UNIVERSITY of RWANDA

Research and Postgraduate Studies (RPGS) Unit

DEVELOPMENT OF A REAL-TIME ROAD ACCIDENT DETECTION SYSTEM USING TINYML

"A Case study of Kigali City"

By: Joseph NTAMBARA

Reference number: 221027911

College of Science and Technology

School of ICT

Supervisors:

Dr. Didacienne MUKANYILIGIRA

Dr. Evariste TWAHIRWA

AUG 2024

ABSTRACT

The number of deaths from car accidents has gone up a lot lately. This trend is likely to continue, especially with more people and cars on the road. Delays in reporting accidents can lead to serious injuries if emergency vehicles and response teams are slow to reach the crash site. Various IoT systems have been developed in order to present a solution to the problem of delayed access to medical care due to the time it takes to alert response teams. One of the most important objectives in developing these solutions is to make sure that accidents are detected accurately to avoid false alerts and alarms while coming up with a solution. This research thesis focuses on the development of a real time road accident detection system using tiny machine learning (TinyML). This technique brings the power of machine learning to resource-constrained devices, enabling real-time, low-latency inference at the edge with minimal power consumption and reduced data transmission needs. Therefore, this research thesis utilized TinyML in order to develop a system that detects road accidents in real time. An image dataset was collected for three classes; Accident Detected, No_Accident_Detected and Other Objects.

Then we used MobileNetV1, a convolutional neural network (CNN) architecture with specific modifications. Input images were sized at 96x96 pixels, balancing efficiency and detail. A width multiplier of 0.2 reduced computational complexity for resource-constrained deployment, trading some accuracy. The final fully connected layer was removed as classification was unnecessary; only high-level feature extraction was required. A 10% dropout rate after the final convolutional layer regularized the model during training to prevent overfitting. This MobileNetV1 96x96 0.2 (no final dense layer, 0.1 dropout) configuration tailored the architecture for the research goals, emphasizing efficiency while removing extraneous components and incorporating regularization. The model was trained using the image dataset and we got an accuracy of 87.7% and an accuracy of 86.12% on new, unseen data. Finally, the model was deployed into Arduino Nano 33 BLE sense with on device performance of 431 milliseconds inferencing time, Peak RAM usage of 97.4KB and 220.9KB Flash usage. Future work will include an integration of a communication model to alert relevant authorities.

Keywords: Road Accidents, machine learning, TinyML, MobileNetV1, IoT, CNN

DECLARATION

I, Joseph NTAMBARA, with Reg. No: **22102791**, do hereby declare that all the work presented in this dissertation is my own original work unless otherwise acknowledged. It has never been presented either in part or in full for publication or award of a degree in any university.

I, therefore, present it for the award of Masters of Science in Information and Communication Technology (Option: Operational Communication) at the College of Science and Technology University of Rwanda, of, School of ICT.

Signature……………...

Joseph NTAMBARA

Date………………….

Supervisor's signature……………..

Dr. Didacienne MUKANYILIGIRA

Date:……………………………..

DEDICATION

I dedicate this work to my Supervisors Dr. Didacienne MUKANYILIGIRA, Dr. Evariste TWAHIRWA and to my lovely wife Mrs. Jane MUTETERI, I genuinely treasure each and every one of you.

God bless you all abundantly!

ACKNOWLEDGEMENT

I wish to acknowledge with profound gratitude my supervisors, Dr. Didacienne MUKANYILIGIRA for her invaluable skilful assistance and tireless support solidified towards my studies and particularly during the preparation and execution of this work.

My sincere acknowledgement extends to the entire leadership of the University of Rwanda, College of Science and Technology for offering me a possibility to carry out this research task.

I would like thank to most sincerely my classmates for the moral support and encouragement that they accorded to me during the programme.

I am indebted to my family for their love, patience and support rendered during the entire time of my studies.

Finally, I would want to express my gratitude to my government for their comprehension and assistance.

May God bless you all!

LIST OF ACRONYMS AND ABBREVIATIONS

CNN: Convolutional Neural Network

TinyML: Tiny Machine Leaning

MCU: Microcontroller

RAM: Random Access Memory

UR: University of Rwanda

RPGS: Research and Postgraduate Studies

LTE: Long Term Evolution

ICT: Information and Communication Technology

GPS: Global Positioning System

GSM: Global System for Mobile communication

SMS: Short Message Service

MHz: Mega-Hertz

IoT: Internet of Things

HIV/AIDS: Human Immunodeficiency Virus/ Acquired Immune Deficiency Syndrome

ITS: Intelligent Transportation System

WHO: World Health Organization

EU: European Union

NPN: Negative Positive Negative

LED: Light Emitting Diode

dB: decibel

DoD: Department of Défense

RF: Radio Frequency

PDA: Personal Digital Assistant

LCD: Liquid Crystal Display

LSI: Large Scale Integrator

ROM: Read Only Memory

VANET: Vehicular Ad Hoc Network

ML: Machine Learning

EEPROM: Electrically Erasable Programmable Read Only Memory

IP: Internet Protocol

Wi-Fi: Wireless Fidelity

API: Application Programming Interface

HTTP: Hyper Text Transfer Protocol

PC: Printed Circuit

USB: Universal Serial Bus

UART: Universal Asynchronous Receiver Transmitter

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Background and Motivation

Referring to [1], automobiles are an important part of our everyday life and wellbeing. They facilitate our travel to work, other places and even communication with loved ones and cargo transportation. The invention of automobile has had a positive impact in almost every aspect of our life and we can proudly say that it’s an invention we needed. However, it might also lead to disaster and loss of life in cases of road accidents. Speed is a key and dangerous part of driving because it not only makes crashes worse but also makes them more likely to happen [2].

According to the European Commission [3], more people are dying in traffic accidents around the world. The World Health Organization's "Global Status Report on Road Safety" says that in 2010 alone, 1.35 million people died in such accidents [4]. Because of this, car crashes now cause more deaths worldwide than AIDS, tuberculosis, or diarrheal diseases. Additionally, road accidents are the leading cause of death for young people aged 5 to 29 [5].

Injuries are a major global health issue, causing about 4.5 million deaths each year and leaving 650 million people with disabilities worldwide [6]. In low and middle-income countries, nearly 90% of these injuries and disability-adjusted life years occur. More than 75% of traffic accident deaths are male, with over 60% involving individuals aged 15 to 44 [5]. Between 1990 and 2013, traffic deaths increased globally, with the majority occurring in low and middle-income countries, and Africa having the highest rate of traffic fatalities [6].

The World Health Organization [7] reports that traffic injuries are a major yet often overlooked public health issue that requires coordinated efforts for effective and lasting prevention. World Health Organization notes that road traffic deaths are increasing. Although low- and middle-income countries have about 54% of the world's vehicles, they account for 90% of all traffic deaths. Most victims are vulnerable road users such as motorcyclists, cyclists, and pedestrians. Road accidents cost most countries 3% of their gross domestic product. If no progress is made, road traffic accidents are expected to become the seventh leading cause of death by 2030 [8].

Over the years, the automobile industry has created smart cars that are now part of the intelligent transportation system. This system uses sensors, communication, and information processing technology in vehicles to enhance the efficiency and safety of transportation [9]. By using this technology to develop smart cars, it can also help reduce death rates on the road.

To achieve new goals and targets in Europe, the European Road Safety Observatory's work on vehicle safety is essential. The EU has installed cameras on roads to reduce the number of serious and fatal injuries from traffic accidents. These cameras record vehicle movements, and if a traffic violation or accident occurs, details about the vehicle, which includes features like airbags, seat belts in all positions, and alcohol interlocks for fleet drivers, are sent to the police and the EU's cooperative system for integrated technology policies. After an incident, automatic crash notifications enable quicker access to emergency medical services [10]. Various countries like the United States, the United Kingdom, and Canada have launched government-funded programs as they embrace intelligent transportation systems (ITS) in the transportation sector to enhance safety after accidents by alerting rescue teams promptly [11].

The Rwandan government has made significant efforts to prevent injuries and deaths from road accidents by implementing intelligent technology. The Rwanda National Police actively manages road traffic through strategies like deploying police officers on highways, installing speed limiters, promoting safety helmets, emphasizing regular vehicle inspections, and launching the GERAYO AMAHORO road awareness program [12].

This research project addresses the pressing issue of increasing death rates on roads by introducing an innovative approach: an image-based TinyML road accident detection system. This system can be deployed in various locations such as streets, highways, and at traffic lights or robots. The incorporation of TinyML enables the system to analyse images in real-time, enhancing its ability to detect accidents accurately and efficiently. The developed System operates quickly with extremely low power and memory consumption on Internet of Things devices that are cost-effective and resource-constrained.

Moreover, future research will focus on expanding the system's functionality to include sending notifications directly to the police, alerting them about the accident and potentially reducing response times even further. This innovative approach aims to detect accidents but also streamline emergency response procedures for enhanced road safety.

1.2 Problem Statement

As more cars fill the roads, the number of accidents rises daily. According to research from the WHO [13], each year, 50 million people get hurt globally, with 1.4 million losing their lives. Most crashes happen because of human mistakes, like driving distracted or eating, drinking, or using a phone while driving. The main causes of death are either lack of medical help at the accident site or slow response times during rescue efforts. With the help of a TinyML based system, accident detection can be faster, potentially saving many lives with further integration of a communication model. The rise of smart cities is bringing attention to intelligent transportation systems (ITS) as a way to enhance traffic safety. The proposed TinyML-based system was designed to mimic human thought processes when detecting an accident by the use of Machine Learning.

Therefore, this research project was focused on the development of a TinyML based road accident detection system that can detect road accidents accurately and in real time.

1.3 Research Objectives

1.3.1 Main Objective

The main objective of this research work was to design and develop a real time accident detection system that uses TinyML technique by analysing images in real time. The main objective was achieved by measuring inference time, which is the time it takes for the system to capture an image, process and to classify accordingly.

1.3.2 Specific Objectives

- ✓ To collect image dataset for machine learning model training
- ✓ To train a model that will accurately classify images accordingly in real time and with high accuracy and performance
- ✓ To implement the hardware setup of the system
- ✓ To deploy the trained machine learning model into a low cost and power constrained device

1.3.3 Hypothesis/Research Questions

- ✓ How can we come up with an image dataset for road accident detection application?
- ✓ What's the most accurate and high-performance Transfer learning-based model that can be used to detect an accident with reference to the data collected?
- ✓ What is the most effective hardware setup and the components to be used in this case?
- ✓ Can a TinyML based system present an effective and efficient solution to the problem of road accident detection?

1.4 Study Scope

The main focus of this research was to apply TinyML in road accident detection systems and find out how fast and accurately it can detect an accident in real time. The primary objective of this project was to train a machine learning model that will classify images as **Accident_Detected** or **No_Accident_Detected**.

Finally, future work can include the integration of a communication or notification model in order to alert the police and relevant authorities.

1.5 Significance of Study

Developing real-time accident detection systems using TinyML holds profound significance in enhancing safety across various domains. By leveraging the power of TinyML, which enables machine learning models to run efficiently on resource-constrained devices, such systems can be deployed directly within vehicles, wearable devices, or even roadside infrastructure, allowing for instantaneous detection and response to potential accidents. This capability not only minimizes the response time of emergency services but also enables proactive measures to mitigate the severity of accidents, potentially saving lives and reducing the societal and economic costs associated with road accidents. Moreover, the localized processing inherent to TinyML facilitates privacy-preserving solutions by processing sensitive data directly on-device, thus addressing concerns regarding data privacy and security. Overall, the integration of TinyML into accident detection systems represents a significant step towards creating safer and more resilient transportation ecosystems.

1.6 Organization of Study

This section gives an overview of how the thesis was structured. Initially, the research was divided into six chapters: Chapter 1, Chapter 2, Chapter 3, Chapter 4, Chapter 5, and Chapter 6.

Chapter one covers the introduction, background, and motivation of the study, the problem statement, general and specific objectives, study scope, significance and organization

Chapter two reviews relevant literature and research on road accident detection systems, identifying gaps and contributions.

Chapter three outlines the research methodology and tools used.

Chapter four presents the System Design and Analysis, including the prototype and data sensor collection methods

Chapter Five details simulation outputs, prototype, and data analysis methods. Finally, **Chapter Six** concludes and offers recommendations based on the results for future research.

CHAPTER 2: LITERATURE REVIEW

2.1 Related Works

In [14], the authors developed a system aimed at detecting car accidents by analysing the behaviour of vehicles. Unlike other studies that focus on detecting objects that might cause accidents, their proposed system identifies when an accident has occurred by collecting and processing data from nearby vehicles. They utilized machine learning tools to differentiate between normal and abnormal driving behaviours, with the main objective being to monitor traffic patterns and flag vehicles exhibiting unusual movements as potential accidents.

Authors propose an Android-based application that monitors vehicles through an On-Board Diagnostics (OBD-II) interface to detect accidents [15]. The application estimates the gravitational force during frontal collisions and uses this information, along with airbag triggers, to identify accidents. Upon detection, it promptly sends accident details via email or SMS to pre-defined contacts and makes an automatic emergency call. Experimental tests on a real vehicle demonstrated the application’s ability to respond to accidents in less than three seconds, showcasing the potential of smartphone-based solutions to enhance road safety.

A system is designed to alert nearby medical centres about road accidents, particularly involving two-wheelers, to facilitate timely medical aid and potentially save lives in [16]. The system utilizes an accelerometer attached to the vehicle to detect its tilt and a heartbeat sensor worn by the user to monitor heart rate abnormalities, thereby assessing the severity of the accident. Upon detecting a serious accident, the system communicates this information via Bluetooth to a connected smartphone. An Android application on the smartphone then sends a text message with the exact location of the accident to the nearest medical centre and the user's friends, aiming to expedite medical assistance.

In [17], authors address the limitations of existing video anomaly detection methods for autonomous driving by developing an unsupervised approach to detect traffic accidents using dashboard-mounted camera videos. Traditional methods assume fixed cameras and static backgrounds, which is not suitable for vehicle-mounted cameras, and rely on one-class classification with hand-labelled datasets that only recognize trained anomalies. To overcome these issues, the authors propose a novel method that predicts the future locations of traffic participants and monitors the accuracy and consistency of these predictions using three different strategies. They evaluated their approach on a new diverse traffic accident dataset, An Accident Detection (A3D), and a publicly available dataset, demonstrating superior performance compared to the state-of-the-art. The code and dataset from this study are publicly available on GitHub.

A framework is proposed in [18] which utilizes Internet of Things (IoT) technology to enhance accident detection and notification. The system integrates smart sensors with a microcontroller in vehicles to detect collisions. When an accident occurs, the system uses GPS and GSM modules to send the location coordinates to pre-registered numbers and nearby ambulances, ensuring immediate notification and timely assistance. This approach aims to mitigate delays in emergency response and improve the chances of survival for accident victims.

In [19], authors aimed to enhance traffic safety by rapidly detecting traffic accidents using the eXtreme Gradient Boosting (XGBoost) machine learning technique. They utilized a comprehensive dataset from Chicago metropolitan expressways, collected between December 2016 and December 2017, which included information on 244 accidents and 6073 non-accident instances. The dataset comprised traffic, network, demographic, land use, and weather features. To interpret the model's results and assess feature importance, SHAP (SHapley Additive exPlanation) was employed. The XGBoost model demonstrated a high detection accuracy of 99%, a detection rate of 79%, and a false alarm rate of 0.16%. Key traffic-related features, particularly the speed difference before and after an accident, were found to significantly impact accident detection. Additionally, the study conducted a feature dependency analysis, examining relationships between average daily traffic and post-incident speed, distance to the Central Business District and residential density, and upstream versus downstream post-incident speeds.

Authors [20] address the problem of fatalities in road accidents due to delays in locating victims and notifying authorities. They highlight data from the National Crime Records Bureau showing a high number of accidents and traffic-related deaths in India, which are increasing annually. To mitigate these fatalities, the authors propose an efficient solution that integrates various sensors, including a GCP transceiver, a Raspberry Pi RFID receiver, and crash sensors, to detect accidents. A central server, equipped with detailed medical information and an auto-emailing module, facilitates the swift transmission of the victim's location and information to emergency contacts and authorities. This system leverages IoT to enhance the current vehicle alert system, aiming to reduce the response time and save lives.

Authors developed an IoT-based vehicle accident detection system utilizing GPS and sensors to improve emergency response and minimize loss of life and property in the event of road accidents in [21]. Despite advancements in vehicle design, road lane design, and traffic control, accidents still occur frequently. This system detects accidents and automatically sends alert messages containing the exact location (latitude and longitude) to the driver's family members, nearby police stations, hospitals, and emergency services via GSM. This embedded system aims to ensure timely medical assistance and notify relevant parties immediately, thereby reducing the negative impacts of accidents.

To quickly identify and respond to traffic accidents, many road traffic management systems employ a variety of autonomous event detection technologies. While many academic researchers have focused on improving VANET routing procedures using advanced algorithms, others have focused on automatic accident detection based on GPS/GSM technology, while still others have used smartphones to identify accidents [22].

For instances, the author [23] describes a solution that uses an Android application that may be used to notify accident victims through messaging. This program uses GPS technology for position mapping and furthermore delivers an accident notice. To ensure that the message's reputation increases, this produced alert message will be delivered to the surrounding registered individuals who are physically present in the accident area. Emergency services that are around the accident scene will get the notification depending on its reputation.

Also, according to [24], the researcher describes a method to prevent two-wheeler accidents that may be recognized and then instantly inform the local medical facilities to request assistance for medical assistance. In order to determine the severity of the collision, an accelerometer has been mounted to the car and a heartbeat sensor has been installed on the user's body. The accelerometer can feel the user's pulse. Additionally, the Android software on the user's smartphone notifies the user's relatives and friends about the accident for support. The location of the accident is also shared by the Android app, which is highly helpful for locating the sufferer.

The road accident was discovered by [25] using an unsupervised Traffic Accident Detection system based on visualization of images captured by cameras placed on the roads. They proposed system uses alteration between present trajectories and predict trajectories, they experimented their model by evaluating the existing accidents that happen and predict the accidents, and their system localized the location for anomaly issues in the case of vehicle deviation. Also, Shoney Sebastian [26] used an IOT-based system for car accident detection and notification. By combining sensors and the vehicle's microcontroller, they were able to quickly identify accidents and provide notifications. GSM delivers the message to specific numbers, including ambulance and using GPS to determine the location and coordinates of the accident site but the system was unable to determine the magnitude of the accident. The road accident and reporting system also has been reviewed by a researcher Kouros [27] where he used the Machine Learning (ML) technique in his research to sense the existence of accidents using a group of real-time data that contains information about road traffic, connections, demographics, terrestrial conditions, and meteorological conditions. The system employed upstream detectors and downstream detectors to calculate the average time spent by cars at the junction, which allowed it to detect the collision based on the average number of vehicles passing through the junction. In the paper [28], based on a Raspberry Pi, GSM and GPS modem, an accident alarm system was constructed. An accident is initially detected by a piezoelectric sensor, which then transmits its output to the microcontroller. A vehicle's latitude and longitudinal position are determined by the GPS. The vehicle's latitude and longitude positions are relayed as a message over the GSM network. The central emergency dispatch server's static IP address is pre-stored in the EEPROM. Every time a mishap occurs, the location is identified, and a message is delivered to the previously saved static IP address. An Automatic Vehicle Accident Detection and Messaging System using GPS and GSM Modems was also proposed by Sri Krishna [29] where the system of an AT89C52 microcontroller was employed. When the system is turned on, the LED turns on to show that the circuit has electricity. The IR sensors send an interrupt to the microcontroller when they detect an obstruction. The GPS returns the information after receiving the location of the accident-related car. A message containing this information is delivered to a cell phone number. The GSM modem built into the circuit is used to receive this message. The message includes the latitude and longitude values. These values can be used to approximate the vehicle's position.

Also referred to [30], there is a system that tracks an accident in real time but it does not give the act time the accident happened because of lack of machine learning technology.

With reference to the above related works, this research study focuses on improving the accuracy of accident detection by employing machine learning and TinyML. Most of the above-mentioned systems makes use of IoT a few utilise machine learning but their models are cloud based and not employing machine learning to give the better results. Therefore, by employing TinyML, the system performance is improved and low cost since it runs on a resource constrained device.

2.2 Research Conceptual Framework

This research was divided into steps according to the specified objectives and the outlined hypothesis. To achieve the first objective, an image dataset was collected that had two different scenarios; Accident and No Accident. The next step was to train a machine learning model that will classify image with good accuracy and performance. The developed model was then deployed onto a microcontroller that supports machine learning.

Finally, testing the model and analysing its performance in the real world was the last step. Figure 1 illustrates conceptual framework.

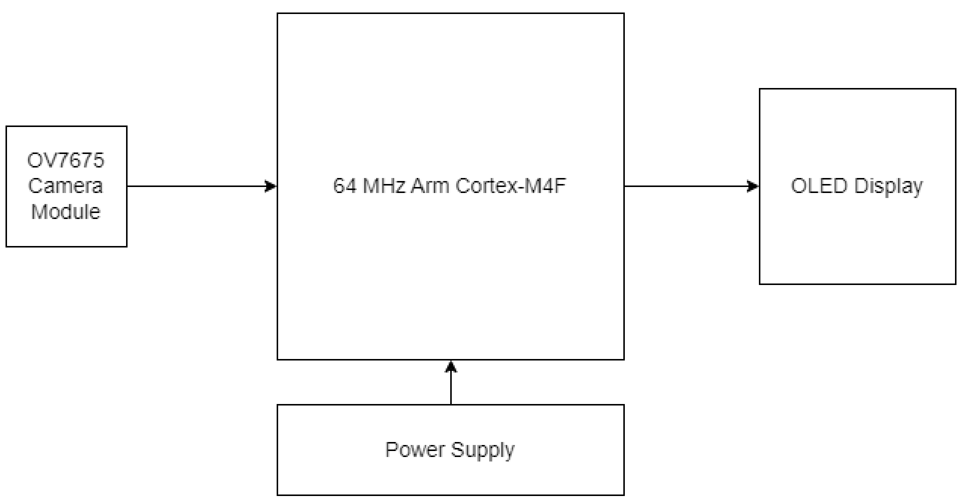

Figure 1: Research Conceptual Framework

CHAPTER 3: RESEARCH METHODOLOGY

This research methodology chapter provides a detailed explanation of the methods and procedures that were used to conduct the research. It will describe the research design, the approach and the rationale for choosing specific methods. It will outline data collection, model training, hardware implementation and model deployment.

3.1 Research Design

This project was divided into seven phases and completed using the Waterfall methodology. The Waterfall model is a sequential process that divides all work into segments to provide precision in problem-solving or system development. We used the Waterfall model in our system because it consists of seven phases, includes a mechanism for collecting user input, and allows for backtracking between steps. Based on our requirements, we could adjust our system at any stage in accordance with the equipment. In this process, each phase must be fully completed before moving on to the next. The project was then completed using the method outlined in Figure 2 below;

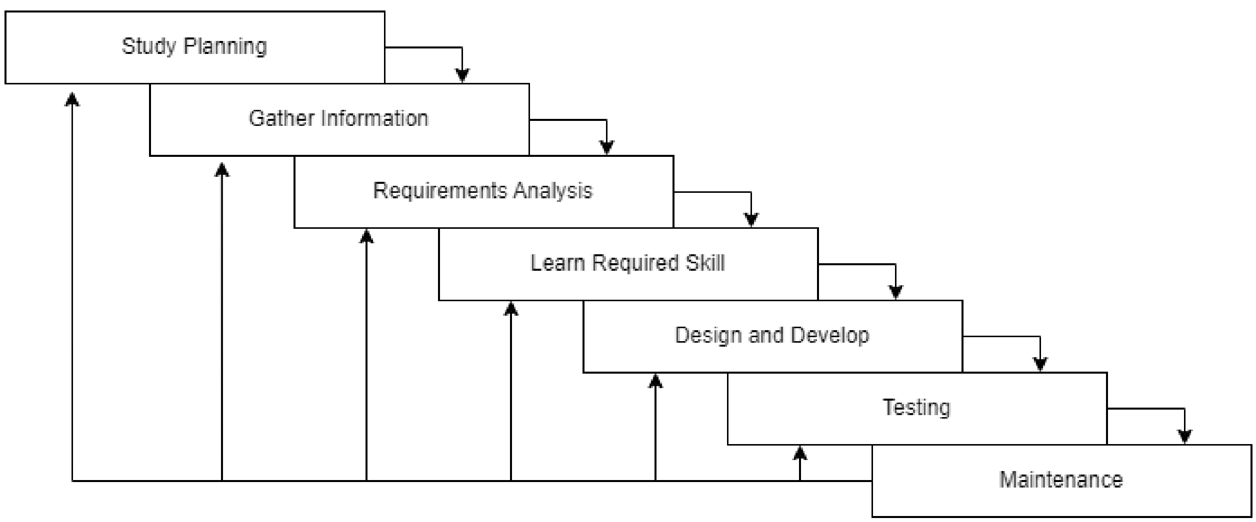

Figure 2: Research Design Flow

The work was completed in the following stages:

✓ Study Planning: The study identified issues in our day-to-day lives, discovered the issue, and devised a strategy to address it.

✓ Information Gathering: The research reviewed various publications on accident detection and used online searches to find solutions.

✓ Analysis of System Requirements: The study utilized an Arduino Nano 33 BLE Sense MCU, OLED display, and OV7670 camera Module.

✓ Learning Required Skills: The study involved learning AI in general, Machine Learning, Model Training, TinyML, and hardware connections necessary to complete the project.

✓ Design and Development: Involved Collecting data, model training and deployment.

✓ Testing: Model Testing and analysis

✓ Maintenance: This stage involved adjusting other parameters in the system to make it perform better.

The Figure 3 shows the System-level diagram of the system that was proposed with Arduino Nano 33 BLE Sense as the Microcontroller, OV7675 as the camera module for input and the OLED Display as an actuator for displaying inferences in real time.

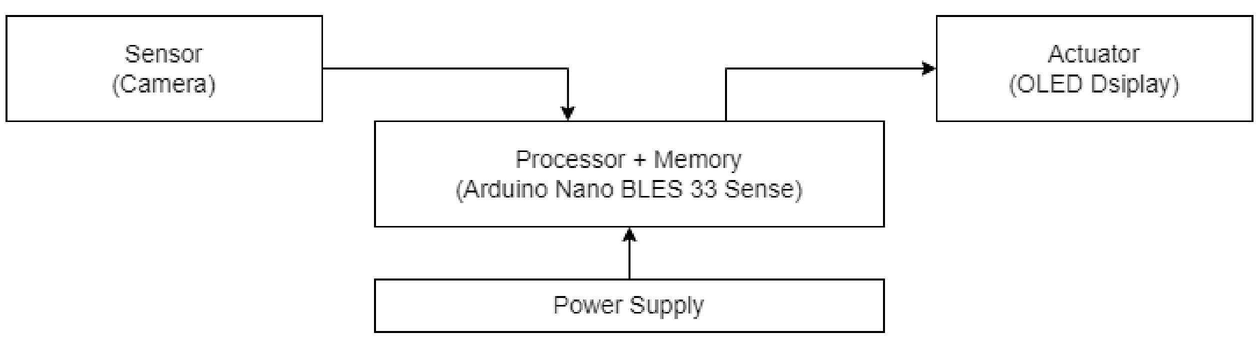

Figure 3: System-Level Design

Likewise, Figure 4 shows the flowchart of the system. Iniatially the system was configured to have detected other objects. Then the model uses the input from the camera to class either as Accident_Detected or No_Accident_Detected. If neither is detected, the system maintains the same output of the OLED Display.

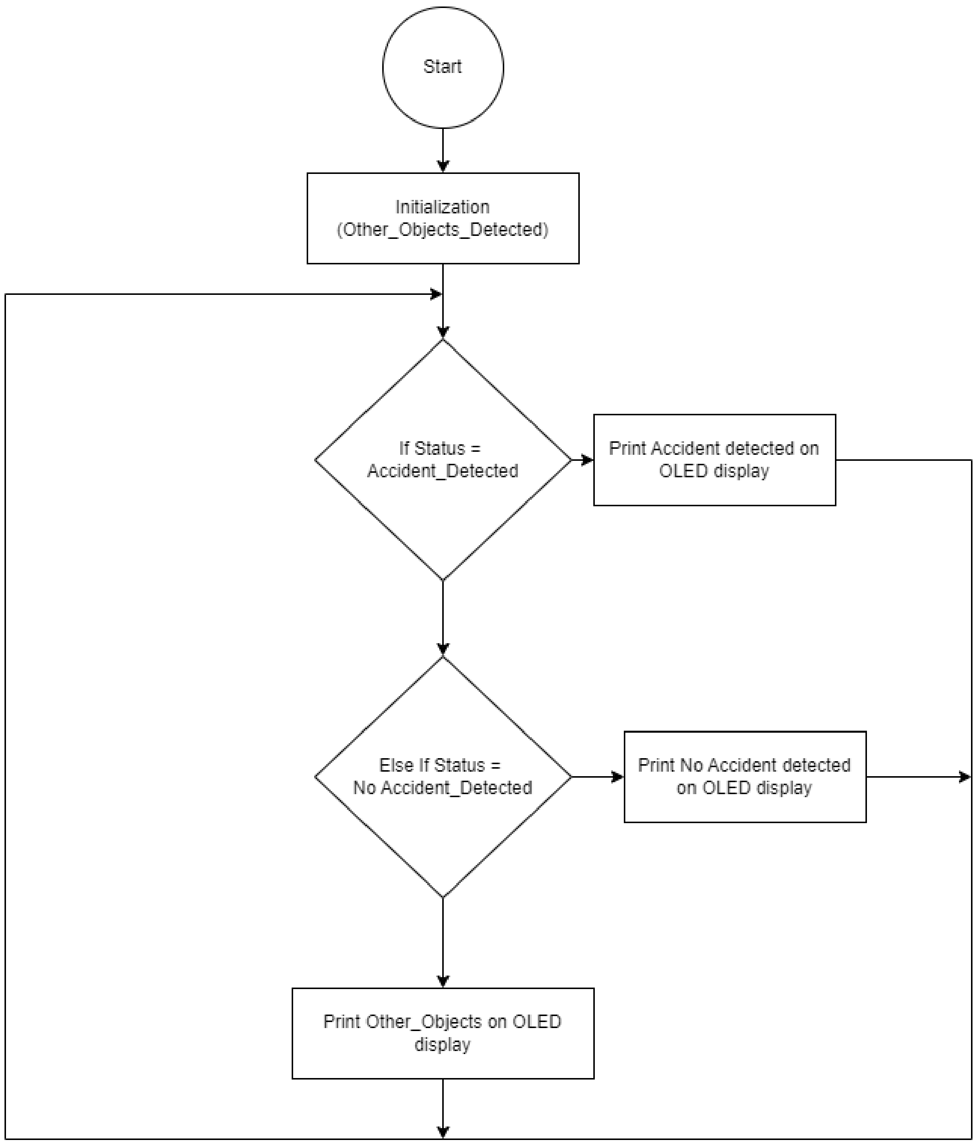

Figure 4: Flowchart

3.2 System Hardware

This section defines the specifications for the hardware components that were used to develop the prototype.

3.2.1 Arduino Nano 33 BLE Sense

The Arduino Nano 33 BLE Sense is perfect for beginners, makers, and professionals to delve into embedded machine learning [31]. This board, built on the nRF52840 microcontroller and operating on Arm® Mbed™ OS, supports Bluetooth® Low Energy connectivity and includes multiple sensors. The board features an LSM9DS1 inertial measurement unit for detecting orientation, motion, and vibrations, supports MicroPython for embedded systems, includes a digital microphone for real-time sound analysis, an APDS9960 sensor for proximity and gesture detection, an LPS22HB barometric pressure sensor for altitude calculation, and an HTS221 sensor for measuring humidity and temperature, all with comprehensive documentation and libraries. The Figure 5 below shows Arduino Nano 33 BLE Sense;

Figure 5 : Arduino Nano 33 BLE Sense [32]

It utilizes nRF52840 microcontroller with an operating voltage of 3.3V and supports an input voltage limit of 21V. It provides 14 digital input/output pins, all capable of PWM output, and features one UART, SPI, and I2C interface each for communication. The microcontroller operates at a clock speed of 64MHz and is equipped with 1MB of flash memory and 256KB of SRAM, though it does not include EEPROM storage. These specifications make it suitable for a wide range of embedded applications, offering robust connectivity and processing capabilities [32].

3.2.1 OV7675 Camera Module

The Arducam OV7675 camera supports resolutions up to 640x480 pixels, ensuring high-quality image capture [33]. Its versatile features enhance functionality across projects like home automation, smart monitoring, and Arduino-based applications requiring visual input. With a user-friendly interface, superior image quality, and compatibility with the Arduino Giga board, the Arducam OV7675 camera is an excellent choice for diverse applications relying on critical visual data. The Figure 6 below shows the OV7675 Camera module;

Figure 6: OV7675 Camera Module

The camera features a 640x480 active array with 2.5μm pixel size and operates with a 38 dB signal-to-noise ratio and a 71 dB dynamic range. It utilizes a 20-pin DVP interface with an electronic rolling shutter, Quad-Bayer RGB colour filter array, and offers RAW/YUV/RGB output formats at resolutions including 640x480, 320x240, and 160x120 at 15 frames per second. The optical characteristics include a 1/9 inch lens size with a 1.75mm effective focal length, F.NO of 2.8, and a 63.9° field of view angle. The camera supports a focus distance range from 0.12 meters to infinity, operates with power supplies AVDD: 2.7V to 3.0V, DOVDD: 1.7V to 3.0V, and DVDD: 1.5V, and functions within an operational temperature range of -30°C to 70°C, housed in a compact 30.5mm x 30.5mm board size.

3.2.3 OLED Display

The OLED displays are compact screens, either 1.12" with 96x96 pixels (V1.0) or 128x128 pixels (V2.0), offering 16 grayscale levels and high contrast through OLED technology [34]. They use SSD1327 (V1.0) or SH1107G (V2.1) drivers, communicating via 4-wire I2C for clock, data, power, and ground connections. Features include normal and inverse colour display, horizontal scrolling, and compatibility with the Grove interface, making them suitable for diverse applications needing efficient visual output.

The item LY120-96096 OLED display features an operating voltage of 3.3/5V, a 96x96 dot matrix with 16 grayscale display, utilizes the SSD1327Z driver chip, and operates within a temperature range of -40 to 70°C. The Figure 1.6 shows the diagram of a 1.12-inch OLED Display;

Figure 7: 1.12-inch OLED Display

3.3 Machine Learning and TinyML

Machine learning allows computers to mimic and adapt human-like behavior. With each interaction and action, the system learns and gains experience for future use [35]. There are several types of machine learning algorithms, including supervised, unsupervised, semi-supervised, and reinforcement learning. Additionally, deep learning, which is a subset of machine learning, excels at analysing large-scale data [36].

TinyML is a type of machine learning (ML) that works on small, low-cost devices with limited resources and power [26]. It aims to bring ML to these devices to make smart technology more widespread, using advances in the Internet of Things (IoT) and edge computing.

3.4 Research Approach (Qualitative Approach)

According to [37], quantitative research involves quantifying and analysing variables using numerical data and statistical tools to answer questions like who, how much, what, where, when, how many, and how. Research typically starts by selecting a sample because studying entire populations is often impractical, ineffective, or unethical. The aim of quantitative sampling is to select a representative sample, allowing the results to be generalized to the whole population. The best sampling strategy depends on the study's purpose [38]. A structured quantitative research approach involves defined variables, hypotheses, and design, enabling researchers to answer research questions and meet study objectives.

In recent years, there has been a significant global movement of people from rural to urban areas as part of the development process. This migration is driven by the search for job opportunities and improved quality of life, especially since rural environments often lack essential social amenities like clinics and schools [39]. This trend is particularly notable in developing countries. Kigali City, the capital of Rwanda, exemplifies this trend with a substantial increase in population from 603,049 people in 2002 to 1,203,725 in 2017, resulting in a high population density of 1,552 people per square kilometre [40] [41]. This rapid urbanization has led to a significant increase in both vehicular and pedestrian traffic, consequently causing more traffic accidents in Kigali [42]. As a response to this growing traffic, the Rwandan National Police has intensified efforts to enhance citizen safety and protect property, fostering a secure environment that supports economic growth and investment [42]. The study was conducted in Kigali City with the aim of reducing road fatalities.

As the major objective of the study which is to design and implement TinyML based Road Accident Detection system that will accurately detect accidents in real time. Further study will involve the inclusion of a communication model to alert relevant authorities as soon as the accident is detected.

3.5 Research Population

Population refers to the group of all items, subjects, or individuals that meet specific criteria. For comparative surveys, it is crucial to fully understand the target populations: 300 pedestrians, 30 motor drivers, 10 car drivers, and 5 police officers before conducting the survey. In this study, the population consisted of pedestrians, car drivers, motor drivers, and police officers located in Kigali City.

3.6 Research Data Collection Methods

The process of collecting data is crucial for statistical analysis. There are two main ways to gather data for research: primary data and secondary data [43]. Primary data is gathered directly by the researcher, while secondary data has already been collected or created by others. Primary data is original and specific, whereas secondary data involves analysis and interpretation of primary data. This distinction is the most significant between the two types. Secondary data serves different purposes from primary data, which is gathered to address current issues. In summary, primary data originates directly from the researcher, while secondary data comes from agencies and organizations conducting investigations.

Primary data, unlike secondary data which refers to historical information, is current and directly addresses the present issue [44]. Gathering primary data involves complex procedures aimed at solving specific problems, whereas secondary data is quickly and easily collected for purposes unrelated to the immediate issue. Examples of primary data sources include surveys, observations, experiments, questionnaires, and in-person interviews. In contrast, secondary data sources encompass official documents, books, journal articles, websites, and internal records [44].

Data collected during a study is called information. In this research, secondary data was utilized. To obtain this secondary data, the researcher consulted the National Institute of Statistics Rwanda

(NISR) and the Rwanda National Police (RNP) for existing data. The aim of the study is to identify the factors that contribute to the high number of deaths from car accidents in Kigali City.

3.7 Research Instruments

We used books, reports, and papers as sources for our research. These resources helped us build the theoretical foundation for our work, analyse data, and interpret results. In developing the physical device, we utilized Arduino Nano 33 BLE sense, OLED Display and Arducam module.

3.8 Research Data Analysis

The secondary data collected was a collection of images for Accident Detected and No Accident Detected scenarios. The image data set was downloaded from Kaggle [45] and after being cleaned, the machine learning model was developed using EdgeImpulse platform [[46].

CHAPTER 4: SYSTEM DESIGN AND ANALYSIS

4.1 System Design Prototype

The research project's system design is structured into two primary components: hardware design and software design. The subsequent section elaborates on each of these aspects.

4.1.1 Hardware Design

OV7675 Camera Module Pin	Arduino Nano 33 BLE sense Pin
VCC	3.3V
GND	GND
SIOC/SCL	SCL/A5
SIOD/SDA	SDA/A4
VSYNC/VS	D8
HREF/HS	A1
PCLK	A0
XCLK	D9
D7	D4
D6	D6
D5	D5
D4	D3
D3	D2
D2	D0/RX
D1	D1/TX

D0	D10
PEN/RST	A2
PWDN/PDN	A3

The OV7675 Camera module was connected to Arduino Nano 33 BLE sense with the following configuration shown in Table 1.0 below;

Table 1: Arduino Nano 33 BLE sense and OV7675 Camera Module interface Connection

The configuration was made like this following the already recommended interface connections for Arduino Nano 33 BLE sense and OV7675 Camera module [47].

Likewise, the OLED display was connected to the Arduino Nano 33 BLE sense using the shield as shown in Table 1.1 below;

OLED Display Pins	**Arduino Nano 33 BLE sense (Through the Shield)**
VCC	VCC
GND	GND
SDA	SDA
SCL	SCL

Table 2: OLED Display to Arduino Nano 33 BLE sense pin configuration

Finally, the whole hardware setup is shown in the Figure 1.7 below;

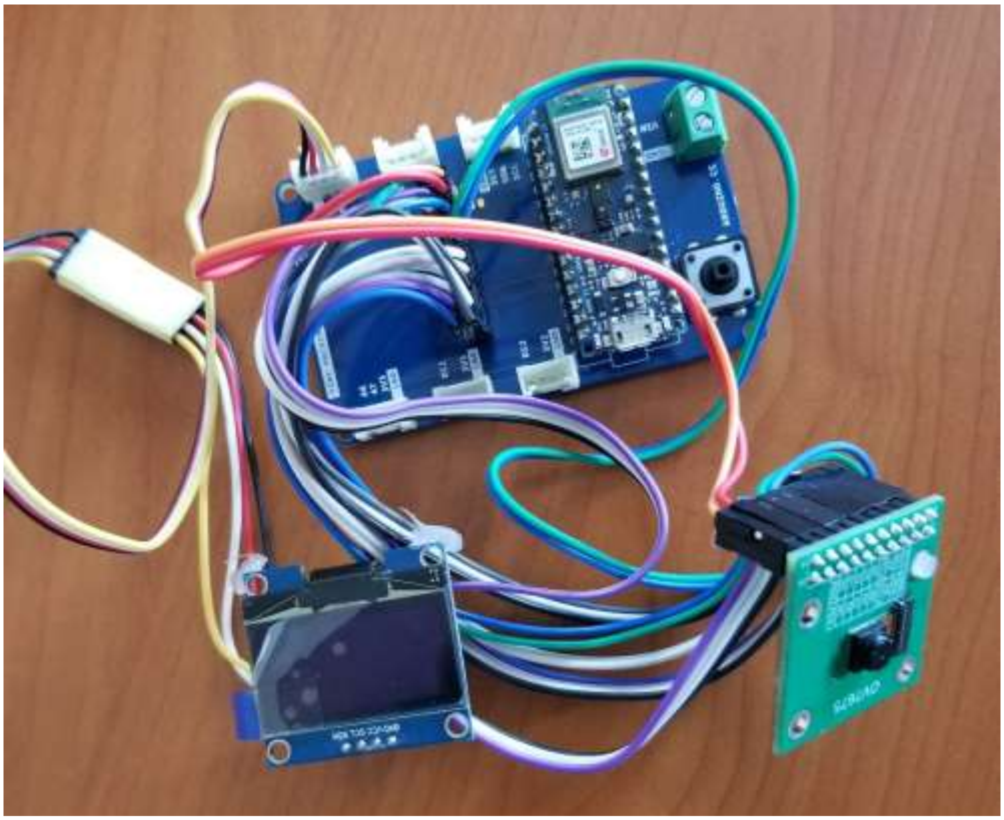

Figure 8: Hardware setup

4.1.2 Software Design

This section describes the machine learning model development, starting from data collection to final model deployment.

4.1.2.1 Data Collection

As pointed out earlier the image dataset used to develop the TinyML was secondary data downloaded from Kaggle.com. As such, these were two classes of images; Accident_Detected and

No_Accident_Detected scenarios. In addition, a third class was added to the collect which contained images of neither of the two. This class was labelled Other_Objects to mean that the model is not detected an Accident or anything related to Road activity involving cars and traffic. For example, if the camera is pointed towards a human being, it was classifying as Other_Objects. In general, this class was intended to include images of the cases where its neither Accident_Detected nor No_Accident_Detected.

4.1.2.2 Data Cleaning

This step involved cleaning the collected images to make sure in Accident_Detected class, there are no images that belong to No/-Accident_Detected class and vice versa. Not only that, but in Other_Objects there should not be any image that looks like the ones in Accident_Detected and No_Accident_Detected class and vice versa. However, we encountered a challenge of insufficient dataset and as such we had 875 images for Accident_Detected class, 870 images for No_Accident_Detected and 867 images for Other_Objects class.

4.1.2.3 Model Design and Training

Using EdgeImpulse platform, the dataset was loaded for processing and model development with the ratio shown in Figure 1.8. In general, we had 2,043 items, the training and Testing split was 80%: 20% ratio.

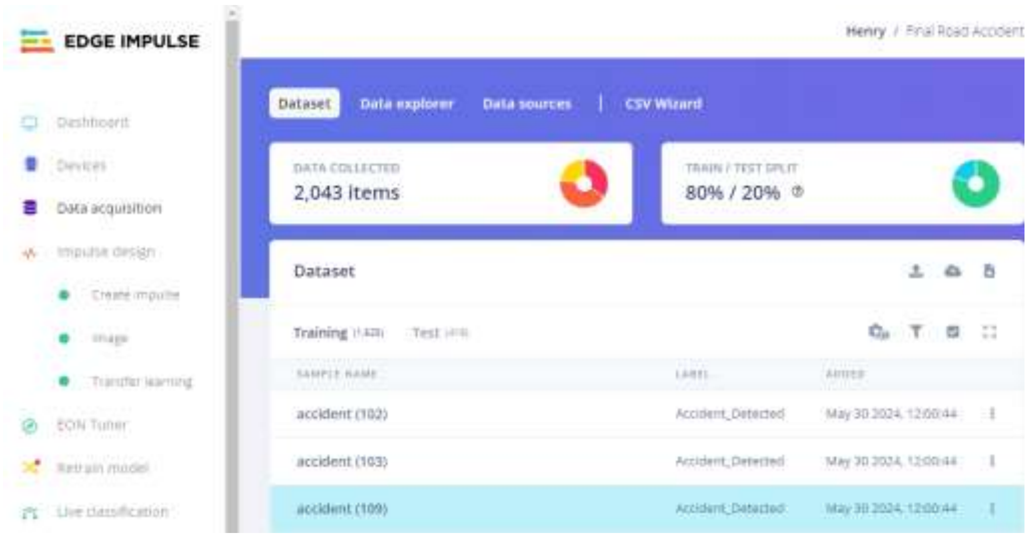

Figure 9: Image dataset uploaded in EdgeImpulse Platform

4.1.2.4 Impulse Design

After gathering data for our project, we formulated an Impulse which includes three essential components: an input block, a processing block, and a learning block. This stage was crucial as it allowed us to construct our own machine learning pipeline. The Impulse Design included the Image Processing Block, Transfer Learning as the Learning Block and finally, we had three Output features as; Accident_Detected, No_Accident_Detected and Other_Objects as shown in Figure 10.

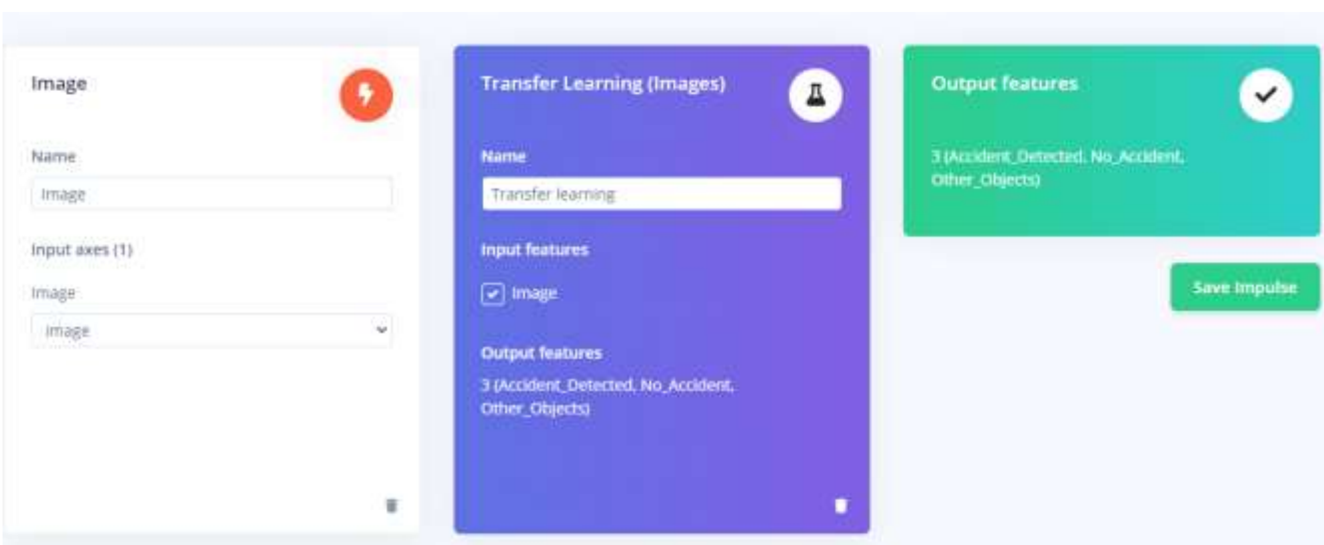

Figure 10: Impulse Design

Finally, the output of the processing block produced the features shown in Figure 11;

Figure 11: Feature Explorer

4.1.2.5 Model Training

The key settings included the number of training cycles set to 60, the option to use a learned optimizer, a learning rate of 0.0005, training on the CPU, and enabling data augmentation. The advanced settings included a validation set size of 20%, the ability to split the train/validation set based on a metadata key, a batch size of 16, the option to auto-weight classes, and the ability to profile the int8 mode.

For the neural network model developed, a pre-trained MobileNetV1 architecture was utilized, designed for efficient deployment on mobile devices and embedded systems. The input layer accepted inputs with 27,648 features. The core component was the MobileNetV1 96x96 model, which took 96x96 pixel images as input. It was configured with a dropout rate of 0.2 applied, but with no final dense layer included. This allowed transfer learning by repurposing the pre-trained weights while customizing the final output layers. For the output, 3 classes were specified, indicating this was a multi-class classification task where the model predicted one of three possible output categories for each input sample.

Finally, the model developed was tested and deployed into the hardware developed for testing.

CHAPTER 5: RESULTS AND ANALYSIS

5.1 Model Results Analysis

After training, the results in Figure 12 were obtained;

Figure 12: Training Output

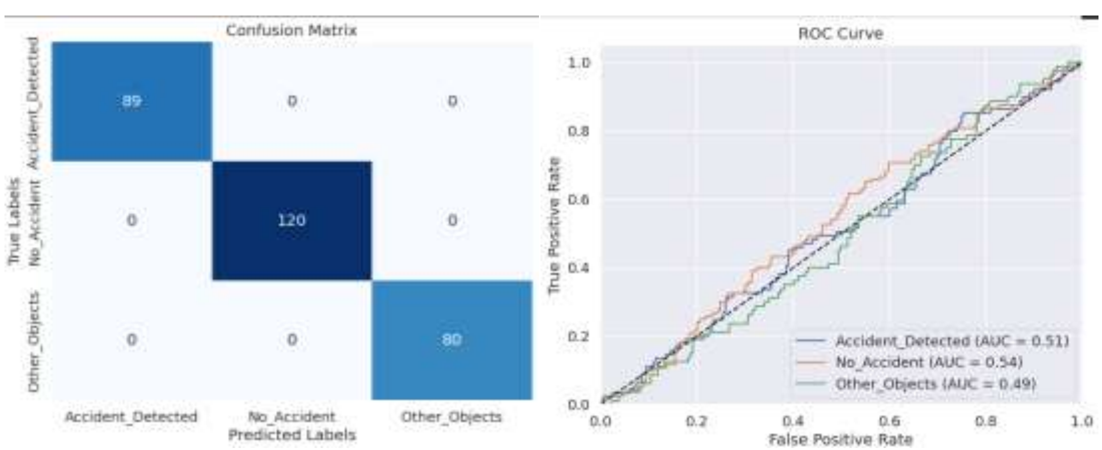

Figure 13: Confusion Matrix and ROC Curve plots

The models evaluated using float32 and int8 precision demonstrate robust performance across multiple metrics. Starting with the float32 model, it exhibits strong accuracy (0.9043 on the test set) and ROC AUC (0.9744), indicating excellent overall predictive capability and class separation. In terms of precision and recall, it achieves notable scores across all classes: for "Accident_Detected" (precision: 0.8867, recall: 0.8693, F1-score: 0.8779), "No_Accident" (precision: 0.9143, recall: 0.9040, F1-score: 0.9091), and "Other_Objects" (precision: 0.9140, recall: 0.9659, F1-score: 0.9392). The confusion matrix further illustrates its ability to correctly classify instances, with balanced support across classes.

Comparatively, the int8 model shows slightly lower accuracy (0.8493) and ROC AUC (0.9707) on the test set, yet still maintains competitive precision and recall for "Accident_Detected" (precision: 0.7513, recall: 0.9281, F1-score: 0.8304), "No_Accident" (precision: 0.9379, recall: 0.7684, F1-score: 0.8447), and "Other_Objects" (precision: 0.9167, recall: 0.8750, F1-score: 0.8953). Its confusion matrix indicates comparable classification performance but with a noticeable decrease in precision for "No_Accident." Both models demonstrate strong capabilities, with the float32 precision model offering marginally better overall performance metrics.

The overall performance of the model is shown in Figure 2.3, with an Inference time of 431ms, Peak RAM of 97.4KB and Flash Memory of 220.9KB;

Figure 14: Model On-device Performance

5.2 Model Testing

Using the 20% that was assigned for testing, results were obtained as shown in Figure 15 below;

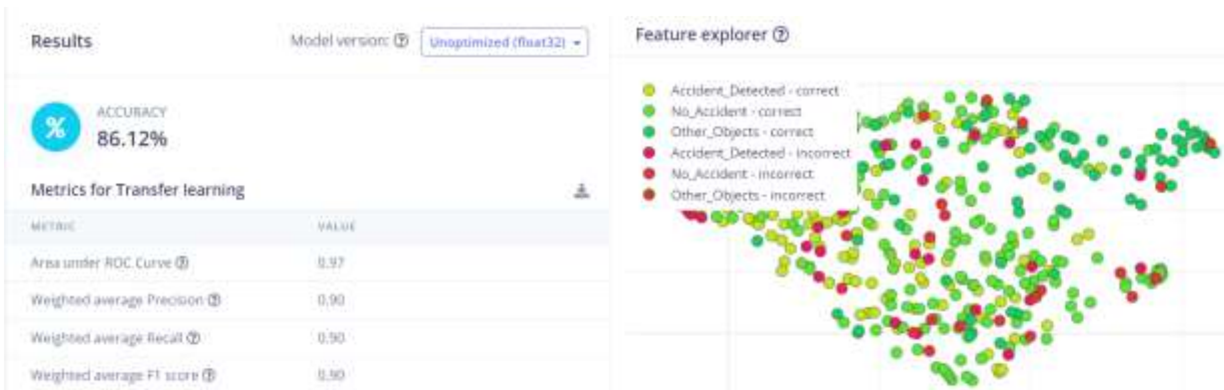

Figure 15: Testing output results

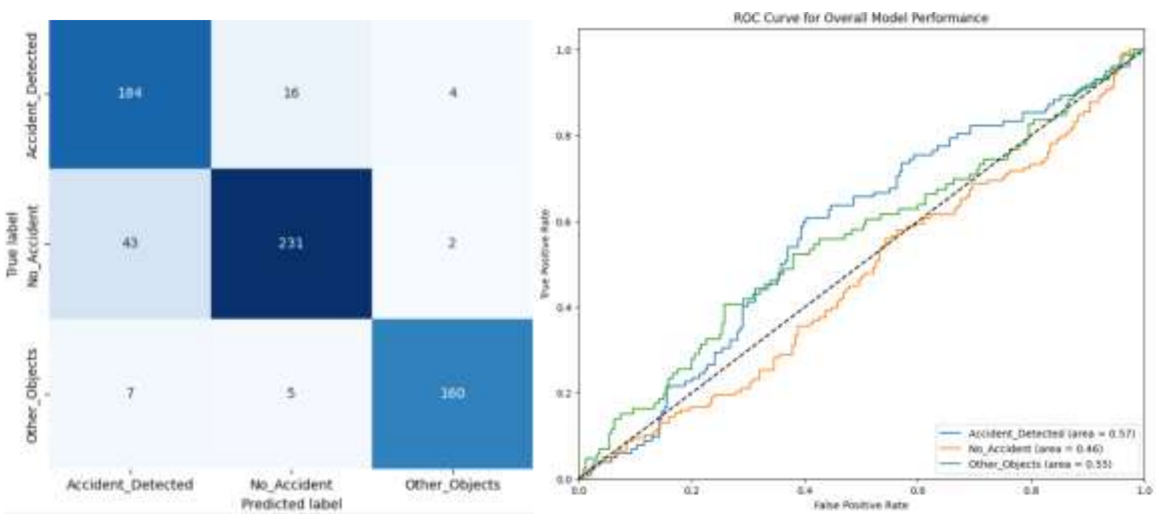

Figure 16: Confusion Matrix and ROC Curve for testing results

Overall, the model demonstrates strong performance across multiple evaluation metrics, including accuracy, precision, recall, and F1-score, indicating its effectiveness in predicting the classes of interest—'Accident_Detected,' 'No_Accident,' and 'Other_Objects.' With an average accuracy of approximately 89% on both validation and test sets, the model exhibits robustness in its predictive capabilities. Furthermore, achieving weighted F1-scores above 88% underscores its ability to balance precision and recall across all classes, crucial for reliable classification. The high ROC AUC scores, consistently above 97%, further validate the model's capability to distinguish between classes effectively. These results collectively highlight the model's strong performance and reliability in classifying instances, making it well-suited for practical applications requiring accurate and consistent predictions across diverse scenarios.

Finally, the model was deployed and live classification is shown in the Figure 2.4 below;

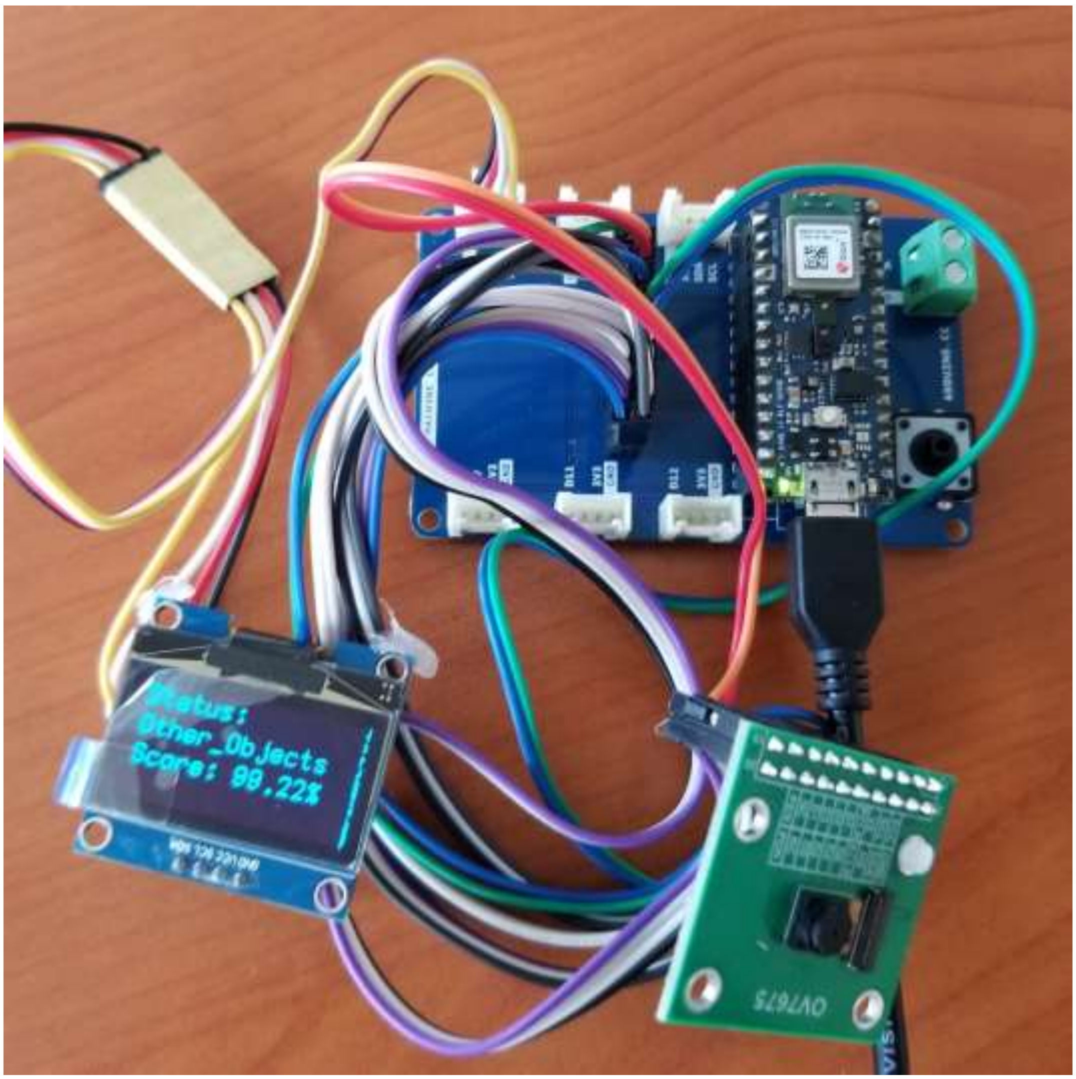

Figure 17: Model Testing showing other objects detected

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The need for a project focused on detecting accidents using image datasets stems from the critical importance of timely and accurate accident detection in enhancing road safety and emergency response. Traditional methods often rely on manual reporting or outdated technology, which can delay the notification of incidents and deployment of emergency services. By leveraging advanced image recognition techniques, this project aims to automate the detection process, significantly reducing response times and potentially saving lives. In areas with high traffic volumes or remote locations where accidents might not be quickly noticed, an automated system can provide real-time alerts, ensuring quicker intervention and reducing the severity of outcomes for those involved.

Implementing this project using TinyML (Tiny Machine Learning) brings substantial benefits, particularly in terms of resource efficiency and deployment flexibility. TinyML enables the use of machine learning models on low-power, resource-constrained devices like the Arduino Nano 33 BLE Sense. This makes it possible to deploy intelligent accident detection systems directly at the edge, closer to the source of data collection, thereby minimizing latency and reliance on constant internet connectivity. The compact size and low power consumption of these devices allow for widespread and unobtrusive installation in vehicles, traffic cameras, or roadside infrastructure. Overall, TinyML not only makes advanced accident detection feasible and cost-effective but also enhances its practicality and scalability, making our roads safer and smarter.

Considering the deployment constraints of using int8 precision for the Arduino Nano 33 BLE sense, the model's performance is respectable but shows some trade-offs compared to the float32 precision model. The int8 model achieves decent accuracy (0.8493 on the test set) and ROC AUC (0.9707), demonstrating its ability to maintain effective class separation and prediction capabilities within the constraints of reduced precision. However, there is a noticeable decrease in precision, particularly for the "No_Accident" class, which may affect applications where precise identification of this class is critical. Despite these limitations, the int8 model still performs adequately across all classes, with strong recall for "Accident_Detected" and balanced support in the confusion matrix, making it a viable choice for deployment on resource-constrained platforms like the Arduino Nano 33 BLE sense, where computational efficiency and memory constraints outweigh minor reductions in performance metrics compared to higher precision models

The on-device performance of the TinyML based system is 431ms Inferencing time, 97.4KB Peak RAM usage and 220.9KB Flash usage.

6.2 Recommendation/Future Work

In our accident detection project using image datasets, the limited size of our dataset significantly impacted the model's accuracy, presenting a challenge in achieving optimal performance. Small dataset can limit the model's ability to generalize and effectively learn the distinguishing features of different classes, leading to reduced accuracy and higher susceptibility to overfitting. For future work, we recommend getting more image dataset and it will help to improve the accuracy and performance.

Additionally, the current phase of our project does not include an interfacing communication model, which is crucial for real-time deployment and alerting mechanisms. For future work, we recommend expanding the dataset to improve model robustness and incorporating a communication interface. This would facilitate real-time transmission of detected incidents to emergency services or relevant authorities, enhancing the system's practical utility and impact on road safety. Integrating technologies such as wireless communication modules or IoT frameworks can ensure seamless and timely data exchange, thereby improving overall system efficacy

REFERENCES

[1] A. Yau, “Accident Detection System with GPS, GSM, and Buzzer,” *Journal of Mathematical Techniques and Computational Mathematics,* 2023.

[2] Aithor, “Aithor,” Aithor, 14 4 2024. [Online]. Available: https://aithor.com/essay-examples/the-evolution-of-the-automobile-its-effects-on-society. [Accessed 19 8 2024].

[3] E. Commision, “EU Road Safety Policy Framework 2021-2030 - Next steps towards "Vision Zero",” *Brussels,* 2019.

[4] Peden, “Research gate,” Research gate, 30 7 2010. [Online]. Available: https://www.researchgate.net/publication/362131153_Global_Road_Safety_2010-18_An_Analysis_of_Global_Status_Reports. [Accessed 19 8 2024].

[5] WHO, “WHO,” WHO, 13 12 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries. [Accessed 19 8 2024].

[6] E. Krebs, “The epidemiology of road traffic injury hotspots in Kigali, Rwanda from police data,” 2016.

[7] WHO, “World report on road traffic injury prevention,” 2004.

[8] WHO, “Road safety development,” *Global status report on road safety,* 2019.

[9] Guerrero-Ibáñez, “Sensor technologies for intelligent transportation systems.,” *Sensors (Switzerland),* 2018.

[10] Kae, “ The Potential for Vehicle Safety Standards to Prevent Deaths and Injuries in Latin America: An Assessment of the Societal and Economic Impact of Inaction,” 2016.

[11] Isaiah, “ Role of Intelligence Transport System in the Fight against Road Accidents in Kenya,” *International Journal of Computer Applications (0975 – 8887),* 2019.

[12] K. City, “Smart City Rwanda Masterplan 2019.,” 10 20 2019. [Online]. Available: https://unhabitat.org/sites/default/files/documents/2019-05/rwanda_smart_citymaster_plan.pdf. [Accessed 12 5 2023].

[13] WHO, “Global Status Report on Road Safety,” *Geneva,* 2018.

[14] Basit, “Road Accident Detection Using Machine Learning, 2021,” *International Conference on System, Computation, Automation and Networking (ICSCAN), , Puducherry, India,* 2021.

[15] Manzoni, “Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones,” *IEEE 36th Conference on Local Computer Networks, Bonn, Germany,* 2011.

[16] Haridas, “Intelligent accident detection and alert system for emergency medical assistance,” *International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India,* 2017.

[17] Crandall, “Unsupervised Traffic Accident Detection in First-Person Videos,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China,* 2019.

[18] S. Sebastian, “IoT based car accident detection and notification algorithm for general road accidents,” *International Journal of Electrical & Computer Engineering,* 2019.

[19] A. Bahador, “Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis.,” *Accident Analysis & Prevention,* 2020.

[20] Yellamma, “Automatic vehicle alert and accident detection system based on cloud using IoT. Embracing Machines and Humanity Through Cognitive Computing and IoT.,” *Singapore: Springer Nature Singapore,* 2023.

[21] Singh, “ An IOT Based Accident Reporting and Tracking System.,” *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY,* 2020.

[22] Manzoni, “Providing accident detection in vehicular networks through OBD-II devices and android-based smartphones,” *Local Comput. Networks,* 2011.

[23] Pamila, “Accident Alert System Application Using a Privacy-Preserving Blockchain-Based Incentive Mechanism,” *5th International Conference on Advanced Computing & Communication Systems (ICACCS),* 2019.

[24] George, “Intelligent accident detection and alert system for emergency medical assistance,” *International Conference on Computer Communication and Informatics (ICCCI),* 2017.

[25] Crandall, “Unsupervised Traffic Accident Detection in First-Person Videos,” *IEEE International Conference Intelligence Robot,* 2019.

[26] Sebastian, “IoT based car accident detection and notification algorithm for general road accidents,” 2019.

[27] Kouros, “Toward safer highways , application of XGBoost and SHAP for real-time accident detection and feature analysis,” 2019.

[28] A. K. Yadav, “Execution of Accident Vehicle Tracking System and A keen application based following framework,” 2017.

[29] Sri, “Plan of Vehicle situating System Based on ARM,” *Department of Physics and Electronic Information Engineering,* 2010.

[30] Goyal, “Research gate,” Research gate, 1 January 2022. [Online]. Available: https://www.researchgate.net/publication/353584281_Realtime_Accident_Detection_and_Alarm_Generation_System_Over_IoT. [Accessed 23 8 2024].

[31] Arduino, “Arduino nano 33 BLE Sense.,” Arduino, 2 7 2017. [Online]. Available: https://docs.arduino.cc/hardware/nano-33-ble-sense/. [Accessed 15 6 2024].

[32] Arduino, “Arduino,” Arduino nano BLE Sense , 2 7 2017. [Online]. Available: https://store-usa.arduino.cc/products/arduino-nano-33-ble-sense.. [Accessed 15 6 2024].

[33] Arducam, “Arducam OV7675 Camera module,” Arducam , 15 6 2016. [Online]. Available: https://store-usa.arduino.cc/products/arducam-camera-module. [Accessed 15 6 2024].

[34] S. Studio, “Seeed Studio OLED Display,” Seeed Studio , 4 3 2018. [Online]. Available: https://wiki.seeedstudio.com/Grove-OLED_Display_1.12inch. [Accessed 14 6 2024].

[35] A. Kumar, “Machine learning from theory to algorithms: an overview,” *Journal of physics: conference series.,* 2018.

[36] Sarker, “Machine learning: Algorithms, real-world applications and research directions,” *SN computer science ,* 2021.

[37] Apuke, “Arabian Journal of Business and Management Review,” *Quantitative research methods: A synapsis approach ,* 2017.

[38] Marshal, “ Sampling for qualitative research,” *Oxford University,* 1996.

[39] K. City, “ Kigali Economic Development Strategy,” 2002.

[40] K. City, “ City development strategy,” 2018.

[41] Manirakiza, “ The rapid urban growth and neighbourhood dynamics of a prosperous city,” 2020.

[42] RNP, “ Rwanda National Police 2014 Review,” 2014.

[43] Douglas, “Meaning of data,” 6 5 2015. [Online]. Available: http://www.onlineetymologydictionary/data.. [Accessed 8 3 2024].

[44] Mesly, “ Creating models in psychological research,” 2017.

[45] Kaggle, “Kaggle,” Kaggle, 20 8 2019. [Online]. Available: https://www.kaggle.com/. [Accessed 13 5 2024].

[46] EdgeImpulse, “EdgeImpulse.,” EdgeImpulse, 8 7 2016. [Online]. Available: https://edgeimpulse.com/. [Accessed 17 05 2024.].

[47] Edgeimpulse, “Connecting Camera ov7675, Edgeimpulse forum,” Edgeimpulse forum, 20 7 2017. [Online]. Available: https://forum.edgeimpulse.com/t/error-to-connect-camera-ov7675/9297/13. [Accessed 05 06 2024].

[48] SDG, 2013. [Online]. Available: https://sustainabledevelopment-rwanda.github.io/3-6-1/.