. These various factors produce expenses, so the profitability of this type of operation must be maintained and taken care of. The key to making decisions for future reconstruction is to estimate the damage from the information collected.

In order to implement preventative rehabilitation, it is necessary to have decision support from a proper evaluation of asphalt conditions. There are different ways to perform tasks regarding potholes, the traditional method is based on human observation, prone to errors and of course low efficiency and precision. Recent methods use digital images and videos to record and classify conditions of road anomalies. These technologies increased the efficiency and reliability of the inspection, but they have several obstacles such as the large storage and computing space for image processing.

The present work is through the investigation of basic concepts on the subject and the comparison of different works on detecting potholes through different technologies. This thesis proposes and develops a pothole detection model that works efficiently and optimally to solve this problem and competes with the related works.

1.1 Research Methodology

A key component to designing an optimized algorithm is understanding the factors that influence it and its characteristics. To carry out the research throughout this thesis, design science was selected as the general methodology; This science has as objectives the study and investigation of artificial things, created by the human being. It also encompasses the way in which these activities are conducted [7].

Peffers K. et al. [30] compiled various researches from information system and other disciplines that have proposed adaptations to this methodology, arriving at the following stages:

- 1. Identification and motivation of the problems: Define in a specific way the problem to be studied and justify the value of the contribution.
- 2. Define the objectives for a solution: For a job to have a purpose, it is necessary to define the objective. This will mark the way forward for the investigation.
- 3. Design and development: This stage is to create the necessary tools to obtain results, that is, to create the model. It includes determining the desired functionality and architecture of the model. During this stage practical and theoretical approaches will be combined.
- 4. Demonstration: Obtain results and solutions for one or more instances of the problem. This phase could involve its use in experimentation, simulation, case study, testing, or other appropriate activities. Seeks to demonstrate and verify that the current work is valid or needs changes to be made.
- 5. Evaluation: It is the observation and measurement of how well the model solves the problem. This stage happens after making all the adjustments and changes to the model, so that they are finally compared with the results of other works.
- 6. Communication / Publication: Publicize the solution and its relevance.

First, a literature survey is included, where the related work is studied and related problems are investigated. Moreover, the factors that influence the design of an optimized algorithm and thereafter formulate a mathematical model of the problem will be taken into consideration. This leads to the next stage when we use key defining features of the model to state our hypothesis. The hypothesis will be further validated and refined through experimental and/or simulation work on various inputs. The results will then be compared with other existing solutions to verify its efficiency.

1.2 Problem definition

The road and highway networks are significant infrastructure developments since they generate economic growth in the country. When there is a good management and maintenance project on the roads. These actions have positive consequences within the transport sector since they increase the comfort and safety of drivers and passengers. The traditional detection and inspection method for potholes relies on human observation, which has low efficiency and many errors. Actual inspection methods include digital video and image analysis to record and identify surface conditions, but especial tools are required, and others are not very accurate, this will see it in the 'Related Work' chapter.

In Mexico, the National Road Network (RNC) [12] integrates the entire paved network and most of the unpaved roads in Mexico. The total length is $780,511$ km, and only $174,779$ km are paved. In 2020, in the Federation's budget of expenses, the Reconstruction and Conservation of Roads program had a budget of \$ $10,289,172,579$ pesos.

INEGI [23] presents the National Survey of Urban Public Safety (Figure 1.1), which provides an estimate of the perception of public safety of the population aged 18 years and over in urban areas. In the results of the last survey carried out in the last quarter of 2020, 96.8% of the population identified some problem in their city, surprisingly the issue of potholes in streets and avenues, the one with the highest frequency nationwide, above the crime. Thus, the conditions of our roads are of deep concern for the population.

In Mexico City, there is the 'Bache 24' program, which aims to attend, within no more than 24 hours, the reports made by citizens regarding potholes located in the primary road network, which is made of 169 roads (including the main roads, roads, avenues, as well as access roads to the city). This program seeks to guarantee the road and pedestrian safety of the inhabitants of Mexico City through an efficient patching system. According to users, this program is deficient due to the poor functionality of the mobile app; it is necessary to emphasize that the capture mode is manual. So it becomes a slow process and not optimal for the user.

Motion sensors from mobile devices will be used to detect data since they are designed for movement, vibration, tilt, shake, and rotation detection. Movements can directly reflect user interaction as well as device location.

For each smartphone-based application, various combinations of sensors can be used according to the desired application criteria, so it will be necessary to identify and select which sensors will be suitable for this project. Previous studies investigating the detection of road surface anomalies using smartphone sensors have used motion sensors extensively. Table 1.1 shows the most common sensors for this type of task.

Maintenance must be carried out in a systematic, effective and planned way, to guarantee that the roads are in the best possible conditions. Therefore, this research can help minimize data collection work and make it more efficient and faster.

1.3 Hypothesis and Research Questions

Hypothesis:

A reliable and accurate asphalt model could be proposed using sensor vibrations on a

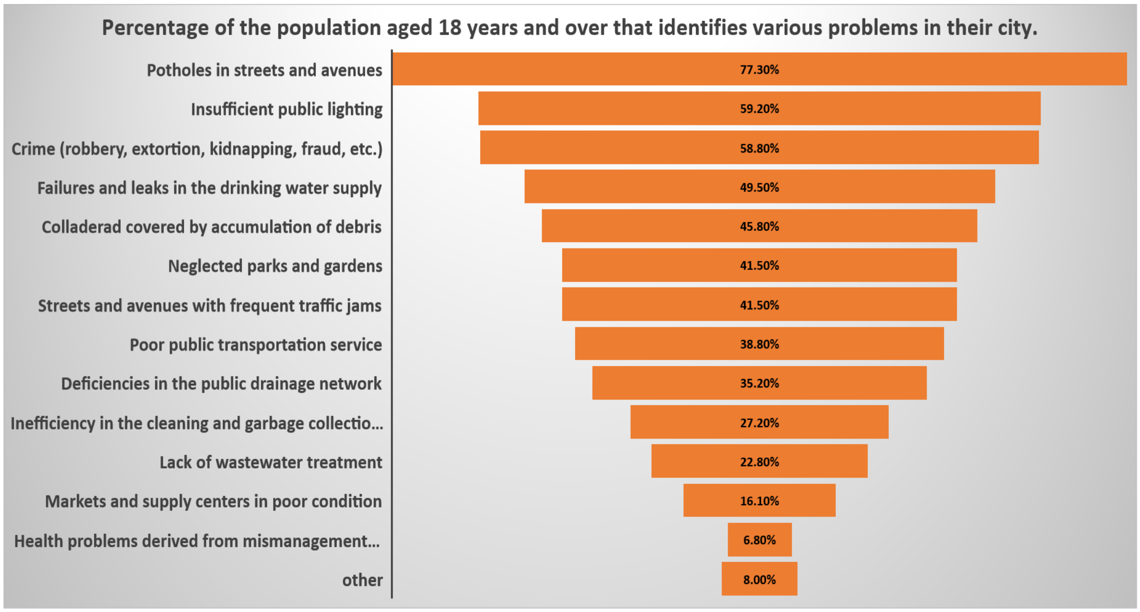


Figure 1.1: Results of National Survey of Urban Public Safety[23]

Sensor Name	Type	Unit	Description
Gyroscope	Physical	rad/s^2	Measures a device's rate of rotation
GPS	Physical	Degree	Obtain location information
Accelerometer	Physical	m/s^2	Measures the acceleration
Linear Accelerometer	Virtual	m/s^2	Measures the acceleration force, excluding the gravity
Rotation	Virtual	rad	Measures the orientation of a device
Gravity	Virtual	m/s^2	Measures the force of gravity

Table 1.1: List of sensors [4]

smartphone, collecting the information with the concept of Mobile Crowdsensing (MCS).

Research Questions

- 1. What is the vibration-based method best suited to pothole detection?
- 2. What are the most suitable sensors to carry out a correct method based on vibrations to detect potholes?
- 3. How to generate and obtain the data from the selected sensors?
- 4. What variables and factors must be taken into account to capture data?
- 5. What machine learning model best fits the collected data from the sensors?
- 6. Regarding pothole detection, what constitutes an accurate model?
- 7. How to design an automated pothole detection model that is also profitable, using machine learning?

1.4 Objectives

Investigate, design, and implement a pothole detection model, with a vibration-based method, through model learning using smartphone sensors and mobile crowd detection techniques. This model can be used by the public and private sectors to accurately monitor the current state of the asphalt surface in terms of the appearance of potholes. The model can provide information about the condition of the asphalt surface.

Sub-objectives

- Carry out a benchmark of the different and similar models applied to this problem.
- Investigate the different ways to increase the system's viability and reliability.
- Develop a mobile application to collect the data and a model to process the data and generate a reliable report on the condition of the asphalt of the street.

Chapter 2

Preliminary

2.1 Pavement

The pavement is a structure, generally made up of a bearing layer for the different types of vehicles, supported on a layer of classified granular material called the base[6]. In turn, said layer rests firmly and coherently on the layer of classified granular material called sub-base. These layers are responsible for distributing and transmitting the loads applied by traffic. There are two types of pavement [35]: flexible and rigid.

2.1.1 Structural elements of the pavement

The pavement is structured by three elements:

- **Base:** . It is the layer located below the folder (flexible pavement). Its function is to be resistant, absorbing most of the vertical efforts and having resistance to deformation due to traffic. Traditional granular bases are used for medium and light traffic, but treated granular materials are used for heavy traffic.

- **Sub-base:** . This element provides the base with a uniform foundation and constitutes a suitable work platform for its placement and compaction. It must be a permeable element to fulfill a draining action. In cases with light traffic, mainly heavy vehicles, this layer can be dispensed with, and the slabs rest directly on the sub-grade layer.

- **Sub-grade:** This layer must be capable of resisting the efforts transmitted to it by the pavement. It provides the necessary level for the sub-base and protects the pavement, preserving its integrity at all times, even in severe humidity conditions. This layer also has the objective of maintaining and providing adequate support to pavement anomalies in different scenario conditions.

- **Surface:** It is the section in direct contact with the traffic loads. It is the layer with the highest bearing capacity of the pavement structure. It is responsible for supporting the efforts and deformations caused by traffic, resisting weather and wear, providing a surface that facilitates the mobility of vehicles, as well as being a waterproofing medium protecting the other layers of the pavement.

2.1.2 Types of pavement

The damages are manifested differently according to the type of pavement in question, there are two types [35]:

- **Flexible pavement:** is characterized by having the surface layer made with asphalt concrete. Furthermore, it is made up of a granular base and a granular sub-base. In this class, the layers are placed so that the upper ones have greater rigidity than the lower ones. Its composition can be seen in Figure 2.1.

Advantages	Disadvantages
Recommended for low vehicle loads.	It is not recommended for high vehicle loads.
Allows complete waterproofing of the tread surface.	It is a less durable pavement.
The faults are generated in the surface layer, so the folder can be easily intervened.	Requires more maintenance or intervention.

Table 2.1: Advantages and Disadvantages of flexible pavement

Figure 2.1: Flexible pavement

- **Rigid pavement:** The surface layer is hydraulic concrete, on a sub-base of granular material, directly on the sub-grade. Thanks to its rigid composition, this type distributes the vertical loads on the surface. In this type of pavement you cannot observe the deformations of the lower layers without structural failure.

Advantages	Disadvantages
Greater structural capacity and durability.	For low vehicle loads it becomes a very expensive solution.
Recommended for high vehicle loads.	Fine materials coming out of the joints towards the surface, when there is no sub-base.
Requires less maintenance.	Design of steel is required.

Table 2.2: Advantages and Disadvantages of rigid pavement

Pavements, both flexible and rigid, do not fail or collapse spontaneously but do so gradually and progressively for different reasons, such as the continuous action of traffic and weather; they always have a manifestation on the surface of the pavement.

Figure 2.2: pavement

2.2 Pavement failures

In pavements it is defined as a failure when it loses the service characteristics for which it was designed [1], there are two types:

- **Structural failure:** . It is the deficiency in the pavement that affects and reduces the load capacity. To correctly classify this type of failure, it is necessary to consider the structural element by which it originated. If it gives correct and early maintenance, it can be very costly; otherwise, they worsen quickly and become economically larger.

- **Functional Failure:** .These are the failures that refer only to the conditions of the pavement surface that is in direct contact with external agents; that is, they are the surface failures of the pavement, which affect the comfort and safety of the user in transit. A functional failure not being properly repaired can turn into a structural failure.

2.3 Agents of deterioration

Once we have seen the two types of possible pavement failures and before we look at the different kinds of pavement impairments that exist with their sub-classes, we must focus on the agents that cause these failures. Herrera et al.[38] mention different agents, starting with the most influential agent on the pavement; since it is the one interacting the longest, we refer to traffic. On the other hand, there is also the phenomenon of humidity, in which they mention that despite being a significantly weak element when it seeps into the pavement, it manages to create distortions in the lower layers of the pavement. The subsoil is an agent that is ultimately a natural element constantly changing, which is why it becomes one of the most important elements to study when carrying out paving projects. Another significant element is the quality of construction. Namely, the materials used to make mixtures; since they are the ones that will give support since the conditions to which they will be exposed need them to be durable. So, having a rigorous quality control and inspection system is crucial. Finally, maintenance is an essential factor; the pavement will wear out naturally due to erosion and the constant use it has. Good maintenance allows the pavement to fulfill its functions during construction, but if it is not correct economically, it represents a considerable expense.

2.4 Types of pavement deterioration

As we saw in the previous section, pavement deterioration depends on various elements and factors. Also, the pavement type has different paving behaviors, so depending on the type of pavement, the damage will be classified and justified differently. The Mexican Institute of Transportation [11], presents us with the most common deterioration or failures in the highways of the Mexican national network, which are:

- Landslides:

  Pavement lifting

  Advanced and total erosion

  Disintegration

  Potholes

- Deformations:

  Bumps

  Corrugations

  Transverse and longitudinal

  Settlement

  Longitudinal Ridges

- Breaks:

  Reflection cracks

  Parabolic cracking

  Cracks

  Crocodile cracking

2.5 Potholes

These are the most common faults and how they are divided [11]. In this work, we will focus only on the 'Potholes,' which will be the type of failure to study and identify. For this, we must define it more thoroughly; for that, we will use the definition provided by the Mexican Institute of Transportation catalog above.

Potholes are defined as cavities of various sizes in the wearing course due to the detachment or initial disintegration of the aggregates that form cavities when passing vehicles.

Possible causes:

- Lack of resistance
- Shortage of asphalt content
- Poor thickness

- Localized transit disintegration
- Weak spots on the surface

2.6 Mobile Crowd-sensing.

Due to the fact that in recent years cell phones contain more and more technology, in this case the use of different sensors is mentioned, from which data can be extracted. Therefore, new concepts are beginning to be studied, as is the case of Mobile crowdsensing (MCS), which has become promising to facilitate road detection applications. MCS is a technique in which many people have mobile devices capable of detecting and computing collectively, sharing data, and extracting information. We want to use opportunistic crowdsensing; this consists of the data being detected, collected, and shared automatically without the user's intervention [24]. The MCS also helps achieve the goal of urban sensing, leveraging users' mobility, sensors embedded in mobile phones, and existing wireless infrastructure to detect and collect data. This way, diverse urban data can be economically detected, even in regions not yet covered by a specialized detection infrastructure. [40].

This research follows the line of methods based on vibrations through different smartphone sensors, collecting data through the concept of MCS.

Measuring the signals from smartphone sensors is challenging as there are differences in sensor properties between various smartphone models and in size, weight, length, and suspension systems. As for the anomalies, which in this case study will be the potholes, we will also find a variety in length, depth, and shape of the potholes in the road.

2.7 Vibrations

When we refer to any movement that is repeated after a while, we will define it as: vibrations or oscillations. The vibration theory deals with different studies where the oscillatory motion and forces are associated with an object.

Vibratory systems are composed of different elementary parts, which include: the storage of potential energy, conservation of kinetic energy, and a means by which energy is gradually lost. The vibration occurs when potential energy is transformed into kinetic energy and potential energy alternately. If the system is damped, a part of its energy will decrease in each vibration cycle.

2.7.1 Classification of vibration

Vibration classification can be given in several ways. Some of the essential classifications are as follows. [31]:

- **Free vibration:**. When a system is allowed to vibrate on its own after an initial disturbance. No external force acts on the system.
- **Forced Vibration:** This vibration happens when a system is subjected to an external force. The oscillation that appears in machines such as motors is an example.

- **Damping and undamping vibration:** It is called undamped vibration if no energy is lost by friction or other resistance during oscillation. However, when energy is lost, it is damped.
- **Linear and Nonlinear Vibration:** If all the aforementioned essential components of a vibrating system behave linearly, it is a linear vibration. Nevertheless, if any essential components behave nonlinearly, it is a nonlinear vibration. Vibratory systems tend to behave nonlinearly with increasing amplitude of oscillation.
- **Deterministic and random vibration:** If the force or movement that acts on the system can be known at any time, it is deterministic. On the other hand, when those values cannot be predicted, it is non-deterministic or random. It is possible to estimate averages as the excitation's mean or mean squared values.

2.8 Sensors

The phone's sensor frame can access various types of built-in sensors. These sensors can be classified into two: based on hardware (physical) and software (virtual) [19]. Software-based sensors use data from one or more sensors to be able to record and calculate values that are occurring in real time. In general, smartphone sensors are classified into: motion sensors and position sensors [20].

- Motion sensors: are recommended to be used to monitor movement, such as vibration, tilt, jolt, rotation, and rocking of a device. The movement of the device is monitored relative to the local level coordinate system with the physical environmental conditions.
- Position sensors: are specifically used to specify the physical position of a device locally. As for the contribution, it turns out to be minimal at the time of detection of different anomalies.

2.8.1 Accelerometer

Accelerometers are sensors used to measure acceleration. Following the logic of its name, it is an instrument to measure the acceleration of an object to which it is attached; it does so by measuring an internal inertial mass[9]. They are inertial sensors that measure the second derivative of the position. Also, it measures the inertial force generated when a mass is affected by a change in speed. There are several types of technologies and designs that, although they all have the same purpose, can be very different from each other depending on the application for which they are intended and the conditions in which they have to work.

2.8.2 Gyroscope

The gyroscope is a mechanical device formed by a body with rotation symmetry. That rotates around its axis of symmetry and whose axis of rotation is not fixed; it allows us to know an angle in time while it is rotating (angular velocity), which allows us to determine the behavior of the mobile in which it is mounted.

Type	Measurement range	Bandwidth	Advantages and Disadvantages	Application
Thermal	1.5g - 250g	0.1 - 1500	- High sensitivity - Average cost - Easy use - Low temperatures	- Impact - ABS - Airbag
Piezoelectric	0g - 2000g	10 - 20000	- Medium sensitivity - Complex use - Low temperatures - Does'nt work continuously	- Vibration - Impact - Industrial use
Piezoresistive	0g - 2000g	0 - 10000	- Continuous and alternate response - Low cost - Minimum size and weight - High sensitivity	- Vibration -Impact - Automotive - Flight tests
Capacitive	0g - 1000g	0 - 2000	- Runs continuously - Low noise - Low power - Low cost	- General use - Industrial use - Alarm systems
Mechanics	0g - 200g	0 - 1000	- High precision in continuous - Slow - High cost	- Inertial navigation - Tools - Leveling

Table 2.3: Types of accelerometers

Gyroscopes used to have a weight and size with which it was complicated to take advantage of their utilities. Currently, through Micro Electro Mechanical Systems (MEMS), size and weight are no longer a concern since it works based on miniaturized electrical and mechanical components. There are different types of gyroscopes[17]:

- Rotary: It uses a mass rotating on an axis, supported by one or several gimbals, depending on the desired degrees of freedom; the precission is minimal; thus, the axis is kept stable, and thus it will always point in the same direction.
- Vibratory: they have a vibrating element that, when rotated, is affected by the Colioris force, which causes secondary vibrations octagonal to the original.
- Optics: use the Sagnac effect to detect rotation.

2.9 Random vibration analysis

Random vibrations are those of which it is not known exactly what causes them. In addition, the magnitude and duration cannot be determined with certainty, so defined functions or values of interest are needed.

Mechanical vibrations are 'random' because of the activities that cause them. The proposed methods for evaluating vibrations assume that the motion is stationary, so a representative average over the acquisition period can be used. That is the theoretical way, but in practice, the conditions change from one moment to another, so it is necessary to occupy short periods considering the static data. There are also non-stationary data, called transient signals, caused by phenomena of concise duration, with a well-defined start and end: impacts, blows, Etc.

Due to our problem's characteristics, the analysis type must be clearly defined as non-linear random vibrations.

2.10 Parameter extraction and pattern recognition

This stage is the procedure by which we reduce the attributes of an entity, retaining the information related to established groups or classes. In this case, since they are signals from the real world, we must make a practical selection to capture the desired characteristics in the best possible way instead of the pure signal.

Anil K. et al.[25] in their work said that effective selection identifies and selects the best features of the input data. Limiting the features simplifies the representation of the input pattern and the classifier; this leads to a faster and more efficient system. Nevertheless, if it made a considerable reduction, it can cause the information to be lost and the reduction of accuracy considerably. The authors also mention that a system requires attention to the following characteristics: definition of pattern classes, representation of patterns, extraction, selection of parameters, design and training of the classifier, testing, and evaluation

Chapter 3

State of the Art

Potholes, the primary type of anomaly in several countries, have become a case study for detecting pavement deterioration. For this reason, detecting these anomalies becomes relevant to form part of road and highways' prevention, maintenance, and rehabilitation strategy.

One way to do this is through manual detection and evaluation. Currently, there are different methods and technologies to analyze and monitor the state of roads, thus detecting and evaluating automatically and achieving an efficient evaluation methodology. These methodologies are done through different types of devices. They rely on one or more cameras to make observations of the road, also through scanners and otherwise through vibrations. Current researches to automate pothole detection can be divided into: vision-based, vibration-based, and 3D reconstruction-based methods [8, 34, 33].

In this chapter, we will have a description of some of the works related to the issue of fault detection on the roads employing sensors in a cell phone. The chapter will consist of two subchapters. In subchapter 3.1, we will talk about the different methods that currently exist to detect potholes, while subchapter 3.2 delves into vibration-based techniques, mentioning capture techniques, as well as machine learning that has been used for this type of case. The results of these techniques are also described in this subchapter. This chapter concludes by mentioning our ideas about the techniques and technologies that were proposed for this work.

3.1 Comparison of Pothole Detection Approaches

The vision-based methods have as input data; images or videos, by which it is determined if there is any type of anomaly on the road, this is achieved through image processing and deep learning. This type of approach turns out to be more affordable than the 3D reconstruction approach, it also has the advantage that the approximate number and shape of the anomaly can be calculated. Since it uses two-dimensional information, it has the disadvantage of a limitation in the measurement of volume and depth. [8, 34]

A vibration-based method, using data retrieved from acceleration sensors, determines the existence of potholes and can predict pothole depth. Vibration-based methods are the most profitable among the different methods. It has advantages such as requiring little storage for data acquisition, and real-time data processing can be applied. It has different limitations, like extracting the correct shape of the potholes. Since they only analyze the vibration information from the sensors, there may also be complications during the data acquisition process. [34, 33]

Finally, we have methods based on 3D reconstruction, which can predict the shape and volume of potholes based on stereoscopic vision technology. This class of technologies predicts the potholes' shape and volume with greater accuracy among the three methods. As a limitation, it is the most expensive method to detect potholes compared to the other two, apart from the fact that they find it challenging to recognize potholes covered with water or dirt [34, 33].

The characteristics of pothole detection approaches are presented in Figure 3.1.

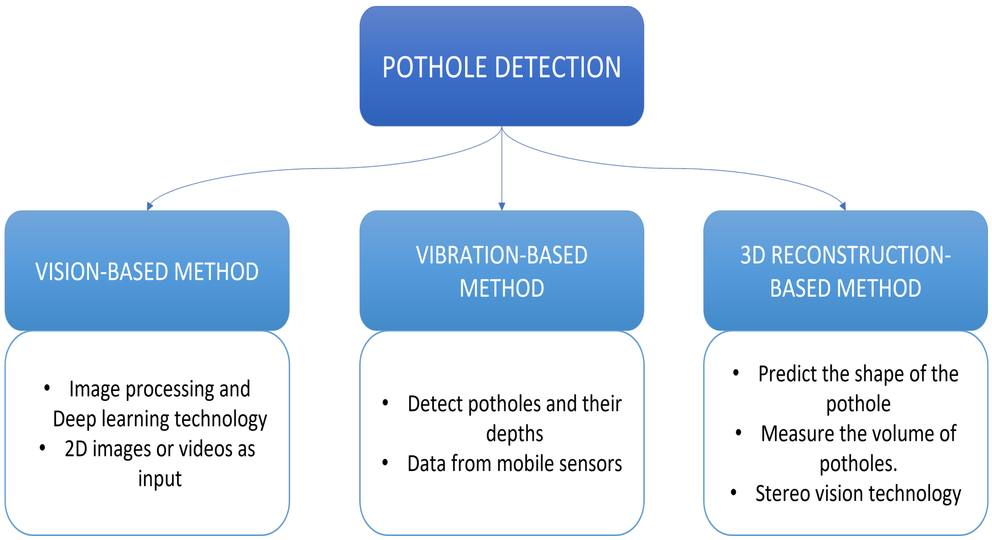


Figure 3.1: The characteristics of pothole detection approaches.

Methods	Strenghts	Weaknesses
Vision-based method	More profitable than 3D reconstruction. - Can calculate the number and shape of potholes.	- it is difficult to measure information, such as the volume and depth of potholes - Lighting and shadows affect this detection approach.
3D reconstruction-based method	- Determines the shape of the pothole more accurately among the three methods.	- It is a much more expensive method.
Vibration-based method	- It is the most profitable method. - Little storage - Can be applied in real time	- You cannot provide the exact shape of the potholes. - There are external factors that affect its performance.

Table 3.1: Strenghts and Weaknesses of methods

3.2 Vibration-based pothole detection.

This class of methods used to detect potholes generally uses the output of 3-axis accelerometers and GPS to locate the potholes, usually via smartphones. Detection happens in such a way that when the wheels of a car go through a pothole, the accelerometer output gives different output than in regular operation (fluctuation that is different from zero), and thanks to this sudden movement, the existence of a pothole. However, some movements include punctures, sudden braking, railway crossings, Etc. These changes also represent fluctuations in the accelerometer values since the road or driving conditions are different, which must be controlled or classified, since it can cause the method to make classification errors. These sensors are cheap and generally included in all current cell phones. The algorithms for this type of problem require comparatively low calculation costs; because potholes and other factors do not have specific properties, the methods are error-prone, mainly when they produce false alarms. Also, drivers run away when going over potholes, and the sensors cannot produce the necessary results. Vibration-based methods offer many essential advantages: they require small storage, can be used in real-time processing, can detect road distresses at any time of the day, is profitable and less costly, have high accuracy, require no-visual inspection, can be automated [26, 5, 41]. This method uses force, rotation, and orientation technology with a low response time. Those are why Vibration-based methods are the main focus to be investigated.

3.2.1 Related Work

Seraj et al.[36], in this work the data collection was done with a Samsung Galaxy S2. The collection setup was established in such a way that the cell phone was fixed to the windshield of the car. All data collection was done in two different countries: The Netherlands and Albania. This was done through five different cars. The sensors used in this work were; an accelerometer, GPS, and gyroscope. The raw data labeling process was done manually, using audio and video. In the case of classification, they used SVM with an RBF kernel. The characteristics in terms of time and frequency were calculated with an FFT algorithm, which included a Hamming window function and on the other hand, the decomposition of the signal was carried out with a stationary wavelet transform. Regarding the characteristics extracted, they were made from two sets: time and frequency. Regarding time, the following were extracted: standard deviation, variance, mean, peak to peak, mean of absolute values, zero crossing rate, the correlation between axes, inclination angles, area of magnitude, and duration of the signal. On the other hand, what it is concerning frequency were: the average frequency and energy of the frequency bands. The authors mention that during the experimentation, tests were made with unprocessed signals and with demodulated signals, to see which had the best performance. Finally, to use the model they used a training base with 3066 windows and one window with a 66& overlap. Regarding the classification process with 2 steps site. In the first step, all the windows were processed and in which anomalies were detected, they went to the second step. In the second step, another classifier classified the type of anomaly in each window. A 10-fold cross-validation was performed to train and test the classifier. The results were that the demodulated functions were more accurate than the unprocessed signals, giving the modeling a classification accuracy of around 90%.

In Sri Lanka, there was a project to create its road network, so they are creating new roads every year, leaving the old roads without maintenance service. In the country, public transport buses have different tools that monitor different parameters of their environment; this project is called BusNet. Originally this project was designed to measure air pollution using various sensors. However, later, De Zoysa et al.[10] proposed adding accelerometers to the network that already exists, which is effectively BusNet, in order to be able to monitor the condition of the road surface. For the development, they used a laptop to collect the information thrown by the accelerometers mounted on the bus network. The BusNet data transfer

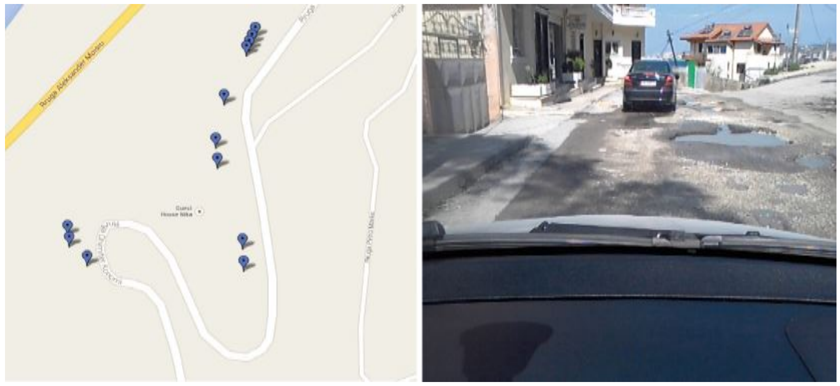


Figure 3.2: Snapshots of map and video anomalies Source: [36]

protocol is implemented and is currently undergoing testing. BusNet lowers the cost of deploying sensors, replacing them with fewer mobile sensors. It also solves the management, maintenance, and security issues associated with a sensor network deployed in a large field.

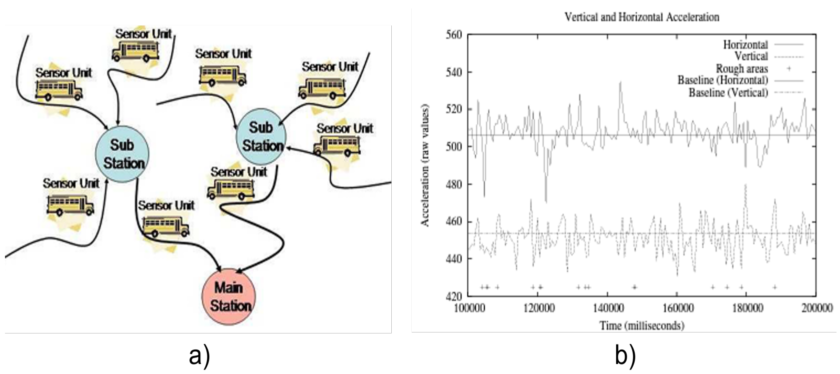


Figure 3.3: a) BusNet System Architecture and b) Horizontal and vertical acceleration for a stretch of road. Adapted source: [10]

Hoffmann et al.[21] in this work the authors chose to use a machine learning classification approach using bicycles. Occupying a Nokia 5800, for data collection. The device was mounted on the handlebars of the bicycle, the data was recorded using the cell phone's accelerometer and GPS, at a frequency of 37 Hz. Three different types of surfaces were distinguished: smooth, irregular or rough. Two types of classification processes were carried out: direct classification and classification based on the detection of potholes. Direct classification refers to the classification of the surface into one of the three aforementioned categories. During this process, the data that was extracted from the sensors were: speed, inclination, mean and variance of acceleration, and standard deviation. During the process, each road section was divided into sets of segments, to later classify each of these, taking into account the characteristics of the previous segments. During the feature optimization stage, they obtained as a result that the speed and inclination decreased the precision or did not contribute anything to the classifier. In this work they used two types of classification algorithms: KNN and Naive Bayesian Classifier. They performed a 10-fold cross-validation to test behavior on the evaluation data. At the end of the process, they had as a result a very similar performance between both classifying algorithms, but the KNN turned out to be superior. However, the overall accuracy of the direct process of 78% turned out to be very unsatisfactory. On the other hand, the potholes approach, where the classifier had to distinguish between a smooth path and a path with potholes or another similar anomaly. In this process the precision was much higher, with a result around 98%.

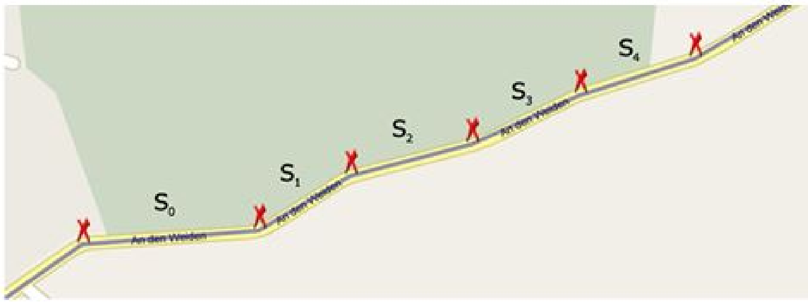


Figure 3.4: Segmented track Source: [21]

RonghuaDu et al. [14] decide to take for this work an approach based on the recognition of anomalies in roads, through the accelerometer of a smartphone, when the car passes through an irregular road surface. The methodology they proposed is divided into three stages: data acquisition and preprocessing anomaly recognition, and classification. Through the accelerometer and GPS, they were in charge of collecting different types of information, including the speed, acceleration, and position of the car. The raw data were preprocessed with a Butterworth filter. After implementing an improved Gaussian model, they recognized the different anomalies, having as reference the z-axis of the accelerometer. The training and test data were collected independently, to differentiate each of the data groups and maintain independence. To classify the different types of anomalies, the KNN algorithm was used, managing to classify the types of anomalies, including bumps and potholes. Finally, in obtaining results, the authors compared the precision of the results of other works against those of identification of the proposed method. The test result shows that the accuracy of the pothole classification is 94.12%.

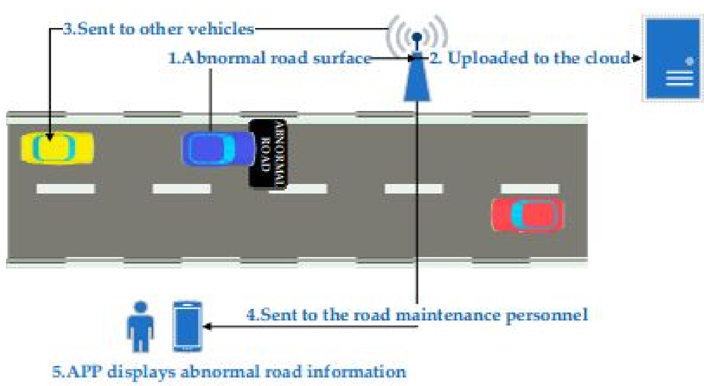
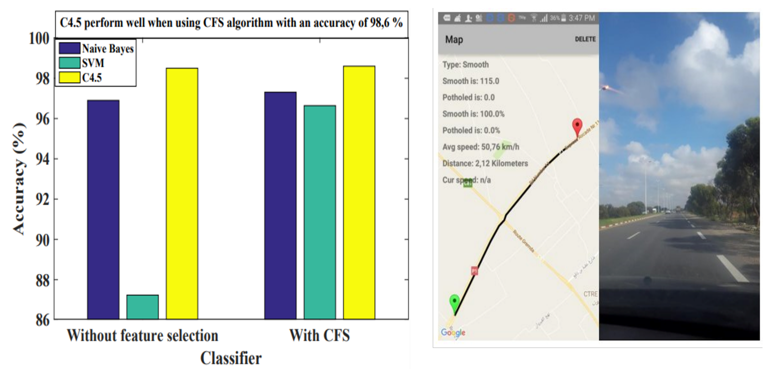


Figure 3.5: Results of the model by Ronghua Dut al. Source: [14]

Azza Allouch et al.[3] opted for a road anomaly classification approach, using an accelerometer and gyroscope, incorporating machine learning techniques. The approach is divided into two general phases: training and prediction. The training phase was subdivided into four steps: data collection, preprocessing, feature extraction and selection, and training of the selected classifier. The data was collected by means of two sensors: accelerometer and gyroscope, to later process them with a low-pass filter. Once the data was processed in the time domain, they were transformed to Fourier coefficients, on the other hand in the frequency domain, they used the Fourier transform to extract specific characteristics from the data. The extraction of the most important characteristics was done by means of a correlation analysis. Once selected, these characteristics were used to carry out the training of the classifying model. The selected classification algorithms were: C4.5 decision tree, SVM and Naive Bayes. The threads for the prediction are the same as for the training phase. Path anomaly prediction was achieved by loading the classifier model. The performance measurement was made through the metrics: accuracy, precision and recovery. The metrics showed that the performance of the C4.5 decision tree (0.9860, 0.985, 0.985) was better than the performance of SVM (0.9525, 0.951, 0.953) and Naive Bayes (0.9690, 0.972, 0.969).



a) b)

Figure 3.6: a) Accuracy of each algorithm. and b) Smooth road. Source: [3]

Chao Wu et al. [43] propose a methodology consisting of four general phases: data acquisition, processing, feature extraction, and classification. For the acquisition of data, it was done through the vibrations registered by the sensors included in a smartphone, which were: the accelerometer and GPS. During data processing, the most useful features for pothole detection were extracted, this was done by means of sliding windows with simple threshold methods. The last classification process, the authors decided to use different machine learning methods to train and test the data: linear regression, SVM and random forest. Precision, recall, F1 score, and accuracy were used as performance metrics. According to the experiments, the time and frequency functions turned out to be the best bump detection functions. The Random Forest (RF) was the one with the best performance of the classifiers, with a bump accuracy of 88.5% and a recovery of 75%. In addition, the RF has an accuracy for the test set of 95.7%. In the figure 3.7 we can see the model proposed by the authors.

Eriksson et al.[15], were guided by a signal processing approach and using machine learning, they detected and classified specific road anomalies. This was achieved through computers integrated with acceleration sensors and GPS. Raw data windows are compiled

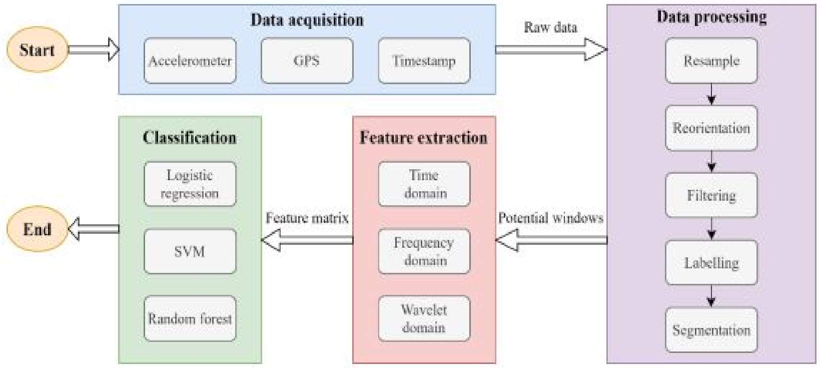


Figure 3.7: Results of the model by Chao Wu al. Source: [43]

380 times per second, including: time, heading, velocity, acceleration, and location. During the experiment they concluded that when the sensor is connected to the car's dashboard, the data signal turns out to be cleaner than in other sections. After real-time manual tagging, done by a person sitting in the car, a total of 7 different event classes were classified. In the work it is mentioned that during the data collection, there were also unwanted anomalies captured by the accelerometer. To avoid these events, the speed characteristic and filters were used. To perform the training, they concentrated on the peaks of the X-axis and Z-axis acceleration values, as well as the instantaneous velocity of the automobile. An important point was how false positives were avoided and precision was increased, and this was thanks to the fact that the anomaly is recorded only when other detectors have also recorded an event in the same place. Thanks to the above process and after an analysis of the data, the result was a false positive rate of 0.2%.

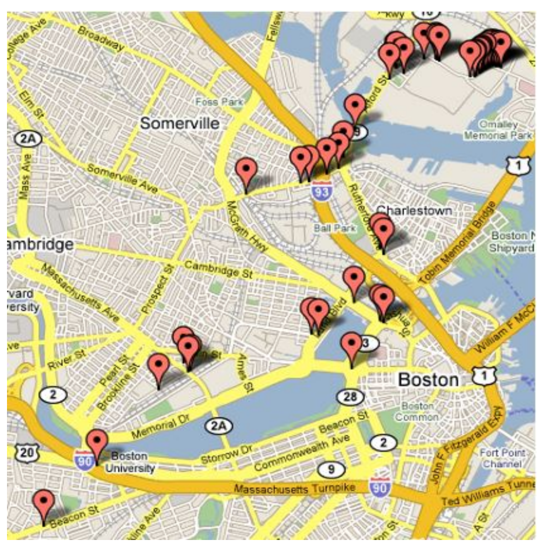


Figure 3.8: The 48 highest-confidence detections. Source: [15]

This thesis proposes to work on some of the aspects of previous works, as well as mix some approaches. One of the approaches that will be addressed will be that of Hoffmann et al.[21], of dividing into three different types of events, since they have quite satisfactory results. This thesis will take an approach based on mounting a data collection system in a car, making it three different categories of events. Furthermore, we observed that the supervised learning approach carried out by Seraj et al.[36], could be successful. This technique will also be applied in this thesis. Using a neural network.

Regarding data collection, as in previous works, a phone with Android operating system and a data collection application developed specifically for this work will be used. An additional function will be implemented, which allows the device not to activate the sensors at all times, only when the values set in a threshold are exceeded, in order to limit the amount of data and have it more isolated. Also, like Eriksson [15], the cell phone holder will be placed on the dashboard of the car, since they mention that it was the place where the best data collection would be obtained.

Chapter 4

Machine Learning

Machine learning is a field of artificial intelligence (AI)[39] and has become the core component of digitization solutions. This is considered the most effective data analysis methodology, in terms of forecasts, through the application and creation of models and algorithms. It encompasses a set of techniques for machine learning, whether in data organization, pattern recognition, or autonomous machine learning. Some application examples may be classification, clustering, regression, and anomaly detection. Depending on the types and categories of data, we will have to choose between unsupervised and supervised learning methods to implement the appropriate algorithm.

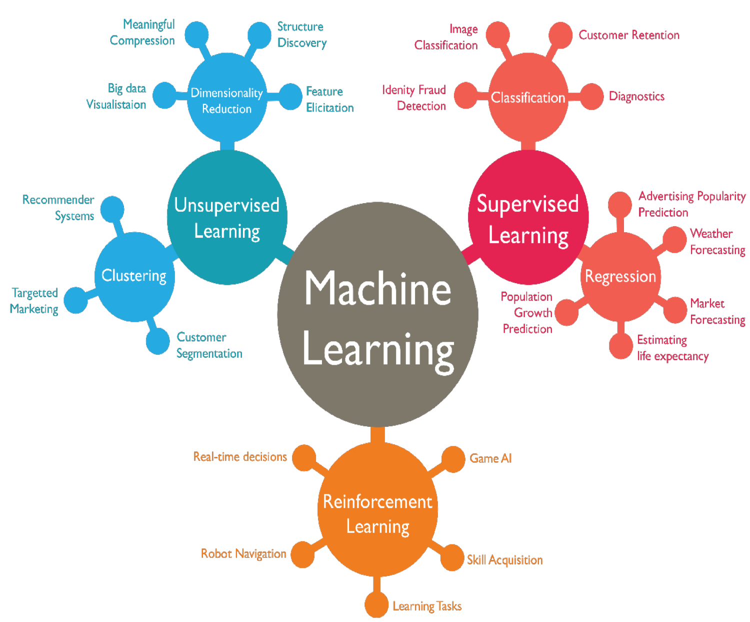


Figure 4.1: Machine learning

In unsupervised learning, it is important to mention that in training, the model will not have input data that is already labeled. In this type of learning, the selected model will try, through statistics, to find hidden patterns in the data set, so that they can finally be classified into the different proposed categories.

In the case of supervised machine learning, the data used contains the input and the desired results. The data set contains a labeled entry. In supervised learning, there are different algorithms, but the most common are SVM, decision trees, and neural networks.

Vapnik. et al.[42] suggest that SVM can be implemented on non-linear data. To define this case, it is necessary to develop a hyperplane. According to the definition of SVM, by dividing the training data into *n* classes, the SVM algorithm will classify these *n* classes.

On the other hand, in the area of Machine Learning, we find Deep Learning; sophisticated models for machine learning are used in this area. This class of algorithms uses neural networks since they try to imitate the behavior of biological neural networks, and their main application is with classification tasks.

4.1 Neural Networks

This part of the project is the central part of the detector and what are its fundamentals. As we move forward, the content is specialized; we start with the most general neural networks and are concretizing until we reach a neural network model.

4.1.1 Neurons

The human brain has between 100,000 million and 1 billion of neurons. The cells are electrically excited to receive, process, and transmit information. Connections called synapses are what allow signals to occur between neurons. Neurons have three main parts: cell body, dendrites, and axon. Neurons are interconnected with each other through these synapses, which are the junction of the axon and the dendrites [28].

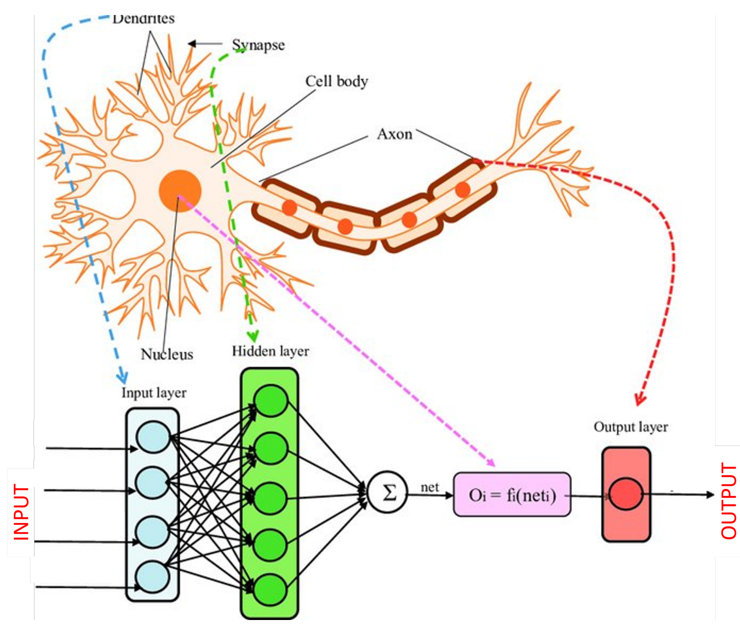


Figure 4.2: Computational neuron adapted from a biological neuron. Source: [18]

Figure 4.2, we can see an example of a simulation between the learning mechanism of the human brain and a computational neural network.

In the computational model of a neuron, the signals traveling along the axons (x0) interact multiplicatively (x0 w0) with the dendrites of the other neuron as a function of the synaptic strength at that synapse (weights 'w'). A biological neuron is modeled through the activation function of the perceptron, where the impulses from other axons are summed, and the sum is evaluated according to the chosen activation function.

4.1.2 The Neural Networks

An artificial neural network is a computational model formed by elements or nodes that try to imitate the human brain as functional to achieve automatic learning. This is achieved from neural layers, where input is validated, and a classified result is provided at the end [13].

In the basic neural layered structure shown in Figure 4.3, the first layer is made up of the input neurons, the next layer is the hidden layer, where we can have multiple layers of neurons, and the last layer is the output layer.

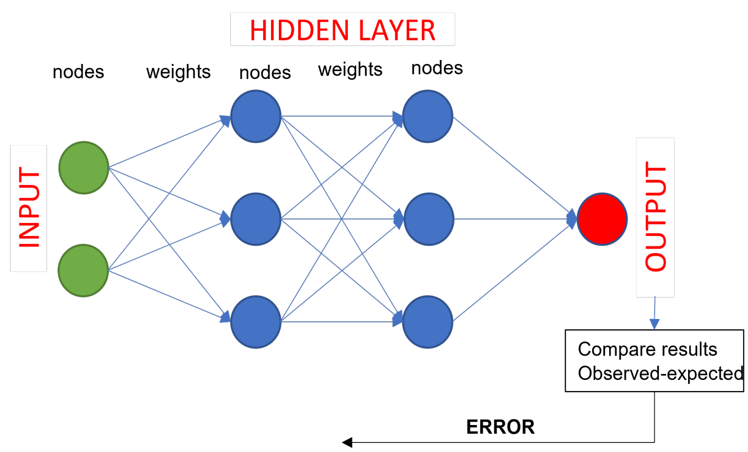


Figure 4.3: Simple Neural Network

Neurons are usually Fully-Connect, meaning that all neurons in one layer communicate with all neurons in adjacent layers. The number of neurons tends to decrease, higher in the input layer and lower in the output layer. Neuron activation results from thresholds and weights assigned to each layer and each connection or link between neurons. This layered structure and the assignment of these weights allow us to recognize patterns.

4.1.3 Types of Networks

They can be classified according to their topology, architecture (Monolayer or Multilayer), and their learning (Supervised or Unsupervised); the types can be [35]:

- Networks with supervised learning: it is necessary to have input data and the desired or reference output values; in this type, the network will learn by correcting and updating the weights that best achieve the desired output.

- Networks with unsupervised learning: here, there is input data, but not the output reference, and here the network will look for patterns or similarities between the input data in order to segment and classify.

- Networks with reinforcement learning: this type learns through feedback; that is, the reference has to be created with the approval or disapproval of the desired result; this makes the network adjust its precision to our responses.

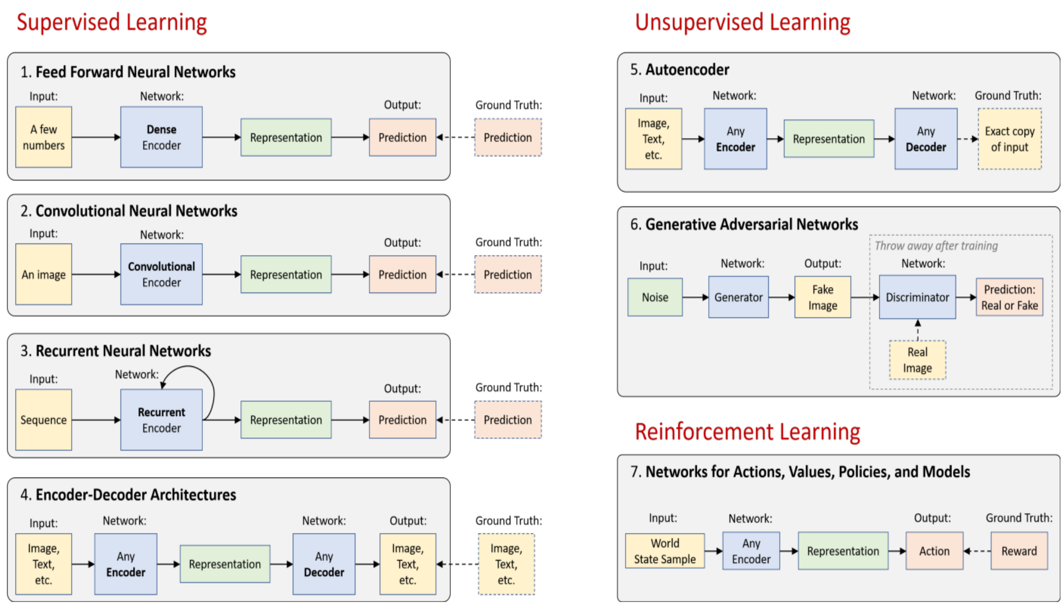


Figure 4.4: Types of Neural Network [35]

4.1.4 Factors Affecting the Success of Neural Networks

As we have seen, neural networks have multiple layers. Each layer comprises a series of interconnected nodes with associated activation functions and other essential factors such as weights and cost functions. All these factors together serve to increase efficiency and performance.

4.1.5 Activation Functions

Neural networks are structures where we find multilayer neurons, which are made up of nodes, which will have the task of classifying and predicting the data that we provide as input to the network. All layers contain nodes, and each of them handles a weight considered when processing information from one layer to the next.

If an activation function is not implemented in the neural network, we will simply have a linear function as output data. We know that a linear equation is very easy to solve, since the complexity of this type of equations is limited and limits the model a lot when it comes to learning and processing complex things from data. Having the absence of activation functions in a neural network, the model will only be constrained so that it will only be linear regression with limited performance and power. Therefore, in order to achieve a final goal and create a model that can model different types of data, with different levels of difficulty and much more complex architectures for knowledge extraction, it is necessary to occupy and select activation functions and neural network techniques. artificial.

We must bear in mind that neural networks calculate and learn any function, since they are finally universal approximations to any problem. So when we introduce an activation function, it gives us the ability to increase the complexity of data extraction. So that an activation function can be differentiable, since this will allow us to create weight loss strategies and their optimization to reduce errors. There are different activation functions that can be:

1. Binary Step Function
2. Linear
3. Sigmoid
4. Tanh
5. ReLU
6. Leaky ReLU
7. Parametrized ReLU
8. Exponential Linear Unit
9. Swish
10. SoftMax

In conclusion, using an activation function, through the processing of the inputs, which are transformed and extracted, we will have a unitary activation as a result.

4.1.6 Cost Functions

Within neural networks, it is vital to define their purpose. So a cost function is defined, a measure that shows how well the network is trained to its target input and output training data. The main objective is to minimize the cost function so that the output is as close to the desired one as possible. To achieve this optimization, the Gradient Decent is used [2].

The gradient is calculated according to the cost function, and concerning the calculated gradients, the weights are updated. The cost functions must be written as averages to calculate gradients and depend only on the outputs. There are many cost functions used, such as:

1. Sum squared error (SSE)
2. The cross entropy (CE)
3. The exponential cost (EXP)

Marquis of Sa. Joaquim P. et al. [27] They explain and compare SSE and CE in terms of their impact on hidden layer reconstruction performance in networks. CE produces very few errors, and SSE provides the best-layered reconstruction performance.

Chapter 5

Methodology

This chapter provides a description of the design and implementation of the road surface classification model and data collection mobile application. The chapter is divided into five subchapters. Subchapters 5.1 show us the framework that will follow all our experimentation, 5.2 focuses on applying data collection to train and test our model, while subchapter 5.3 focuses on filtering the data obtained to have the best base. Training subchapter 5.4 shows the model road surface classification that was performed. Finally, at 5.5 we have the evaluation and classification performance of our model.

5.1 Anomaly detection using smartphone sensors

The methodology to be used in this work according to a road anomaly detection approach will consist of five general stages (Figure 5.1): (1) sensing (data collection), (2) preprocessing, (3) processing ,(4) Post-processing and (5) performance evaluations.[32].

5.2 Data Collection

This section provides an overview of the data collection application's design, architecture, and implementation.

The data collection was through a mobile application created in Android Studio, the programming language used was Java and it was designed to run on smartphones or tablets with the Android operating system. The application was configured for cell phones that have built-in accelerometer and gyroscope, it is very important to emphasize that this mobile application was created and used specifically for this work. Another important point is that it was only downloaded and used on a single smartphone. Most Android devices contain by factory default different types of sensors that allow measuring movement, orientation, etc. These different types of sensors can provide data, which will allow us to monitor different movement behaviors, which can have high precision and accuracy.

The Android operating system has three broad categories of sensors[20]: motion, environmental and position sensors.

The developed application is specifically based on motion sensors, which are the ones that measure acceleration and rotation forces in three axes, more specifically we are going to

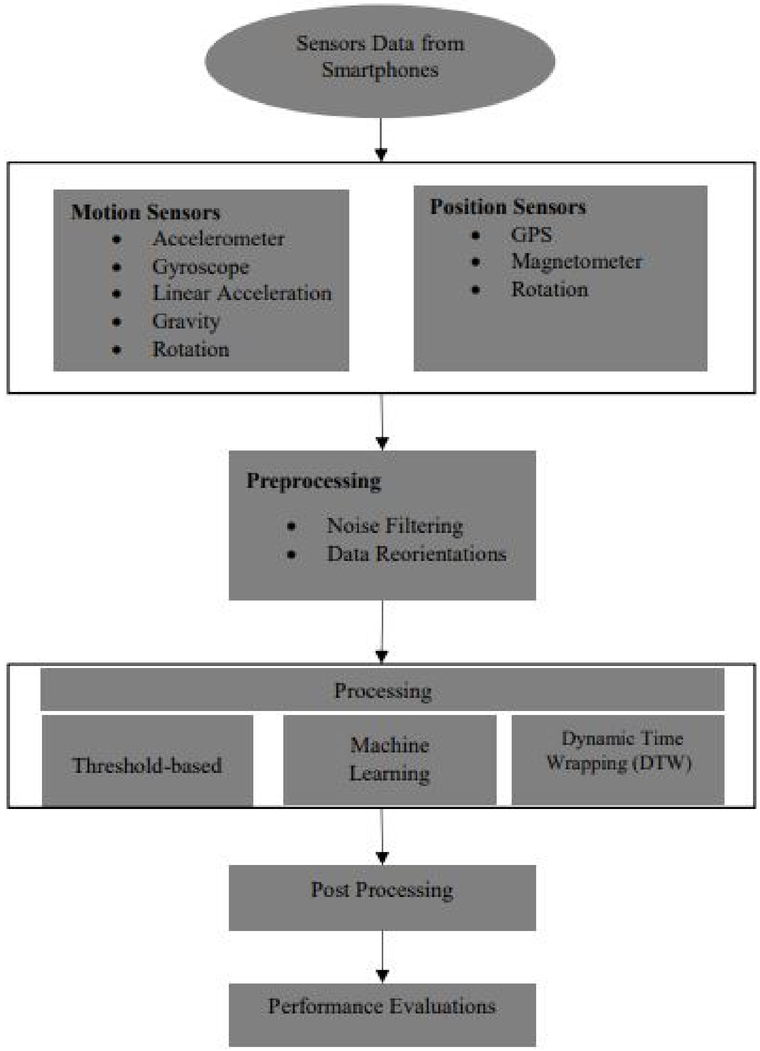


Figure 5.1: Overall process for road surface anomaly detection using smartphone sensors[32]

use two sensors: an accelerometer and a gyroscope. The sensors included in a smartphone can depend on the model and the manufacturer, and these can be of two types: software or hardware. Software-type sensors are derivatives of hardware-type ones, while hardware sensors extract data directly from the sensor. The sensors used in this case are of the movement type, which provide us with information that can help with specific things; in this case, it will be for pothole detection. The gyroscope can give us the measurement of the acceleration, with respect to gravity, this can be useful for the reorientation of the device in relation to the Earth. On the other hand, the accelerometer will provide us with acceleration data, excluding gravity. This can be used for a task like motion detection.

5.2.1 User interface

The application for collecting raw data is developed under a simple user interface. Figure.5.2 illustrates the main and initial view of the application, on the other hand Figure. 5.3, we can see a view of what the mobile app looks like when recording data. Remember that it is a mobile application that, as mentioned in previous sections, is only designed for data collection, which will serve as input for our model, so this task will be done manually. So every time the user fell into one of the possible scenarios (pothole, bump, normal), he had to click on the buttons that we can see at the top of the screen.

According to the presented event, data is saved in different tabs, so once each record is finished, the data can be copied simply by pressing and holding the tab from which the data will be copied. This function records tagged acceleration data from different road surfaces, which the road quality detection model would then use.

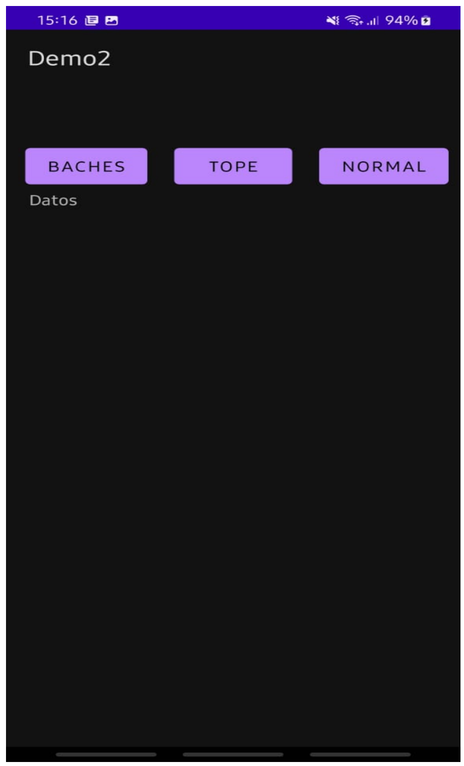


Figure 5.2: The main view of the app of Data Collector

5.2.2 Data recording

A concept that is important to explain is that of SensorEvent, which is an object in the Java language, which represents that the sensor detected an event and therefore contains the data created by the sensor in use. Every time the sensor receives a change or an update of values, a new SensorEvent is created. To start and monitor SensorEvents, Android Studio has the Android SensorManager class which contains four different types:

- SENSOR_DELAY_FASTEST
- SENSOR_DELAY_UI
- SENSOR_DELAY_GAME

Figure 5.3: The main view, while collecting data

- SENSOR_DELAY_NORMAL

These ranges of use can be variable, so a slightly different solution was implemented for the road surface data collector. In this work, the SENSOR_DELAY_FASTEST type was used, which can give us less variability in the data collection. Every time a new event (SensorEvent) is received, the previous value for that sensor will be overwritten. Using the sensors at their fastest execution interval in this work shows that occupying a Samsung Galaxy S21 Ultra with Android version 11, the interval between consecutive SensorEvents from each sensor is low enough while sampling is done at 10 Hz.

According to previous works on the subject, they used data windows based on time; in this work, a different way of creating these windows is without the need to collect an excess of data, creating data windows of 238 values per axis of each sensor used. This is achieved by adding the absolute values of the three axes per sensor, having a threshold of 10, meaning that if this sum turns out to be less than the threshold, no event will be recorded. However, if it turns out to be equal or greater, the event will be recorded; observe it more explicitly below:

$$sum = |x| + |y| + |z| \tag{5.1}$$

$$threshold = 10 \tag{5.2}$$

$$\text{if sum} >= threshold \tag{5.3}$$

Then, Save sum in window

$$window = \begin{bmatrix} x_1 & y_1 & z_1 \\ . & . & . \\ . & . & . \\ . & . & . \\ x_2 38 & y_2 38 & z_2 38 \end{bmatrix} \tag{5.4}$$

Every time we want to record a type of event, it is crucial to select which one it belongs to; this can be seen in Figure 5.3, where the location of the buttons is available at the top of the screen. We can also observe an example of data that has been recorded, and we can also observe the format of the data.

5.2.3 Collection Setup

We worked with a 2006 Nissan Altima vehicle (Figure. 5.4) with five seats. The collection configuration was done through the smartphone, which ran the application that was previously made, while the phone was kept with a support on the car's dashboard. This can be seen in Figure. This position helped minimize the movement of the cell phone and, consequently, the noise in the measurements. No more restrictions were introduced since, in this way, the actual application is more feasible and reproducible. The experiments were done in different parts of Mexico City and the metropolitan area, trying to do it when there was a low traffic load and an approximate speed of 25 ± 5 km/h was maintained.

Figure 5.4: Vehicle used in the investigation

5.2.4 Chosen Roads

Deciding the characteristics that the roads should have to obtain essential data was done through the investigated literature, reference works, and own experience. Therefore, three scenarios were finally divided (pothole, bump, normal). Different paths were traveled and recorded to obtain the necessary and correct data, which would be necessary to build our database of each event. In Figure 5.5, we can see the data collection routes with three different colors, each corresponding to the event that occurred the most on that street or avenue. The red color belongs to the potholes, the blue to bumps, and the green to normal roads.

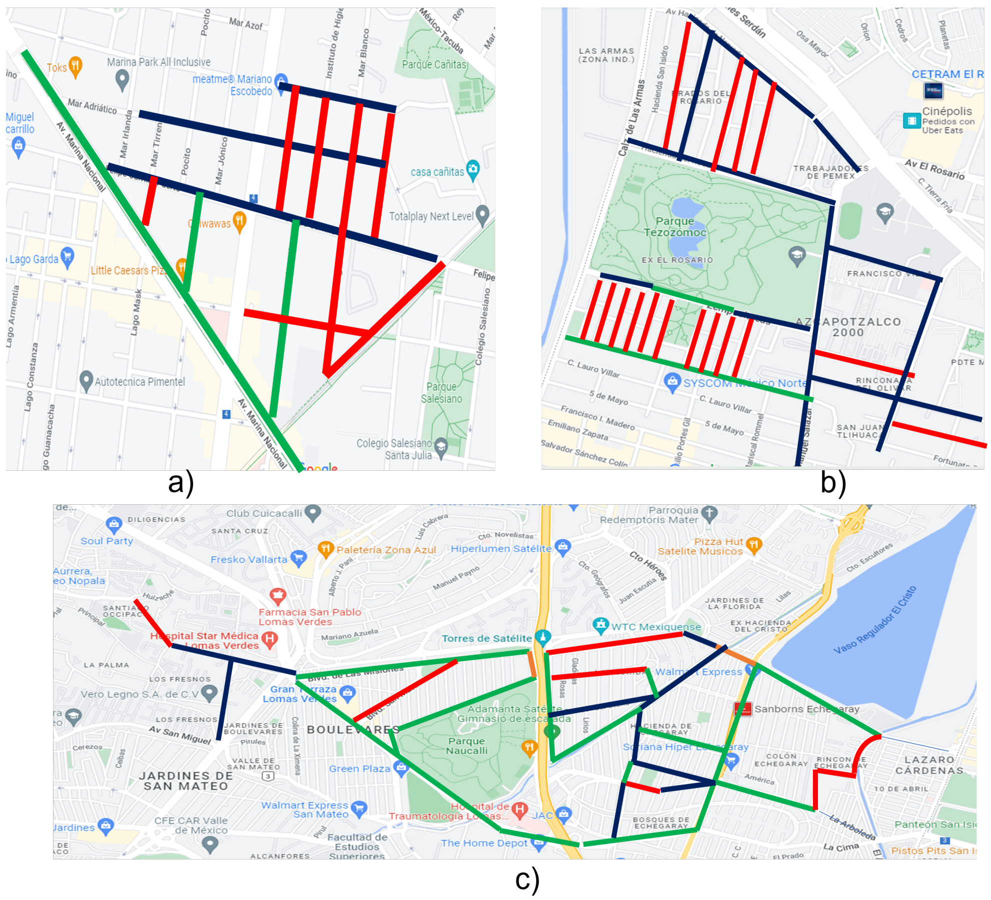


Figure 5.5: Areas, where the data collection was carried out. Images a) and b) show different sections of CDMX, while image c) is a trip to the outskirts of the State of Mexico. The segment colors vary from green, red, and blue, describing the type of surface sampled as: green - normal, red - bump, blue - pothole.

For the 'bump' type, a clear example that can have satisfactory data is like the one in Figure 5.6(a). For the 'potholes' type, different sizes and types of bumpers were chosen to have variety in the data; an example is Figure 5.6(c). Figure 5.6(b) shows us the 'normal' type, which refers to streets that may be rough or not in good condition, presenting slightly damaged surfaces with a moderate frequency, which, when circulating, provide acceleration data.

(a) A road with potholes.

(b) A road with conditions that turn out to be normal in Mexico.

(c) A road with bumps.

Figure 5.6: Examples of the different types of classified events.

5.2.5 Final Dataset

This section will explain how our database is composed before being processed and occupied by our selected model. We made three different datasets, each corresponding to the type of event selected, which has been mentioned in previous sections. Each dataset was obtained by exporting the data obtained through the collection app developed; these data were converted into '.csv' documents. Each dataset contains six columns; we can see an example in Figure 5.7, each one corresponding to the three different axes of the occupied sensors (accelerometer and gyroscope); the columns are the following:

- AX (x-axis of the accelerometer)
- AY (y-axis of the accelerometer)
- AZ (z-axis of the accelerometer)
- GX (x-axis of the gyroscope)

- GY (y-axis of the gyroscope)
- GZ (z-axis of the gyroscope)

	AX	AY	AZ	GX	GY	GZ
0	-0.606590	1.560097	-1.025081	0.215254	-0.022984	0.097051
1	-0.540882	2.096340	-0.096130	0.217086	-0.012599	0.098884
2	-0.575768	0.995061	-0.984110	0.197539	-0.007101	0.100106
3	-0.545888	1.004465	-1.026807	0.184710	-0.010156	0.100716
4	-0.542176	1.004201	-1.047549	0.160887	-0.020540	0.101327

Figure 5.7: Example of 'pothole' database.

The 'pothole' database has a total of 25,704 data points per axis, so as explained, this is an equivalent of 108 events, each of 238 data points per axis. On the other hand, the 'bump' one has a total of 23,086, equivalent to 97 events, and finally, we have the 'normal' one with a total of 33082, equivalent to 139 windows. One important factor to emphasize is that these chosen roads are for the experimental purpose of this thesis. However, other vehicles probably perform differently due to different factors, such as the type of driving, car suspension, size and weight of the vehicle, and its tires. This factor needs to be considered for future jobs.In Table 5.1, we can see a resume of this.

Name of Database	Number of Data	Number of events
Pothole	25,704	108
Bump	23,086	97
Normal	33,082	139

Table 5.1: Databases

5.3 Pre-Processing

This stage will consist of transforming the raw data that was obtained from the aforementioned sensors. The purpose will be to convert that data into a clean and organized data set. Our objective will be to smooth the data and filter the noise¡ that is presented in the different measurements. To reach this result, different forms of smoothing and filtering can be used, this time we choose the digital filters that are generally used for two purposes: separation of signals that have been mixed and restoration of signals that may be distorted by the type of road from which they were drawn. [37]

In this work, we mention three types of events that represent different types of road anomalies, which will be used for the learning process.

Python is the programming language that we selected to implement this algorithm. We choose this language for its excellent support in science and research. It allows us to use different libraries, packages, and extensive IDEs available, which will help us create applications for scientific purposes. This thesis uses Pandas, NumPy, TensorFlow, and the matplotlib library. Using NumPy for data structuring aspects, with Tensorflow, we implement our neural network implementation. Then we used the Matplotlib library to create visualizations of different results.

First, the databases mentioned in the last section were selected, divided, and uploaded to Python. Before we start modifying and manipulating the data, we must take into account that we have different amounts of data for each of the databases; remember that for the pothole database, it is 25,704; for the 'bumps' database, it is 23,086 and finally the 'normal' base with a total of 33,082. Figure.5.8, shows an example of how the three databases are out of phase since they contain different amounts of data.

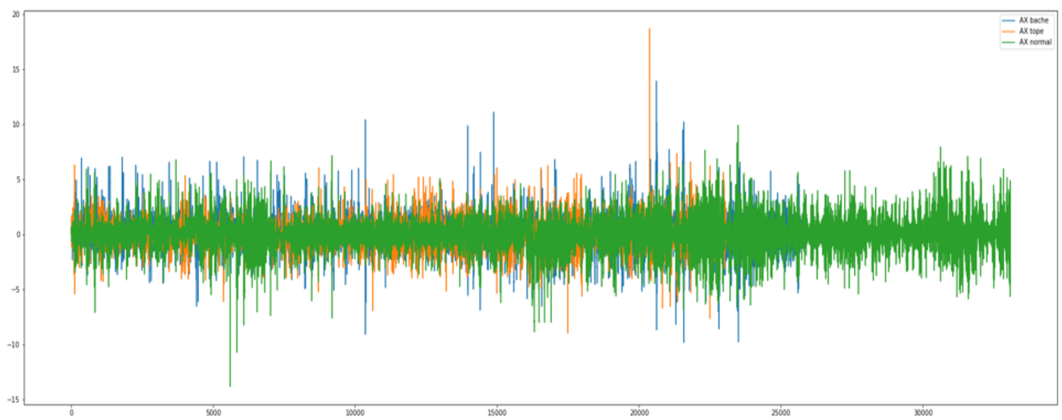


Figure 5.8: Example of a gap in AX (X-axis of the accelerometer) in the different databases, the data in blue represents the 'potholes,' orange represents the 'bumps,' and in green, we have the data from the 'normal' database.

Therefore, the data must be balanced to achieve the correct manipulation and development of the model. So, in this case, we decided to match all databases concerning the one with the least amount of data, So the 'bumps' database was the chosen one. Making all the databases contain a total of 23,086 data, which is equivalent to 97 events for each of the databases. In Figure.5.9, we can see an example of how the data is already balanced, so we can see that there is no longer a gap between data in terms of quantity.

5.3.1 Feature extraction

In machine learning, feature extraction calcifies, or determines specific attributes, from input data[29]. In order to occupy any classifier, a process of extraction of these attributes is required. All the features are extracted from the accelerometer and gyroscope data, so the model will be trained with six features.

As we can see in Figures(5.8, 5.9), the data with which we will work, despite being balanced among themselves, contain much noise, which makes it more difficult for the model

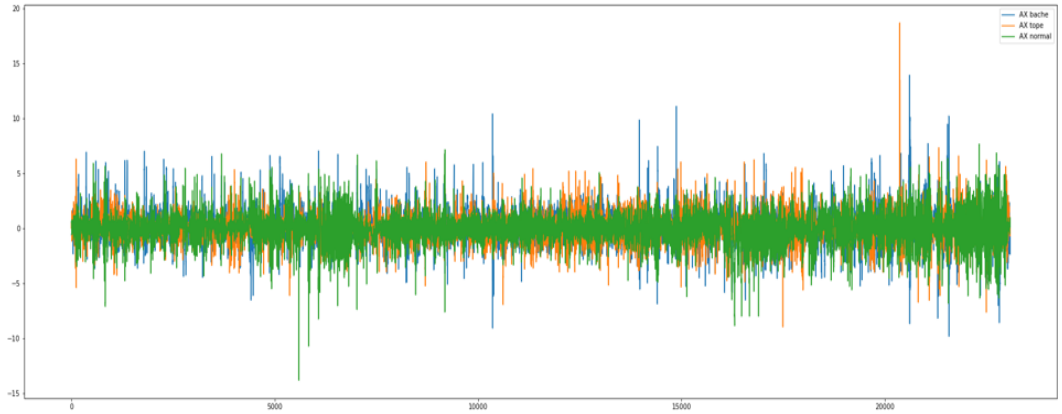


Figure 5.9: Example of balanced data in AX (X-axis of the accelerometer), the data in blue represents the 'potholes,' orange represents the 'bumps,' and in green, we have the data from the 'normal' database.

to classify. So employing the **moving average method** seeks to reduce noise and have a better distinction between one database from another.

This method consists of making the arithmetic mean of the demand of *'N'* periods but the most recent to avoid random fluctuations and has the objective of demonstrating a forecast for the following period. On the other hand, this method is used more for those demands that do not present pronounced trends. For the calculation of this method, we will use the *'N'* demands of the most recent periods, for which the averages will move from one period to another. It is calculated using the following formula.

$$F_{t+1} = \frac{\text{Sum of the last 'N' data}}{N} = \frac{D_t + D_{t-1} + ... + D_{t-n+1}}{N} \tag{5.5}$$

Where,

- D_t: Real value in period *t*.
- N: Total number of data included in the average.
- F_{t+1}: Value for $t + 1$.

After applying the moving average method to each database, we have 237 empty spaces filled with the average of all the data resulting from the moving average. After applying these mathematics, it is necessary to concoct the three databases to have only one database with the three types of events. In the Figure 5.10, we can see how our new database is composed. We can see the addition of six new columns due to the operations performed. Figure 5.11 shows the accelerometer data in the X-axis; in the upper part of the image, it is the data without manipulation, and above the figure, after applying the moving average.

5.4 Processing and Post-processing

The processing step analyzes the values from the pre-processed sensors to detect anomalies on the road surface. There are three main approaches [32]:

	AX	AY	AZ	GX	GY	GZ	Activity	MeanAX	MeanAY	MeanAZ	MeanGX	MeanGY	MeanGZ
0	-0.606590	1.560097	-1.025081	0.215254	-0.022984	0.097051	bache	0.081981	-0.050445	0.885124	0.011605	-0.005167	-0.019353
1	-0.540882	2.096340	-0.096130	0.217086	-0.012599	0.098884	bache	0.081981	-0.050445	0.885124	0.011605	-0.005167	-0.019353
2	-0.575768	0.995061	-0.984110	0.197539	-0.007101	0.100106	bache	0.081981	-0.050445	0.885124	0.011605	-0.005167	-0.019353
3	-0.545888	1.004465	-1.026807	0.184710	-0.010156	0.100716	bache	0.081981	-0.050445	0.885124	0.011605	-0.005167	-0.019353
4	-0.542176	1.004201	-1.047549	0.160887	-0.020540	0.101327	bache	0.081981	-0.050445	0.885124	0.011605	-0.005167	-0.019353
...	...	...	...	...	...	...	...	...	...	...	...	...	...
23081	0.657609	-0.386220	-0.381561	0.016875	0.127976	-0.279776	normal	0.036567	-0.147766	-0.227763	-0.008876	-0.063946	0.033441
23082	0.651542	-0.383676	-0.364877	0.013821	0.138972	-0.298713	normal	-0.041115	-0.133374	-0.230548	-0.012749	-0.064942	0.032481
23083	0.967596	0.049865	-0.401591	0.079794	0.165850	-0.296270	normal	0.043392	-0.121351	-0.231961	-0.015755	-0.065896	0.031434
23084	0.474794	-0.520039	0.428092	0.060857	0.159130	-0.290161	normal	0.047818	-0.111720	-0.229872	-0.018239	-0.066777	0.030063
23085	0.463725	-0.518051	0.439116	0.106061	0.208000	-0.280387	normal	0.052088	-0.103691	-0.226718	-0.019328	-0.067460	0.028816

Figure 5.10: Rolling mean data

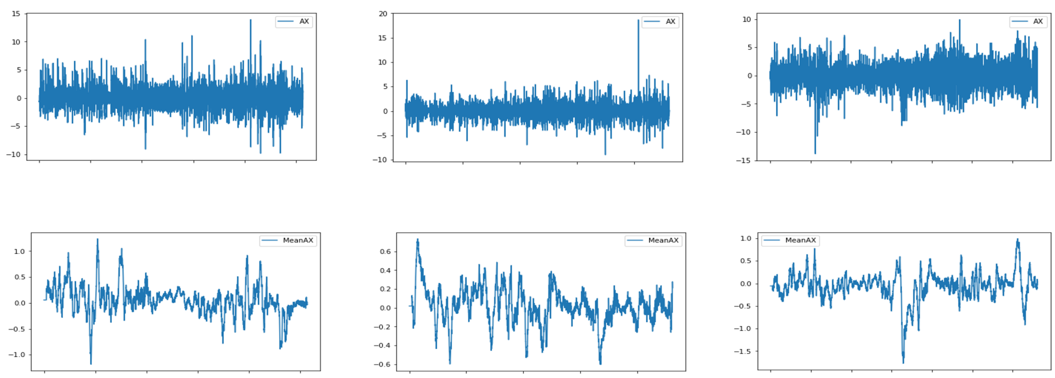


Figure 5.11: Rolling mean data

- The threshold-based approach: a data log is performed after the detected values exceed a set value.
- The machine learning approach: uses more advanced classification techniques to detect anomalies in the road surface.
- Dynamic time warping (DTW): using pattern matching, the similarity between different sequences is temporally measured.

This thesis will deal with the machine learning approach, using supervised learning for all the implemented methodologies.

Supervised learning[29] is a technique that uses learning data produced by a pair of values. It should be known that each pair has a set of input variables, and one output. Depending on how well the training data is, the main idea of this technique is that the chosen algorithm learns to predict the output variable, for new and unknown input data.

5.4.1 2-CNN Model

In order to feed the data into our neural network, a reshape is applied to the data so that each event has multiple two-dimensional records which hold 238 slices for each of the three accelerometer and gyroscope readings (x,y,z axis). One record is associated with one label (e.g., Pothole). Those records are fed into the neural network during training.

The input layer is a vector with 1428 elements (flattened representation of 238 slices for the six accelerometer and gyroscope readings, three for each). Then, we have three hidden layers with 100 nodes each. One traditional layer upfront for reshaping the input into a 238x6 matrix and a Softmax activation layer as the final layer.

Finally, the output layer with the available labels, the network will provide the probability regarding each output class. Probabilities will add up to 1.

Once the different data types and structures were created, the input and output training data structures were used to train a 2-CNN classifier. In the next subchapter, the performance of the classifier is tested. Upon completion, the method returns the expected values, as well as the correct labels. In Figure 5.12 we can observe the operation of our model in a more general way.

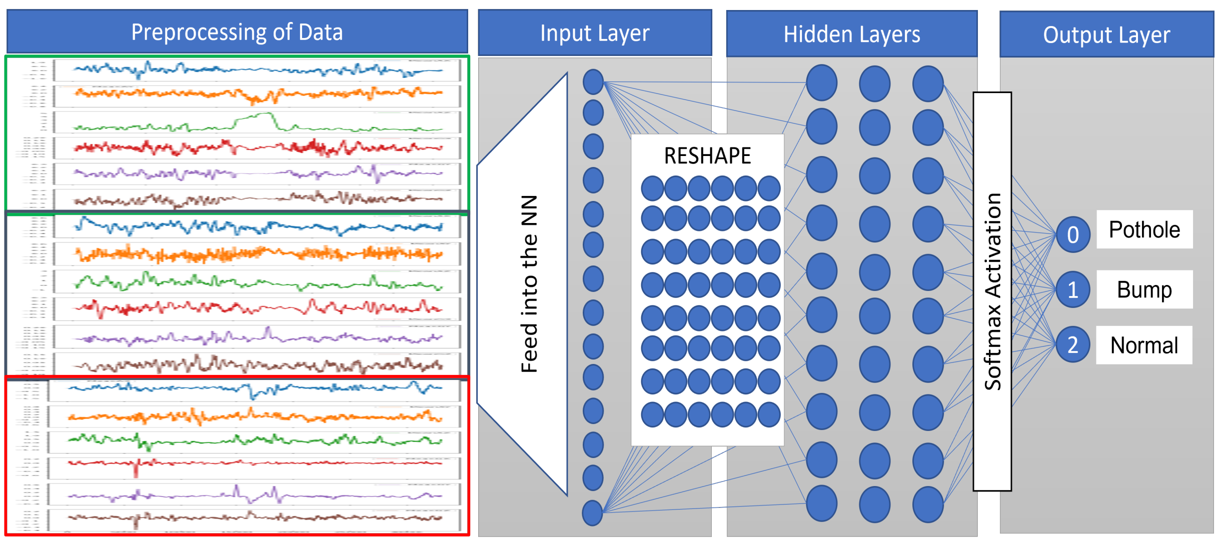


Figure 5.12: IMPLEMENTED 2-CNN

5.5 Performance Evaluations

To carry out a correct evaluation of the performance of the model created for the detection of anomalies on roads, for this we will require performance metrics, which are included: accuracy, precision, false positive rate, and false negative rate. Combining and doing all these metrics is important and will be necessary since it will allow us to have a better panorama and reality of the performance of our model.

It is very difficult to know which are the characteristics that will most affect the selected classifier. This thesis has decided to take the solution of making subsets of the different characteristics. This will help us since we have a broader picture of the possible scenarios and this suction will allow us to more easily select the subset that contributes the most to our model.

In this thesis, the complete feature set is made up of six different types. The classifier model was trained and evaluated during 30 epochs. For this evaluation process, the data was divided as follows: 80% as training data and 20% as test data.

Since we had many characteristics, we divided into different subsets to observe which would have a better model performance; the accuracy results for each subset can be observed in the Table 5.2.

Database	Number of Features	Features	Accuracy
Pothole - bumps	1	AX	64%
Pothole - bumps - normal	1	AX	48%
Pothole - bumps	1	AY	83%
Pothole - bumps - normal	1	AY	62%
Pothole - bumps	1	AZ	68%
Pothole - bumps - normal	1	AZ	51%
Pothole - bumps	1	GX	76%
Pothole - bumps - normal	1	GX	60%
Pothole - bumps	1	GY	75%
Pothole - bumps - normal	1	GY	52%
Pothole - bumps	1	GZ	71%
Pothole - bumps - normal	1	GZ	53%
Pothole - bumps	2	AX - AY	83%
Pothole - bumps - normal	2	AX - AY	70%
Pothole - bumps	2	AX - AZ	82%
Pothole - bumps - normal	2	AX - AZ	66%
Pothole - bumps	2	AX - GX	82%
Pothole - bumps - normal	2	AX - GX	69%
Pothole - bumps	2	AY - AZ	89%
Pothole - bumps - normal	2	AY - AZ	73%
Pothole - bumps	2	AY - GX	86%
Pothole - bumps - normal	2	AY - GX	72%
Pothole - bumps	2	AY - GY	88%
Pothole - bumps - normal	2	AY - GY	72%
Pothole - bumps	3	AX - AY- AZ	92%
Pothole - bumps - normal	3	AX - AY- AZ	84%
Pothole - bumps	3	AY - GX - GY	92%
Pothole - bumps - normal	3	AY - GX - GY	81%
Pothole - bumps	**6**	**AX - AY - AZ - GX - GY - GZ**	**98%**
Pothole - bumps - normal	**6**	AX - AY - AZ - GX - GY - GZ	92%

Table 5.2: Accuracy results with the different propose subsets

The subset of the highest precision (98%), shown in Table 5.2, was the subset in which only the occurrence of potholes and bumps along with six features were included, which is extremely high, but it is important to note, however, that the 'normal' event was not included, because the data appears to be too noisy, and it cannot be properly processed by the neural network, so it was decided to simply take into account the two above-mentioned.

In Figures 5.13, 5.14, we have different metrics showing how well our model performs.

On the one hand, Figure 5.13 shows the loss graph, made of the values of the measurement of the cross entropy; this graph represents the sum of errors in our model, and we can observe that values are continuously decreasing in the training values and in those of validation, which indicates that our model works well and makes few errors. The model uses the loss function to learn.

Then Figure 5.14 is the accuracy graph, which is easier to understand. It measures how well our model predicts by comparing model predictions to actual values in percent- We can observe a constant growth in the training and validation values. We also have the accuracy value that turned out to be the best.

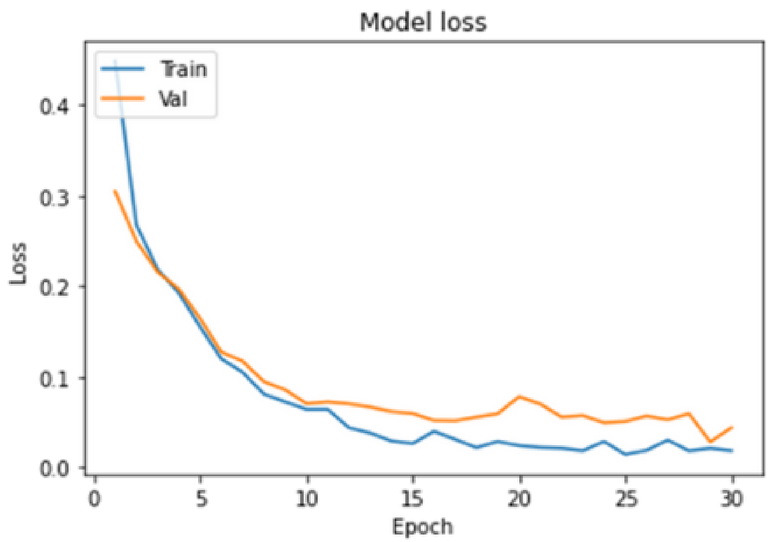


Figure 5.13: Model loss graph.

Finally, we provide our confusion matrix, which will enable us to see how a supervised learning system performs. The number of predictions for each class is represented in each column of the matrix, whereas the instances in the actual class are represented in each row. We can see how the model is predicting each of the categories by looking at Figure 5.15, which displays two values to predict: potholes (0) and bumps (1). The results for each type of event can then be seen; for potholes, accuracy values were 98%, hitting 226 cases; for bumps, accuracy values were 99%, hitting 227 true positives.

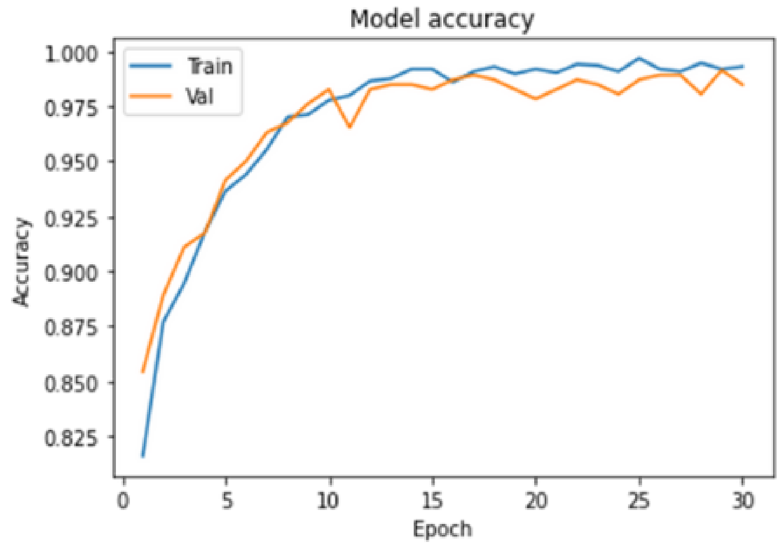


Figure 5.14: Model Accuracy graph.

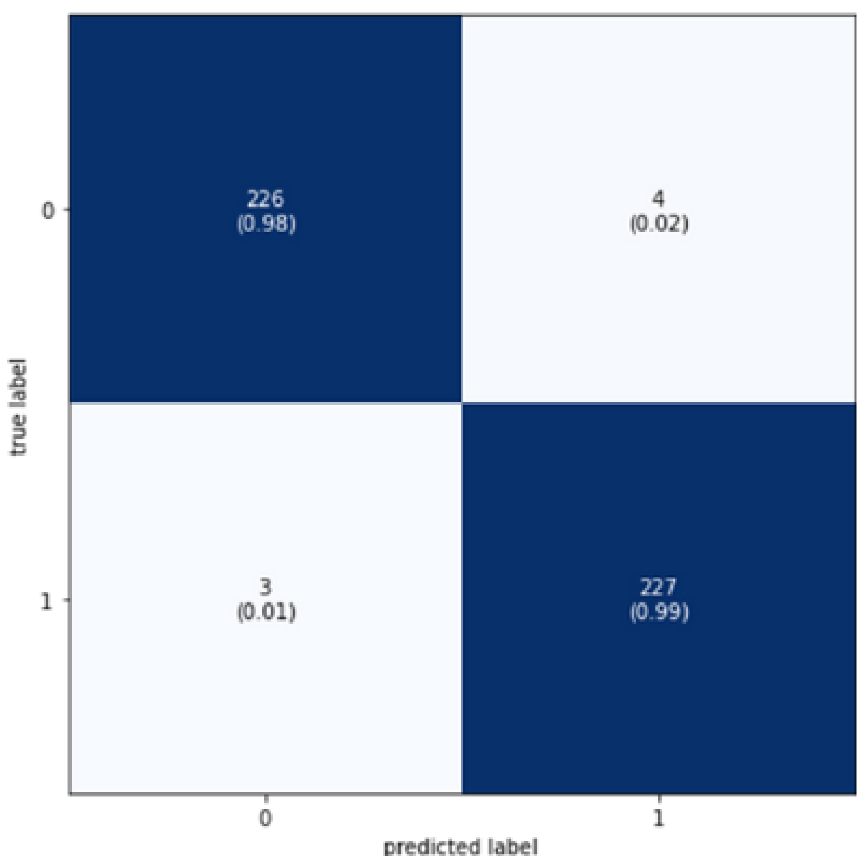


Figure 5.15: Confusion matrix.

Chapter 6

Conclusion

As a result of this thesis, it was possible to develop an Android application for linear acceleration data collection. The app uses the Android Sensor framework to access the gyroscope and linear accelerometer sensors.

We use the application to collect acceleration data in different parts of Mexico City and its surroundings. Three types of events were distinguished: potholes, bumps, and normal. The 'normal' database produced much noise. So, to have an algorithm with the highest performance, we decide not to use it. We also develop a classification algorithm for the proposed events. Implementing the idea of supervised learning to predict what type of event is occurring, calculated from the acceleration data of the different sensors. After the calculations, we extracted six features: Mean AX, Mean AY, Mean AZ, Mean GX, Mean GY, and Mean GZ.

The detection algorithm was evaluated on a total dataset containing 2,297 data points. For the 2-CNN model, we separated the data into two; training (80%) and testing(20%). A set of features, which according to our results are the ones that provide the best classifier accuracy in comparison to the initial configuration of the feature set, was selected for further testing. During the different tests, the classifier demonstrated an accuracy of 98% in correctly classifying events. A great classification model was obtained since it is quite competitive with the models proposed by other authors. However, the initial objective of this thesis is not completed since, unfortunately, the concept of crowdsensing could not be used since the information was only obtained through a single mobile device.

6.1 Contributions

Asphalt surface conditions could be estimated from the test vehicle's recorded responses when driving on asphalt. A system was set up to collect the data. By analyzing the test data, we can find the correlation between vibration responses and asphalt conditions. The vibration-based road evaluation model described in this paper has the advantage that it requires little storage, and it turns out to be very suitable for data processing. It is true; the model does not provide complex details of the selected events, as it only determines if there is a match or not. It can become a feasible and correct model for road inspections in a simpler way. If a higher level of detail is desired, video could be used, but it would be a very different approach than today. We can say that a model was obtained that fulfills its main task, since it classifies the selected events correctly and with high precision.

6.2 Comparisons

In comparison with other works on recognition methods based on a smartphone's sensors mentioned in this thesis. We start with Ronghua Du et al.[14], who attempted a similar approach to classifying surface quality. Through a KNN algorithm, they include bumps and potholes in their classification of abnormal pavement including bumps and potholes, obtaining an accuracy of 94.12%. Azza Allouch et al.[3] occupied the accelerometer and gyroscope data, and they tested through the different models. The model with the best performance was the decision tree model, with an accuracy of 98%. Also, the work of Chao Wu et al.[43], through accelerometer and GPS, only classified potholes, and the model with the best performance was Rain Forest, with an accuracy of 88.5% in potholes and an accuracy of 95.7% in testing. Thanks to the above, we can see that our model has a high accuracy since it stands out above two of the mentioned works and ties with one of them.

6.3 Future Work

The main issue to consider for future work is a reorientation. There must be an algorithm to reorient the axes regardless of the position of the smartphone, this detail can be the difference so that the model can be used in an ideal way in a real world scenario. Another approach that I consider should be done is the option of occupying more than one type of car, since this will provide us with different information, therefore we will have more points of comparison. So, there is a chance to find a better solution. If all these problems are solved, then the proof of concept solution of the thesis could become a cloud-based application. In such an application, we could receive data from multiple people, and we would increase the quality of the classification. This would help us to achieve the objective of using the crowdsensing concept, since we use more than one data source to classify different points on the road. Having more robust databases will help us filter spontaneous events and possibly false data. In addition, we may have the possibility of expanding the number of different types of pavements.

Appendix A

Appendix

```
!pip uninstall tensorflow==2.8.0
!pip install tensorflow==2.0.0

from urllib.request import Request, urlopen  # Python 3
import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense, Dropout,
    BatchNormalization
from tensorflow.keras.layers import Conv2D, MaxPool2D
from tensorflow.keras.optimizers import Adam
import sklearn.metrics

url1 = ("https://raw.githubusercontent.com/rossel4/pothole_thesis/master/
    bachesf.csv")
url2 = ("https://raw.githubusercontent.com/rossel4/pothole_thesis/master/
    topef.csv")
url3 = ("https://raw.githubusercontent.com/rossel4/pothole_thesis/master/
    normalf.csv")

bache= pd.read_csv(url1)
bache['Activity']='bache'
tope= pd.read_csv(url2)
tope['Activity']='tope'

normal= pd.read_csv(url3)
normal['Activity']='normal'

print(bache['Activity'].value_counts())
print(tope['Activity'].value_counts())
print(normal['Activity'].value_counts())
```

```
df1 = pd.concat([bache,tope,normal])

def plotsaxis (df1,df2,df3):
    plt.figure( figsize= (30,10))
    plt.plot(df1.AX, label='AX bache')
    plt.plot(df2.AX,  label='AX tope')
    plt.plot(df3.AX,  label='AX normal')
    plt.legend()
    plt.show()
    plt.figure( figsize= (30,10))
    plt.plot(df1.AY, label='AY bache')
    plt.plot(df2.AY,  label='AY tope')
    plt.plot(df3.AY,  label='AY normal')
    plt.legend()
    plt.show()
    plt.figure( figsize= (30,10))
    plt.plot(df1.AZ, label='AZ bache')
    plt.plot(df2.AZ,  label='AZ tope')
    plt.plot(df3.AZ,  label='AZ normal')
    plt.legend()
    plt.show()
    plt.figure( figsize= (30,10))
    plt.plot(df1.GX, label='GX bache')
    plt.plot(df2.GX,  label='GX tope')
    plt.plot(df3.GX,  label='GX tope')
    plt.legend()
    plt.show()
    plt.figure( figsize= (30,10))
    plt.plot(df1.GY, label='GY bache')
    plt.plot(df2.GY,  label='GY tope')
    plt.plot(df3.GY,  label='GY normal')
    plt.legend()
    plt.show()
    plt.figure( figsize= (30,10))
    plt.plot(df1.GZ, label='GZ bache')
    plt.plot(df2.GZ,  label='GZ tope')
    plt.plot(df3.GZ,  label='GZ normal')

    plt.legend()
    plt.show()

plotsaxis(bache,tope,normal)

countactivity= (df1['Activity'].value_counts())/238
countactivity

countactivity.plot(kind='bar')

bache = bache[bache['Activity']=='bache'].head(23086).copy()
tope = tope[tope['Activity']=='tope'].head(23086).copy()
normal = normal[normal['Activity']=='normal'].head(23086).copy()

df2 = pd.concat([bache,tope,normal])
```

```
plotsaxis(bache,tope,normal)

def axisn(activity,df):
  rowactivity = (df['Activity']==activity)
  data = df[rowactivity]
  data = data[['AX','AY','AZ','GX','GY','GZ']]
  ax = data.plot(subplots=True, figsize= (7,30), title = activity)

axisn('bache',df2)

def mean (df):
  df['MeanAX']= df['AX'].rolling(238).mean()
  df['MeanAY']= df['AY'].rolling(238).mean()
  df['MeanAZ']= df['AZ'].rolling(238).mean()
  df['MeanGX']= df['GX'].rolling(238).mean()
  df['MeanGY']= df['GY'].rolling(238).mean()
  df['MeanGZ']= df['GZ'].rolling(238).mean()


bache = bache.copy()
bache["MeanAX"].fillna(bache["MeanAX"].mean(skipna=True), inplace=True)
bache["MeanAY"].fillna(bache["MeanAY"].mean(skipna=True), inplace=True)
bache["MeanAZ"].fillna(bache["MeanAZ"].mean(skipna=True), inplace=True)
bache["MeanGX"].fillna(bache["MeanGX"].mean(skipna=True), inplace=True)
bache["MeanGY"].fillna(bache["MeanGY"].mean(skipna=True), inplace=True)
bache["MeanGZ"].fillna(bache["MeanGZ"].mean(skipna=True), inplace=True)

tope = tope.copy()
tope["MeanAX"].fillna(tope["MeanAX"].mean(skipna=True), inplace=True)
tope["MeanAY"].fillna(tope["MeanAY"].mean(skipna=True), inplace=True)
tope["MeanAZ"].fillna(tope["MeanAZ"].mean(skipna=True), inplace=True)
tope["MeanGX"].fillna(tope["MeanGX"].mean(skipna=True), inplace=True)
tope["MeanGY"].fillna(tope["MeanGY"].mean(skipna=True), inplace=True)
tope["MeanGZ"].fillna(tope["MeanGZ"].mean(skipna=True), inplace=True)

normal = normal.copy()
normal["MeanAX"].fillna(normal["MeanAX"].mean(skipna=True), inplace=True)
normal["MeanAY"].fillna(normal["MeanAY"].mean(skipna=True), inplace=True)
normal["MeanAZ"].fillna(normal["MeanAZ"].mean(skipna=True), inplace=True)
normal["MeanGX"].fillna(normal["MeanGX"].mean(skipna=True), inplace=True)
normal["MeanGY"].fillna(normal["MeanGY"].mean(skipna=True), inplace=True)
normal["MeanGZ"].fillna(normal["MeanGZ"].mean(skipna=True), inplace=True)

def plotsmean (df1,df2,df3):
  plt.figure( figsize= (30,10))
  plt.plot(df1.MeanAX, label='Mean AX bache')
  plt.plot(df2.MeanAX, label='Mean AX tope')
  plt.plot(df3.MeanAX, label='Mean AX normal')
  plt.legend()
  plt.show()
  plt.figure( figsize= (30,10))
  plt.plot(df1.MeanAY, label='Mean AY bache')
  plt.plot(df2.MeanAY, label='Mean AY tope')
  plt.plot(df3.MeanAY, label='Mean AY normal')
```

```
  plt.legend()
  plt.show()
  plt.figure( figsize= (30,10))
  plt.plot(df1.MeanAZ,label='Mean AZ bache')
  plt.plot(df2.MeanAZ, label='Mean AZ tope')
  plt.plot(df3.MeanAZ, label='Mean AZ normal')
  plt.legend()
  plt.show()
  plt.figure( figsize= (30,10))
  plt.plot(df1.MeanGX,label='Mean GX bache')
  plt.plot(df2.MeanGX, label='Mean GX tope')
  plt.plot(df3.MeanGX, label='Mean GX tope')
  plt.legend()
  plt.show()
  plt.figure( figsize= (30,10))
  plt.plot(df1.MeanGY,label='Mean GY bache')
  plt.plot(df2.MeanGY, label='Mean GY tope')
  plt.plot(df3.MeanGY, label='Mean GY normal')
  plt.legend()
  plt.show()
  plt.figure( figsize= (30,10))
  plt.plot(df1.MeanGZ,label='Mean GZ bache')
  plt.plot(df2.MeanGZ, label='Mean GZ tope')
  plt.plot(df3.MeanGZ, label='Mean GZ normal')

  plt.legend()
  plt.show()

plotsmean(bache,tope,normal)

#df = pd.concat([bache,tope,normal])
df = pd.concat([bache,tope])

countactivity= (df['Activity'].value_counts())/238


def axis(activity,df):
  rowactivity = (df['Activity']==activity)
  data = df[rowactivity]
  data = data[['MeanAX','MeanAY','MeanAZ','MeanGX','MeanGY','MeanGZ']]
  ax = data.plot(subplots=True, figsize= (7,30), title = activity)

axis ('bache',df)

df['Activity'].value_counts()

balanced_data = pd.DataFrame()
balanced_data = balanced_data.append([bache,tope])
#balanced_data = balanced_data.append([bache,tope,normal])


from sklearn.preprocessing import LabelEncoder

label = LabelEncoder()
```

```
balanced_data['label'] = label.fit_transform(balanced_data['Activity'])
balanced_data.head()

label.classes_

df_embarked_one_hot = pd.get_dummies(balanced_data['label'],
                                     prefix='label')

df_new_enc = pd.concat([balanced_data,
                        df_embarked_one_hot],
                        axis=1)

X = balanced_data[['MeanAX', 'MeanAY', 'MeanAZ', 'MeanGX', 'MeanGY', '
    MeanGZ']]
y = balanced_data['label']

scaler = StandardScaler()
X = scaler.fit_transform(X)

scaled_X = pd.DataFrame(data = X, columns = ['MeanAX', 'MeanAY', 'MeanAZ',
     'MeanGX', 'MeanGY', 'MeanGZ'])
scaled_X['label'] = y.values

scaled_X

import scipy.stats as stats

Fs = 10
frame_size = 238
hop_size = Fs*2 #

def get_frames(df, frame_size, hop_size):

    N_FEATURES = 6

    frames = []
    labels = []
    for i in range(0, len(df) - frame_size, hop_size):
        ax = df['MeanAX'].values[i: i + frame_size]
        ay = df['MeanAY'].values[i: i + frame_size]
        az = df['MeanAZ'].values[i: i + frame_size]
        gx = df['MeanGX'].values[i: i + frame_size]
        gy = df['MeanGY'].values[i: i + frame_size]
        gz = df['MeanGZ'].values[i: i + frame_size]

        # Retrieve the most often used label in this segment
        label = stats.mode(df['label'][i: i + frame_size])[0][0]
        frames.append([ax, ay, az, gx, gy, gz])
        labels.append(label)

    # Bring the segments into a better shape
    frames = np.asarray(frames).reshape(-1, frame_size, N_FEATURES)
    labels = np.asarray(labels)
```

```
    return frames, labels

X, y = get_frames(scaled_X, frame_size, hop_size)

X.shape, y.shape

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
    random_state = 0, stratify = y)
X_train.shape, X_test.shape

X_train[0].shape, X_test[0].shape

X_train = X_train.reshape(1837, 238, 6, 1)
X_test = X_test.reshape(460, 238, 6, 1)

X_train[0].shape, X_test[0].shape

model = Sequential()
model.add(Conv2D(16, (2, 2), activation = 'relu', input_shape = X_train
   [0].shape))
model.add(Dropout(0.1))

model.add(Conv2D(32, (2, 2), activation='relu'))
model.add(Dropout(0.2))

model.add(Flatten())

model.add(Dense(64, activation = 'relu'))
model.add(Dropout(0.5))

model.add(Dense(2, activation='softmax'))
model.compile(optimizer=Adam(learning_rate = 0.001),
              loss = 'sparse_categorical_crossentropy', metrics = ['
                 accuracy'])

history = model.fit(X_train, y_train, epochs = 30, validation_data= (
   X_test, y_test), verbose=1)

print(model.evaluate(X_test, y_test))

from sklearn.metrics import classification_report
predictions = model.predict(X_test)

def plot_learningCurve(history, epochs):
  # Plot training & validation accuracy values
  epoch_range = range(1, epochs+1)
  plt.plot(epoch_range, history.history['accuracy'])
  plt.plot(epoch_range, history.history['val_accuracy'])
  plt.title('Model accuracy')
  plt.ylabel('Accuracy')
  plt.xlabel('Epoch')
  plt.legend(['Train', 'Val'], loc='upper left')
  plt.show()
```

```
    # Plot training & validation loss values
    plt.plot(epoch_range, history.history['loss'])
    plt.plot(epoch_range, history.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Val'], loc='upper left')
    plt.show()

plot_learningCurve(history, 30)

from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score

predictions = model.predict(X_test)

!pip install mlxtend

import mlxtend
from mlxtend.plotting import plot_confusion_matrix
from sklearn.metrics import confusion_matrix

from mlxtend.plotting import plot_confusion_matrix

label.classes_

print(tf.__version__)

print(mlxtend.__version__)

y_pred = model.predict_classes(X_test)

mat = confusion_matrix(y_test, y_pred)
fig, ax = plot_confusion_matrix(conf_mat= mat , show_normed=True, figsize
    =(7,7))
plt.show()
```

Appendix B

Appendix

```
package com.example.demo2;

import static java.lang.Math.abs;
import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.PermissionChecker;

import android.Manifest;
import android.content.ClipboardManager;
import android.content.Context;
import android.content.pm.PackageManager;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Build;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;

import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

import org.json.JSONArray;
```

```java
import org.json.JSONException;
import org.json.JSONObject;

import java.io.DataOutputStream;
import java.lang.Math;
import java.lang.reflect.Array;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Arrays;


public class MainActivity extends AppCompatActivity implements
    SensorEventListener, View.OnClickListener {
    private static final String TAG = "MainActivity";
    SensorManager sensorManager;
    Sensor accelerometer;
    Context context;
    Sensor giro;
    String sensorName;

    double[] arrGps = new double[2];

    int flag = 0;

    int numSamples = 119;
    String[][] accBuff = new String[3][238];
    String[] accBuffX = new String[238];
    String[] accBuffY = new String[238];
    String[] accBuffZ = new String[238];
    int take = 0;
    int samplesRead = numSamples;
    float accelerationThreshold = 10;
    long counter = 0;

    // GIROCOPIO
    int girnumSamples = 119;
    float[][] girBuff = new float[3][238];
    float[] girBuffX = new float[238];
    float[] girBuffY = new float[238];
    float[] girBuffZ = new float[238];
    int girtake = 0;
    int girsamplesRead = girnumSamples;
    float giroscopeThreshold = 19;
    long gircounter = 0;

    //INTERFAZ
    String textViewContent = "";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        context = this;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        TextView letrero1 = (TextView) findViewById((R.id.letrero1));
        letrero1.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                ClipboardManager clip = (ClipboardManager)
                    getApplicationContext().getSystemService(Context.
                    CLIPBOARD_SERVICE);
                clip.setText(letrero1.getText().toString());
                Toast.makeText(getApplicationContext(),"Ya:p",Toast.
                    LENGTH_SHORT).show();
            }
        });
        letrero1.setMovementMethod(new ScrollingMovementMethod());
        letrero1.append("AX,AY,AZ,GX,GY,GZ\n");

        TextView letrero2 = (TextView) findViewById((R.id.letrero2));
        letrero2.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                ClipboardManager clip = (ClipboardManager)
                    getApplicationContext().getSystemService(Context.
                    CLIPBOARD_SERVICE);
                clip.setText(letrero2.getText().toString());
                Toast.makeText(getApplicationContext(),"Ya:p",Toast.
                    LENGTH_SHORT).show();
            }
        });
        letrero2.setMovementMethod(new ScrollingMovementMethod());
        letrero2.append("AX,AY,AZ,GX,GY,GZ\n");

        TextView letrero3 = (TextView) findViewById((R.id.letrero3));
        letrero3.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                ClipboardManager clip = (ClipboardManager)
                    getApplicationContext().getSystemService(Context.
                    CLIPBOARD_SERVICE);
                clip.setText(letrero3.getText().toString());
                Toast.makeText(getApplicationContext(),"Ya:p",Toast.
                    LENGTH_SHORT).show();
            }
        });
        letrero3.setMovementMethod(new ScrollingMovementMethod());
        letrero3.append("AX,AY,AZ,GX,GY,GZ\n");

        Button botonBache = (Button) findViewById((R.id.botonBache));
        Button botonTope = (Button) findViewById((R.id.botonTope));
        Button botonNormal = (Button) findViewById((R.id.botonNormal));

        LocationManager locationManager = (LocationManager) MainActivity.
            this.getSystemService(Context.LOCATION_SERVICE);
        LocationListener locationListener = new LocationListener() {
```

```
            int permissionCheck = ContextCompat.checkSelfPermission(
                MainActivity.this, Manifest.permission.ACCESS_FINE_LOCATION
                );

            @Override
            public void onLocationChanged(Location location) {
                String cords = "Coordenadas => Longitud: " + location.
                    getLongitude() + "Latitud: " + location.getLatitude();
                arrGps[0] = location.getLongitude();
                arrGps[1] = location.getLatitude();
                Log.d(TAG, cords);
            }

            public void onProviderEnabled(String provider) {
            }

            public void onStatusChanged(String provider, int status,
                Bundle extras) {
            }
        };

        botonBache.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(textViewContent!="") {
                    letrero1.append(textViewContent);
                    //letrero1.append("\n");
                    textViewContent = "";
                }
                if(letrero1.getVisibility() == View.INVISIBLE) {
                    letrero2.setVisibility((View.INVISIBLE));
                    letrero3.setVisibility((View.INVISIBLE));
                    letrero1.setVisibility((View.VISIBLE));
                }
            }
        });

        botonTope.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(textViewContent!="") {
                    letrero2.append(textViewContent);
                    //letrero2.append("\n");
                    textViewContent = "";
                }
                textViewContent="";
                if(letrero2.getVisibility() == View.INVISIBLE) {
                    letrero1.setVisibility((View.INVISIBLE));
                    letrero3.setVisibility((View.INVISIBLE));
                    letrero2.setVisibility((View.VISIBLE));
                }
            }
        });
```

```
        botonNormal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(textViewContent!="") {
                    letrero3.append(textViewContent);
                    //letrero3.append("\n");
                    textViewContent = "";
                }
                if(letrero3.getVisibility() == View.INVISIBLE) {
                    letrero1.setVisibility((View.INVISIBLE));
                    letrero2.setVisibility((View.INVISIBLE));
                    letrero3.setVisibility((View.VISIBLE));
                }
            }
        });

        locationManager.requestLocationUpdates(LocationManager.
            NETWORK_PROVIDER, 0, 0, locationListener);


        sensorManager = (SensorManager) getSystemService(Context.
            SENSOR_SERVICE);

        Log.d(TAG, "Iniciando acelerometro");
        accelerometer = sensorManager.getDefaultSensor(Sensor.
            TYPE_LINEAR_ACCELERATION);
            sensorManager.registerListener(this, accelerometer,
                SensorManager.SENSOR_DELAY_FASTEST);
        Log.d(TAG, "Acelerometro iniciado");

        Log.d(TAG, "Iniciando giroscopio");
        giro = sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
        sensorManager.registerListener(this, giro, SensorManager.
            SENSOR_DELAY_FASTEST);
        Log.d(TAG, "Giroscopio iniciado");

        int permissionCheck = ContextCompat.checkSelfPermission(this,
             Manifest.permission.ACCESS_FINE_LOCATION);
        if (permissionCheck == PackageManager.PERMISSION_DENIED) {
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
                Manifest.permission.ACCESS_FINE_LOCATION)) {

            } else {
                ActivityCompat.requestPermissions(this, new String[]{
                    Manifest.permission.ACCESS_FINE_LOCATION}, 1);
            }
        }
    }

    boolean isAccelData = false;
    boolean isGyroData = false;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
```

```java
    public void onSensorChanged(SensorEvent sensorEvent) {
        if (sensorEvent.sensor.getType() == Sensor.TYPE_GYROSCOPE) {
            isGyroData = true;
            sensorName = sensorEvent.sensor.getName();
            //Log.d(TAG,sensorName + " Accel X: "+ sensorEvent.values[0] +
                " Accel Y: "+ sensorEvent.values[1] + " Accel Z: "+
               sensorEvent.values[2]);
            //buffer
            while (girsamplesRead == girnumSamples) {
                float gSum = abs(sensorEvent.values[0]) + abs(sensorEvent.
                    values[1]) + abs(sensorEvent.values[2]);
                if (gircounter < 118)
                    gircounter++;
                for (int i = 0; i < 119; i++) {
                    for (int j = 0; j < 3; j++) {
                        girBuff[j][i] = girBuff[j][i+1];
                        if(j==0){
                            girBuffX[i] = girBuffX[i+1];
                        }else if(j==1){
                            girBuffY[i] = girBuffY[i+1];
                        }else if(j==2){
                            girBuffZ[i] = girBuffZ[i+1];
                        }
                    }
                }

                girBuff[0][118] = sensorEvent.values[0];
                girBuff[1][118] = sensorEvent.values[1];
                girBuff[2][118] = sensorEvent.values[2];
                girBuffX[118] = sensorEvent.values[0];
                girBuffY[118] = sensorEvent.values[1];
                girBuffZ[118] = sensorEvent.values[2];

                //Log.d(TAG, "Suma " + Float.toString(aSum));

                if (flag == 1 && gircounter >= 118) {
                    girsamplesRead = 0;
                    break;
                } else {
                    break;
                }
            }

            while (girsamplesRead < girnumSamples) {
                girsamplesRead++;
                girBuff[0][girsamplesRead + 118] = sensorEvent.values[0];
                girBuff[1][girsamplesRead + 118] = sensorEvent.values[1];
                girBuff[2][girsamplesRead + 118] = sensorEvent.values[2];
                girBuffX[118+girsamplesRead] = sensorEvent.values[0];
                girBuffY[118+girsamplesRead] = sensorEvent.values[1];
                girBuffZ[118+girsamplesRead] = sensorEvent.values[2];

                if (girsamplesRead == girnumSamples) {
                    for (int i = 0; i < 238; i++) {
```

```
                    //Log.d(TAG, sensorName + i + " X: " + Float.
                        toString(girBuff[0][i]) + " Y: " + Float.
                        toString(girBuff[1][i]) + " Z: " + Float.
                        toString(girBuff[2][i]));

                    // \n
                }
                // \n
                //samplesRead = 118;
                flag = 0;
                break;

            }
            break;
        }

    }
    if (sensorEvent.sensor.getType() == Sensor.
        TYPE_LINEAR_ACCELERATION) {
        isAccelData = true;
        sensorName = sensorEvent.sensor.getName();
        //Log.d(TAG,sensorName + " Accel X: "+ sensorEvent.values[0] +
            " Accel Y: "+ sensorEvent.values[1] + " Accel Z: "+
            sensorEvent.values[2]);

        //buffer0
        while (samplesRead == numSamples) {
            float aSum = abs(sensorEvent.values[0]) + abs(sensorEvent.
                values[1]) + abs(sensorEvent.values[2]);
            if (counter < 118)
                counter++;
            for (int i = 0; i < 119; i++) {
                for (int j = 0; j < 3; j++) {
                    accBuff[j][i] = accBuff[j][i+1];
                    if(j==0){
                        accBuffX[i] = accBuffX[i+1];
                    }else if(j==1){
                        accBuffY[i] = accBuffY[i+1];
                    }else if(j==2){
                        accBuffZ[i] = accBuffZ[i+1];
                    }
                }
            }

            accBuff[0][118] =  Float.toString(sensorEvent.values[0]);
            accBuff[1][118] =  Float.toString(sensorEvent.values[1]);
            accBuff[2][118] =  Float.toString(sensorEvent.values[2]);
            accBuffX[118] =  Float.toString(sensorEvent.values[0]);
            accBuffY[118] =  Float.toString(sensorEvent.values[1]);
            accBuffZ[118] =  Float.toString(sensorEvent.values[2]);

            //Log.d(TAG, "Suma " + Float.toString(aSum));

            if (aSum >= accelerationThreshold && counter >= 118) {
```

```
                    flag = 1;
                    samplesRead = 0;
                    break;
                } else {
                    break;
                }
            }

            while (samplesRead < numSamples) {
                samplesRead++;
                accBuff[0][samplesRead + 118] = Float.toString(sensorEvent
                    .values[0]);
                accBuff[1][samplesRead + 118] =  Float.toString(
                    sensorEvent.values[1]);
                accBuff[2][samplesRead + 118] =  Float.toString(
                    sensorEvent.values[2]);
                accBuffX[118+samplesRead] =  Float.toString(sensorEvent.
                    values[0]);
                accBuffY[118+samplesRead] =  Float.toString(sensorEvent.
                    values[1]);
                accBuffZ[118+samplesRead] =  Float.toString(sensorEvent.
                    values[2]);
                if (samplesRead == numSamples) {
                    for (int i = 0; i < 238; i++) {
                        textViewContent=textViewContent+(accBuff[0][i] + "
                            ," + accBuff[1][i] + "," + accBuff[2][i] + ","
                            + girBuff[0][i] + "," + girBuff[1][i] + "," +
                            girBuff[2][i] + "\n");
                        // \n
                    }
                    //Log.d(TAG, textViewContent);
                }
                break;
            }
        }

        if (isAccelData & isGyroData) {
            isAccelData = false;
            isGyroData = false;
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int i) {

    }

    @Override
    public void onClick(View view) {

    }
}
```

Bibliography

[1] Fallas en pavimentos flexibles. USON.

[2] ALGARICI, N. Lecture 4- neural networks. urlchrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/ https://www.ee.bgu.ac.il/ haim-p/ml/lectures/lec4/lec4.pdf, 2020.

[3] ALLOUCH, A., KOUBAA, A., ABBES, T., AND AMMAR, A. Roadsense: Smartphone application to estimate road conditions using accelerometer and gyroscope. *IEEE Sensors Journal PP* (05 2017).

[4] ANDROID. Documentation for app developers. url https://developer.android.com/docs/ .

[5] APARNA, BHATIA, Y., RAI, R., GUPTA, V., AGGARWAL, N., AND AKULA, A. Convolutional neural networks based potholes detection using thermal imaging. *Journal of King Saud University - Computer and Information Sciences* (2019).

[6] ARROYO HILTON, N. F. Diseño y conservación de pavimentos rígidos. Facultad de Ingeniería, UNAM.

[7] AZASOO, J., AND BOATENG, K. A retrofit design science methodology for smart metering design in developing countries.

[8] COENEN, T. B., AND GOLROO, A. A review on automated pavement distress detection methods. *Cogent Engineering 4*, 1 (2017), 1374822.

[9] DE SEVILLA, U. *Sensor medidor de Aceleración. ACELERÓMETRO*. Universidad de Sevilla.

[10] DE ZOYSA, K., KEPPITIYAGAMA, C., SENEVIRATNE, G. P., AND SHIHAN, W. W. A. T. A public transport system based sensor network for road surface condition monitoring. In *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions* (New York, NY, USA, 2007), NSDR '07, Association for Computing Machinery.

[11] DEL TRANSPORTE, I. M. *CATALOGO DE DETERIOROS EN PAVIMENTOS FLEXIBLES DE CARRETERAS MEXICANAS*. Secretaria de Comunicaciones y Trasportes, 1991.

[12] DEL TRANSPORTE, I. M. Red nacional de caminos. urlhttps://www.gob.mx/imt/acciones-y-programas/red-nacional-de-caminos: :text=1%2C969%20km%20son%20carreteras%20 pavimentadas%20y%2024%2C768%20km%20son%20caminos%20rurales., 2020.

[13] DIPANJAN, S., RAGHAV, B., AND TUSHAR, S. *Practical Machine Learning with Python : A Problem-Solver's Guide to Building Real-World Intelligent Systems.* Apress, 2018.

[14] DU, R., QIU, G., GAO, K., HU, L., AND LIU, L. Abnormal road surface recognition based on smartphone acceleration sensor. *Sensors 20*, 2 (2020).

[15] ERIKSSON, J., GIROD, L., HULL, B., NEWTON, R., MADDEN, S., AND BALAKRISHNAN, H. The pothole patrol: Using a mobile sensor network for road surface monitoring. pp. 29–39.

[16] ESPARZA, G. R. *GUÍA DE PROCEDIMIENTOS Y TÉCNICAS PARA LA CONSERVACIÓN DE CARRETERAS EN MÉXICO.* Secretaria de Comunicaciones y Trasportes, 2014.

[17] ESPAÑA, M. *Sistemas de Navegación Integrada con Aplicaciones.* COMISIÓN NACIONAL DE ACTIVIDADES ESPACIALES, 2019.

[18] FAROBIE, O., HASANAH, N., AND MATSUMURA, Y. Artificial neural network modeling to predict biodiesel production in supercritical methanol and ethanol using spiral reactor. *Procedia Environmental Sciences 28* (12 2015), 214–223.

[19] FENG, M., FUKUDA, Y., MIZUTA, M., AND OZER, E. Citizen sensors for shm: Use of accelerometer data from smartphones. *Sensors 15*, 2 (2015), 2980–2998.

[20] GOOGLE. Sensors overview. urlhttps://developer.android.com/guide/topics/sensors, 2019.

[21] HOFFMANN, M., MOCK, M., AND MAY, M. Road-quality classification and bump detection with bicycle-mounted smartphones. *CEUR Workshop Proceedings 1088* (01 2013), 39–43.

[22] INEGI. Accidentes de tránsito terrestre en zonas urbanas y suburbanas. urlhttps://www.inegi.org.mx/sistemas/olap/consulta/general$_ver4/MDXQueryDatos.asp?c=$ $, 2019.$

[23] INEGI. Encuesta nacional de seguridad pública urbana. Tech. rep., INEGI, 2020.

[24] ISHIKAWA, T., AND FUJINAMI, K. Smartphone-based pedestrian's avoidance behavior recognition towards opportunistic road anomaly detection. *ISPRS International Journal of Geo-Information 5*, 10 (2016).

[25] JAIN, A. K., DUIN, R. P. W., AND MAO, J. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell. 22* (2000), 4–37.

[26] Jo, Y., and Ryu, S. Pothole detection system using a black-box camera. *Sensors 15*, 11 (2015), 29316–29331.

[27] Joaquim P. Marques de, S., Luís M.A., S., Jorge M.F., S., and Luís A., A. *Minimum Error Entropy Classification.*, vol. 2013 of *Studies in Computational Intelligence*. Springer, 2012.

[28] LECUONA, M. Sistema nervioso. UNAM.

[29] Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning.* Adaptive computation and machine learning series. The MIT Press, 2018.

[30] PEFFERS, K., TUUNANEN, T., ROTHENBERGER, M. A., and CHATTERJEE, S. A design science research methodology for information systems research. *Journal of Management Information Systems 24*, 3 (2007), 45 – 77.

[31] Rao, S. S., and Navarro Salas, R. *Vibraciones mecánicas.* Pearson, 2012.

[32] Sattar, S., Li, S., and Chapman, M. Road surface monitoring using smartphone sensors: A review. *SENSORS 18*, 11 (2018).

[33] Sattar, S., Li, S., and Chapman, M. Road surface monitoring using smartphone sensors: A review. *Sensors 18*, 11 (2018).

[34] Sattar, S., Li, S., and Chapman, M. Developing a near real-time road surface anomaly detection approach for road surface monitoring. *Measurement 185* (2021), 109990.

[35] Saucedo Vidal, A. Concreto hidraulico permeable, una alternativa para la recarga de los mantos acuiferos del valle de mexico. Facultad de Ingeniería, UNAM.

[36] Seraj, F., van der Zwaag, B. J., Dilo, A., Luarasi, T., and Havinga, P. Roads: A road pavement monitoring system for anomaly detection using smart phones. p. 128–146.

[37] Smith, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[38] Tacza Herrera, E. B., and Rodriguez Paez, B. O. Evaluación de fallas mediante el método pci y planteamiento de alternativas de intervención para mejorar la condición operacional del pavimento flexible en el carril segregado del corredor javier prado, 2018.

[39] Tidjon, L. N., and Khomh, F. Threat assessment in machine learning based systems, 2022.

[40] Wang, L., Zhang, D., Wang, Y., Chen, C., Han, X., and M'hamed, A. Sparse mobile crowdsensing: challenges and opportunities. *IEEE Communications Magazine 54*, 7 (2016), 161–167.

[41] WANG, L., ZHANG, D., WANG, Y., CHEN, C., HAN, X., AND M'HAMED, A. Sparse mobile crowdsensing: challenges and opportunities. *IEEE Communications Magazine 54*, 7 (2016), 161–167.

[42] WESTON, J., MUKHERJEE, S., CHAPELLE, O., PONTIL, M., POGGIO, T., AND VAPNIK, V. Feature selection for svms. *Advances in neural information processing systems 13* (2000).

[43] WU, C., WANG, Z., HU, S., LEPINE, J., NA, X., AINALIS, D., AND STETTLER, M. An automated machine-learning approach for road pothole detection using smartphone sensor data. *Sensors 20*, 19 (2020).

Curriculum Vitae

Master student was born in México City, México, on February 6, 1998. He earned the Industrial Engineering degree from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus in December 2020. He was accepted in the graduate programs in Computer Science in January 2021.

This document was typed in using $\LaTeX\,2_\varepsilon$[a] by Carlos Alonzo López Castañeda.

[a]The style file `phdThesisFormat.sty` used to set up this thesis was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus