Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2021

NUNO FILIPE SANTOS APARÍCIO

# SERVIÇOS DE MONITORIZAÇÃO E ALERTA METEOROLÓGICOS PARA SISTEMAS COOPERATIVOS DE TRANSPORTE INTELIGENTES

# ROAD WEATHER MONITORING AND ALERT SERVICES FOR COOPERATIVE INTELLIGENT TRANSPORTATION SYSTEMS

**Universidade de Aveiro** Departamento de Eletrónica, Telecomunicações e Informática
**2021**

**NUNO FILIPE SANTOS APARÍCIO**

# SERVIÇOS DE MONITORIZAÇÃO E ALERTA METEOROLÓGICOS PARA SISTEMAS COOPERATIVOS DE TRANSPORTE INTELIGENTES

# ROAD WEATHER MONITORING AND ALERT SERVICES FOR COOPERATIVE INTELLIGENT TRANSPORTATION SYSTEMS

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor Joaquim José de Castro Ferreira, Professor adjunto da Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro, e do Doutor Paulo Jorge de Campos Bartolomeu, Investigador Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

**o júri / the jury**

| | |
|---|---|
| presidente / president | Prof. Doutor José Nuno Panelas Nunes Lau<br>Professor Associado, Universidade de Aveiro |
| vogais / examiners committee | Prof. Doutor José Carlos Meireles Monteiro Metrôlho<br>Professor Adjunto, Instituto Politécnico de Castelo Branco |
| | Prof. Doutor Joaquim José de Castro Ferreira<br>Professor Adjunto, Universidade de Aveiro |

**agradecimentos / acknowledgements**

Agradeço a ajuda de todos os que me acompanharam no meu percurso académico, principalmente à minha família que sempre me apoiou de forma incansável e facilitou para que tudo fosse possível.

Este trabalho é financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER), através do Programa Operacional Competitividade e Internacionalização (COMPETE 2020) do Portugal 2020 [Projeto TRUST com o nº 037930 (POCI-01-0247-FEDER-037930)];

Cofinanciado por:

COMPETE 2020

PORTUGAL 2020

UNIÃO EUROPEIA
Fundo Europeu
de Desenvolvimento Regional

**Palavras Chave**

Estrada, Clima, Meteorologia, Veículo, Dispositivo móvel, Monitorização.

## Resumo

Com o aumento de veiculos nas estradas e do consequente tráfego rodoviário em todo o mundo, inúmeros problemas foram surgindo, tais como a poluição do ar, o congestionamento ou os acidentes rodoviários, sendo necessárias estratégias inteligentes para minorar estes problemas. Uma das principais situações que, infelizmente, continua a causar bastantes vitimas são os acidentes rodoviários, que tem muitas das vezes como causa condições meteorológicas adversas, sejam situações como chuvas fortes, nevoeiro, pouca luminosidade, ou gelo. O projeto TRUST procura soluções para evitar que estas situações causem incidentes rodoviários. Para tal, promove o desenvolvimento de um sistema de monitorização meteorologica e ambiental com o objetivo de identificar potenciais riscos, causados por condições atmosféricas adversas e alertar os condutores, recorrendo a tecnologias de sistemas cooperativos de transporte inteligentes. O projeto aponta como uma das suas soluções mais importantes, uma aplicação móvel capaz de alertar condutores em tempo real sobre estas condições bem como servir de ferramenta de monitorização destes parametros. Esta dissertação está inserida no contexto deste projeto e visa o desenvolvimento de novas funcionalidades e ferramentas da aplicação para telemóvel, tais como a recolha de dados de luminosidade e de videos da estrada, permitindo o desenvolvimento de mecanismos automáticos de deteção e identificação de condições adversas para condução através do processamento das imagens recolhidas, como a existência de chuva ou pouca luminosidade na estrada, um algoritmo de recomendação de velocidade em tempo-real, com base nos incidentes existentes, entre outros. Com estas funcionalidades implementadas, a aplicação móvel tem o potencial de aumentar a segurança rodoviária, ao fornecer uma velocidade recomendada real e assertiva para a localização do condutor e também melhora o serviço de monitorização do projeto TRUST, ao analisar e identificar automaticamente eventos meteorologicos perigosos para os condutores, partilhando rapidamente esta informação com o resto do sistema, que pode prevenir e diminuir o número de acidentes nas estradas.

**Keywords** Road, Weather, Meteorology, Vehicle, Mobile Device, Monitoring.

## Abstract

With the increase in vehicles on the roads and the consequent road traffic around the world, numerous problems have arisen, such as air pollution, congestion and road accidents. Smart strategies are needed to minimize these issues. One of the main situations, that unfortunately continues to cause many victims, is road accidents, which are often caused by adverse weather conditions, such as heavy rain, fog, insufficient light, or black ice. The TRUST project seeks to prevent these conditions from being the cause of road accidents. It also promotes the development of a meteorological and environmental monitoring system in order to identify potential risks that may lead to dangerous driving conditions, through cooperative communication between vehicles and infrastructures and using sensors and technologies for intelligent transport systems. The project points out, as one of its most important solutions, a mobile application capable of alerting drivers in real time about these conditions as well as serving as a tool for monitoring these parameters. The following dissertation occurs within the scope of this project and aims to develop new functionalities and tools for the application and system. Some of these functionalities/tools are: the collection of luminosity data from the environment, videos of road and weather conditions via smartphone, the use of image processing to aid in the development of automatic mechanisms for the detection and identification of adverse weather conditions, such as rain or insufficient light on the road, a real-time speed recommendation algorithm based on existing incidents, among others. With these features implemented, the application increases user safety by providing a real and assertive recommended speed limit for the driver's location. It also improves the TRUST project monitoring service, by automatically analyzing and identifying dangerous weather events for drivers and by sharing this information quickly with the rest of the system, which can prevent and reduce the number of road accidents.

# Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| **ITS** | Intelligent Transport System |
| **C-ITS** | Cooperative Intelligent Transport System |
| **TRUST** | Transportation and Road Monitoring System for Ubiquitous Real-Time Information Services |
| **CAM** | Cooperative awareness message |
| **DENM** | Decentralized Environmental Notification Message |
| **PASMO** | Plataforma aberta para o desenvolvimento e experimentação de soluções para a mobilidade |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **OBD** | On-Board Diagnostics |
| **OBU** | OnBoard Unit |
| **RSU** | Road side Unit |
| **GPS** | Global Positioning System |
| **WEBRTC** | Web Real-Time Communication |
| **RGB** | Red, Green, Blue |
| **API** | Application Programming Interface |
| **USB** | Universal Serial Bus |
| **MQTT** | Message Queuing Telemetry Transport |

# CHAPTER 1

# Introduction

*One of the greatest innovations of all times was the creation of road networks capable of connecting practically the whole world. With the increased use of roads and vehicles, it became necessary to continuously improve the road system and seek innovative solutions that maintain the safety of all its users. Nowadays, more than electronic panels and traffic signs, systems capable of monitoring, predicting and alerting are needed in order to be able to make meaningful decisions quickly, using one of the greatest weapons of today, technology.*
*Intelligent transport systems aim to improve transportation, introducing communication and information technologies, focused on monitoring and sharing parameters that influence driving, with the purpose of making mobility more connected and more cooperative.*
*The goal is to keep improving and developing road system solutions that form a C-ITS (Cooperative Intelligent Transportation Systems) which strives for a strong bond between vehicles, infrastructures and drivers. C-ITS combines different services and communication technologies for cooperation between vehicles and their surrounding environment, through the use of vehicle communications (V2V, V2I and V2X) that support data sharing without the direct intervention of managing entities.*
*This dissertation is part of a project called TRUST, that asserts itself as an intelligent transport system (ITS) and aims to facilitate the implementation of technology in the transport area in order to make roads, and consequently cities, more intelligent. TRUST contributes to achieve maximum road traffic efficiency and safety by focusing on constantly monitoring adverse weather conditions and disseminate information to drivers in advance, to reduce negative road situations. The project presents, as one of its main solutions, an application for mobile devices that will help in the monitoring of weather conditions and in the sharing of road incidents to drivers in the form of notification. This dissertation presents the continuation of the development of the application, introducing new features implemented in the mobile application and tools for the TRUST project, such as collecting luminosity levels on the road, collecting and detecting weather events through the smartphone's camera, a speed recommendation algorithm based on events generated by the system, among others. In a global context, the technological tools presented improved the monitoring and alert services of the project and returned the most positive results to the intelligent transport system.*

## 1.1 Motivation

Currently, road safety is very important. Unfortunately, it still hasn't earned the respect and prominence deserved. It is something to invest in and find new solutions to improve the way the road system functions. As we know, technology is naturally present and essential in every feature of our daily lives.

Thus, the development of new technologies, which keep the entire road system connected and exchanging information automatically and in real time, is an almost mandatory step to keep all drivers safe and comfortable. This was my great motivation for participating in this project, helping to develop something that can truly make a difference.

## 1.2 Objectives

In the near future, both cars and driving are expected to be much safer than they are today. For this purpose, it is intended that cars anticipate, or inform their driver in advance, about events that may possibly affect their driving and safety. For this to be possible, all vehicles and infrastructures must be connected and work cooperatively.

The TRUST project focuses on monitoring weather conditions that can affect driver safety. One of the project's ambitions is to develop new technologies for both vehicles and infrastructures, capable of collecting, analyzing and sharing information, at any time and any place.

One of the solutions presented, to improve the collection of information and cooperation between vehicles and infrastructure, is a mobile application capable of monitoring driving conditions and alerting drivers about events in certain locations. In addition to being able to alert other drivers about the incidents that have occurred, the intent is for the application to contribute itself to the monitoring of weather conditions and to improve the driver's safety, by developing new tools and functionalities. The first through:

- The collection of important road data like luminosity values of the surrounding environment, through the smartphone light sensor.
- The collection of video images of the road, simulating the driver's point of view, through the smartphone camera.
- The development of image processing models to analyze the collected images and be able to determine existing hazards quickly and automatically.
- The development of a video stream service in case of accident.

The second through:

- The integration of new specific weather events to describe each situation in detail.
- The development of a voice interaction service to validate events generated by the system.
- The implementation of an algorithm that provides the driver with a safe speed recommendation in real time based on the type of road, type of vehicle and existing events in the vicinity of the user.

Therefore, in addition to fulfilling the basic objective of being an interface and sharing events with the driver, the application also plays a fundamental role both in the driver's safety and in the collection of road information for the TRUST project.

## 1.3 Document Organization

The rest of the dissertation is organized as follows:

- Chapter 1 - Introduction: Presents the motivation, objectives and challenges of the presented work.

- Chapter 2 - Fundamental Concepts and State-of-the-Art: This chapter presents an overview and a detailed description of concepts related to the Intelligent Transport System and informs the reader about the current technologies that are used in this field which are crucial for the development of this dissertation.
- Chapter 3 - System Architecture: This chapter presents the overall architecture of the TRUST project, the mobile application solution and every component that interacts with it.
- Chapter 4 - Implementation: In this section we discuss how all the application's features and system programs were implemented and developed, following the line of thought and purpose in which they were created.
- Chapter 5 - Tests and Validation: This chapter presents the methods used to test the tasks described during the dissertation and the results obtained.
- Chapter 6 - Conclusions and Future Work: A conclusion about the work, the tasks developed and future work.

CHAPTER 2

# Fundamental Concepts and State-of-the-Art

## 2.1 Intelligent Transportation Systems

Road transport is fundamental and widely used today. Almost everyone needs to move from one location to another and use the roads to do so. With the wide use of this mechanism and the constant increase in traffic on the roads, there is also an increase in the dangerous and alarming situations that drivers face. Since it is essential, now and in the future, it is up to the road transport system to find solutions so that safety is maximized as much as possible. Although there are rules, namely those of the road code, they are hardly enough to keep roads safe, for many situations drivers face derive from conditions that are not theoretically controllable, such as behavior from other drivers or poor driving conditions.

With a focus on solving these problems, Intelligent Transport Systems emerge as developed systems which, through innovative technologies and applications, seek to provide users with a safer, more efficient and more intelligent type of transport. ITS applies all types of technologies to the transport system, such as applications for detection, analysis, control and communication of road data, that allow travelling in better and safer conditions.[1]

Climatic issues such as floods, rain, strong winds and ice formation are a major factor that influences drivers, leads to poor conditions and can cause great instability. The fact that in many situations the driver is not prepared to face these conditions and is caught off guard makes driving extremely dangerous both for himself/herself and those around him/her. The main reason why this situation is recurrent is due to the lack of communication between the driver and the rest of the road system. As a consequence the lack of information creates uncertainty, instability and insecurity. Thus, an intervention is needed that addresses this issue and presents solutions for these gaps.

### 2.1.1 TRUST

TRUST is a project that seeks to respond to these flaws by developing a system capable of monitoring environmental and meteorological factors in order to identify potential risks in road driving and share that information with the drivers. It intends to do so, by creating a communication bond between vehicles and everyone/everything pertaining to the road system, such as pedestrians, cyclists, traffic lights, traffic control center and others. This is necessary in order to be able to rely on real-time data exchange to alert drivers of possible risks and exact situations in certain locations.

So, the first main focus is the monitoring of weather conditions and variables that could possibly influence drivers, by gathering sensor data, either through roadside infrastructures or through the vehicles themselves. Once collected, assertive and careful analysis of all data coming from the sensors will be provided, in order to identify potentially negative consequences for drivers, such as lack of visibility, precipitation, strong winds, floods, fires, lack of light, among others.

Although it is extremely important to collect and analyze data, it is equally important to minimize the information gap that drivers constantly suffer. To this end, and as the second main focus, the project presents solutions for making the information collected available in a quick and coherent way, through the generation of events that will be disseminated to drivers through alerts in the form of notification. While traditional road systems are only able to make this information available through basic services such as electronic panels on highways or through radio communications, the TRUST project wants to present platforms that are able to do it faster and in real time, making sure that all drivers can act upon this important information anytime and anywhere.

In order to achieve these goals, the project has two clear ambitions: to develop a driver notification interface and to present an easy, functional and effective interoperability with third party systems.[2]

*Driver Notification Interface*

The driver interface is presented as a solution of the TRUST project and is part of the development of a fast and connected system meant to improve the connection between all the participants of the road transport system.

It is a proposal of an interface that will provide relevant notifications related to potential risks to the driver, based on the data collected and processed by different parts of the system. This interface is, in essence, a mobile application that will allow the driver to receive alerts, in the form of notifications about road events associated with certain locations. This solution has a bi-directional role, as it serves the purpose of disseminating events generated by the system's processing tools, as well as being itself a sensor that collects real data when passing through different locations.

This is, therefore, a solution that challenges traditional systems as it speeds up the process of disseminating and sharing information among all participants in the system, which is undoubtedly essential and necessary today.

The solution will be addressed at a further stage, in a more detailed way, to clarify its involvement with the entire system architecture. [2]

*Interopability with Third Party Systems*

One of the main and most useful goals of the TRUST project is the ability of the system to integrate data from different sources. As such, the greater the information, the greater the ability to conclude with assertiveness and confidence about risk events to alert drivers. Thus, it is important to integrate external systems to complement the data already collected by the physical components of the system itself.

The main focus of the system is to keep roads and drivers safe. Therefore, it is important to collect as much information as possible. External systems strictly focused on different areas are an asset integrated into the TRUST project with this clear intention. HERE and TomTom are two important sources of data about road traffic, that allow access to relevant information about this subject and to produce alerts of events related to traffic conditions, in real time. The national authority itself allows the collection of certain events, such as road accidents, enabling, thus, the creation of events related to incidents in certain locations. The OpenWeather service is an example of a service which provides real-time information on weather conditions in specific regions and based on this information drivers can be warned about possible road hazards related to weather events.

As we have seen, the integration of third-party systems is important and essential to collect interesting and reliable data, which makes the entire TRUST project work smoothly and in harmony towards its goal of providing drivers with the necessary road information in advance that can be crucial to prevent incidents. [2]

### 2.1.2 Vehicular Communications

An intelligent transport system seeks to improve transportation applying services that increase road safety and road traffic efficiency, forming a system in which all parties collaborate and communicate with each other, for a quick and assertive collection, analysis and sharing of information. To take a road system to a higher level, it is necessary to involve connectivity and cooperation between services, forming a Cooperative-ITS, which combines different protocols of communication technologies. The basis of a C-ITS communication is the ability to share data directly, in an autonomous way, between vehicles and their surrounding environment, whether with other vehicles, infrastructures, or other devices, and are known as vehicular communications, in which they assume the form of V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure) and V2X (Vehicle-to-Everything).

**Vehicle-to-vehicle** communication technology (V2V) allows cars to communicate with each other and share information in real time. In relation to other technologies, it has the advantage of giving prior notice to other vehicles, which share the same V2V technology, about data collected or even potential dangers. For instance, a vehicle can warn others that it will change lanes and the ones that may be affected can know beforehand.[3]

**Vehicle-to-Infrastructure** communication technology (V2I) allows cars and infrastructures to exchange data where the infrastructure supports all connected cars and shares information received by both vehicles or other systems. Infrastructure support is more focused on sharing information about driving conditions, such as current traffic status, weather and road conditions. With this type of information, infrastructures are able to generate countless possibilities to create and make resources, such as speed recommendations, detours, traffic light alerts, arrival times, among others, available.[3]

However, a combination of these two would be ideal, and **Vehicle-to-Everything** communications (V2X) combines both V2V and V2I. This makes the road system much safer, allowing all vehicles to communicate with other vehicles and infrastructures simultaneously. These technologies are essential and the basis for a future of fully autonomous road systems. The greater the number of system solutions with the V2X technology, the more noticeable the advantages of an intelligent road system will be. The cooperation between infrastructures that collect parameters regarding road conditions and the connectivity of the information shared between the infrastructures and vehicles, allow for a faster and more efficient flow of information, which will make the road system more secure.[3]

These communications allow the sharing of information such as location, speed, or even incident alerts and they need a communication technology that allows this data exchange to be done in real time. Thus, short distance radio communications, called Dedicated Short-Range Communications, were developed and these allow the quick sharing of information between vehicles and infrastructures that are at a relatively close distance. In Europe, specifically, the European Telecommunications Standards Institute (ETSI), has developed a communication technology with the same principles and objectives which is called **ITS-5G**.[4]

This technology acts on the 5.9 GHz band, which is reserved for the use and support of ITS applications only and is normally used for communication between Vehicle and Roadside ITS station units. The reservation was granted due to the fact that applications require real-time and low latency communications, and cannot run the risk of interference. A possible delay in receiving a message, due to interference, can prevent a driver from being warned in time about an accident 100 meters beforehand, therefore not being able to brake safely.

The use of the 5.9GHz band suits the needs of ITS applications because it:

1. Allows that a large amount of data be transferred simultaneously, due to its large bandwidth.
2. Provides low interference communications, due to its short wave range and the use of 10 different channels, 5.85 Ghz - 5,925 Ghz.

Initially, short range communications may be seen as a negative point, but this is not the case, as vehicles communicate with infrastructures on the side of the road that are relatively close to their radius of action. So, Dedicated Short-Range Communications are in fact a perfectly suitable choice in these situations.[5][6]

The sharing of information between the main players happens specifically through two types of messages, detailed below, CAMs and DENMs, and they are transmitted quickly and safely due to the characteristics of this technology network.

*Vehicle and Roadside ITS station units*

When approaching an intelligent system that is based on vehicle communications, it is extremely important to mention the devices that are part of and make this communication possible. We can say that each system needs some specified hardware for better development, integration and continuous communication of all parts. In the case of ITS, two fundamental units for V2X vehicle communications are **Vehicle ITS station** unit and **Roadside ITS station** unit, also know as On-Board Unit (OBU) and RoadSide Unit (RSU).

Vehicle ITS station units are devices on board of each vehicle that assume their importance in several aspects: collecting information on driving conditions and parameters of the vehicle, and allowing the transmission and reception of road alerts generated by the management platform, presented to the driver at a later stage. Viewed in a general context, it bridges the gap between the driver and the road system.

Roadside ITS station unit is the hardware that is present in the infrastructures that gives support to the vehicles. It obtains information on traffic, accidents, locations, weather conditions and others, and has the responsibility to disseminate that information with the vehicle's OBU.

The hardware used in these units is essential for these communications to take place and these units play a key role in assuring the rapid fluidity of road information. This allows, that the main objective of informing the drivers of adverse conditions and possible accidents beforehand and maintaining the road system connected and cooperative, be achieved.

### 2.1.3 C-ITS Protocol Stack and Basic Services

Since, the TRUST project, in which this dissertation occurs, falls within a C-ITS and works toward a connected road system, in the following subchapter we will approach the structure of a C-ITS, as well as its components, services and rules. The current state of the art on messages and basic services used in vehicular communications, DENM and CAM, as well as the protocol stack itself are set out below.

*ITS Protocol Stack*

The OSI model is a reference model for computer networks in which the system's architecture is divided into seven layers, each one with different protocols and functionalities. The goal of this model is that all seven layers work and communicate with each other in order to transmit data from one layer to another.[7]

Although an intelligent transport system (ITS) is based on the OSI model, a few changes are apparent in the figure below. An ITS architecture only has four horizontal protocols layers and two vertical entities.

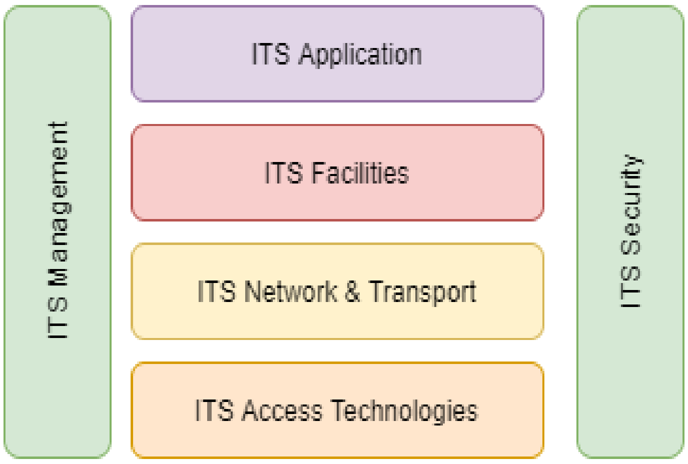


**Figure 2.1:** ITS Protocol Stack

Like the OSI model, each layer has its own tasks and executes them individually, later interacting with the other layers. The system's architecture is subdivided and organized in a way that is simpler and less complex. Finally, by interacting with each other and sharing information across the system, the stack works as a whole to fulfill its responsibilities.

Next, we will analyze how each layer behaves and how they interconnect and communicate with each other, in order to understand the importance of this model.[8][9]

1. The horizontal layers of the protocol stack are divided as follows:
    - The lowest layer of the protocol stack is the *ITS access technologies* layer and it is responsible for connecting the different ITS stations with the access technologies used, to ensure that the data transfer is made without errors, from one node to another, in the physical layer.
    - The second layer is called the *ITS network and transport* layer, and is responsible for delivering and transmitting information between ITS stations, and also from ITS stations to other nodes on the network. This layer includes both the network layer and transport layer of the OSI model.
    - The ITS *Facilities* layer creates data structures, received from different sources, capable of storing and maintaining data of different types, whether they are sensors or external communications. It is also responsible for granting access to common functionalities to applications by providing specific message management for ITS applications such as CAMs or DENMs. The facilities layer provides basic support services for ITS applications to build, manage, and process these type of messages. This layer, besides being very important in the protocol stack, takes on an extra emphasis in this dissertation and will be focussed on later on in this chapter.
    - Finally, we have the last layer called *ITS application* layer, which simply refers to ITS applications and their use in road safety, traffic efficiency, and others.
2. In addition to the horizontal layers, we can see that there are two entities in the vertical of the stack: the *ITS management* entity and the *ITS security* entity.

- *ITS management* entity is responsible for the management of each ITS station as well as the exchange of information between the different layers of the stack.
- *ITS security* entity is responsible for ensuring the privacy of services, good authentication protocols and security in the exchange of messages between different layers of the stack.

*Cooperative awareness message*

A cooperative awareness message or CAM is a specific periodic message for road users (drivers, trucks, bicycles) to communicate with other users and roadside infrastructures (traffic lights, gates, traffic signs) and inform them about each other's multiple parameters, such as their position, speed or headings, etc. The CAM is generally used by vehicles in certain areas to prevent accidents.

The CA's basic service is an entity which acts upon the ITS facilities layer and manages the CAM protocol and it is responsible for the construction, management and processing of CAMs, this means that two services are provided: sending and receiving CAMs.

The service of sending CAMs consists of two processes: generating and transmitting these messages. There is an ITS station that composes the message, and then the ITS transport layer is used to disseminate it. The dissemination process uses point-to-multipoint communication and varies from system to system, since CAM messages can be sent from one ITS station to another ITS station, or from one ITS station to all ITS stations, within the network. This also means that a CAM message is never forwarded from a receiving ITS station to other stations.

The process of receiving messages is simpler than the process of sending messages. In this phase, the CA's basic service provides the ITS application layer with the CAM information that has to be processed. At a later stage, each application decides on how to proceed with its content.

The CA's basic service interacts directly with the ITS Network and Transport layer's for the exchange of information between ITS stations, with the Security entity (ITS Security Service Access Point), in order to ensure a secure message exchange (sending and receiving CAMs). Both the ITS Management entity and ITS Applications layer make the message data available to ITS applications.[10]

The figure below shows how CAMs fall within and relate to the rest of the system, making the mechanisms presented above possible.

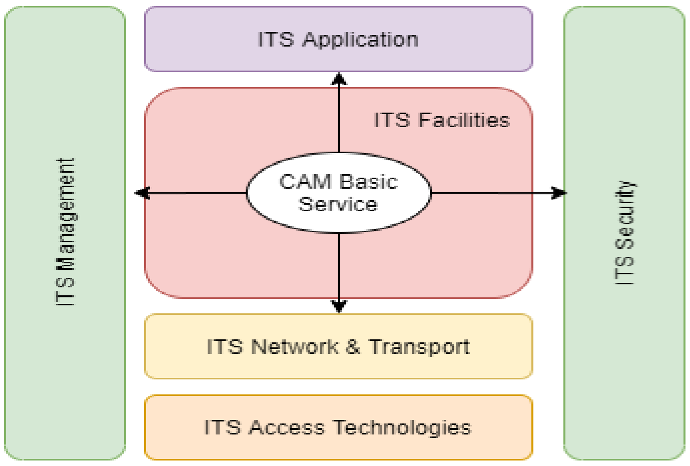


**Figure 2.2:** CA basic service within the ITS-S architecture

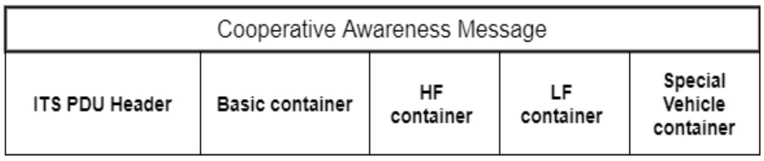
These messages contain a specific structure that must be followed by the generation process of the sending ITS station. It is composed of an ITS PDU header and several containers. First, the header presents information about the protocol version, the message type and the ITS-S ID of the originating ITS-S. In relation to containers, a CAM from a vehicle's ITS station must present:

- A basic container that presents general information about the ITS station from which the CAM originates, such as the type of the ITS station and its last geographical position when the CAM was generated.
- The high frequency container that contains highly dynamic information about the ITS station from which the CAM comes, such as the speed or direction of the ITS station.

In addition to the mandatory containers, the CAM message may display:

- The low frequency container that presents static or little dynamic information of the ITS station from which the CAM comes, such as the current state of the vehicle's lights.
- A special container that presents information specific to the role of the vehicle of the ITS station from which the CAM comes, as for example in the case of a public transport or emergency vehicle.

When CAMs are generated by a Road side unit, they must include only the basic container, the other containers are optional.[10]



| Cooperative Awareness Message | | | | |
|---|---|---|---|---|
| ITS PDU Header | Basic container | HF container | LF container | Special Vehicle container |

**Figure 2.3:** CA message structure

*Decentralized Environmental Notification Message*

A DENM or Decentralized Environmental Notification Message is a message based and triggered by specific events. The main purpose of these messages is to describe and alert drivers (and their vehicles) of events which were detected, that may possibly affect driving.[11]

The DENM protocol is responsible for the the exchange of events between ITS stations and follows a series of steps:

1. The ITS station that detects the event (known as the originating ITS-S) transmits the message, so that it can disseminate it as quickly as possible to all the stations that may be affected within the area of relevance.
2. The transmission process of a DENM starts and ends with an ITS-S application, and may be repeated. In addition, the DENM can subsist while the event is active.
3. The event termination can be achieved in two ways, either by having reached the expired time or by the originating ITS-S application asking for the termination of the event.
4. After processing the information shared by the DENM, receiving ITS stations decide whether it is appropriate to notify the driver.

The DENMs also have an entity that operates on the DENM protocol, the DEN basic service, which provides services to ITS applications (in the application layer). This entity provides services for ITS-S applications such as:

- Report, update or end a certain event, if it is an originating ITS-S.
- Receive and process the message data and make this information available to applications, if it is a receiving ITS-S.
- Message forwarding service.

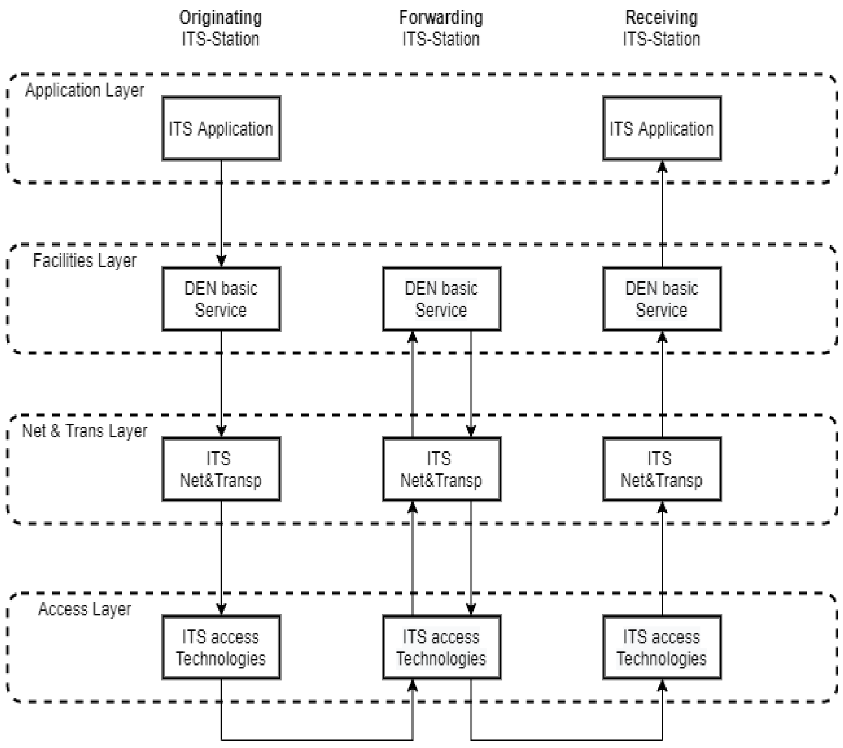


**Figure 2.4:** Forwarding process by DEN basic service

There are some differences between CAMs and DENMs regarding services and data flow between ITS stations. One being that a DENM may be forwarded, using intermediate stations to alert other ITS-S stations that are within the relevance radius of the generated event, when it is not possible to establish a direct communication between the originating ITS-S and other receiving ITS-Ss. The basic DEN service, located in the ITS facilities layer, is in charge of forwarding the DENM, slightly altering the basic data flow that is present in the CAMs. Thus, an ITS-S can be used to receive the DENM and send it to other ITS-Ss, without even processing the information it contains.

Since DENMs report various types of events that may have a negative impact on road safety, it is presumed that such events will occur in the most varied ways and situations. For a complete event report, a DENM structure consists of the type of event and its position, as well as the detection time and duration.

Therefore, to cover the different situations in which these messages can be used, the type of DENM can be the following:

- New DENM - An event is detected for the first time.
- Update DENM - Message with new information about an already existing event.
- Cancellation DENM - Message, sent by the originating ITS-S, informing the end of the event in question.
- Negation DENM - Message, sent by other than an originating ITS-S, informing the end of an event.

The DENM's basic service is an entity which acts upon the ITS facilities layer and manages the DENM protocol, whose functionality consists of communicating with the adjacent layers and the two vertical entities (ITS Management and ITS security) present in the architecture of the ITS station, through its interfaces and access points. The service provides APIs for ITS-S applications (ITS applications layer), allowing message generation, message forwarding and message reception.

As the exchange of DENMs has to be something universal and easy to integrate, the message always follows its own structure, consisting of the ITS PDU header and multiple containers, which forms the DENM payload. The ITS PDU Header is a common header, as it exists in CAMs, which presents information such as the protocol version, the message type and the ITS station identifier. In addition to the header, a DENM from a vehicle's ITS station features: [11]

- The management container presents information related to the DENM management, such as the action of the message (actionID), when the event was detected and its validity, the position of the event, the type of ITS-S that generated the message etc.
- The situation container presents information about the event that was detected and reported in the DENM, such as the event type, which includes the cause and the sub-cause of why the event is happening, some linked cause that may be related to this event, or even several other locations where this event may have occurred.
- The location container contains information about the event location and describes directions, locations and road types.
- The à la carte container is where additional data that is not appropriate to be in any other container is inserted, such as external temperature, road works, or others. It Is very specific data of the event.

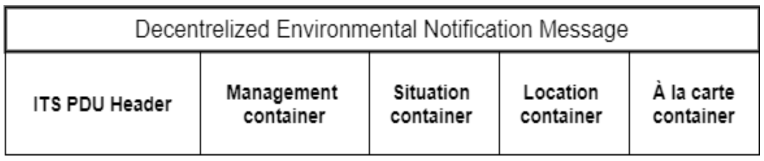
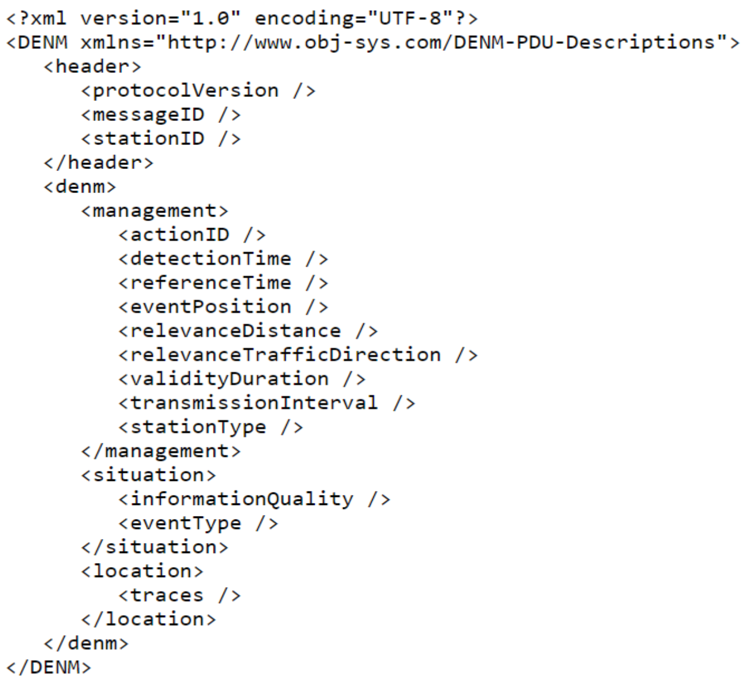


**Figure 2.5:** DEN message structure

Finally, the following figure shows an example of a DENM generated with its main essential parameters, for a better understanding of the topic covered. In this particular example, the message presents the header together with three containers, which can be seen in each tag individually.



```
<?xml version="1.0" encoding="UTF-8"?>
<DENM xmlns="http://www.obj-sys.com/DENM-PDU-Descriptions">
   <header>
      <protocolVersion />
      <messageID />
      <stationID />
   </header>
   <denm>
      <management>
         <actionID />
         <detectionTime />
         <referenceTime />
         <eventPosition />
         <relevanceDistance />
         <relevanceTrafficDirection />
         <validityDuration />
         <transmissionInterval />
         <stationType />
      </management>
      <situation>
         <informationQuality />
         <eventType />
      </situation>
      <location>
         <traces />
      </location>
   </denm>
</DENM>
```

**Figure 2.6:** DEN message example

### 2.1.4 Advanced Driving Assistance Mobile Applications

As we saw earlier, the TRUST project presents, as part of the solution, a platform for mobile devices, which serves as a human-machine interface. The main purpose of the application is to provide drivers with a platform for quick and easy visualization of road weather warnings so that it is an extra help to prevent certain incidents. Since this platform poses as an important solution of the project, it is essential to analyze the current state of the art, in order to develop the best possible solution and try to address flaws found in other applications.

Within driver-assistance mobile applications, each application tries to focus on one of the road themes, which is why there are several applications with so many different features. While some applications try, for example, to focus on calculating and presenting the best routes from one place to another, others focus, for example, on alerting the driver to events that may impact their driving.

*Google Maps*

The most well-known and used driving assistant application is, without doubt, **Google Maps**.Due to the enormous power of Google, this application has developed a large number of features, many of which are outside the scope of driving, but which nevertheless are able to help the driver, through the mobile application, find the best routes, calculate the fluidity of traffic relevant to the user, as well as report events that may have occurred.

Google Maps is not entirely focused on fully assisting the driver in this way, allowing a small number of events such as accidents, vehicles stopped on the road, road works, cut lanes to be reported and consequently emitting an alert to slow down the vehicle. However, it is still not possible for events regarding weather conditions to be reported.

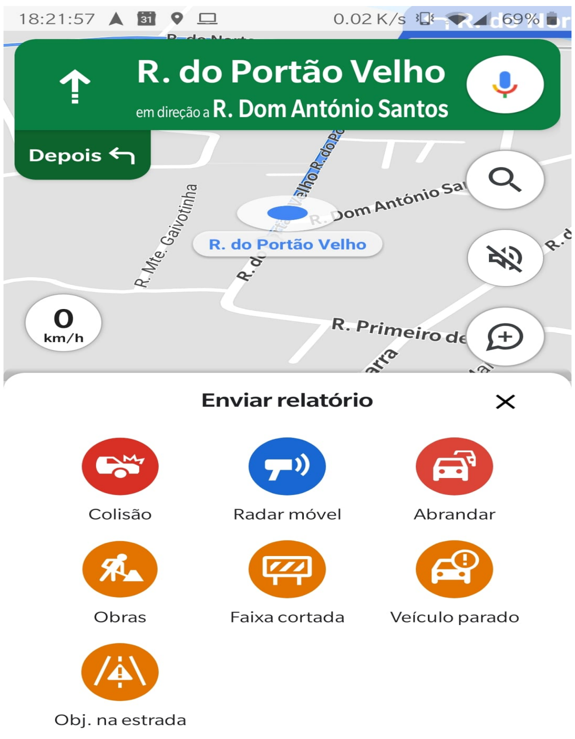


**Figure 2.7:** 'Google Maps' application

*Waze*

**Waze** is another well known and widely used mobile application. In addition to providing users with the best routes to their destinations, this application also offers diverse services, regarding events and road traffic, similarly to Google Maps. It is possible to receive alerts about traffic status in real time as well as other types of events such as accidents, dangerous situations, roadside assistance or road closures.

Waze is a recent application considered to be one of the greatest applications in the market. It continues to make great progress, presenting good measures and features for many areas, including road safety and road accident prevention. The voice command, developed by Waze, is an excellent example of an innovative feature which enhances road safety by allowing users to send hand-free real-time alerts. The following image shows how simple and intuitive it is to use the WAZE mobile application, and warn other drivers about potentially hazardous driving events.

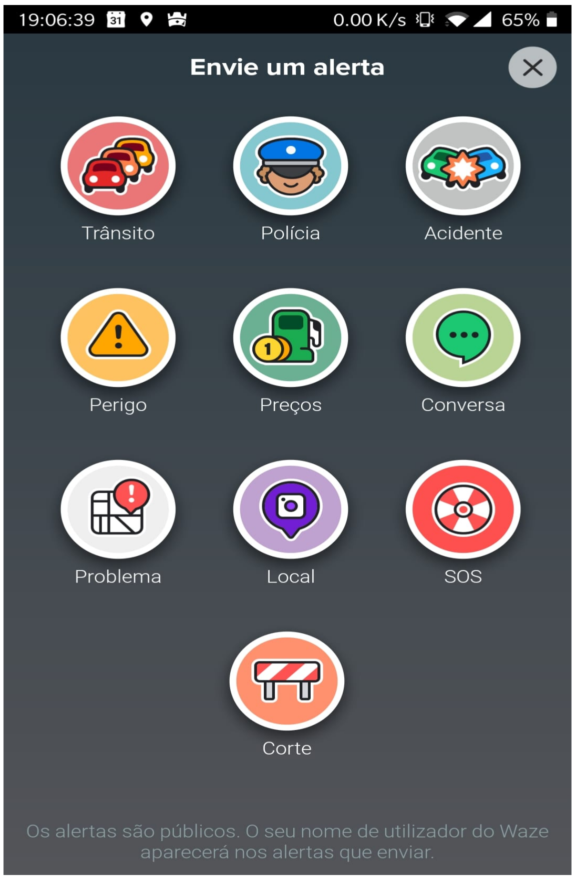
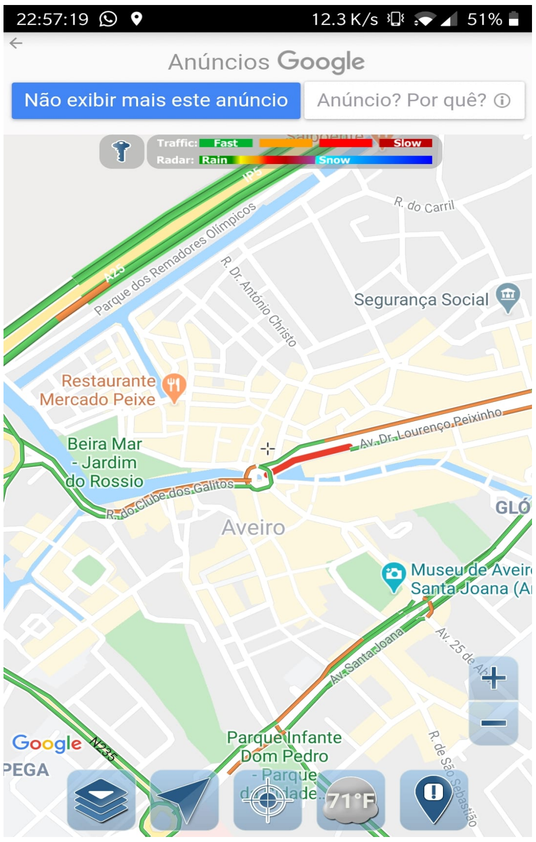


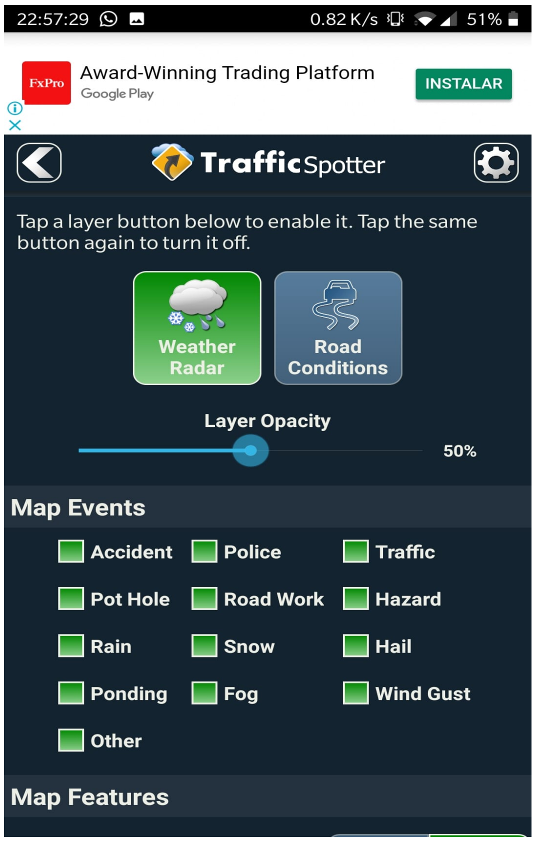
**Figure 2.8:** 'Waze' application

*Traffic Spotter - Traffic Reports*

There are other less widely used applications that exchange information for a more intelligent road system such as the app **'Traffic Spotter - Traffic Reports'**. This application's main features are to provide information about road traffic and road incidents, combining it with events and weather conditions, the latter being a very important factor often associated with accidents. The application's motivation to grow an intelligent and cooperative road system, in the future, seems to be going in the right direction. Despite already providing several features that are the basis of an ITS, such as showing traffic congestion and multiple map events, where weather conditions are included, the application is far less intuitive, in comparison with the previous two. The following images give us an idea of how the combination of information regarding road traffic, road incidents and weather conditions can have a great impact in the future.


22:57:19
12.3 K/s 51%
Anúncios Google
Não exibir mais este anúncio
Anúncio? Por quê?
Traffic: Fast Slow
Radar: Rain Snow
Segurança Social
Restaurante Mercado Peixe
Beira Mar - Jardim do Rossio
Aveiro
Museu de Aveiro Santa Joana
Parque Infante Dom Pedro - Parque da Cidade
71°F
Google

(a) Map activity



Tap a layer button below to enable it. Tap the same button again to turn it off.

(b) Event activity

**Figure 2.9:** 'Traffic Spotter - Traffic Reports' application

In the table below, it is possible to conclude that there are many key aspects the TRUST mobile application focuses on which are absent in all other applications. As mentioned, all of the applications specified above provide the driver with the functionality to report events, but in a very limited way. However, when meteorological events or adverse conditions are taken into consideration, the capacity of these applications to report these events are limited. We can also verify that other safety aspects, such as, data collection, monitoring, speed recommendation and voice interaction, which are implemented and focused on by TRUST, are also practically non-existent in the previous applications.

| **Key-Feature** | **Google Maps** | **Waze** | **Traffic Spotter** |
|---|---|---|---|
| Report incident | YES | YES | YES |
| Report adverse driving conditions | NO | NO | YES |
| Provide weather Information | NO | NO | YES |
| Autonomous weather monitoring | NO | NO | NO |
| Road conditions data collection | NO | NO | NO |
| Recommended Speed based on road conditions | NO | NO | NO |
| Voice commands interaction | NO | YES | NO |

**Table 2.1:** Available applications comparison

Even though these applications provide some meteorological information, and a few of them have the ability to generate alert events, they do so with insufficient detail and emphasis, and with some latency, due to the fact that weather conditions are constantly changing. These systems are not prepared to provide this type of information to drivers in real time and, therefore, are not efficient in preventing accidents related to adverse weather conditions. This is due to the fact that all of these systems operate in a centralized manner and with long-range communications, without great efficiency in terms of the velocity in which information is presented. The TRUST project proposal, on the contrary, supports an onboard unit connected directly to the application.

The TRUST project aims to tackle this problem, by combining a meticulous and efficient collection of weather information, obtained from diverse sources, with a quick dissemination of alerts by vehicle communications, which guarantees low latency and real-time information.

## 2.2 Communications Technologies

The next step towards working for a truly safe and efficient road system is to guarantee the cooperation between all road users. As such, ITS applications are developed to connect vehicles and infrastructure through vehicular communications. The greater the number of vehicles cooperating with each other, the better the entire road system will operate provided that the choice of the type of communication is the most adequate for each situation.

### 2.2.1 Communications between the smartphone and the OBU

As previously mentioned, the mobile application communicates directly with the OBU to report and receive event notifications about adverse driving conditions. This communications can happen via Bluetooth low energy, WiFi or USB.

*Bluetooth*

Bluetooth is a communication technology that allows the connection and exchange of information via radio waves, where the main objective is to connect devices quickly, wirelessly and with low energy consumption.

As the transmission is done through radio waves and one of the characteristics is low energy consumption, the devices surrounding the connection must be relatively close to each other due to its low range. Since both the mobile application and OBU are in the same vehicle the situation can be considered ideal.

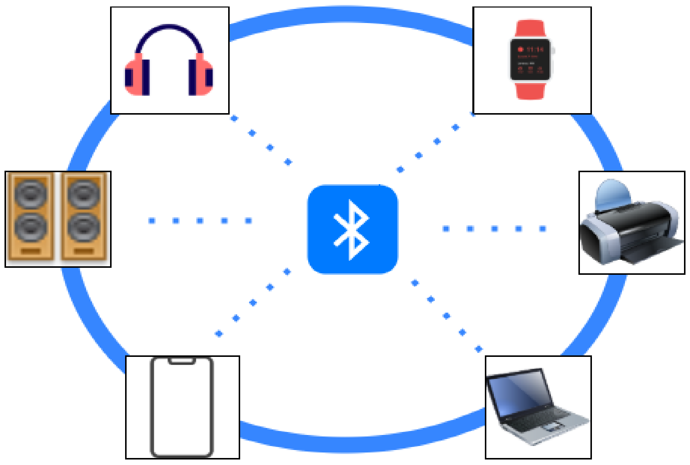


**Figure 2.10:** Bluetooth multi-device environment

Although the Bluetooth technology has undergone changes as improvements in several aspects, it maintains its form of connection and mode of operation. Bluetooth sends and receives radio waves in the frequency band centered at 2.45 GHz, being able to alternate in 79 different frequencies.[12] One of the great advantages of Bluetooth being a short-range communication is the fact that, since it has low power requirements, it cannot send signals far, which means that they will not interfere with connections on the same frequency that someone may already be using. In addition, it becomes a very secure connection, as the connection is very small and restricted and cannot be accessed from distant access points, such as WI-FI.[13][14]

As we can see, Bluetooth is a communication technology recommended for exchanging information from devices that are very close in a secure and fast way and without requiring much energy from the devices, which is ideal in this case. In addition to Bluetooth, there are other possible communication technologies for the exchange of information over relatively short distances, whether by wireless or by cables, addressed below.

*Wi-Fi*

Wi-Fi is a wireless communication technology that allows the connection between closeby devices, through radio frequencies, with the main purpose of exchanging information. It is pointed out as one of the great technological developments, because this wireless network standard allows a quick and easy communication between various devices that are within the range of the network. In this situation, the communication technology presents itself as an alternative to Bluetooth by establishing a short-range wireless connection between the OBU and the application.[15]

In the case of communication between these two devices, the OBU is the one who supports and provides conditions for the connection to be established and for it to be possible to exchange the information desired. The connection is possible because the onboard unit has the necessary tools to generate a Wi-Fi access point, also known as a Hotspot, so that the smartphone can establish a direct connection and, from that point, send and receive messages in both directions. There are no major security concerns as the connection is composed only of the two devices that are inside the vehicle, where the OBU is responsible for the Wi-Fi access point, which is the one that can control access to the content of the connected devices. So, once connected, both devices can send messages with the known predefined structures, the CAMs and DENMs, quickly, without interference and with low latency.

It is remarkable the importance that Wi-Fi has in the vehicular environment of this project, because despite being used in two different ways, it can serve as a basis for external to vehicles communications, that is, between cars and infrastructures, as well as inside the vehicle, between OBU and the smartphone. Therefore, Wi-Fi is another technology used in the communication of these two devices and that may be an alternative to other wireless technologies.[16]

*USB*

Once the wireless communication technologies have been analyzed, we can approach communication via USB cable. Since, in this specific case, the communication between the two devices takes place a few centimeters away, the use of cables is justifiable. In fact, the use of cables in a vehicle is not unfamiliar, as it is common to charge phones in the car through the car's cigarette lighter, or even connect to the radio to play our own music.[17]

USB communications have a simple but very effective architecture, capable of providing a connection to many devices simultaneously and allowing communications to be carried out asynchronously. As we can see in the figure below, there are three main entities in the USB architecture:

- The **Host**, usually the computer, is the main element and controller of the system, as it is the only device that can communicate with all the others.
- The **Hub** is a device with the ability to extend the number of available ports and consequently expand the number of connections between devices in the system. It acts as an intermediary between the devices and the Host with the simple functionality of copying the message it receives from the host and replicating it to other I/O devices. One of the characteristics of this device is the ability to send messages from the host to all devices in the system, whereas messages coming from devices are only forwarded to the Host.
- Each **I/O Device** is a peripheral that is connected to the Host via USB, it can be keyboards, printers, speakers, Monitors, among others. As said before, each device can send messages to the Host but cannot communicate with other peripherals within the system.

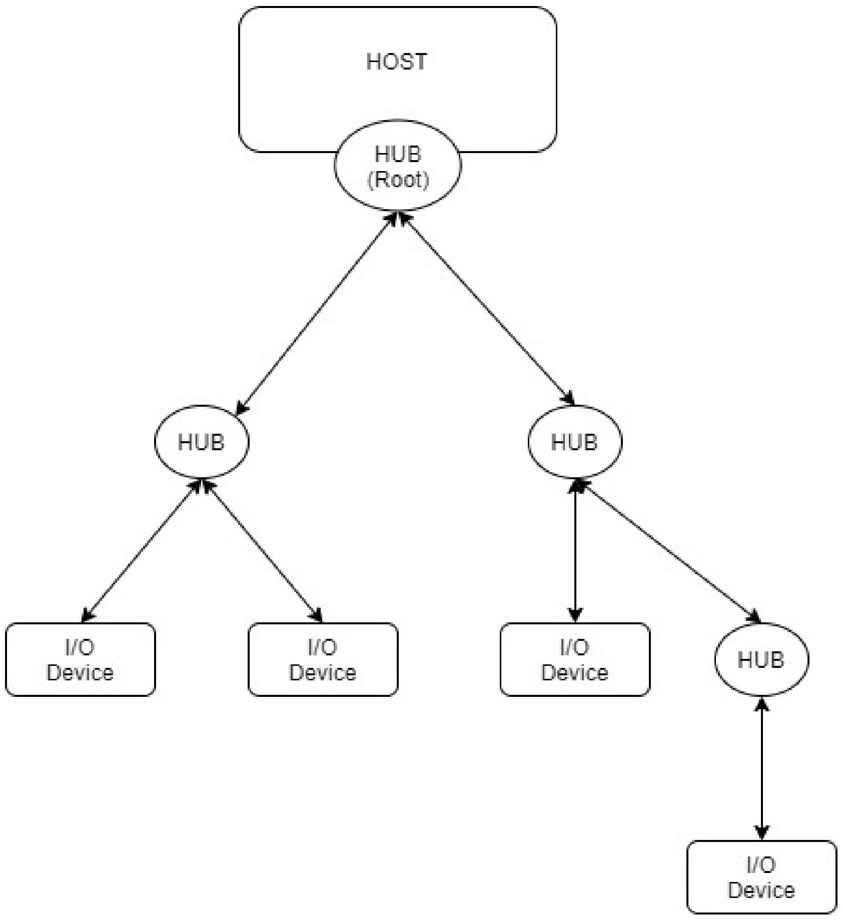


**Figure 2.11:** USB architecture

In the communication process between the host and the device in question, the host needs to know both where the data is to be sent, through the device's address, and the type of transfer to be used. There are four types of transfers:[18]

- **Control** transfers are those used for parameter setting and status control of the USB device. For example, in the process of initializing a USB communication, the Host sends a request to the I / O device, which returns the requirements of each of the endpoints to exchange information.
- **Bulk** transfers are associated with transfers of large volumes of data, such as printers or digital scanners. It is a process that can present some latency, as it requires a large bandwidth and can only be used when that band is available, losing transfer priority. In addition, data integrity is constantly checked through cyclic error checking redundancy.
- **Interrupt** transfers are related to infrequent transfers and with a low volume of data that, unlike the previous one, requires priority over others. It is mainly used in mice or keyboards. When connecting to the Host, the device determines how often the Host should request data from the device endpoints.
- **Isochronous** are referred to continuous data transfers. It happens when data needs to be transferred at all times and as quickly as possible. Due to the existence of this requirement, this type does not present any data verification and data loss is preferable

to forwarding packets that would delay the communication delivery time. Loudspeakers are an example of this type, as data transfers take place in real time.

Although new technologies, such as wireless communications, are attractive, USB continues to evolve and present major improvements that make it unique and very useful. The main reason for its creation was due to the simplicity of its connection, where no configuration is necessary to connect any two devices. By simply connecting the cable between the OBU and the smartphone, everything comes together, this is known as Plug and Play. In addition, the data transmission capacity is high and the power requirement is low, which is a plus for the communication between these two devices.

# CHAPTER 3

# System Architecture

In this chapter and in the sections that follow, the entire architecture of the proposed solution is presented, as well as the components that comprise it. It is a fundamental chapter to understand how the system works as a whole and how information flows into it.

## 3.1 TRUST

This dissertation addresses the work developed and focuses on one of the proposed solution of the TRUST project, so it is important to approach the system and its architecture.

As discussed in the previous chapter, the TRUST project aims to eliminate gaps in traditional road systems by focusing on constantly monitoring weather conditions and providing warning services of adverse situations for drivers, in order to make the roads safer and more efficient. The project has infrastructures equipped with ITS technologies, which enables communication between vehicles and infrastructures, to successfully and quickly alert all users of the road system about dangerous incidents in specific locations. The possibility of communication between vehicles and infrastructures points to a cooperative and intelligent road system, capable of bringing together joint efforts from different entities and achieving a system with more secure and with less road accidents.

For the main purpose to be achieved, the project follows a list of objectives[2]:

- Collection of meteorological data through sensors available in vehicles and road infrastructures.
- Use of vehicular communication protocols for fast forwarding of the collected data to intelligent services.
- Remote analysis of monitored data and identification of dangerous weather events that put drivers at risk.
- Sharing and disseminating events through vehicle communications, in real time.
- Development of a notification interface for drivers to receive hazard alerts and interact with the road system.

The interface tool for reception of events by drivers is one of the great solutions of this project and the main focus of this dissertation, which addresses new implementations and features to achieve the objectives described above. It is through the project's mobile application that the driver is connected to the TRUST system and will receive alerts on road hazards, but for this to be possible, all entities in the system must work in a cooperative and coordinated manner.

### 3.1.1 System Architecture

The TRUST project has several both hardware and software components which are all fundamental to the system due to the role that each one plays. In order to provide multiple services related to road mobility, the system uses sensors and infrastructures already implemented by the Telecommunications Institute (IT) under the PASMO project, which is an open platform that provides physical sensors and communication networks. In addition to the infrastructures and sensors that were developed by PASMO (https://pasmo.pt) and the new ones developed by TRUST, we must emphasize that the uniqueness of the project lies within the development of an on-board unit (OBU), that not only collects vehicle data but also establishes direct communication with infrastructures, other vehicles and the mobile application

The following figure shows the architecture of an intelligent and cooperative road system, into which TRUST fits perfectly. In the image, we can see various actors that are walking, driving or riding bicycles, as well as buildings, institutes and roadside infrastructures, which allow that communication be established not only between vehicles and infrastructures but also between vehicles themselves, therefore forming a connected road system.[2]

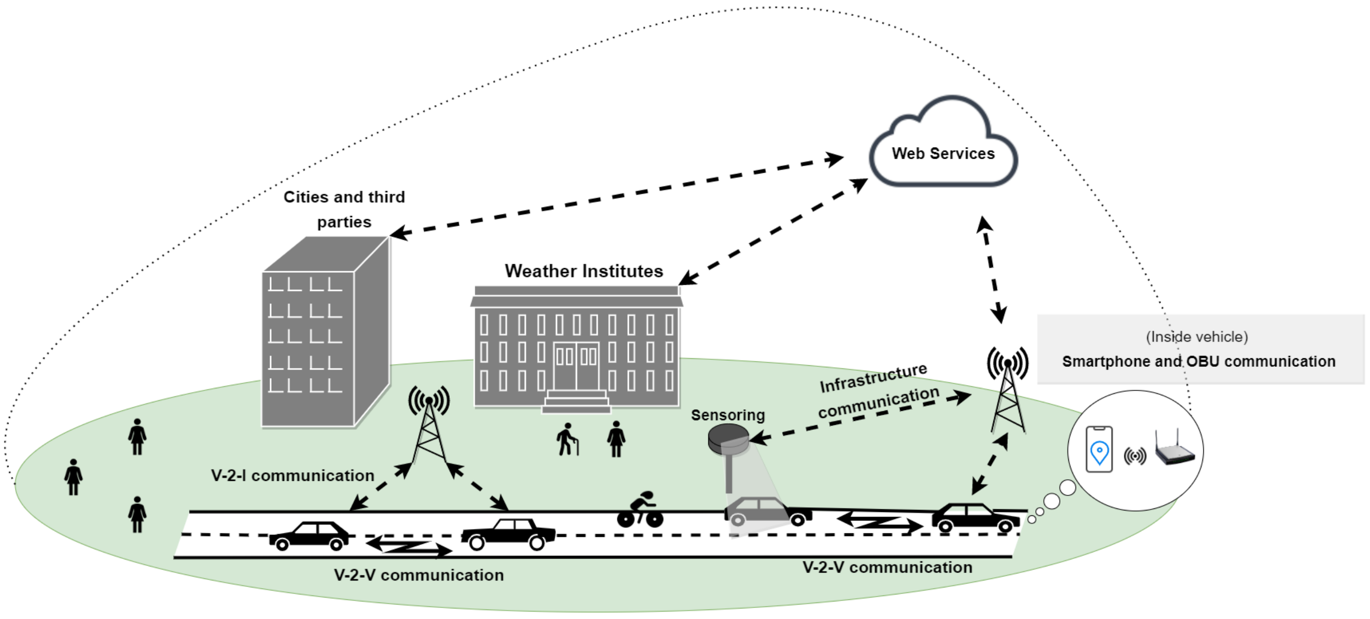


**Figure 3.1:** TRUST Architecture

The project architecture promotes the integration of components and communication scenarios that involve data access to vehicles, and sensors, through external devices such as other vehicles, road equipment, meteorological stations or on board devices, by using wireless communications, such as location-based communications (V2X) or communication networks for cloud connectivity, such as cellular networks. It combines a variety of access technologies and communication protocols composed of different characteristics, such as communication range, delay, bandwidth, among others, and can be divided into **networked communications** and **localized communications**. While networked communications refer to communications supported by infrastructures, such as cellular or satellite networks, localized communications refer to the ability to exchange information directly between vehicles and their surrounding environment, without the intervention of any managing telecommunication network. Both of them are the backbone of the services of a cooperative ITS.

In a vehicular environment, localized communications are called V2X, where there is direct communication between vehicles and vehicles, vehicles and infrastructures or other actors, such as pedestrians. As discussed in the previous chapter, V2X is supported by a dedicated frequency band, belonging to the WiFi variant 802.11p, so that this communication is as secure as possible. The distinctiveness of a C-ITS is due to the variety of ITS stations present in its architecture, of which the following are considered fundamental:

- Vehicle ITS station unit.
- Roadside ITS station unit.

Through V2X vehicle network communications, it is possible to assign critical characteristics of the TRUST project, which include the ability to associate specific locations with meteorological data, such as fog, rain or air pollution. This architecture has 4 main phases, the first being the acquisition of meteorological data and traffic information by using all the contributing sensors; the second being the listing and deployment of all the contents collected for the ITS-S central and for brokers; the third being the handling, storage and analysis of received data, and lastly, the generation of high-level data to interact with end-users.

In this architecture, the mobile application, which is the focus of this dissertation encompasses two fundamental roles: the collection of data and meteorological information as well as the direct interaction with end-users, i.e. drivers.

The system architecture is divided into 3 layers: the sensor layer, the communication layer and the information layer[2].

- **Sensor Layer** - Refers to physical devices that collect, process or share meteorological data that are of interest to the project. For example, sensors that collect additional information about weather conditions around specific areas of the road or stations, that work as units for monitoring weather parameters, such as, determining temperature, relative humidity, precipitation, wind speed.
- **Communications Layer** - Responsible for the interface between physical devices, such as sensors, applications, stations, OBUs and the cloud infrastructure.
- **Information Layer** - The responsibility of this layer is to analyze and store the information collected by the sensor layer so that it is possible to share it with the stakeholders involved.

*Sensors and Infrastructures*

The TRUST project had some crucial sensors and infrastructures developed, in order to achieve the determined objectives. These have the role of collecting interesting data for the project in a clear and consistent way. Based on the data collected, the entire system operates and proceeds with the remaining tasks. For example, flood sensors have been developed to obtain extra information about weather conditions on certain parts of roads that can cause incidents such as floods, aquaplaning, among others. There are also weather stations that monitor weather conditions, such as the temperature, precipitation, wind, visibility, and report if any abnormalities are detected.

*Vehicle ITS station Unit*

As mentioned earlier, a very important sensor is the vehicle itself. It is used for other purposes but it is also able to collect data in a quick, consistent and concrete way.

There are multiple devices inside the vehicle, the most important of which are represented in the image below, the CAN/OBD reader, the OBU and the smartphone. When working cooperatively they are capable of providing different types of data which can be useful for detecting accidents beforehand and which the TRUST project seeks to benefit from. The CAN/OBD reader monitors the vehicle system and reports problems with the vehicle. This device, in combination with the OBU and the application enables the sharing of information regarding the vehicle, such as the condition of the brakes or engine, position of the accelerator, condition of the lights, among others. This information can be very useful in certain situations which occur outside the scope of accidents caused by meteorological causes.

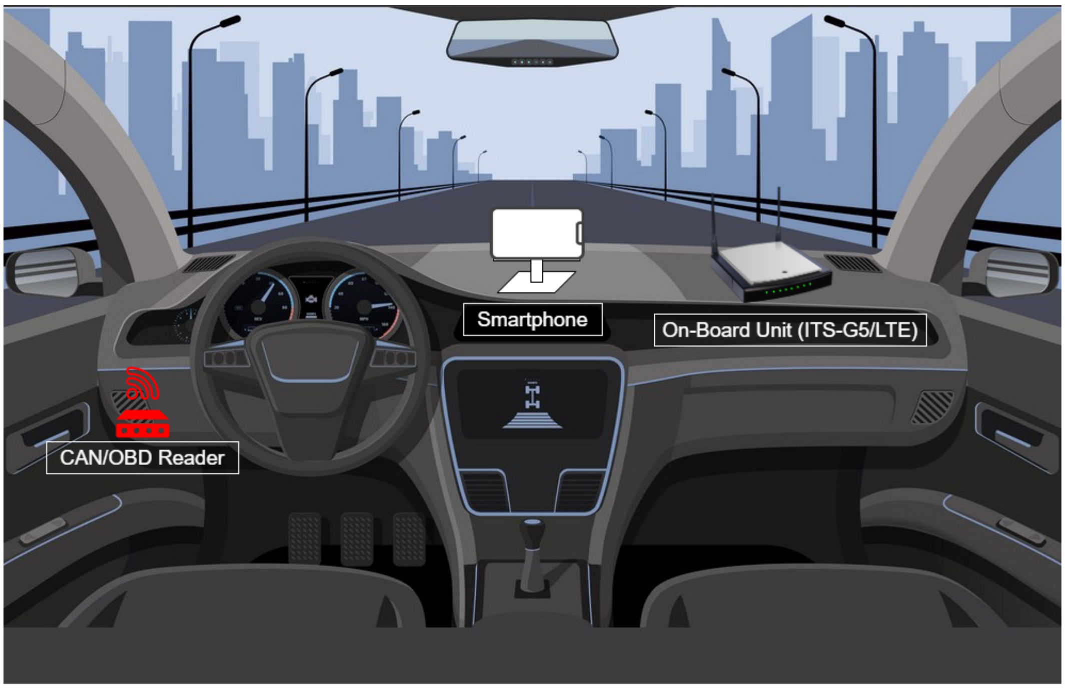


**Figure 3.2:** On Board devices

Furthermore, the OBU, presented as one of TRUST's main solutions, plays a very important role in the entire system as it communicates directly with the application. All the information that the application intends to send to the rest of the system is the OBU's responsibility, and it assumes itself as being an intermediary between the application and the remaining components. [2]

In the following figure, it is possible to identify the different interfaces and physical components of an OBU, which allow communication with the application, with other vehicles and with infrastructures.

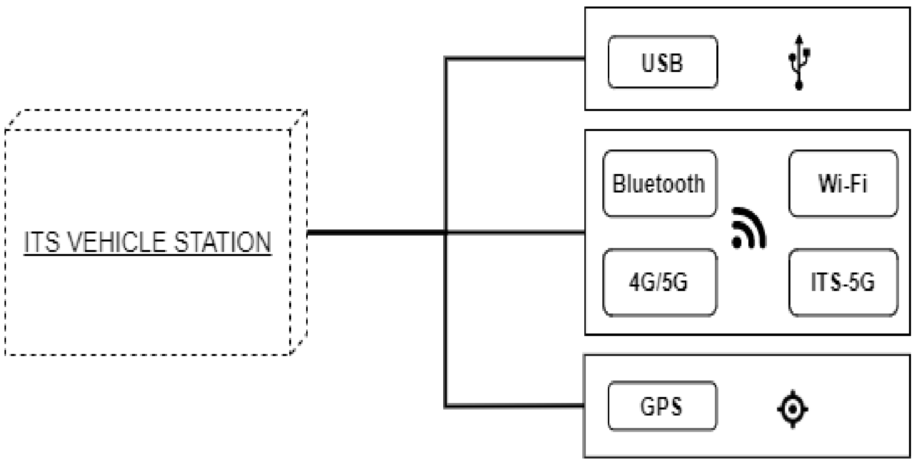


**Figure 3.3:** Vehicle ITS station interfaces

In this sense, when the system generates a weather alert, based on the data collected by the sensors, it shares it with the OBU and forwards it to the corresponding applications. The same happens in reverse, this is: the driver reports an event, such as ice formation, that is sent from the application to the OBU, and forwarded to the rest of the system. As previously mentioned, the communication between the application and the on-board unit takes place over WIFI, Bluetooth or USB. As mentioned in the previous chapter, the communication established between the OBU and the rest of the system, i.e, other OBUs or RSUs, is done by using the ITS-5G technology.

## 3.2 Application Overview

This dissertation focuses on one of the TRUST project solutions: the development of an interface, between the driver and the system, that can receive notifications generated by the system about adverse conditions and road hazards that may endanger people's lives.

This application, not only assists a driver by warning him/her about possible accidents that may occur in the surroundings areas, thus enabling the driver to prepare in advance for dangerous events, but it also gathers data from road and weather conditions, as well as from the vehicle itself.

The collaboration between IT and BRISA resulted in the development of the overview application, which has several functionalities, such as, receiving meteorological events generated by the system, providing a real-time location of the driver's vehicle and direct communication with the OBU. It also provides a clear interface of the alerts generated by the system, favoring driver safety. However, and as referred beforehand, the vehicle and the smartphone have several tools that are also capable of receiving information, acting as sensors, collecting interesting data for the system to process, and, in certain cases, identifying dangerous situations autonomously.

### 3.2.1 Application Architecture

The main purpose of this application is to serve as an interface for the driver, who receives events generated by the system and is notified of the location in question, so that he/she is aware of the situation and, consequently, the risk of an accident is greatly reduced. So, before the development of this dissertation, the application already included some functionalities that focused on the main goal of this solution, such as:

- Establish a connection between the mobile application and the OBU via Bluetooth, Wi-Fi or USB.
- Display the smartphone's current location on the map.
- Receive and display events sent by the OBU on the map.
- Report specific events in a specific location.

However, this application aims to play a key role in the entire project and be an important tool for monitoring weather conditions as well as providing the driver with safety services. In this sense and within the scope of this dissertation, new features were developed:

- Video background recordings that collect weather information and road conditions.
- Collection of luminosity values through smartphone sensors.
- Recommended speed limit algorithm based on alerts received and vehicle category.
- Video stream service when passing through an accident zone.
- Voice validation of alerts received.
- Road condition monitoring and event identification service through image processing.

These functionalities represent only what is visible to the user. But in reality there is a background system that makes it possible to achieve these goals, hence it is important that its architecture be addressed. Therefore, the following image depicts the architecture and components that facilitate the flow of information through the mobile application.

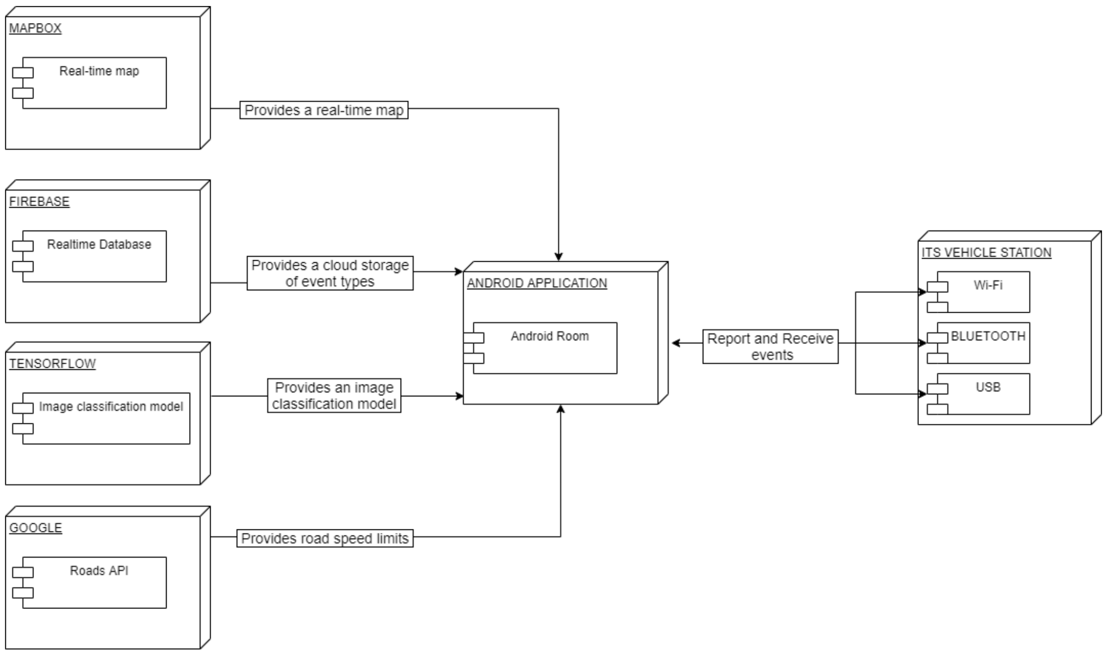


**Figure 3.4:** Application Architecture

*Communication Technologies*

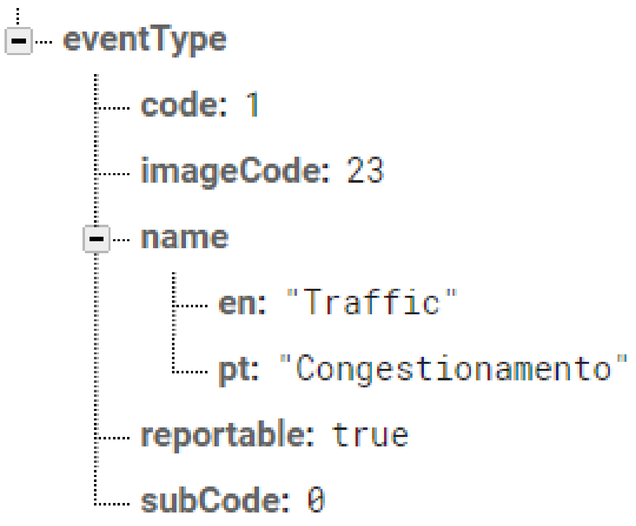
As mentioned before, the application can only communicate with the OBU through three types of communication technologies: WiFi, Bluetooth and USB. However, the only technology that is currently in a more advanced development phase is WiFi, whose communication between the application and the OBU is made via the **MQTT** (Message Queue Telemetry Transport) protocol. This communication protocol is a very simple, lightweight and flexible network protocol, widely used in IoT communications and consists of machine-to-machine (M2M) communication connectivity, based on a publish/subscribe message model. This publish/subscribe model defines two entities on the network: the broker and its customers, who can take the form of subscribers or publishers. The **Broker** is an information hub that stores messages received from clients with the purpose to, later, forward that messages to others. Each message is referring to or falls within a certain subject, so all information is organized under topics. So, a **topic** is a subject identifier, where all messages related to that subject will be sent.

As clients, we have **Publishers** who connect to the broker, regarding a specific message topic, and publish/send messages related to that topic. [19] Thus, all messages that a client sends to a topic are received by all the others that have indicated that they want to receive it, enabling one-to-one, one-to-many or many-to-many communications. In this case, the application is a client and the OBU works as the broker. The messages sent and received, regarding the incidents generated by the TRUST project happens, through the types of messages addressed and belonging to the intelligent transport system, DENMs and CAMs. However, the application shares other information, like sensor values, through other message types.

Since the application only interacts with the OBU of the vehicle in question, it serves as an intermediary and communicates with other vehicles or infrastructures. So, for example, a driver and user of the application who detects a fire in a forest along the road, reports it in his/her application. The application sends this information, via WIFI (MQTT), to the OBU of the same vehicle. The OBU establishes a connection with other vehicles and infrastructures and sends a DENM with information about the fire, also via WIFI (ITS-G5). A driver, who is using this application and passing through the area will receive a DENM, sent by the OBU of the vehicle in question, notifying him/her of the location of the fire on the smartphone map.

*Firebase*

The most important feature of the application is to receive alerts of events that can affect driving safety. When it comes to weather conditions, the application receives DENM messages, in code format, with a specific structure, namely the type and subtype of the event. To access information about the type of event the application uses an external source, Firebase, to query event data. [20]



```
eventType
    code: 1
    imageCode: 23
    name
        en: "Traffic"
        pt: "Congestionamento"
    reportable: true
    subCode: 0
```

**Figure 3.5:** Firebase Event Information Structure

Firebase is a platform developed by Google, which helps in the support and development of mobile and Web applications. The platform provides various products, authentication and security methods, cloud repositories, service quality systems and monitoring, among others. The *'Realtime Database'* product provided by the platform is the main reason why it was chosen and used.

The Firebase *Real time Database* is a non-relational database hosted in a cloud. The platform provides a simple way to store and synchronize data between users in real time, which makes it easy for users to access information and collaborate with each other. The system is designed so that a change in the database is updated in the cloud and automatically sent to all interested users.

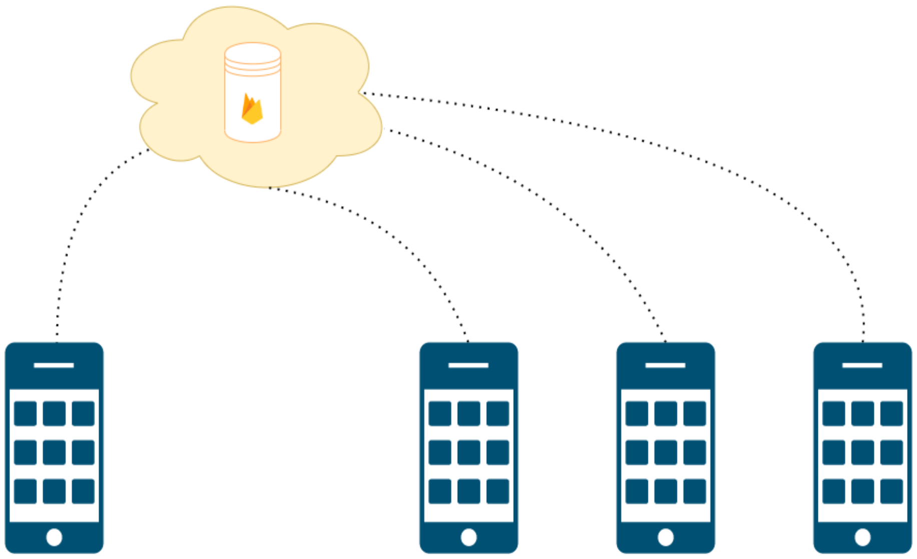


**Figure 3.6:** Firebase Realtime Database

*Android Room*

In addition to receiving the data and presenting it on the map, it is necessary to store the events in an offline database for internal treatment in the application. In this context, the persistent data library Room is used. It offers a system for caching data that has already been processed, which allows the application to access data, in this case events, even if the communication with the OBU is no longer active.

The Room presents itself as an abstract SQLite layer that allows fluent access to a database. It's structure has three components: [21]

1. **Room Database** - It contains the database and serves as the main access point for connecting to the application data.
2. **Data Access Objects** - Also known as DAO, are the objects that contain the methods to access the database.
3. **Entities** - Refers to database tables.

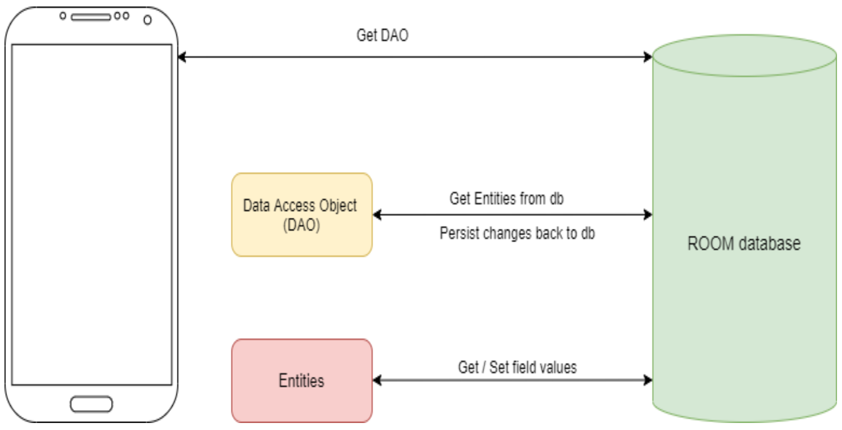


**Figure 3.7:** Android Room Architecture

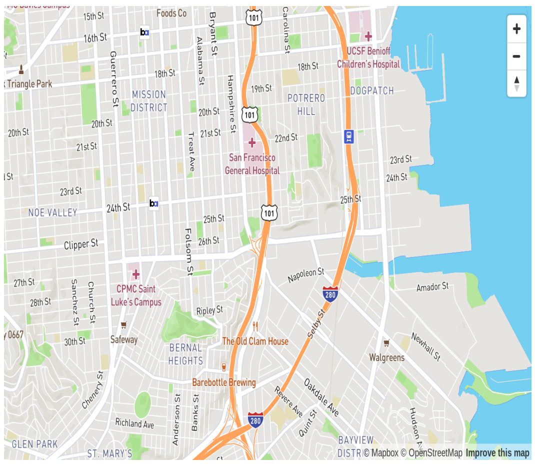
The Room database is used to persistently store the events received until their validity ends, which means that they do not need to be processed several times, and can be accessed without internet connection.

*MapBox*

The open source map application platform **MapBox** provides a map display, which shows both the user's location as well as all the impacting events. This platform is reliable and used by several companies in the geographic market as it offers several features accompanied by good documentation, making it simple to integrate into the application.

mapbox

**(a)** Logo


MISSION DISTRICT
POTRERO HILL
DOGPATCH
NOE VALLEY
BERNAL HEIGHTS
GLEN PARK
BAYVIEW
San Francisco General Hospital
UCSF Benioff Children's Hospital
CPMC Saint Luke's Campus
© Mapbox © OpenStreetMap Improve this map

**(b)** Map

**Figure 3.8:** Mapbox

Within this platform, product solutions such as real-time maps, navigation assistant, among others, are provided. It is a very complete platform that serves the intended objective perfectly, which, for the time being is to provide high quality maps in real time.

### 3.2.2 Activities and Components

The application provides the user with several features, presenting a simple and easy-to-use interface as it aims to ensure that the driver uses the application as little as possible and as quickly as possible. As such, the application provides a simple and intuitive interface, capable of fulfilling its functions and requiring minimum effort by the user. For this reason there are only two activities with an interface, and most of the application's services work in the background. It is important to remember that, although on-board assistance while driving is an asset for the driver, he/she must remain fully focused on the road and on those around him/her.

In its main space the application displays a map with the driver's real-time location and that is where the reported or received events will appear.

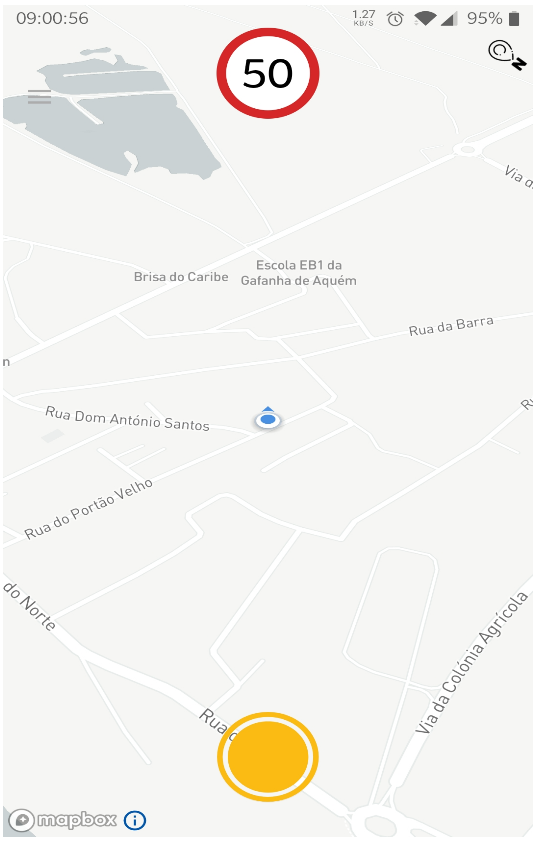


**Figure 3.9:** Application Map Activity

The application's functionality and interface are indispensable, because, in addition to receiving events, the driver can report incidents. In this case, after the driver detects an incident that may have a negative impact on others, he/she accesses this fragment of the application, visible in the following image, where he/she can choose the type of event that best describes the situation and share it with other drivers.

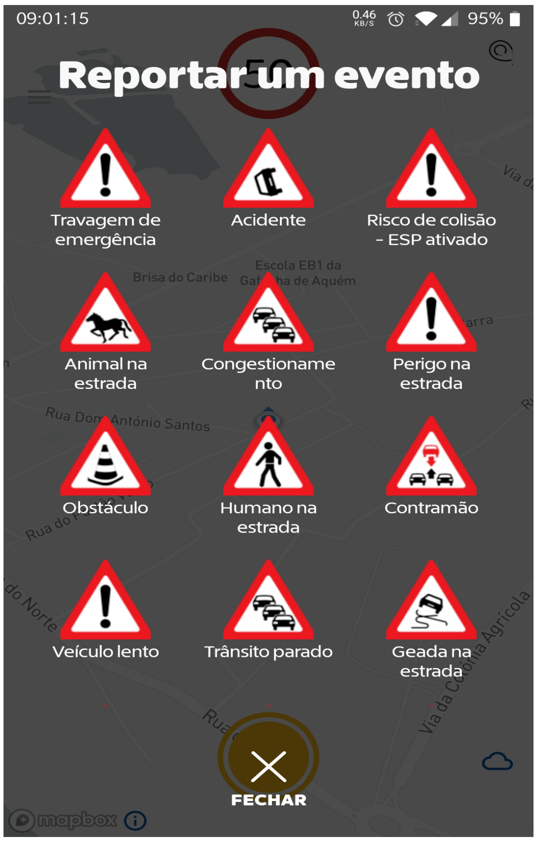


**Figure 3.10:** Application Report Activity

The possibilities of events are diverse and the symbols of each one allow the driver to send a precise alert, quickly, without interfering with his/her concentration on the road. After being reported, the event is received in the applications of other drivers, and appears on the map, as shown in the map activity image.

As it is a driver assistant mobile application, its interface must be as simple as possible. In the background, the application works as a sensor and collects various types of information, in addition to services that detect adverse weather conditions.

## 3.3 System and Application Developments

Considering that we now have detailed knowledge of the TRUST system and its supporting mobile application, we will focus directly on the developments made. Therefore, this section presents what was developed during this dissertation, regarding the functionalities integrated in the application and the services external to the application but within the scope of the TRUST project.

### 3.3.1 Integration of additional weather alerts

The TRUST project focuses mainly on monitoring weather conditions with the intention of providing that information to users. The application, which represents one of the solutions of this previously initiated project, shares and receives harmful road and weather information by means of alerts and notifications. However, road accident type catalogues were quite limited and superficial, mainly those related to meteorological causes, focusing on accidents, road works, among others.

In order to surpass this problem a proposition to analyze and improve the information given to the driver, in an accurate and coherent way, about what he/she could expect on the road, was made. Thus, after detailed discussion, new weather and meteorological events were introduced that seemed mandatory and essential to complete the application. This process and the way events were thought of is described in the next chapter.

### 3.3.2 Extra data collection

The application serves mainly as a simple alert interface for drivers. It establishes fast and direct communication between the various drivers and infrastructure of the road system. But in fact, since there are many drivers and seeing that they are located in different geographical areas, it is appropriate to benefit from this situation, and collect data about different road sections. Just as vehicles collect information automatically, the same can happen with the application that the driver is using, all of which without driver intervention.

*Luminosity Data*

One of the major causes of road accidents is the lack of visibility often due to the insufficient light on the road. Therefore, since each smartphone contains a light sensor, it can be used to monitor the amount of light available and be a reference for geolocation.

Thus, one of the goals was to constantly collect information of natural light, in most diverse areas, where users of the application pass and automatically capture luminosity values. This data is sent to external entities which with other data, referring to other factors and sources, are able to judge the driving conditions and share it in advance with other drivers.

*Video Recording Data*

Nowadays, one of the greatest advantages of a smartphone, is its ability to record videos quickly and with quality, due to the strong investments that many brands have made on their cameras.

The application intends to benefit from this capability by regularly collecting data on the driving conditions of each geographical area, in a conscious, automatic and balanced way. This can be a big breakthrough in determining the existence of adverse conditions, even if they have not been reported by any user/driver.

A practical example of this feature may be the existence of fog, in which several drivers may decide not to alert the application about this weather condition. Still, the smartphone automatically collects this data and sends it to the system, so that it can be processed, all without user intervention. Thus, other drivers who pass by the same location, later on in the day, can be warned in advance about the conditions they can expect.

Knowing that a driver's main focus is to drive with maximum safety, it is essential that automatic processes able to collect this type of data and draw conclusions, which are an asset for the road ecosystem, exist.

### 3.3.3 Speed Limit Advisor

Despite all the interferences that may occur with the system, such as adverse road and weather conditions or poor visibility, drivers are primarily responsible for their vehicles and everything that happens to them. It is well known that speeding is one of the main causes of accidents in everyday life. Speeding happens regardless of the conditions and the area in which we are driving. However, there are speed limits whose main purpose is to keep drivers safe, maintaining them at a safe speed limit, henceforth providing them with enough time to react to any events that may occur.

Thus, the application incorporated a feature which informs the driver of the recommended speed limit for that area. It is logical that when drivers are confronted with road incidents and adverse road conditions, they should slow down and approach them with extreme caution. Thus, the feature provides the driver with a recommended real-time speed limit for the type of road in question, taking into account all the events that surround this event as well as those that can influence it.

### 3.3.4 Stream Service

The TRUST system collects and analyzes large amounts of data, not only through the application, but also through the use of sensors and infrastructures. It is safe to say that there are quite a few parties involved in the system which make a large amount of information available to both drivers and the system itself. But, although it's possible to associate incidents with geolocations, the information that is gathered is basic and superficial, lacking much detail. Often additional information is not necessary, as previous knowledge of the affected area is sufficient, an example being heavy rainfall. However, there are cases where several factors are associated and the more information we have about a specific incident, the better, being this the case of road accidents in which cars and people are directly involved.

Thus, one of the proposed features, which intends to add information to the event, is the video stream. The mobile application captures the event through the driver's smartphone camera as the vehicle passes the road accident. It is possible for traffic control to access these images in real time, obtain more information and draw conclusions about the specific event. This can be important to assess the current state of the situation, such as how many cars are involved, whether it is necessary to call an ambulance, among others.

### 3.3.5 Event validation through speech recognition

Prior to the development of this dissertation, the application already received events sent by the OBU, making them available on the map, for the driver to observe. These events are generated by the TRUST system based on the data collected and the information received by the drivers. Even though each event has a specific expiration date which is not entirely predictable, it may or may not be shortened.

So, in order to retrieve information about events already generated, the application asks the driver, when he/she passes a certain event, if it is still active or if it has already expired.

The process of questioning the driver and receiving the answer is hands-free thus completely safe.

Hence, without distracting the driver, events and information are kept as accurate and consistent as possible, preventing, for example, drivers from receiving alerts for events that actually no longer exist.

### 3.3.6 Recognition of road conditions through image processing

There are numerous concerning and important situations to focus on when it comes to road safety. In this context the TRUST project focuses on weather conditions that may affect drivers. For the mobile application to have as much information as possible, in addition to collecting and analyzing data, it is necessary to do it quickly and thus, the more automated the data analysis process is, the better.

Therefore, since the application has the ability to collect real-time images about his/her surroundings, the system intends to take advantage of this implemented functionality using the collected images to process data and generate weather alerts autonomously, whether incidents have been reported, or not.

To make this possible, image processing models were implemented for two recurring danger situations on the roads: poor luminosity and rainfall.

The luminosity and rainfall model were developed to be used by two distinct parts of the system, both within the realm of the application or not. The models were designed to be used by other entities with more processing capacity and more information. However, nowadays, smartphones are also capable of using the developed models, even if not so accurately, due to an increase of their processing capacity in the last few years.

The application is one of the devices that collects videos of the road. As soon as videos are captured, it is important that instant feedback be given to the system. Thus, in addition to collecting videos, the application also classifies the video, telling the system immediately whether or not it is raining and whether or not there is sufficient light by using the models described below.

*Luminosity model*

An image processing model which determines the degree of visibility of the driver, due to the luminosity on part of the road, was developed. By combining the application to collect images of good quality with a trained image classification model, capable of quickly identifying hazardous road and weather conditions, it is possible to associate a specific event to a specific location, fundamental for preventing accidents.

With the lack of visibility, our reaction time drastically decreases and although it is not possible to improve the luminosity on parts of the road, other than through the headlights, it is best to be prepared for this scenario beforehand. The application intends to be able to generate and alert drivers about incidents of this type and to do so as quickly as possible.

Therefore, in this case, the objective of the system is to process the images collected by the application quickly, using a previously constructed classification model. The model is capable of making valuable predictions and clearly stating whether the road section where the videos were collected show visibility problems.

*Rainfall model*

In addition to the luminosity of the road, other road conditions are extremely important too. There are several external factors that can influence driving and the more intense the monitoring, the better. One of the weather factors that most limits the driver and consequently causes accidents, is rainfall. Driving while it is raining is harmful because it reduces the adhesion of the vehicle to the road and, if very intense, can cause visibility problems.

Thus, and since rain is recurrent in seasons such as winter, another image classification model was developed, with the aim of detecting the existence of rain in certain parts of the road and disseminating this information amongst drivers. It can be crucial to know this beforehand because the driver may be close to an area of intense rain and therefore, being unaware of the danger ahead, drive into it at a much higher than recommended speed. Therefore, the rainfall model presents itself as a tool of the TRUST project capable of automatically detecting the type of situations described above.

CHAPTER 4

# Implementation

The goals of the implementations developed in the mobile application are to complement the monitoring services of the TRUST project and to increase driver safety.

In addition to the inclusion of new types of potential hazardous situations, and since the project focuses on monitoring and controlling weather conditions, the application acts as another sensor and source of information, through the collection of videos by the smartphone camera and brightness sensor data. The collected videos are automatically analyzed by intelligent image processing models also by the application, in order to quickly identify two dangerous situations for drivers, lack of visibility due to the low light of the road and the existence of rain, which are often the cause of road accidents.

For the safety of all drivers and an improvement of the information flow within the system, a video stream service was developed in case of an accident, in which the traffic control management entity can access and analyze the situation in real time and take the necessary measures. In addition, the user can contribute to the assertiveness of the system information as it can confirm the validity and veracity of events received through a voice interaction service. Finally, the driver has available a safe speed recommendation in real time based on various parameters and situations, which can greatly reduce the number of accidents.

Thus, this chapter describes and demonstrates how each component was integrated, in a more technical and detailed way.

## 4.1 Integrated development environments

First of all, it should be noted that two integrated development environments were used to develop this project: Android Studio and Pycharm.

The first one is widely used all over the world and considered as the main platform for developing Android applications, as it presents a simple and pleasant display that has all kinds of essential integrations, such as being able to see the layout of activities, having an emulator to run the program and possible integrations like git, which was also used as a repository for this project. So, since the main developments were on the mobile application, Android Studio was the major platform used and with the most focus through the entire journey.

One other focus of the thesis was the development of trained models capable of processing images and every step from split videos into frames, split the data into training and testing groups, to training the model and testing the inputs, was done using the integrated development environment Pycharm.

## 4.2 DENM Integration process

In an earlier phase, the application presented a list of events that it was able to receive and present to users. These events can have many causes or sub-causes and be of different types.

There are many kinds of events and the ways they affect the safety of vehicles nearby vary. For example, events related to traffic and congestion, accidents that may have occurred in the vicinity, road works, problems related to road conditions, humans or animals on the roads, among others. Each event is part of a general topic, distinguished by its specific sub-code. The application has more than 150 events of the most diverse types. Some of which can be viewed and analyzed in general in the following table:

| WEATHER EVENTS APPLICATION | | | |
|---|---|---|---|
| **Cause description** | **Cause code** | **Sub-cause code** | **Road sign** |
| Traffic | 1: Traffic Condition | 0: unavailable | |
| Multi-Vehicle Accident | 2: Accident | 1: VehicleAccident | |
| Road Marking Works | 3: Roadwork | 2: roadMarkingWork | |
| Ice on the Road | 6: Adhesion | 5: iceOnRoad | |
| Earthquake damage | 9 : SurfaceCondition | 2: earthquakeDamage | |
| Obstacle on the road | 10: ObstacleOnTheRoad | 0: unavailable | |
| Animal on the road | 11: AnimalOnTheRoad | 4: largeAnimals | |
| Cyclist on roadway | 12: HumanPresenceOnRoad | 2: cyclistOnRoadway | |
| Wrong way driving | 14: WrongWayDriving | 0: unavailable | |
| Strong Winds | 17: ExtremeWeatherCondition | 1: strongWinds | |
| Fog | 18: Visibility | 1: fog | |
| Heavy rain | 19: Precipitation | 1: heavyRain | |
| Queue after the bend | 27: DangerousEndOfQueue | 3: Queue after bend | |
| Collision Warning | 99: DangerousSituation | 7: collisionkWarning | |

**Table 4.1:** Weather events application

One of the most important type of event that the TRUST project focuses on are meteorological and environmental-related events that may influence driving conditions and compromise the safety of vehicles in the affected area.

Meteorological events may range from heavy rainfall, icy roads, strong winds, fog, among others. Since this is a very important aspect, it is essential that we focus solely on this type of event, analyzing weather events that already exist in the application, as well as those that do not. So, let's describe the process of gathering information, analyzing it, and choosing new events, step by step.

So, first we must consider information about all the weather related events that are already present in the mobile app.

| WEATHER EVENTS APPLICATION | | | |
|---|---|---|---|
| **Cause description** | **Cause code** | **Sub-cause code** | **Road sign** |
| Slippery Road | 6: Adhesion | 0: unavailable | ⚠ |
| Heavy Frost on the Road | 6: Adhesion | 1: heavyFrostOnRoad | ❄ |
| Fuel on the Road | 6: Adhesion | 2: fuelOnRoad | ⚠ |
| Mud on the Road | 6: Adhesion | 3: mudOnRoad | ⚠ |
| Snow on the Road | 6: Adhesion | 4: snowOnRoad | ❄ |
| Ice on the Road | 6: Adhesion | 5: iceOnRoad | ❄ |
| Black Ice on the Road | 6: Adhesion | 6: blackIceOnRoad | ❄ |
| Oil on the Road | 6: Adhesion | 7: oilOnRoad | ⚠ |
| Loose Chippings | 6: Adhesion | 8: looseChippings | ❗ |
| Instant Black Ice | 6: Adhesion | 9: instantBlackIce | ❄ |
| Road Salted | 6: Adhesion | 10: roadsSalted | ❄ |

**Table 4.2:** Weather events application

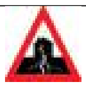
| WEATHER EVENTS APPLICATION | | | |
|---|---|---|---|
| **Cause description** | **Cause code** | **Sub-cause code** | **Road sign** |
| Earthquake damage | 9: SurfaceCondition | 2: earthquakeDamage |  |
| Snow drifts | 9: SurfaceCondition | 5: snowDrifts | |
| Storm Damage | 9: SurfaceCondition | 6: stormDamage | |
| Volcano Eruption | 9: SurfaceCondition | 8: volcanoEruption | |
| Falling ice | 9: SurfaceCondition | 9: fallingIce | |

**Table 4.3:** Weather events application

| WEATHER EVENTS APPLICATION | | | |
|---|---|---|---|
| **Cause description** | **Cause code** | **Sub-cause code** | **Road sign** |
| Extreme Weather | 17: ExtremeWeatherCondition | 0: unavailable | ⚠ |
| Strong Winds | 17: ExtremeWeatherCondition | 1: strongWinds | 🌬 |
| Damaging Hail | 17: ExtremeWeatherCondition | 2: damagingHail | 🌧 |
| Hurricane | 17: ExtremeWeatherCondition | 3: hurricane | 🌬 |
| Thuderstorm | 17: ExtremeWeatherCondition | 4: thuderstorm | ⚠ |
| Tornado | 17: ExtremeWeatherCondition | 5: tornado | 🌬 |
| Blizzard | 17: ExtremeWeatherCondition | 6: blizzard | 🌧 |
| Low Visibility | 18: Visibility | 0: unavailable | 🌫 |
| Fog | 18: Visibility | 1: Fog | 🌫 |
| Smoke | 18: Visibility | 2: smoke | 🌫 |
| Heavy Snowfall | 18: Visibility | 3: heavySnowfall | 🌫 |
| Heavy Rain | 18: Visibility | 4: heavyRain | 🌧 |
| Heavy Hail | 18: Visibility | 5: heavyHail | ❄ |
| Low Sun Glare | 18: Visibility | 6: lowSunGlare | 🌫 |
| Sandstorm | 18: Visibility | 7: sandStorm | 🌫 |
| Swarm of Insects | 18: Visibility | 8: swarmofInsects | 🌫 |

**Table 4.4:** Weather events application

| WEATHER EVENTS APPLICATION | | | |
|---|---|---|---|
| **Cause description** | **Cause code** | **Sub-cause code** | **Road sign** |
| Precipitation | 19: WeatherCondition-Precipitation | 0: unavailable | |
| Heavy Rain | 19: WeatherCondition-Precipitation | 1: heavyrain | |
| Heavy Snowfall | 19: WeatherCondition-Precipitation | 2: heavySnowfall | |
| Soft Hail | 19: WeatherCondition-Precipitation | 3: softHail | |

**Table 4.5:** Weather events application

Now that we know all the kinds of meteorological events that the application encompasses, we have to analyze the data, see what is missing and if anything should be added. This is an important step that deserves special attention. On the one hand it is important to allow the user to be able to receive or send precise/specific events. On the other hand we must avoid the abundance of information and choice of events as the process of reporting events must be kept fast and simple.

This step in the selection process was carried out in conjunction with the Ubiwhere, Research and Development software company, where events that were essential and that might not exist in the application were discussed. The next figure shows a later phase of the whole process when all the events discussed were reduced to just a few, awaiting approval to integrate the application.

The table below shows several events that were pointed out as extremely essential, such as strong winds, fog, fires, tsunamis, etc. In the last step of this phase, a verification was made as to whether the events, which had not yet been eliminated, were already included in one of the tables from one of the previous steps. After analyzing the events presented in the tables above, a conclusion was reached so as to which events should be integrated into the application, as shown in the following table. Two different examples are presented, the tornadoes that were eliminated because they already existed in the application, and the floods, which were inexistent, but essential as they are a very common event.

| NEW WEATHER EVENTS VALIDATION | | | | |
|---|---|---|---|---|
| **Event** | **Exists** | **Code** | **Subcode** | **Img code** |
| Rain fall | YES | - | - | - |
| High Temperature | NO | 17 | 500 | 31 |
| Low Temperature | NO | 17 | 501 | 26 |
| Heat Wave | NO | 17 | 502 | 31 |
| Cold Wave | NO | 17 | 503 | 26 |
| Ice | YES | - | - | - |
| Snow | YES | - | - | - |
| Wind | YES | - | - | - |
| Fog | YES | - | - | - |
| Tornado | YES | - | - | - |
| Tropical Cyclone | NO | 17 | 504 | 25 |
| Hurricane | YES | - | - | - |
| ThunderStorm | YES | - | - | - |
| Fire Risk | NO | 17 | 505 | 32 |
| Avalanche Risk | NO | 17 | 506 | 8 |
| Flood Risk | NO | 17 | 507 | 35 |
| Flood | NO | 17 | 508 | 35 |
| Tsunami | NO | 17 | 509 | 34 |
| Coastal Event | NO | 17 | 510 | 25 |
| Earthquake | NO | 17 | 511 | 33 |

**Table 4.6:** Weather events application

This is the last step of the whole process of finding new weather events. At this stage we already know which events will be integrated into the application, in other words these are the events which have been approved, leaving us to define their types, codes and subcodes, as well as the best images that define them.

Thus, we were able to obtain the final table that represents the new meteorological events of adverse road conditions, making the application more responsive regarding information to share with all users.

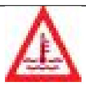
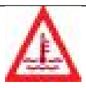
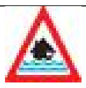
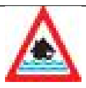
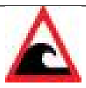
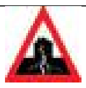
| NEW APPROVED WEATHER EVENTS | | | |
|---|---|---|---|
| **Cause description** | **Cause code** | **Sub-cause code** | **Road sign** |
| High temperature | 17: ExtremeWeatherCondition | 500: highTemperatures |  |
| Low temperature | 17: ExtremeWeatherCondition | 501: lowTemperatures | |
| Heat Wave | 17: ExtremeWeatherCondition | 502: heatWave |  |
| Cold wave | 17: ExtremeWeatherCondition | 503: coldWave | |
| Tropical Cyclone | 17: ExtremeWeatherCondition | 504: tropicalCyclone | |
| Fire risk | 17: ExtremeWeatherCondition | 505: fireRisk | |
| Avalanche Risk | 17: ExtremeWeatherCondition | 506: avalancheRisk | |
| Flood Risk | 17: ExtremeWeatherCondition | 507: floodRisk |  |
| Flood | 17: ExtremeWeatherCondition | 508: flood |  |
| Tsunami | 17: ExtremeWeatherCondition | 509: tsunami |  |
| coastal Event | 17: ExtremeWeatherCondition | 510: coastalEvent | |
| Earthquake | 17: ExtremeWeatherCondition | 511: earthquake |  |

**Table 4.7:** Weather events application

Countless important events related to the environment can happen and are, therefore essential to this dissertation. As such, we foregrounded new meteorological events that deserve and need to be included in the application, carefully analyzing and adding them to the previous list.

We can quickly think of multiple events that occur regularly in the winter, when the rain is intense, such as floods or coastal events. If we think about an opposite phase of the year, summer, we quickly come to the conclusion that fires are a constant, both in our country and in the rest of the world, and often affect driving conditions. These types of events have a huge negative impact on national and international roads and it is crucial to inform all vehicles that drive through these areas of possible dangers. This is the main reason why they were integrated.

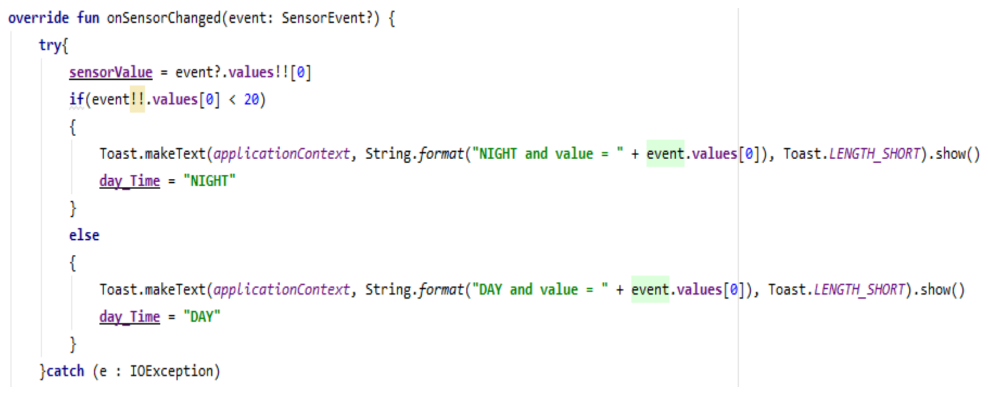
## 4.3 Extra data collection

In addition to facilitating and speeding up communication between vehicles and infrastructures, the application seeks to automatize the detection of events, and possibly share them, without the intervention of the driver himself/herself. In order to do this, the application takes into account pre-existing components that incorporate a smartphone and seeks to collect extra data that can make this automation possible, either through sensors or images that can be further processed by other entities of the road system.

### 4.3.1 Luminosity

One of the major causes of road accidents is poor visibility. This is due to several factors: it may be a road where other vehicles have limited visibility, fog, rain, or insufficient light in a particular section of the road. These are a few of the reasons why a great majority of accidents happen at night. It is obvious that we cannot change the lighting of a road with the application but we can collect data and work on it preventively.

Thus, the application collects data on the environment's luminosity conditions at all times by using the light sensor built into the smartphone. [22]



```
override fun onSensorChanged(event: SensorEvent?) {
    try{
        sensorValue = event?.values!![0]
        if(event!!.values[0] < 20)
        {
            Toast.makeText(applicationContext, String.format("NIGHT and value = " + event.values[0]), Toast.LENGTH_SHORT).show()
            day_Time = "NIGHT"
        }
        else
        {
            Toast.makeText(applicationContext, String.format("DAY and value = " + event.values[0]), Toast.LENGTH_SHORT).show()
            day_Time = "DAY"
        }
    }catch (e : IOException)
```

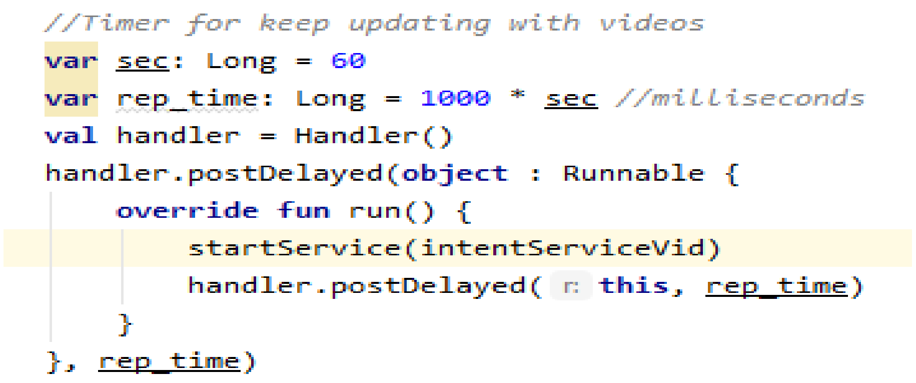
**Figure 4.1:** Collection of luminosity sensor values

The idea is for the application to collect the luminosity data and then send it to an MQTT topic related only to this subject. This way, an external entity that processes this data, has more information to be able to process and report preventive events, with precision.

### 4.3.2 Video Recording

Although the light sensor is a valuable help, the application takes advantage of one of the great weapons of today's smartphones, the camera. The application aims to use the smartphone's great image quality and high-speed video recording system to collect visual data about the roads and the environmental conditions, as cars go by, making it possible to associate geographic locations with road conditions in real time.

For this purpose, a service was created in the background, which is accessed every 120 seconds allowing the application to record a short video to collect data. It was created in the background and without an activity interface so as not to take the driver's focus off the road.

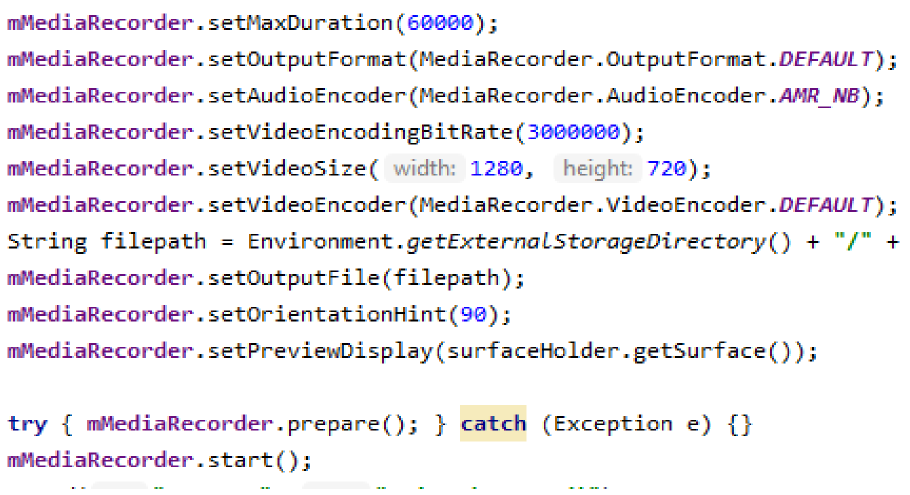


```
//Timer for keep updating with videos
var sec: Long = 60
var rep_time: Long = 1000 * sec //milliseconds
val handler = Handler()
handler.postDelayed(object : Runnable {
    override fun run() {
        startService(intentServiceVid)
        handler.postDelayed( r: this, rep_time)
    }
}, rep_time)
```

**Figure 4.2:** Recording service call

The service activates the camera in the background and records a 10-second video of the road. It automatically makes the video, stores it and eliminates it, which makes this operation quick and without great battery loss. Since the video is recorded in the background, the service presents the user with a notification about when it is recording so that the user is aware of what is happening.

For this purpose, **MediaRecorder** was used which makes the implementation of a recording method very simple and intuitive, making it easy to define video parameters and start, pause or stop the recording.



```
mMediaRecorder.setMaxDuration(60000);
mMediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.DEFAULT);
mMediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
mMediaRecorder.setVideoEncodingBitRate(3000000);
mMediaRecorder.setVideoSize( width: 1280, height: 720);
mMediaRecorder.setVideoEncoder(MediaRecorder.VideoEncoder.DEFAULT);
String filepath = Environment.getExternalStorageDirectory() + "/" +
mMediaRecorder.setOutputFile(filepath);
mMediaRecorder.setOrientationHint(90);
mMediaRecorder.setPreviewDisplay(surfaceHolder.getSurface());

try { mMediaRecorder.prepare(); } catch (Exception e) {}
mMediaRecorder.start();
```

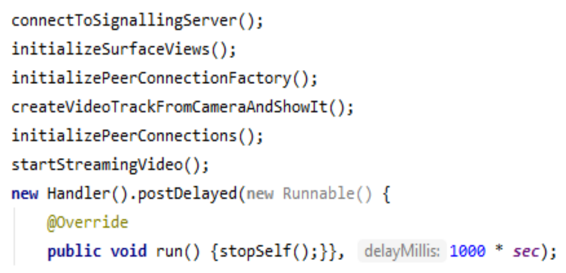
**Figure 4.3:** Media Recorder configuration

## 4.4 Stream Service

For more serious road incidents, it is crucial that traffic control receive more information about several impacting factors related to an accident, to understand if there are injuries, the number of cars involved, if there is a roadblock, etc. For this purpose, an application stream platform was developed and can be accessed by the traffic control entity.

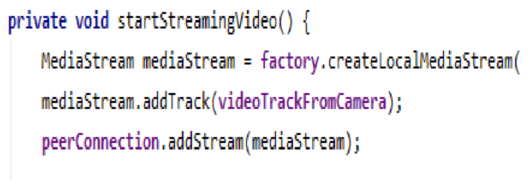
A real-time web communication, known as WebRTC, was implemented, which makes it possible for the application to stream video and audio. WebRTC allows you to configure the connection between two peers via the web, which is usually done through the browser between two computers, but in this case it has been adapted for the mobile application.[23] [24]

Thus, in case the application is crossing an accident zone, the service is launched and it starts a connection with the server, creating a room with a unique id, which is generated and sent, by MQTT, to the OBU. From this moment on, the application starts to stream, only video, in the indicated room, where the traffic control can join and watch. The stream lasts for 3 minutes or until the guest understands that he no longer needs more information about the accident and decides to leave the room.



```
connectToSignallingServer();
initializeSurfaceViews();
initializePeerConnectionFactory();
createVideoTrackFromCameraAndShowIt();
initializePeerConnections();
startStreamingVideo();
new Handler().postDelayed(new Runnable() {
    @Override
    public void run() {stopSelf();}}, delayMillis: 1000 * sec);
```

(a) Full service process



```
private void startStreamingVideo() {
    MediaStream mediaStream = factory.createLocalMediaStream(
    mediaStream.addTrack(videoTrackFromCamera);
    peerConnection.addStream(mediaStream);
```

(b) Start streaming method

**Figure 4.4:** Stream service process - Application side.

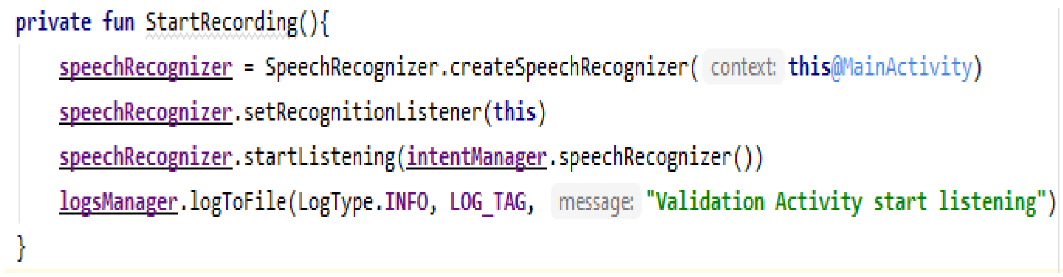
Even though this service and streaming happens in the background, in order not to interfere with the driver's concentration, the application launches a notification to report what is happening in the back-end.

## 4.5 Event validation through speech recognition

he TRUST project needs as much information as possible to keep drivers informed and updated on what is happening on the roads. Sensors, automatic event determination processes, third-party platforms and application reports are some of the great sources of information for generating events. However, to keep drivers updated it is just as important to insert events into the application as it is to remove them when they are finished, preventing drivers from receiving notifications of events that, in fact, no longer exist.

Therefore, in addition to the expiration date or other event cancellation processes, a way for the system to receive real and updated information on these cases is for the application to confirm whether the event is still active or not. With this method, if a system receives multiple cancellation messages about the same event, then it is probably because the event ended before the expiration date.

In this case, the class "TextToSpeech" was used, allowing the application to ask the user, when he is passing the event, if it is still active.



```
private fun StartRecording(){
    speechRecognizer = SpeechRecognizer.createSpeechRecognizer( context: this@MainActivity)
    speechRecognizer.setRecognitionListener(this)
    speechRecognizer.startListening(intentManager.speechRecognizer())
    logsManager.logToFile(LogType.INFO, LOG_TAG,  message: "Validation Activity start listening")
}
```

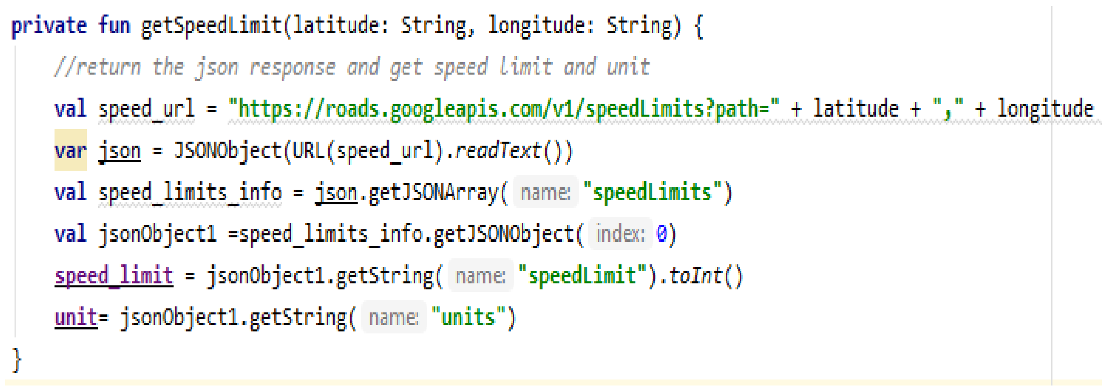
**Figure 4.5:** Speech Recognition method

The application then awaits for a negative or positive response from the user, something like "Yes"or "No". The process of speech recognition begins with the initialization of the SpeechRecognizer, as shown in the figure above, and only after does it consider the results and validate the user response. The results are converted to text and, based on the response, allow feedback of the generated event to be given to the system.

## 4.6 Speed Limit Advisor

As discussed earlier, a direct approach to real-time speed recommendation was considered, because speed is a major influence on road safety and is often identified as the major cause of accidents. The application then presents the user with a recommended speed limit based on the local speed limits, as well as the type of vehicle and the events that influence the surrounding area.

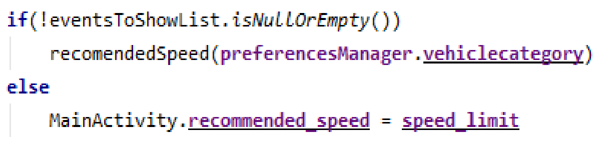
First, the application regularly checks which road the vehicle is on (a rural or local area, a highway, etc.) and registers the speed limit set for that same road. This is achieved by the Google's Roads API which provides this type of information based on the vehicle's location. [25]



```
private fun getSpeedLimit(latitude: String, longitude: String) {
    //return the json response and get speed limit and unit
    val speed_url = "https://roads.googleapis.com/v1/speedLimits?path=" + latitude + "," + longitude
    var json = JSONObject(URL(speed_url).readText())
    val speed_limits_info = json.getJSONArray( name: "speedLimits")
    val jsonObject1 =speed_limits_info.getJSONObject( index: 0)
    speed_limit = jsonObject1.getString( name: "speedLimit").toInt()
    unit= jsonObject1.getString( name: "units")
}
```

**Figure 4.6:** Speed limit getter method

After this, the application verifies which incident events are active in the vicinity and which of these can influence the driver and his/her driving. If there are active events, a new recommended speed limit is calculated based on the type of road and on the type of vehicle in question. If there are no active events, the recommended speed limit remains the same.



```
if(!eventsToShowList.isNullOrEmpty())
    recomendedSpeed(preferencesManager.vehiclecategory)
else
    MainActivity.recommended_speed = speed_limit
```

**Figure 4.7:** Conditioning event existence verification

If the application considers that a new recommended speed limit needs to be calculated, the app invokes the*recommendedSpeed* method. A speed limit adjustment is made according to the current limits, reducing the speed limit to 1/3, 1/2, or other values, and based on the vehicle in question, for instance very heavy goods vehicles have to be extra careful in dangerous situations.

At the same time that the application checks for speed limits, it updates the interface with the recommended values, so that the user always has real-time and very accurate data.

This suggestion, as mentioned, appears as a speed limit sign at the top of the map activity and we'll be able to see how it behaves in the next chapter where tests will be performed in different roads and in situations with different events.

## 4.7 Road conditions recognition through image processing

The mobile application is essential to the driver for it presents him/her with the results of the data that was collected by different components. The faster the information is processed, the more accurate and truthful the alerts generated by the system are.

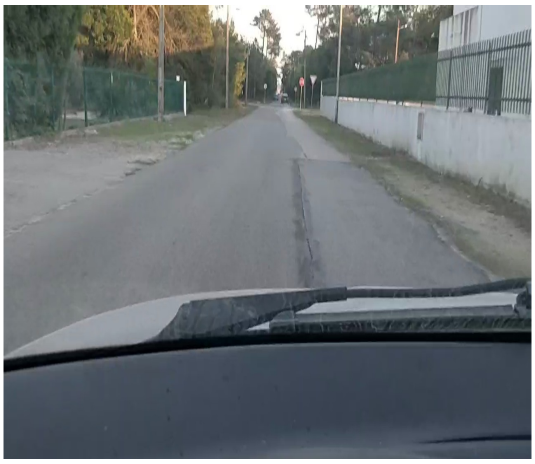
Thus, with the collection of video images by the application, we thought about how to handle this data in order to quickly determine weather alerts and associate them with the corresponding road section. Therefore, it was determined as important to analyze this data using both the application and other entities of the system, by receiving images from the driver's point of view and classifying them in relation to the luminosity of the road and the existence of rainfall, as these are two main causes of road accidents. To this end, two classification models were created for each particular purpose.

### 4.7.1 Luminosity model

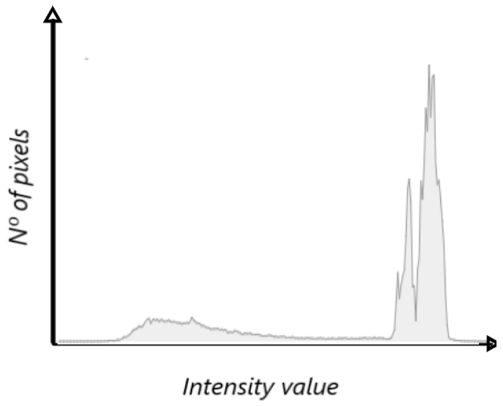
In this case, the trained model focused on the colors of all of the pixels in each image. In the beginning of the training phase, the model receives a set of images, from the driver's point of view, regarding road sections with sufficient or insufficient light, represented by 10,000 categorized images each. The model then analyzes the set of images and forms its own opinion on how to classify them. [26]

To train the model, the RGB color level of every pixel, for each image, was counted. After this the average color intensity of all pixels was calculated and a representative value was associated to each image. [27] This method was performed thousands of times for all images, associating and separating the values of images with sufficient light from those with insufficient light.

The difference of the color intensity and the values obtained in the two training figures are presented below. The first one, is an image taken during the day and presents very high color intensity values for most of the pixels. In the second image there is insufficient light, so the values of all the pixels appear clustered at a much lower intensity.

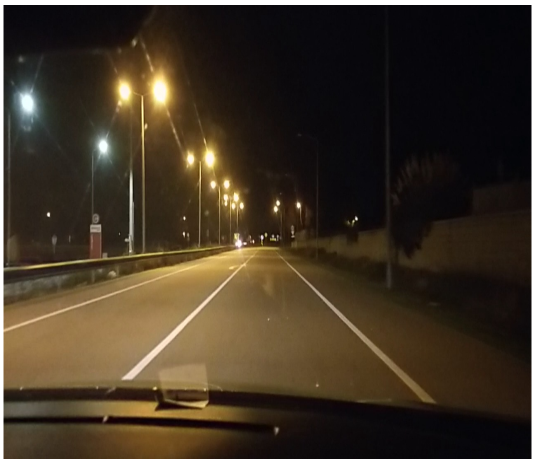


**(a)** Day sample

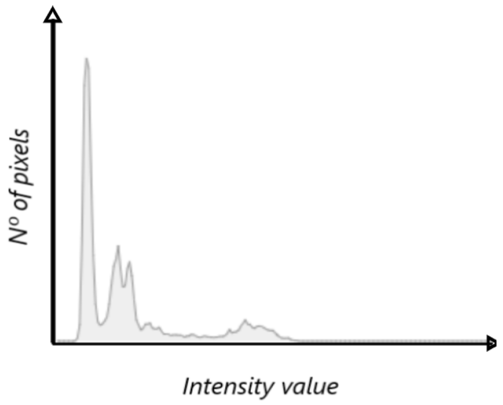


**(b)** Image histogram

**Figure 4.8:** Distribution of luminosity values in a daylight image



**(a)** Night sample



**(b)** Image histogram

**Figure 4.9:** Distribution of luminosity values in a night image

We then obtained the average of the values within all the images, reaching a final classification value on which the model is based.

Once trained, the program receives a 10-second video, which is converted into about 300 images, which are classified individually. The model first analyzes the brightness of each pixel for each image. Then it assigns an individual value, and calculates the average of all pixels in the image. Based on this average, a final rating, either bright or dark, is assigned to the image.

Lastly, a final count of all the images is made, and a result is obtained concluding whether the video presents sufficient visibility or not. It is important that the classification of all images is made individually and that a global result is obtained so as to eliminate any possibility of some samples showing wrong indicators. For example, a vehicle can pass by an area of insufficient light, but the lights flashed from other vehicles can confuse the final result for a place with sufficient light, which in fact, is not true.

### 4.7.2 Rainfall model

The rainfall model and the luminosity model are a little different from each other. The rainfall model's objective is to detect if it is raining, which makes the model's task more complex, in comparison. In this case, a more complex analysis of each image is necessary in order to be able to detect various characteristics that determine whether or not it is raining. Thus, to build the most capable and assertive model, the TensorFlow platform was used, which provides various tools and libraries to create machine learning models, through the study of patterns. In this specific case, a high-level API, called Keras, was used to create neural networks with simplicity. The intention was to create a binary image classification model that determines, whether it is raining or not.

The first step was to collect the necessary dataset, composed of images with and without rain, to train the model. To achieve a better behavior from the model, about 30,000 samples were collected from each output, that is, 60,000 images in total.

When the dataset was completed and ready, the process of implementing the image classification model began. The first decision made was to build a convolutional neural network, which is widely used in the detection of image patterns.

A convolutional neural network consists of combinations of: [28]

- Convolutional layers - a process in which a convolutional filter passes and operates on the image matrix, creating a new and slightly smaller matrix filtered by a feature.

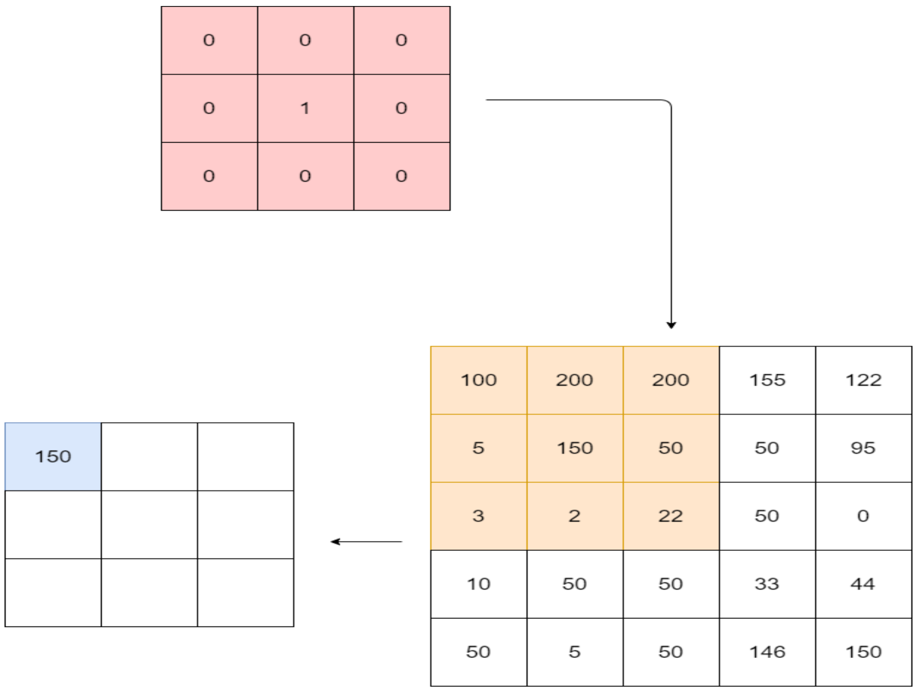


**Figure 4.10:** Example of a convolutional filter that passes over an image matrix

- Pooling layers - the process of reducing a matrix created by a convolutional layer to a smaller matrix. The highest value or the average value of a selected region is used to represent that section of the matrix.

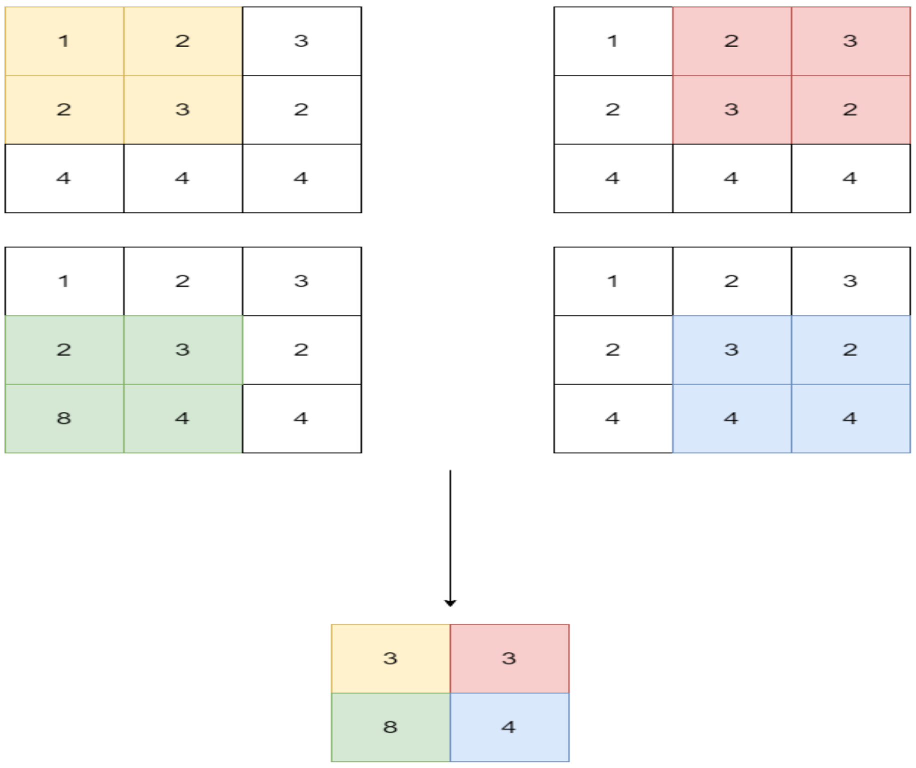


**Figure 4.11:** Example of pooling matrix values

- Dense layers, or fully connected layer - each layer node is connected to all nodes of the next layer.

The basic structure of a convolutional neural network, which has been applied here, is: Convolution -> Pooling -> Convolution -> Pooling -> Fully Connected Layer -> Output. Each convolutional step is followed by pooling, and together they form a hidden layer. The rain classification model that was implemented was based on this network. [29]

Firstly we decided to use a sequential model, a neural network model composed of several stacked layers, where the output of one layer is the input of the next, knowing that each output only has a single result. [30]

After loading the data, deciding on the type of model to be used and, in accordance with the structure of the convolutional neural network, it was imperative that at least one hidden layer composed of a convolution, followed by a pooling, created through the methods add(Conv2D (...)) and add(MaxPooling2D()), be introduced.

An activation function was also required to receive the sum of all inputs from the previous layer and then generate and pass the output to the next layer. In this case, the default activation function 'ReLu' - Rectified Linear Unit - was used, which unless the input value is negative or zero, returns the input value.

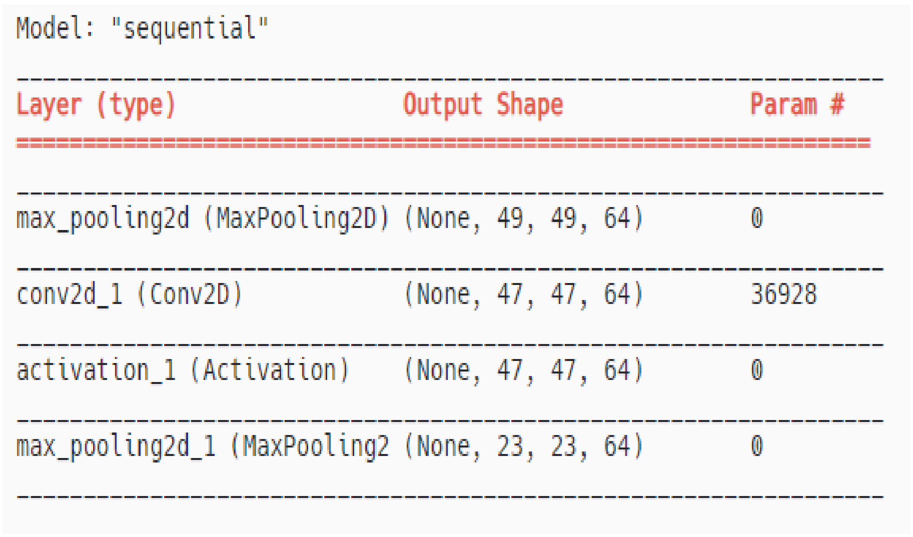
```
model = Sequential()
model.add(Conv2D(layer_size, (3,3), input_shape=X.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
```

**Code 1:** Adding conv2D and Pooling2D methods

Each layer, composed by a (3.3) convolutional filter, that passes over every pixel of the image, and a (2.2) pooling that reduces the matrix to only half.

In the image below we can see a summary of one of the generated models, and the changes that the matrix undergoes when going through this layer:

- The image matrix reaches the layer with the size of 49x49.
- A convolutional matrix filter with a 3x3 size is passed, reducing the matrix in 2 rows and 2 columns, resulting in a 47x47 matrix.
- The pooling receives the previous input and reduces the matrix by (2.2), that is, half of its previous value,47/2 = 23.5, forming a new matrix of 23x23. However, we see that 0.5 corresponds to 1 row and 1 column and that this situation is not taken into account, therefore image information is lost.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 49, 49, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 47, 47, 64)        36928
_________________________________________________________________
activation_1 (Activation)    (None, 47, 47, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 64)        0
_________________________________________________________________
```

**Figure 4.12:** Model summary

Before the output node presents a final result, a dense layer must be added to the model. However, dense layers receive vectors as input, which are presented in 1D, so the Flatten layer is introduced first, to flatten the matrix into a single vector. Once this procedure is done, it is ready to be processed by the dense layer, that in this case produces an output of size 1. The result is always a binary number (1 or 0), either there is rain or not.

```python
model.add(Flatten())
model.add(Dense(1))
model.add(Activation('sigmoid'))
```

**Code 2:** Adding flatten and dense layer

For the compilation of the models, 10 complete training sessions, or epochs, were repeated. A sum of 32 training images were used each time, of which 70% of the dataset was for training and 30% was for tests.

After doing tests with different amounts of layers, it was discovered that when 3 convolutional hidden layers were used with only one dense layer before the output, the results were quite superior, with less data loss and a higher success rate. However, although the number of nodes in each layer changed between 32 nodes, 64 nodes and 128 nodes, the final results were all very similar. Nonetheless, the most effective model was the one in which each layer had 64 nodes, with a total efficiency of 90%.

In the next chapter, we will talk about the simulated tests that were performed on the model and see the results obtained.

### 4.7.3 Using the application for the processing of road conditions

The development of the application not only included pre-existent image processing models to determine adverse weather conditions, but it also took advantage of the huge processing capacity that smartphones have nowadays. Thus, both the model for luminosity and the rainfall model were subject to a conversion and adaptation process, so that they could be used by the application.

After collecting the images, the next step consisted in implementing a listener to launch the process of checking and determining the weather conditions shown on the referred images, based on the prediction of the models, of course. From this point on, each model was processed independently and implemented in a different way.

*Luminosity prediction application*

This model, adapted the same strategy as the model created outside the context of the application; it receives the images of the recorded video and analyzes the lights' intensity. The model passes over each pixel of the image and analyzes each ones' RGB (red, green, blue) colors. Lastly, to obtain the images' brightness value, the model adds the red, green and blue colors, of all the pixels, and divides it by the total number of pixels. If the luminosity value obtained is low, the model returns a message that indicates that there is insufficient light and sends a message to the OBU with this information. In this case, a message is sent in json, via the MQTT protocol, with the precise location of the mobile phone's gps, the image generation timestamp, the model prediction and the environment light value collected by the mobile phone sensor, which, as mentioned, is constantly being monitored.

Thus, the OBU receives a message with all the event information from the application and from two different sensors (camera and light sensors), that may or may not generate events for all vehicles.

*Rain prediction application*

The image classification done by the rain model required a few extra steps. First, the model, which was created using the Tensorflow library, was converted to TensorflowLite, which is more suitable for smartphones. Once converted and integrated into the project, the application opens for the first time and loads the model. This way it will be easier to access it, when needed. The model is loaded by the Interpreter class, using the tensorflowLite library, and points directly to the converted model. This class provides the necessary methods to obtain the shape of the input, the shape of the output and to make a prediction. [31]

After this step, the model is ready to be used. The process is very similar to the luminosity prediction. The Interpreter class provides the run() method, that passes the image as an argument, and returns the result of the model prediction. As previously mentioned, the model assigns the image with only one label to the image, whether it is raining or not.

Based on the model's response, if the result returned is "rain", the model sends a message, also by MQTT, to the OBU with the model's prediction information, the event's timestamp and the exact location of the vehicle. If the answer is negative, that is, it is not raining, the application simply ignores the response and does not report any result.

CHAPTER 5

# Tests and Validation

Once the overall TRUST project, its system, application components and implementation are covered, we will address the tests performed and the obtained results. This chapter is important because it not only validates what has been developed but it also generates an exact idea of what is correct or incorrect and what can be improved or added.

For each component of the application, tests were performed on the road and in real time in order to draw conclusions about how the application behaves. The behavior obtained from each component will be analyzed and explained in detail in this chapter.

## 5.1 Speech Recognition

One of the features developed in the application refers to the collection of information regarding the validity of alerts generated by the system. As mentioned, it is another source of information that can be useful if applied correctly. This functionality consists of two tasks: the analysis and verification of the alert data by the application and the elaboration of the response to the system. Since the system is not currently able to process this type of information from the application's response, the message structure or its content has not been elaborated. Note however, that the method of sending messages from the application to OBU is already operational through MQTT messages.

Even though it's important to send the message to OBU, it is also essential to analyze and verify the alert data. In relation to this part of the process, two tests were considered: application-user interaction, using the "voice"of the application, and the reception and perception of information from the user's voice, user-application.

### 5.1.1 Application - Driver

This function is activated whenever the user is alerted for the first time about any type of warning, in coordination with the appearance of the application event alert window, which can be seen in the next image.

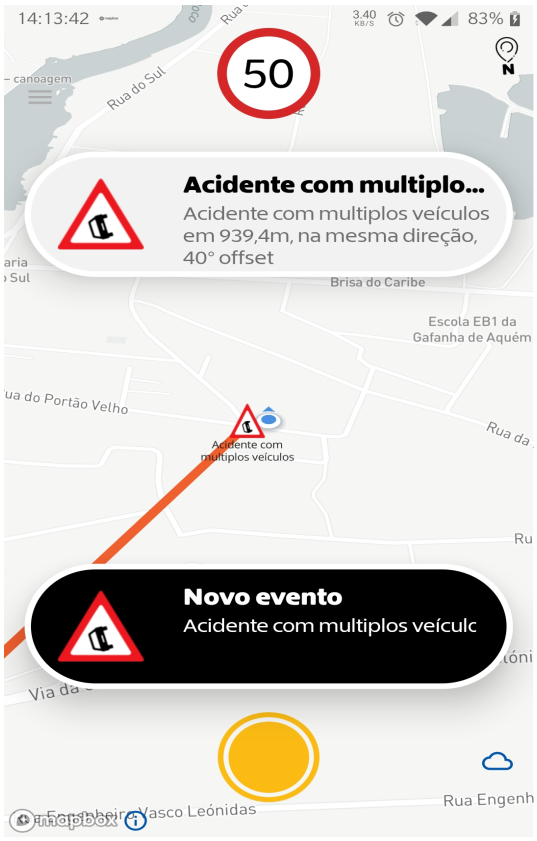

14:13:42
83%
50
Acidente com multiplo...
Acidente com multiplos veículos em 939,4m, na mesma direção, 40° offset
Acidente com multiplos veículos
Novo evento
Acidente com multiplos veículc
mapbox

**Figure 5.1:** Incident notification

When the pop-up appears, the application says: "We have the information that the event 'event type' is still valid. Can you confirm?". Only the type of the event is changed, depending on the situation. When the sentence ends, the application creates a speech recognition process and waits for the user's response.

### 5.1.2 Driver - Application

When the speech application finishes, it is up to the driver to answer the question asked by the application. The answer can be either positive or negative, depending on the situation. The application recognizes phrases based on the word "Yes", such as "Yes it is valid", "Yes I confirm", "I confirm the event" and associated phrases, validating the event. It also understands denial phrases such as "Not valid", "I do not confirm" and associated phrases without major problems, invalidating the event.

The application's behavior is as expected when the driver's response is that of agreement or denial. The problem arises when the user's response is unforeseen, does not fit any of the solutions above, or when there is no response.

Based on the result, the application should send a message to OBU, in case the event is no longer valid, or ignore and take no action, in case of a negative response, but that part of the process has not yet been implemented.

## 5.2 Luminosity Values

The data that is collected from the light sensors, depends on the sensor and its ability to truly perceive the lighting, because the data is updated and collected each time the sensor detects a change in luminosity.

These variations are more natural during the day. At night it is normal for the data to remain very low, without major changes. If we compare the two, during the day we have situations in which the sun radiates light directly on the sensor, others in which there are trees or buildings that block the light, changing the values, justifiably.

Since there is no natural light at night, the light usually comes from electricity poles or from other vehicles. This makes the variation in values not so frequent and the results much more constant.

The following graph shows the variance of the data collected from the sensor, both at night and during the day.

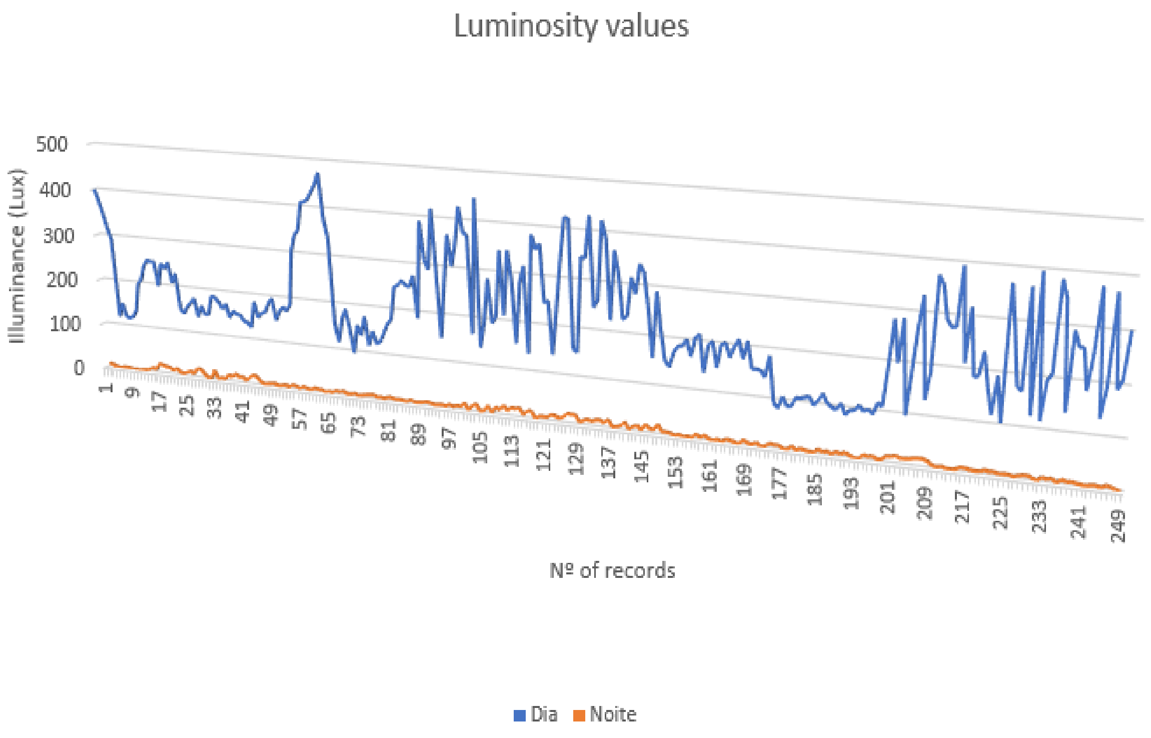


**Figure 5.2:** Variation in luminosity values

It is obvious that there is little variation in the values collected at night, with the light becoming very low, but very consistent. During the day, we see the opposite, very high and sometimes intermediate values. The values that are collected during the day are never very low, so they are not comparable to those collected at night, which makes it easy to manage this code during both parts of the day.

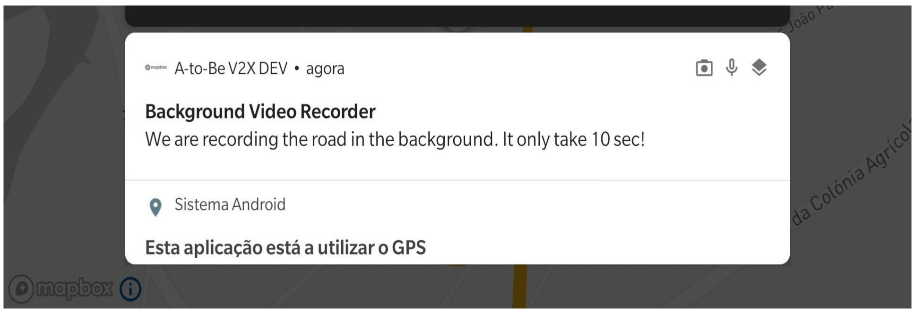
This graph has 250 values collected during the night, and 250 during the day, which is enough to conclude a generalization of this data collection.

After having collected all the data, it was decided that when the light is below 20 lux, the location in question is dark and, therefore, of possible low visibility. Values above 80 lux are of sufficient light and pose no interference in driving. Since there are no results with intermediate values, when they arise, the outcome is inconclusive.

## 5.3 Video Recording

As presented and discussed in previous chapters, a feature for collecting data on the environmental conditions was added to the application, using the smartphone camera. This action takes place every 2 minutes, making it relatively cheap to use on the mobile phone, but equally effective in constantly collecting data. These 2 factors were taken into account when the 2 minute bar was established.

Since it is very important that the driver not be distracted the video is recorded in the background. For privacy reasons, the user is alerted about this action through a notification at the top of the screen. This way, the user, receiving this notification, is automatically aware that the smartphone camera is being used by the application


A-to-Be V2X DEV • agora

Background Video Recorder

We are recording the road in the background. It only take 10 sec!

Sistema Android

Esta aplicação está a utilizar o GPS

**Figure 5.3:** Recording action notification

The duration of this action is short, generating videos of only 10 seconds. It is important to note that it was decided to record videos instead of images, because this method captures much more information from a single location. The video, once captured, will be divided into frames and converted into images. The videos are filmed at 30 frames per second, which means that each video can collect about 300 samples from the same location, much more than if it were captured image by image.

The next example shows the result of converting a video, collected by the application, into images. Each image represents a frame of the video. These are all the images that will be processed to determine the driving conditions of a specific location.

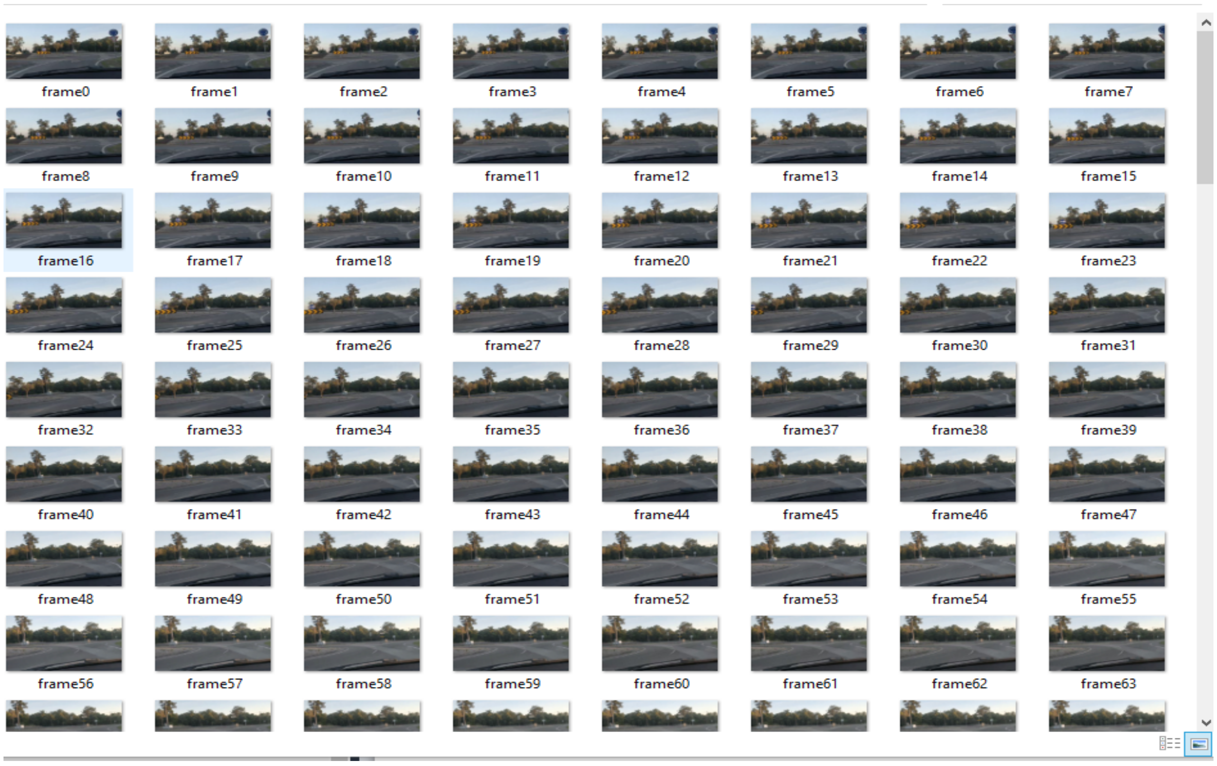


**Figure 5.4:** Evidence collected from the application

### 5.3.1 Battery life issue

One of the major concerns regarding this feature is the battery issue. We know that all applications that require intensive use of the smartphone location have a certain cost for its battery, due to the constant updates they undergo. It is thought that the excessive use of the camera can reduce life battery significantly.

Thus, tests were carried out to analyze the impact of this functionality, when activated or deactivated. The graph below shows a comparison of the results obtained when using the application with the recording functionality or without it, for 30 minutes.

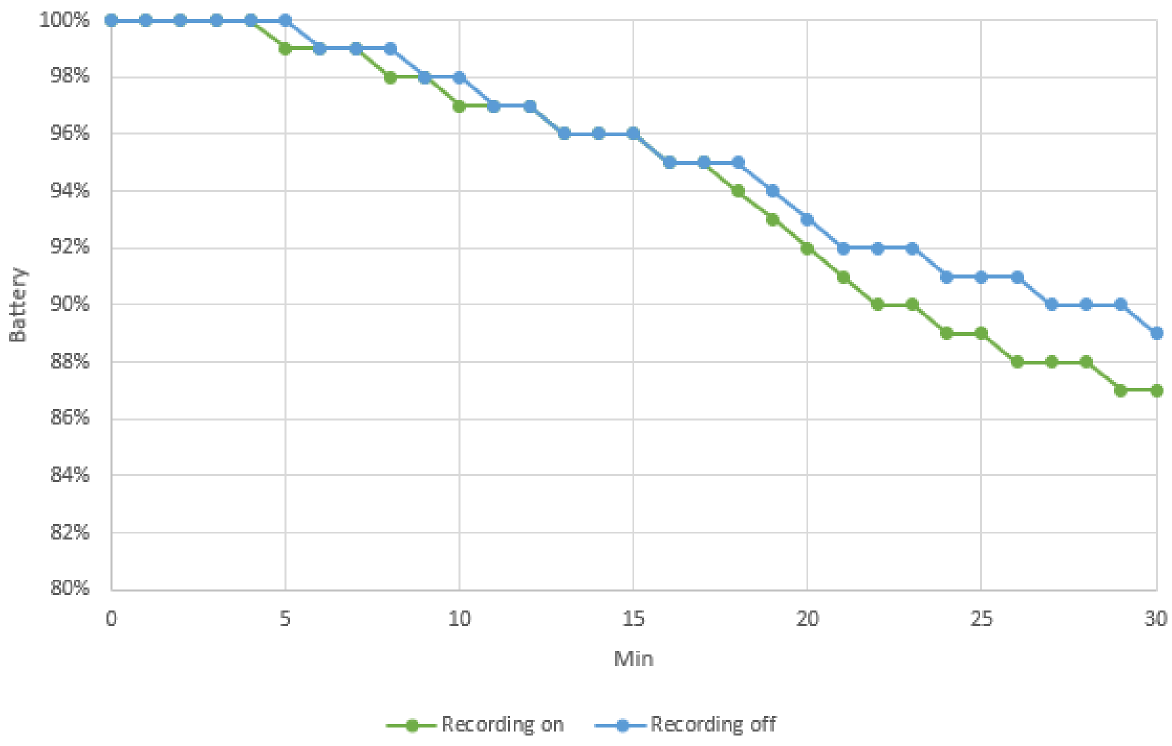


**Figure 5.5:** Variation of battery values

After observing the curve of both lines, we can conclude that the application uses less battery when this feature is inactive, which is its expected behavior. The important thing is to assess whether this difference is so noticeable that it could jeopardize the integration of this requirement.

When the camera is not activated, the line in the graph shows a loss of battery of 0.36% per second, which is expected given the constant location updates. After using the application for about 30 minutes, the loss of only 11% of battery is not, at all, a major concern.

Regarding the behavior of the application, when this feature is active we can see that there is a noticeable difference from the latter situation. However, the difference is not very pronounced and the data obtained shows a loss of battery of only 0.43% per second, compared to the previous 0.36%. Throughout the use of the application, the battery loss is faster when this functionality is active, but it is not as significant as initially thought. At the end of the test, the application only lost 13% of the total battery, results which were very similar to the previous ones.

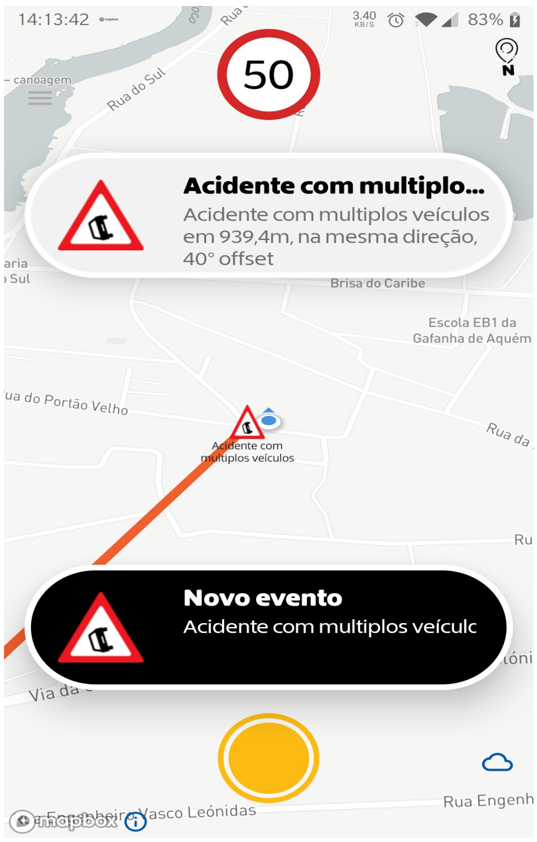
Following this rate, if the application is used for 2 hours, without activating the recording feature, there will be a loss of about 43% of the battery. If the feature is activated, there is about a 51% battery loss. Regardless of the situation, both results are positive when the conditions and type of application are taken into account.

## 5.4 Stream Service

One of the latest features developed was the incorporation of a video stream service between the application and a browser. This allows the responsible traffic control entity to watch, in real time, the event that the vehicle is undergoing. In order to determine the optimal behavior of this functionality, a real road accident was simulated, for this is the only type of event that activates this service. The following steps were used to verify if the functionality of the stream service behaves as expected.

- Place the smartphone on the dashboard phone holder, facing the road, while driving.
- Generate an accident alert at a given location.
- Cross that same location with the application and wait for the alert to be received.
- Receive the ID of the room where the stream of this accident will be transmitted.
- Check for the existence of a notification that represents the beginning of the stream, in the smartphone notification bar.
- Through the browser, enter the stream room of the accident in question.
- Check the reception of the video and analyze its content.
- Leave the room and verify if the stream service ended.

As shown in the two figures below, the following describes the simulation of the reception of an accident alert. An ‘accident with multiple vehicles’ event was generated in a specific location. Moments before passing through the location the driver receives that event alert. When the driver passes by that location, and receives the alert event. that a driver, user of the application, passed moments having therefore received the alert. On the application side, notice the reception of the event notification, as well as the appearance of the icon at the top of the notification bar that refers to the beginning of the stream.


14:13:42
83%
50
Rua do Sul
Brisa do Caribe
Escola EB1 da Gafanha de Aquém
Rua do Portão Velho
Acidente com multiplos veículos
Novo evento
Rua Engenheiro Vasco Leónidas

Acidente com multiplo...
Acidente com multiplos veículos em 939,4m, na mesma direção, 40° offset

Novo evento
Acidente com multiplos veículc

**Figure 5.6:** Accident notification

When the entity watches the event in real time, we can see that, after entering the corresponding room, in the center of the image there is a window with the video stream. Even though it is night time, the image is well centered, giving traffic control the possibility to draw its own conclusions about the accident.

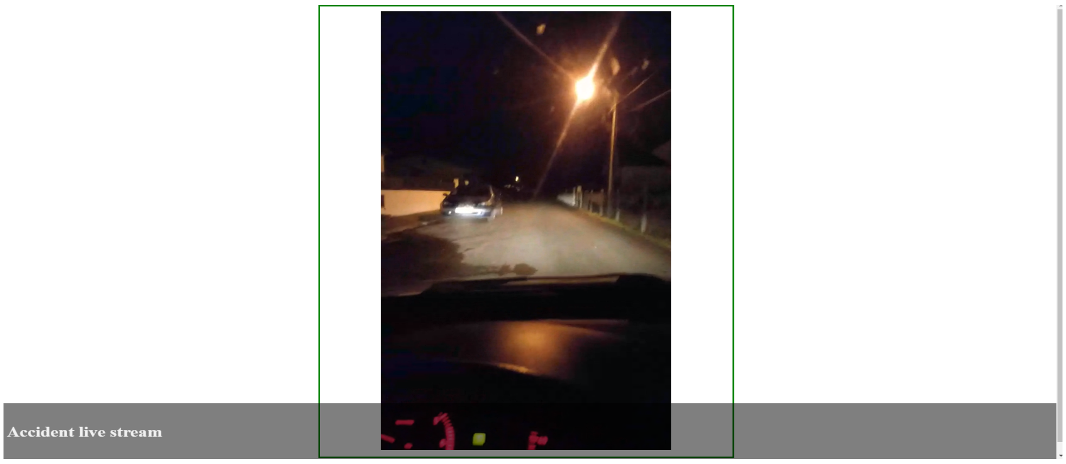


**Figure 5.7:** Real time video stream

However, some problems and limitations have been encountered in the development and testing of this functionality, such as:

- Relatively good internet connection quality is required.
- Camera quality is required. The better the quality of the smartphones' camera, the better the quality of the video .
- As soon as a connection is lost, for some reason, this room becomes useless as the stream ends and can not be joined again.

Although the basic and general concept is working as required, it is not at all perfect or ideal. If it is in the interest of the project, in the long-term, the current solution and possible improvements should be analyzed to create a useful and important feature.

## 5.5 Speed Limit Advisor

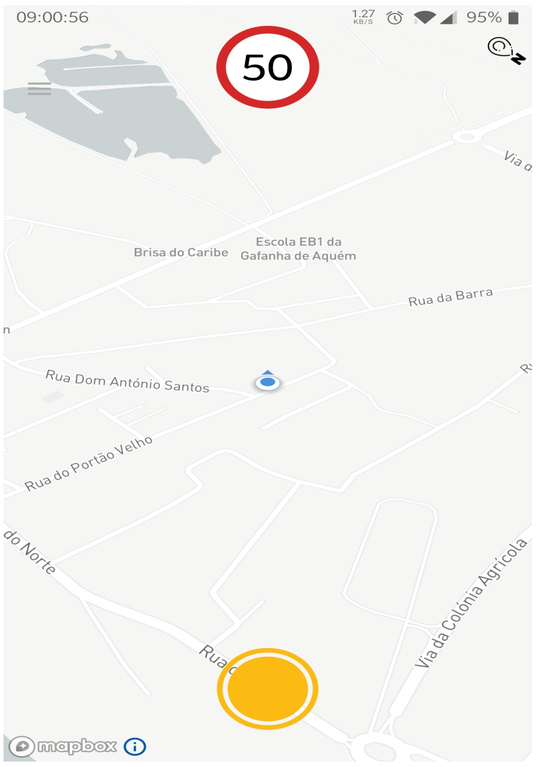
A feature was developed to provide the user with the recommended speed in real time. The application takes into account the type of road the vehicle is using, analyzing the speed limits, the alerts generated within the surroundings and the type of vehicle inserted in the application by the user.

Therefore, since the algorithm presents different types of behavior depending on these parameters, tests were carried out for each case.

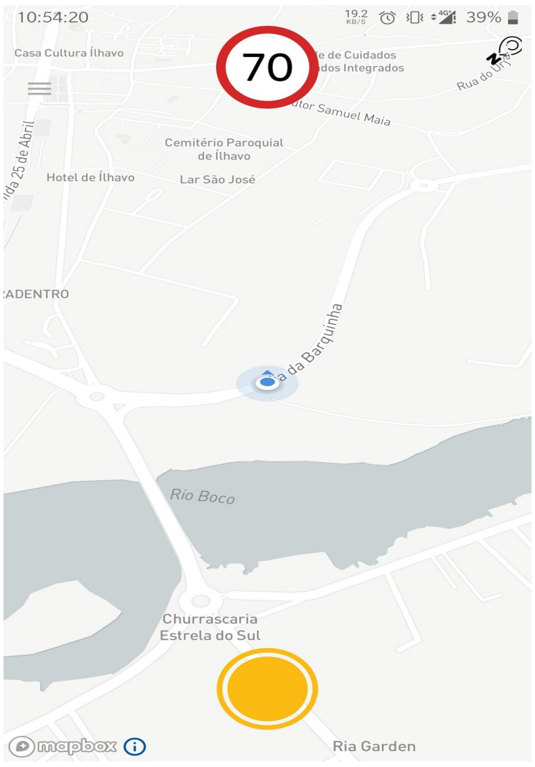
### 5.5.1 Road Type

The road type is the main parameter, as speed limits vary accordingly, therefore, even when a driver is not faced with any incident, the recommended speed limit should always be shown in real time.

In the tests carried out, the outcome was as expected: when the vehicle used different road types, the application varied the values of the recommended speed limit. For example, within built-up areas it presented 50km/h and outside built-up areas it presented 70km/h, as shown in the following figure.



**(a)** Inside Built-Up Area

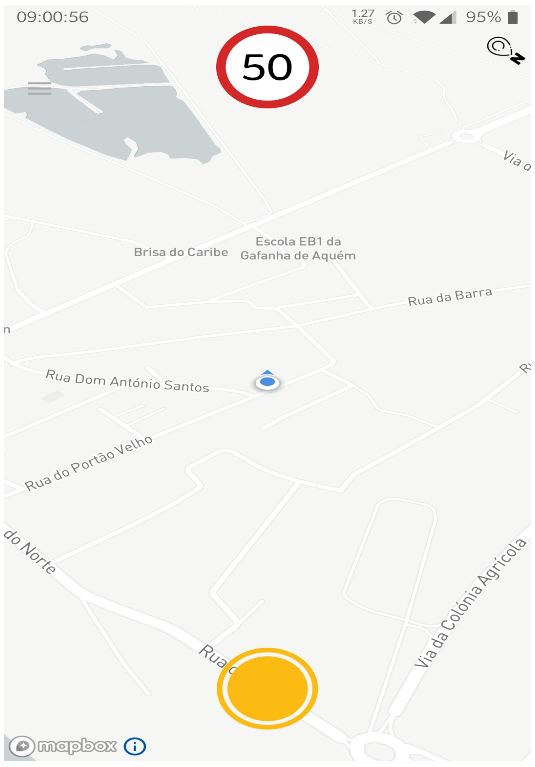


**(b)** Outside Built-Up Area

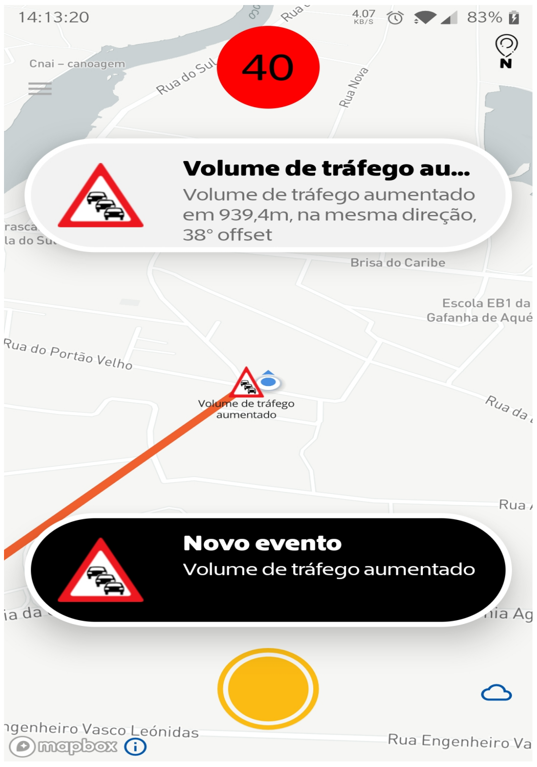
**Figure 5.8:** Recommended speed limit without any alert

### 5.5.2 Alert Type

In addition to the constant collection of speed limits, the algorithm checks for alerts in the vicinity to keep the driver informed, if necessary. In this case, the moment the application receives the alert, the recommended speed limit is reduced. The following image shows an example within a built-up area, where the speed is 50km/h and changes to 40km/h when the alert is received.



**(a)** Before event reception



**Volume de tráfego au...**
Volume de tráfego aumentado em 939,4m, na mesma direção, 38° offset

**Novo evento**
Volume de tráfego aumentado

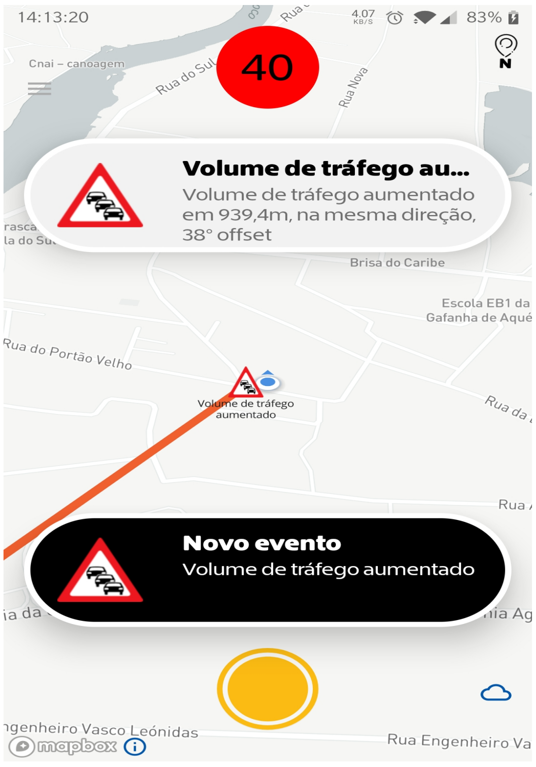
**(b)** Speed advisor update after event reception

**Figure 5.9:** Speed advisor before and after event reception

### 5.5.3 Vehicle Type

The last parameter takes into account the type of vehicle. Larger vehicles, such as trucks, have to be more concerned and careful in relation to speed limits than for instance light goods vehicles or heavy goods vehicles. The application must take into consideration each vehicles' behavior in different situations.

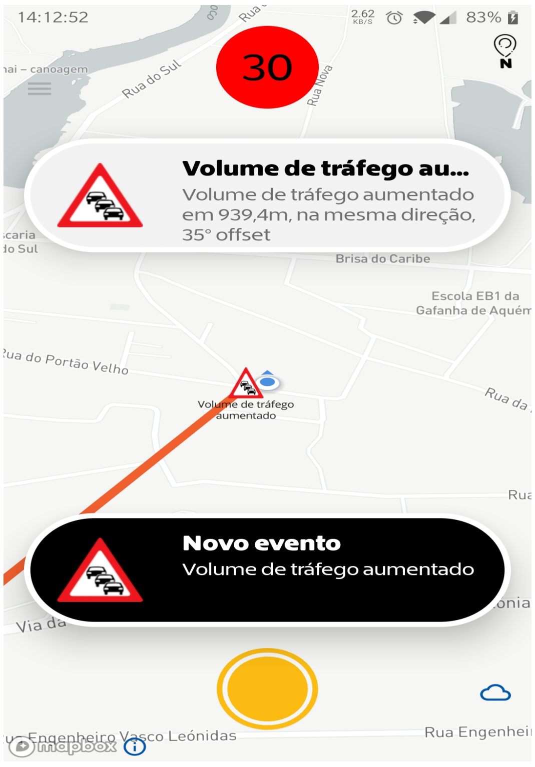
Various tests were carried out, in which the same event was launched in the same area but with different types of vehicles. The following image shows the applications' behavioral difference when an event is launched within a built-in area, with a speed limit of 40km/h for light goods vehicles and 30km/h for heavy goods vehicles.



Volume de tráfego au...
Volume de tráfego aumentado em 939,4m, na mesma direção, 38° offset

Novo evento
Volume de tráfego aumentado

**(a)** Speed advisor for light goods vehicle



Volume de tráfego au...
Volume de tráfego aumentado em 939,4m, na mesma direção, 35° offset

Novo evento
Volume de tráfego aumentado

**(b)** Speed advisor for heavy goods vehicle

**Figure 5.10:** Speed advisor for different types of vehicles

The major concern and limitation of this feature is the use of the internet. Since the algorithm needs to ask the Google API (Application Programming Interfaces) to inform it of the required speed limits, an internet connection is mandatory. Without an internet connection the application will not be able to display the speed limit.

Furthermore, the user has to give the permission to attain the smartphones' current location. In the unlikely event that no permission is given, the majority of the functionalities of the application and the algorithm will not work correctly, for the whole application revolves around the vehicle/smartphone's location.

## 5.6 Image classification

Outside the scope of the application, two image classification models were developed to automatically detect weather events, namely insufficient light and intense rain.

The usage and testing process is similar in both cases:

- The received videos are converted into frames.
- Each image is classified individually.
- The model assigns a final result to the video based on all classified images.

Several videos were collected, and used to test the developed models, in different meteorological conditions, such as rain, night time, daylight, sun, fog, amongst others. These samples were tested in both models. The final results of each one of these samples will be presented in the next section.

### 5.6.1 Luminosity Tests

For this model, the intention is to infer whether there is sufficient light on the road or not. In order to accomplish this, different videos were tested with different types of lighting. We will see the accuracy of the different tests and the confidence that the model has in each result obtained.

During the tests, we noticed that when videos were taken at night, sometimes the model did not distinguish between sufficient or insufficient light. The reason being that, at night, on some parts of the road there was sufficient light, due to streetlights, while on other parts light was insufficient; however, that does not mean that the area in question is of poor visibility.

Thus, in order to resolve this situation, it was decided that the model would only conclude, with certainty, that the light is insufficient in situations where more than 70% of the video had little light. Below, we can see two examples of videos that were collected at night:

- The following video was filmed at night, with the presence of streetlights, which does not result in insufficient light and visibility. The model, in this case, considers that only 56% of the video presents insufficient light. So, although it is night time, there are no visibility problems due to the luminosity on the road.

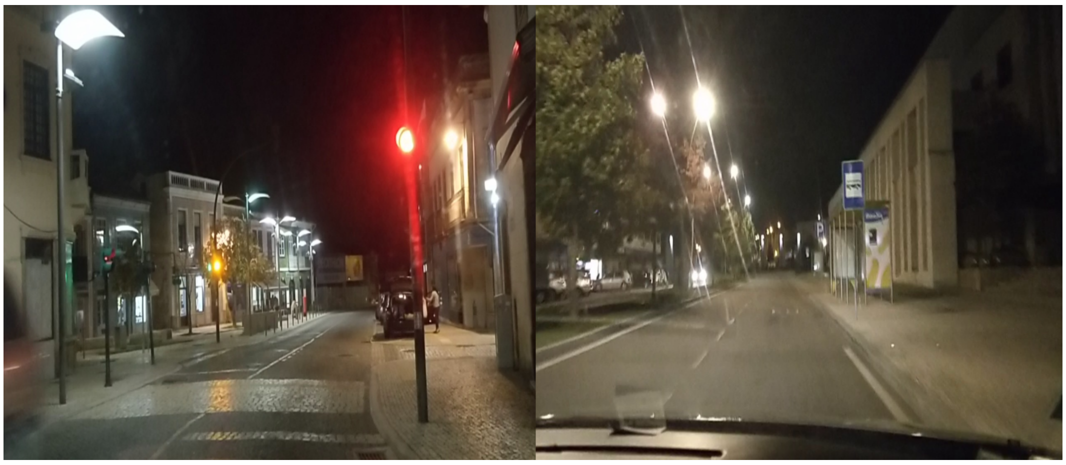


**Figure 5.11:** Night images with sufficient light

- In the following example most of the video was filmed on a very dark road, with no light at all. In this case the model considers that 94% of the video does not have enough luminosity, therefore the result obtained is insufficient light.

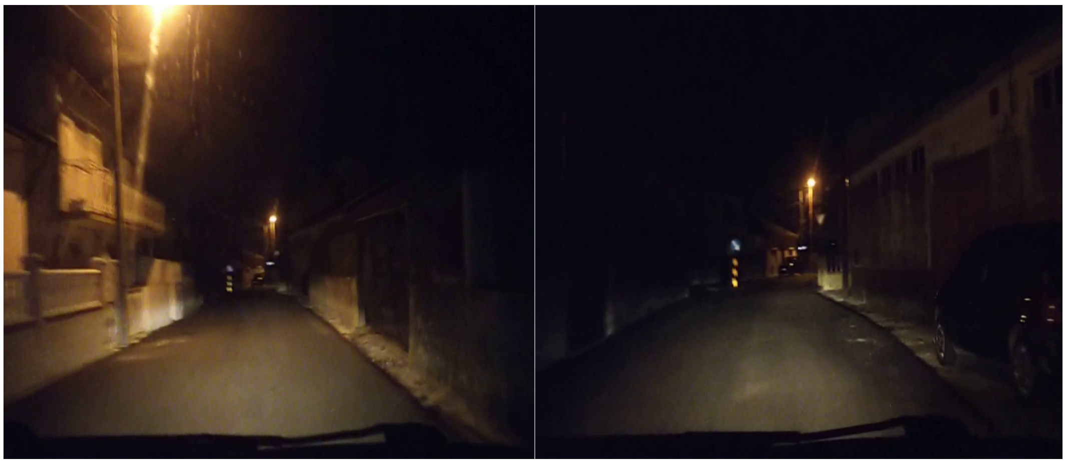


**Figure 5.12:** Night images with low visibility due to insufficient light

When the samples were taken during the day, the model had no difficulty in recognizing that it was daylight, posing no visibility problems. Furthermore, the model never confused the videos that collected data during the day or during the night.

**Figure 5.13:** Daylight images

At the end of several tests performed, we can conclude that the model has good classification results. Although the issue of poor visibility is subjective, for some may consider it to be quite dark, causing visibility problems, while others may not. Most importantly, the model clearly detects when the lighting is really limited and defines it as dangerous.

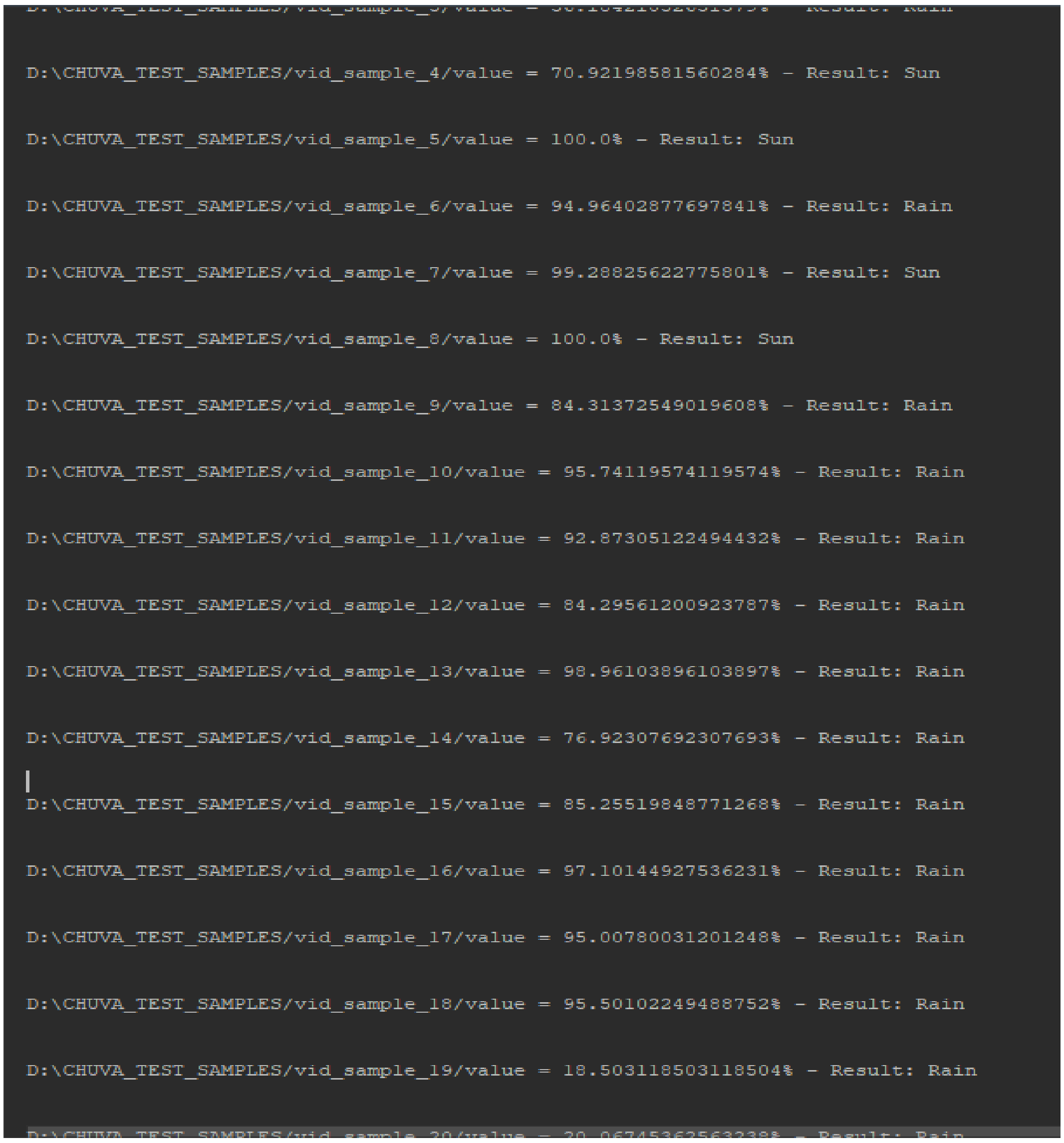
When there is a difference of opinion regarding the lighting other decisive factors are taken into consideration, but the model will undoubtedly be instrumental in the final result.

### 5.6.2 Rainfall Tests

The second model's goal is to identify rain situations, based on the images provided by the application, so that the system can generate alerts and disseminate this information as quickly as possible to drivers.

The process used to execute tests on the model was similar to the luminosity model: simulating the collection of images taken from the application and inserting it as input for the model to classify. In the end, we compared the results obtained with the real classification, along with the confidence of the answers presented by the model. Thus, for this case, videos were collected during the day, during the night, with or without rain.

In the following figure we can see the results of some of the tests that were made on the model and it is clear that, contrary to the luminosity tests, the confidence levels are considerably lower, almost never showing results with 100% confidence. However, this kind of behavior is expected for there are several factors such as road lights, blinding car headlights, traffic signs, road signs, amongst others, that can influence and confuse the model.



```
D:\CHUVA_TEST_SAMPLES/vid_sample_3/value = [illegible] - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_4/value = 70.92198581560284% - Result: Sun

D:\CHUVA_TEST_SAMPLES/vid_sample_5/value = 100.0% - Result: Sun

D:\CHUVA_TEST_SAMPLES/vid_sample_6/value = 94.96402877697841% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_7/value = 99.28825622775801% - Result: Sun

D:\CHUVA_TEST_SAMPLES/vid_sample_8/value = 100.0% - Result: Sun

D:\CHUVA_TEST_SAMPLES/vid_sample_9/value = 84.31372549019608% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_10/value = 95.74119574119574% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_11/value = 92.87305122494432% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_12/value = 84.29561200923787% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_13/value = 98.96103896103897% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_14/value = 76.92307692307693% - Result: Rain
|
D:\CHUVA_TEST_SAMPLES/vid_sample_15/value = 85.25519848771268% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_16/value = 97.10144927536231% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_17/value = 95.00780031201248% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_18/value = 95.50102249488752% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_19/value = 18.503118503118504% - Result: Rain

D:\CHUVA_TEST_SAMPLES/vid_sample_20/value = 20.06745362563238% - Result: Rain
```

Figure 5.14: Confidence of the classification results of the rainfall model

Based on the results of other samples, the model does not have difficulties in identifying, with great confidence, that there is no rain fall when it is sunny and the sky is clear. Mainly because it rarely happens to be raining and sunny at the same time, except perhaps when there are light showers that do not influence driving. The images below correspond to samples from "vid_sample_5"and "vid_sample_8", where the model showed 100% confidence that there would be no rain. This clearly shows that the model is not at all confused with these weather conditions, discarding any of the rainy day patterns obtained during the training phase.

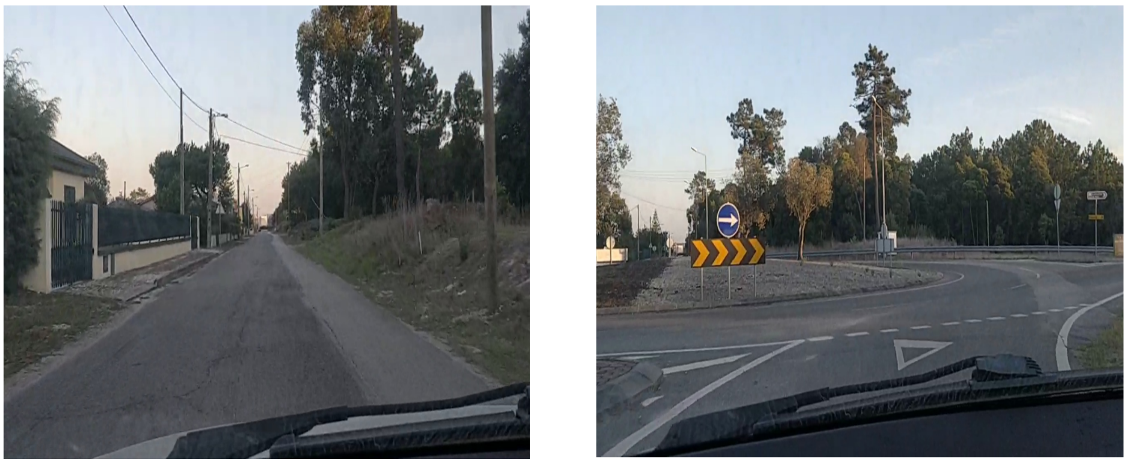

**(a)** Vid_sample_5 **(b)** Vid_sample_8

**Figure 5.15:** Testing samples on a sunny day, with no rain

When faced with night images, the model tends to become more confused and less accurate in its decisions. During the day, the model's confidence is between 80%-90%. Sometimes, on darker days, it drops to 20%-30%. It is normally more difficult for the model to learn more about night patterns because rainy days are generally darker, which confuses the model. The following images, although taken in different weather conditions, show some similarities, which makes the lower confidence level understandable.

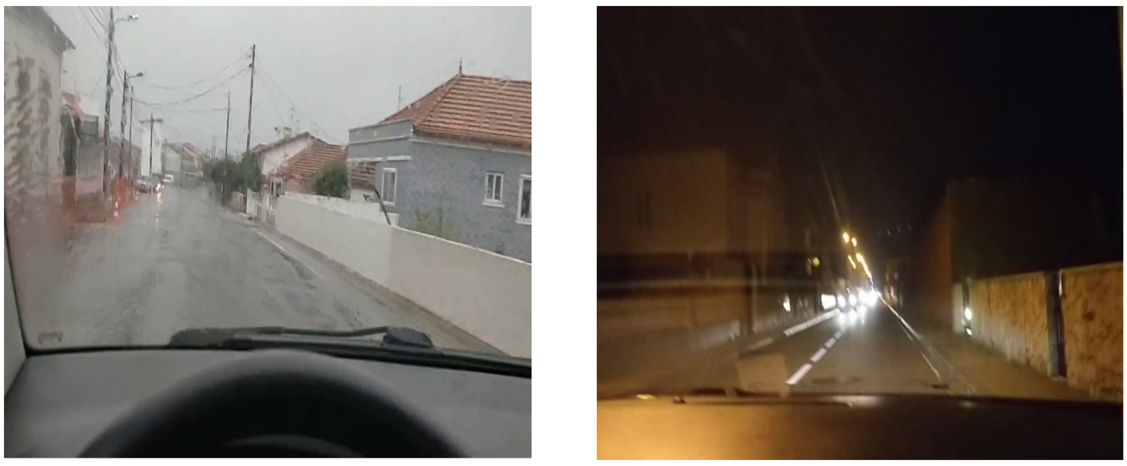

**(a)** Rainy day sample **(b)** Dark day sample

**Figure 5.16:** Testing samples with and without rain during the night

At times, during the night, the model has difficulty in identifying rain patterns, due to the image's low quality. Even humans have difficulty in determining whether it is raining or not when faced with this situation, as presented in the above figures. This is one of the major problems that this model currently faces, for it gets confused when it is confronted with situations in which there is a combination of insufficient light and low quality, producing at times imprecise classifications with a low level of confidence, making it difficult to obtain an accurate final result, contrary to what happens with daylight samples.

During testing, the samples that were obtained in daytime showed only one incorrect classification, due to the reasons already stated above. Since everything is analyzed through the windshield, the obstacles that are found on it can be decisive. In the following image, the windshield appears to be very dirty, so, when the sun hits it, the model confuses it with rain drops.

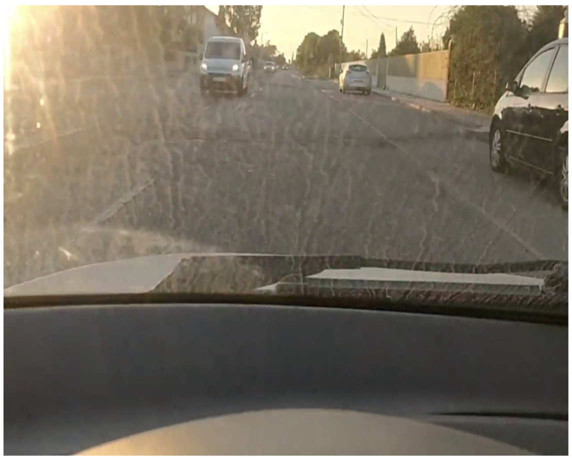
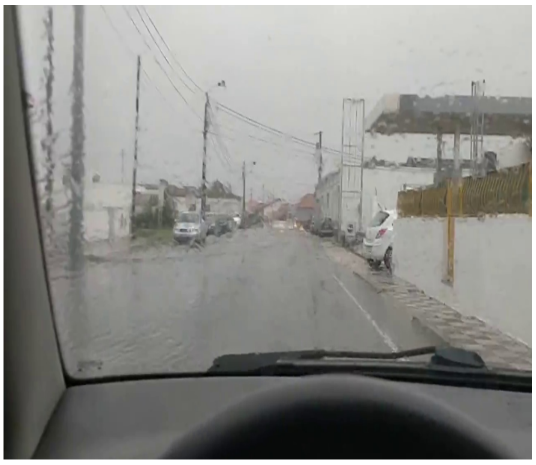


**Figure 5.17:** Day sample of a dirty windshield

The results improve exponentially when analyzing samples of heavy rainfall on the vehicle's windshield. In this case the model obtains results with great confidence. The raindrops falling down on the windshield combined with the continued use of the windshield wiper is another factor that helps the model to detect rain. Although the model hardly ever shows 100% confidence levels, it presents values between 85%-95%, which is a very comfortable result.

In the following images we can see that it is a gloomy day and that it is raining constantly, causing the driver to have blurred vision and engage in the constant use of the windshield wipers. When the model is faced with these patterns, it is clear that it is, in fact, a rainy day.


**(a)** Rainy day sample

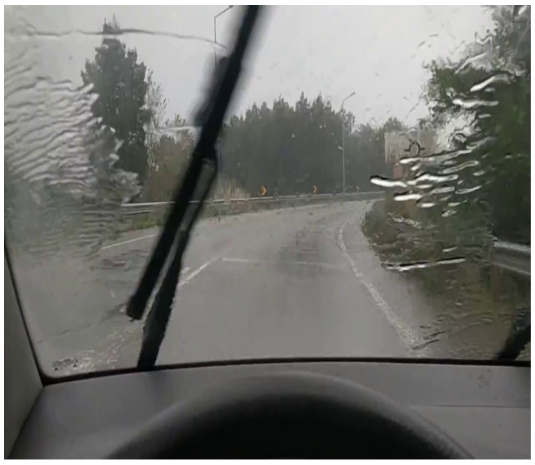

**(b)** Active windshield wipers on a rainy day sample

**Figure 5.18:** Testing samples collected on a rainy day

Based on the set of tests used, we can say that the model's behavior is normally good. Most of the time it is able to manage and distinguish, whether we are dealing with extreme weather conditions, a rainy day, or a sunny day, with more confidence. However, even though the model's classification was good, it is clear that more work is needed in order to improve some of its aspects, namely the night samples, where it is more difficult for the model to draw accurate conclusions.

In this particular case, in order to and improve the model, it is necessary that more information be collected at night, on rainy days, in order for the model to be able to study the patterns better, as well as keep the data sets balanced. However, we must be careful because introducing too many training variables may be a problem, for the model can become confused and analyze patterns that are not the main focus and that we do not wish it to analyze. We may, in fact, even risk ending up with a model that classifies luminosity instead of rain.

### 5.6.3 Application image prediction

The luminosity and rainfall models were also integrated into the application to process the information as quickly as possible. Although one of the goals of the image prediction application is to send the collected videos to other entities of the system so that they can be processed, it is faster for the application to collect the videos and process them immediately, giving instant feedback to the OBU.

So, the ideal behavior would be to launch the image classification process right after the video was collected, and to send an MQTT message to the OBU based on the results.

*Luminosity prediction*

In relation to the luminosity model, it was assessed that the video processing begin immediately after the video recording. It takes an average of 1200 milliseconds (1.2 seconds) for a video, with a duration of 10 seconds, to be received by the algorithm, and make its final prediction, which is naturally very positive. But all this depends on the smartphone's processing capacity.

In addition to the duration of this process, the assertiveness of the classification model was notorious and presented very positive results. Based on the tests performed, and after analyzing videos collected through the application, the model presented a final rating of 95%. Incorrect results were only attributed during nighttime situations when the roads' only light came from streetlights, on the verge of sufficient or insufficient luminosity. One of these examples is visible in the image below.

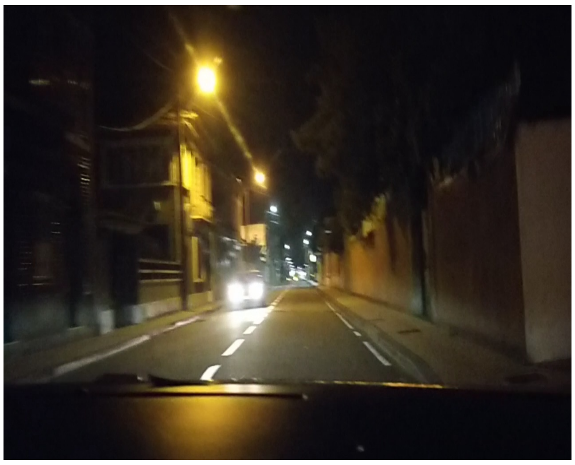


**Figure 5.19:** Misleading night sample

In the second part of the process, the message is formatted in Json, where the values collected by the sensor are attached, along with the timestamp and the exact location where the incident occurred. Afterwards the message is successfully sent by MQTT. We can see an example of a message sent from the application to the OBU, indicating insufficient luminosity.

```json
{
  "id_station": "STATION_ID_001",
  "timestamp": "1609960205058",
  "type": "luminosity_value",
  "result": "Insufficient luminosity",
  "result_sensor": "38.0",
  "lat": "40.6121297",
  "long": "-8.6896758"
}
```

**Code 3:** Luminosity reporting message

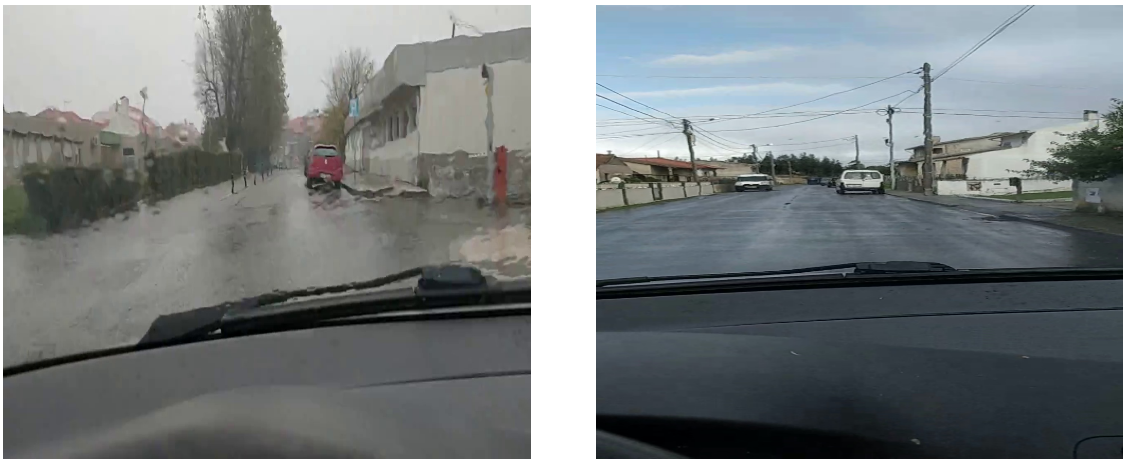
As we can see, the message presented was the result of the model prediction, that used the keywords "Insufficient luminosity", along with the value collected for the environmental luminance sensor, at that precise moment, 38.0 luxs.

When the model detected sufficient light, the application adopted the correct behavior, that is, ignoring and proceeding.

*Rain prediction*

The rain model also adopts a process very similar to the luminosity one. However, in comparison to the luminosity prediction, the rain prediction takes more time, because the classification of the TensorflowLite model takes on average 1700 milliseconds, or 1.7 seconds, due to the fact that it requires more processing power from the smartphone.

The results obtained are quite similar to those found in the Keras model, in which the level of efficiency is close to 85%, a little lower than the brightness luminosity prediction. As expected, the model has better results when it rains intensely and the weather is gray or when the day is clear and bright. However, when it is raining mildly the opposite happens, making it difficult for the model to generate an confident prediction, presenting an answer with less confidence. The two samples, in the following example, were tested on distinct days. On one of the samples it was raining intensely and on the other sample it was just sprinkling. The model was able to detect rain in both situations, but it had a lot of difficulty in seeing rain in image b) so it is natural that the model's response be less confident.



**(a)** Heavy rain sample **(b)** Light rain sample

**Figure 5.20:** Testing samples collected during distinct rain intensity

The main difficulty perceived is when the videos are collected during the night, which causes the model to become confused because of the insufficient light. So it ends up removing elements of comparison and features from the images, in which for example the raindrops are much thinner and not so notorious.

The second part of the process consists in the construction and sending of the json message which is done quickly and without any problem. Below, we can see an example of a message sent by the application to the OBU, in which we see the introduction of the timestamp and the coordinates of the event, along with a very clear message of what the model detected. However, the message would never be sent if the model had classified it as "not rain", consequently ignoring this result.

```
{
  "id_station": "STATION ID",
  "timestamp": "1609155423661",
  "type": "rain_prediciton_results",
  "result": "rain",
  "lat": "40.6121295",
  "long": "-8.6896699"
}
```

**Code 4:** Rainfall reporting message

When we take into account the processing of the videos and and predictions of the two models, the execution and sending time is between 2.5 and 4 seconds, which is acceptable and not very expensive, neither in terms of processing nor in terms of battery consumption. Since the regularity of the video capture and consequent prediction of conditions occurs every 2 minutes, there is a good balance between battery and processing consumption, data collection of road conditions, as well as formalized opinions of specialized models.

CHAPTER 6

# Conclusions and Future Work

## 6.1 Conclusions

Despite the increasing developments in road systems over the years, there are still problems due to the lack of information given to drivers, as well as to the methods in which the information is transmitted and the time consumption involved, which can make the difference in preventing accidents and saving lives. There are still too many road accidents, which unfortunately cause a lot of fatalities, so a urgent intervention is needed, because a large part of these accidents happen due to adverse road conditions, and namely due to meteorological issues, of which the drivers aren't informed beforehand, leaving them unprepared to act upon them.

Thus, to fill these gaps in traditional road systems, the TRUST project is developing a system to observe and monitor weather conditions and a mobile interface capable of providing information to drivers in near real-time, so that they never lack information about possible hazards in their surroundings. The project points to a connected and cooperative system in which vehicles and infrastructures communicate with each other, to combat the data sharing latency problems that currently occur, providing the drivers with the necessary information in advance. To make this possible, in addition to the constant monitoring of these conditions, using sensors installed in infrastructures, new services and tools are developed to determine and identify possible risk situations faster and more assertively.

The implementations, outlined as an objective and which have been integrated, concentrate the application on two very important aspects:

- First of all it monitors the weather conditions by collecting data of the luminosity, background videos, and by, automatically identifying dangerous phenomena such as poor visibility, due to the light, and adverse road conditions, due to rain, through image processing models that analyze the collected images;
- Secondly, the application also provides the user with a variety of events identified by other entities, which may pose a risk to the driver's safety. The list of types of events has been increased after a study on which are the best events to describe dangerous weather

situations. As most of the times, accidents happen due to speeding, the application also provides the recommended speed limit by taking into account the driver's location, the type of vehicle, and the existence of generated events related to driving conditions. In addition to the goal of reducing accidents, if they happen, the application provides information to the traffic control about the specific situation through a video stream service in real time. Finally, to keep the information as real as possible, a voice interaction service was implemented between the driver and the application, which promotes the verification of the veracity of the events generated by the system.

The extreme importance of developing tools, such as those proposed by TRUST, and of platforms such as the mobile application, are essential for road safety because with the increase in vehicles, there are more and more dangers, and a continuous supply of crucial information in real-time is necessary to prevent accidents and fatalities on the road.

## 6.2 Future Work

One of the main goals of this dissertation was to improve and complete the mobile application, which is being developed within the scope of the TRUST project, through the introduction of tools for monitoring adverse weather conditions and services to increase the driver's safety. However, it is necessary to improve the connectivity and cooperativeness between the vehicle and the rest of the road system, through new ideas such as:

- Implementing a process of sending videos between the application and the OBU quickly and without latency.
- Complementing and improving the voice interaction service between the application and the user with new ideas.
- Developing more models for automatic identification of other adverse weather or road conditions, such as fog or ice on the road.

These are extremely useful ideas, that should be taken into consideration in the follow up of the work developed, at a time when new tools are essential for road safety.

# References

[1] M. Choudhary, “What is intelligent transport system and how it works?”, *Geospatial World*, vol. 15, 2019.

[2] J. Almeida, J. Rufino, F. Cardoso, M. Gomes, and J. Ferreira, “Trust: Transportation and road monitoring system for ubiquitous real-time information services”, in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–7. DOI: 10.1109/VTC2020-Spring48590.2020.9129634.

[3] K. Aries. (2020). Connected vehicle technology (v2v, v2i, v2x), [Online]. Available: https://www.verizonconnect.com/resources/article/connected-vehicle-technology-v2v-v2i-v2x/.

[4] A. Festag. (2015). Standards for vehicular communication – from ieee 802.11p to 5g, [Online]. Available: https://festag-net.de/wp-content/uploads/2019/11/2015_from11p-to-5G_festag.pdf.

[5] 5GAA. (2018). Coexistence of c-v2x and its-g5 at 5.9ghz, [Online]. Available: https://5gaa.org/wp-content/uploads/2018/10/Position-Paper-ITG5.pdf.

[6] ETSI. (2010). Intelligent transport systems (its); european profile standard for the physical and medium access control layer of intelligent transport systems operating in the 5 ghz frequency band, [Online]. Available: https://www.etsi.org/deliver/etsi_es/202600_202699/202663/01.01.00_60/es_202663v010100p.pdf.

[7] GeeksforGeeks. (2020). Layers of osi model, [Online]. Available: https://www.geeksforgeeks.org/layers-of-osi-model.

[8] ETSI. (2010). Intelligent transport systems (its); vehicular communications; geonetworking; part 3: Network architecture, [Online]. Available: https://www.etsi.org/deliver/etsi_ts/102600_102699/10263603/01.01.01_60/ts_10263603v010101p.pdf.

[9] Q. Technologies. (2019). Its stack, [Online]. Available: https://www.qualcomm.com/media/documents/files/c-v2x-its-stack.pdf.

[10] E. T. S. Institute. (2019). En 302 637-2 - v1.4.1 - intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service, [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf.

[11] ETSI. (2019). En 302 637-3 - v1.3.1 - intelligent transport systems (its); vehicular communications; basic set of applications; part 3: Specifications of decentralized environmental notification basic service, [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.03.01_60/en_30263703v010301p.pdf.

[12] S. American. (2007). How does bluetooth work?, [Online]. Available: https://www.scientificamerican.com/article/experts-how-does-bluetooth-work.

[13] A. Attias. (). Bluetooth and wifi comparisons, [Online]. Available: https://www.streetdirectory.com/travel_guide/117214/technology/bluetooth_and_wifi_comparisons.html.

[14] E. Ferro and F. Potorti, “Bluetooth and wi-fi wireless protocols: A survey and a comparison”, *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[15] E. P. Cus. (). What is a wifi technology  how does it work?, [Online]. Available: https://www.elprocus.com/how-does-wifi-technology-work/.

[16] J. Jansons, E. Petersons, and N. Bogdanovs, “Wifi for vehicular communication systems”, in *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, 2013, pp. 425–430. DOI: 10.1109/WAINA.2013.17.

[17] E. Alecrim. (2019). O que é usb? (velocidades, conectores e versões), [Online]. Available: https://www.infowester.com/usb.php.

[18] E. Notes. (). Usb operation: Protocol, data transfer  packets, [Online]. Available: https://www.electronics-notes.com/articles/connectivity/usb-universal-serial-bus/protocol-data-transfer.php.

[19] Pplware. (2019). Mqtt: Protocolo de comunicação para pequenos dispositivos móveis, [Online]. Available: https://pplware.sapo.pt/tutoriais/networking/mqtt-protocolo-de-comunicacao.

[20] Firebase. (2020). Firebase realtime database, [Online]. Available: https://firebase.google.com/docs/database.

[21] Google. (2020). Save data in a local database using room, [Online]. Available: https://developer.android.com/training/data-storage/room.

[22] G. Developers. (2020). Sensors overview, [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview.

[23] WebRTC. (2019). Introdução ao webrtc, [Online]. Available: https://webrtc.org/getting-started/turn-server.

[24] Google. (2020). Real time communication with webrtc, [Online]. Available: https://codelabs.developers.google.com/codelabs/webrtc-web/#0.

[25] G. Roads. (2020). Speed limits, [Online]. Available: https://developers.google.com/maps/documentation/roads/speed-limits.

[26] N. Rai. (2020). Day-night classification, [Online]. Available: https://medium.com/@mneonizer/day-night-classification-a01a7d9af695.

[27] M. Vanga. (). Averaging rgb colors the right way, [Online]. Available: https://sighack.com/post/averaging-rgb-colors-the-right-way.

[28] Google. (2020). Machine learning glossary, [Online]. Available: https://developers.google.com/machine-learning/glossary/.

[29] MissingLink. (). Convolutional neural network tutorial: From basic to advanced, [Online]. Available: https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-tutorial-basic-advanced/.

[30] TensorFlow. (2020). The sequential model, [Online]. Available: https://www.tensorflow.org/guide/keras/sequential_model.

[31] M. Maynard-Reid. (2019). E2e tf.keras to tflite to android, [Online]. Available: https://margaretmz.medium.com/e2e-tfkeras-tflite-android-273acde6588.

[32] M. Yuan. (2017). Getting to know mqtt, [Online]. Available: https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot.

[33] ACEA. (2018). Frequency bands for v2x, [Online]. Available: https://www.acea.be/uploads/publications/ACEA_position_paper-Frequency_bands_for_V2X.pdf.

[34] FIWARE. (2019). Alert data model, [Online]. Available: https://fiware-datamodels.readthedocs.io/en/latest/Alert/doc/spec/index.html.

[35] G. Firebase. (2020). Firebase realtime database, [Online]. Available: https://firebase.google.com/docs/database.

[36] S. Maindola. (2020). Step by step guide to build webrtc native android app, [Online]. Available: https://medium.com/@shivammaindola07/step-by-step-guide-to-build-webrtc-native-android-app-47898caa1594.

[37] A. Yousef. (2019). Webrtc for android, [Online]. Available: https://amryousef.me/android-webrtc.

[38] R. Loza. (2016). Classify day and night images, Youtube, [Online]. Available: https://www.youtube.com/watch?v=dfcrYIu5LNo.

[39] Tensorflow. (2018). How to convert your ml model to tensorflow lite, [Online]. Available: https://www.youtube.com/watch?v=MZx1fhbL2q4.

[40] TensorFlow. (2019). Introducing convolutional neural networks (ml zero to hero - part 3), [Online]. Available: https://www.youtube.com/watch?v=x_VrgWTKkiM.

[41] V. Garg. (2020). Mobilenet model is deployed on android application., [Online]. Available: https://github.com/vasugargofficial/Image-Classification-Mobilenet-AndroidDemo.

[42] Vinicius. (2018). Modelo sequencial do keras, [Online]. Available: https://www.monolitonimbus.com.br/modelo-sequencial-do-keras.

[43] Pythonprogramming. (2018). Introduction to deep learning - deep learning basics with python, tensorflow and keras p.1-p.7, [Online]. Available: https://pythonprogramming.net/introduction-deep-learning-python-tensorflow-keras.