Dissertation submitted to the
UNIVERSITY OF MOHAMED BOUDIAF - MSILA

1985
جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

FACULTY OF MATHMATICS AND COMPUTER SCIENCE
DEPARTEMENT OF COMPUTER SCIENCE
Master in Computer science

In partial fulfillment of the requirements for the degree of
Master in **Computer Science**
By

Leyla Makri
Zeyneb Khelifi

Theme of the thesis

Crowdsensing monitoring

Under the supervision of :
Bilal Lounnas

in front of the jury

Benazi Makhlouf	University of M'sila	**President**.
Bilal Lounnas	University of M'sila	**Sypervisor**.
Cheriat Hanane	University of M'sila	**Examinator**.

June 2024

Acknowledgements

This study would have been impossible without the support and encouragement of many people.
This was a journey worthwhile taking.

First of all, we would like to thank **Allah**, the merciful, for giving us the health and the strength to finish this thesis.

Second, we would like to thank our supervisor **Mr. Bilal Lounnas** , for his insightful direction and support during the period of writing this thesis.

Finally, our deep and gratitude to our **Family** and **Friends** for their help and support they give us.
Thank you, All.

Contents

List of abbreviations

AI *Artificial Intelligence*
ML *Machine Learning*
DL *Deep Learning*
IOT *Internet of Things*
MCS *Mobile CrowdSensing*
RFID *Radio Frequency Identification*
NFC *Near Field Communication*
SVM *Support Vector Machine*
MOG2 *Mixture of Gaussian 2*
OARC *Obstacle Avoidance and Road Condition detection system*
LIDAR *Light Detection And Ranging*
RGB *Red Green Blue*
FPS *Frames Per Second*
VSN *Vehicle Sensor Network*
IT *Information Technology*
YOLO *You Only Look Once*

List of Figures

List of Tables

General introduction

In recent years, there has been a growing emphasis on enhancing road safety and transportation efficiency. Traditional methods for detecting road obstacles have often been time-consuming, expensive, and limited in their spatial coverage. This limitation has resulted in delayed responses to hazards and increased safety risks for drivers. However, with the widespread adoption of smartphones and ubiquitous computing technologies, a promising opportunity has emerged to collect real-time data on a large scale, paving the way for the development of more efficient and cost-effective solutions for road obstacle detection.

This dissertation presents a novel obstacle detection system that leverages the capabilities of the Microsoft Kinect sensor and computer vision techniques and approaches. By utilizing the Kinect's and computer vision ability to capture real-time depth and color information, the proposed system enables accurate and efficient detection of various road obstacles, such as potholes, debris, and uneven pavement. The system's primary objective is to address the shortcomings of traditional methods and offer several significant benefits to road users and transportation authorities.

One of the key advantages of the proposed system is improved safety for drivers. By detecting and alerting drivers to potential obstacles in real-time, the system provides timely warnings, allowing motorists to take appropriate measures to avoid accidents or damage to their vehicles. This proactive approach to obstacle detection can significantly reduce the likelihood of collisions and other hazardous incidents on the road.

Furthermore, the system also contributes to reducing costs associated with road damage. Potholes and other road hazards can cause significant harm to vehicles, leading to expensive repairs and maintenance. By identifying and reporting these obstacles promptly.

In conclusion, the dissertation presents a pioneering obstacle detection system that utilizes Microsoft Kinect and computer vision to capture real-time depth and color information. By offering improved safety, reduced costs, and enhanced transportation system resilience, the system represents a significant step forward in addressing the challenges associated with road obstacle detection. As a result, it has the potential to revolutionize the way we approach road safety and transportation efficiency in the future.

Motivation

The growing emphasis on road safety and transportation efficiency has highlighted the need for more effective methods of detecting road obstacles. Traditional approaches have limitations in terms of time, cost, and spatial coverage, leading to delayed responses and increased safety risks. The motivation behind this dissertation is to leverage the advancements in ubiquitous computing technologies, such as smartphones, and explore the potential of real-time data collection for developing efficient and cost-effective solutions for road obstacle detection.

Objectif

The objective of this dissertation is to propose a novel obstacle detection system that utilizes the Microsoft Kinect sensor and computer vision techniques and approaches.By leveraging the Kinect's and computer vision capabilities to capture real-time depth and color information, the system aims to achieve accurate and efficient detection of various road obstacles, including potholes, debris, and uneven pavement. The primary goal is to address the limitations of traditional methods and provide significant benefits to road users and transportation authorities.

Dissertation structure

This dissertation is divided into three chapter organized as follows: The first chapter delves into the realms of Artificial Intelligence (AI) and the Internet of Things (IoT). It defines AI, explores its characteristics, and delves into its applications across various sectors like healthcare, finance, transportation, and education. The chapter further elucidates machine learning techniques, encompassing supervised, unsupervised, and reinforcement learning, while also introducing deep learning and distinguishing it from traditional ML methods. The discussion then transitions to IoT, defining it, outlining its architecture, applications, and challenges. The role of sensing in IoT is emphasized, followed by an exploration of future research directions.

Chapter 2 focuses on crowdsensing monitoring, defining crowdsensing and its importance in monitoring applications. It outlines the lifecycle of mobile crowdsensing systems, and explores environmental, infrastructure, and social applications, providing examples such as air quality monitoring and traffic prediction utilizing crowd data.

Lastly, Chapter 3 addresses obstacle detection research, outlining its motivation and objectives. It presents a Kinect sensor-based methodology and computer vision techniques and approaches for road obstacle detection systems, detailing the developmental methodes. It concludes by highlighting avenues for future research in the field.

Chapter 1

Artificial intelligence and Internet of things

This chapter explores the fields of Artificial Intelligence and the Internet of Things. It delves into applications of AI across domains and key concepts like machine learning.

1.1 Introduction

In today's rapidly advancing technological landscape, Artificial Intelligence (AI) stands at the forefront of innovation, reshaping industries and revolutionizing the way we interact with the world around us. This chapter delves into the multifaceted realm of AI and its application in various domains, from healthcare to finance, transportation, education, and beyond.

Artificial Intelligence, often referred to as the simulation of human intelligence by machines, encompasses a diverse array of technologies aimed at enabling computers to perform tasks that traditionally require human intelligence. From problem-solving and decision-making to natural language processing and perception, AI systems continue to evolve, driven by advancements in machine learning, deep learning, and the Internet of Things (IoT).

The chapter begins with an exploration of the fundamental concepts of AI, elucidating its significance and potential impact on society. It then delves into the diverse applications of AI across different sectors, showcasing its transformative power in healthcare, finance, transportation, and education. Furthermore, the chapter elucidates key concepts such as machine learning, including supervised, unsupervised, and reinforcement learning, as well as the emerging paradigm of deep learning.

Moreover, the chapter sheds light on the Internet of Things (IoT), a paradigm that interconnects physical devices and enables them to collect and exchange data seamlessly. It delineates the characteristics, architecture, and applications of IoT, highlighting its role in enabling smart, interconnected systems that enhance efficiency, productivity, and convenience.

Through a comprehensive examination of AI, machine learning, deep learning, and the Internet of Things, this chapter aims to provide readers with a holistic understanding of these transformative technologies and their implications for the future.

1.2 Artificial Intelligence

Artificial intelligence (AI) is a field that encompasses a wide range of approaches and definitions, making it challenging to provide a precise definition. the term "artificial intelligence" can be misleading as it implies that artificial agents possess the same level of intelligence as humans, which may not always be the case. in this paper, the definition of AI adopted is as follows: "the discipline that focuses on studying and developing computational artifacts that exhibit some aspects of intelligent behavior" [73].

These computational artifacts, known as artificial agents [38], display facets of intelligent behavior and operate autonomously. artificial agents have the capacity to perform flexible actions that align with their design objectives [103].in simpler terms, artificial agents have the ability to make appropriate decisions within the boundaries of their intended purposes. [37] describes three main characteristics of artificial agents from an agency perspective: autonomy, adaptability, and interactivity, as summarized in Table1.1

Characteristic artificial agent	**Definition**
Autonomy	An artificial agent that is capable of reactive and proactive action and has task autonomy and goal autonomy, but only in a limited and well-defined context.
Adaptability	An artificial agent that has the capability to learn (with machine learning) and interact with virtual agents and embodied systems.
Interactivity	An artificial agent that has the ability to perceive and interact with other (virtual) agents.

Table 1.1: Main characteristics of AI agencies

1.3 Application of AI

Artificial intelligence (AI) has the potential to revolutionize various industries, and its impact is already being felt in healthcare, finance, transportation, and education. Here are some examples of AI's applications in each of these industries[18]:

1.3.1 AI in Healthcare

In the field of healthcare, AI is proving to be a game-changer with its ability to analyze vast amounts of data and make accurate predictions.

Here are some key applications of AI in healthcare:

- **Advanced Medical Diagnosis:** AI is revolutionizing medical diagnosis by enabling advanced analysis of medical images such as X-rays, CT scans, and MRIs[31, 30]. by leveraging machine learning algorithms, AI can detect anomalies and aid in diagnosing various illnesses. for example, AI can identify cancerous cells in mammograms or detect signs of stroke in brain scans[5]. this technology has the potential to improve early detection and save lives.

- **Predictive Modeling:** Another significant application of AI in healthcare is predictive modeling. by analyzing patient data, including medical history and lab results, AI can predict disease risks and identify individuals who require early intervention[95]. this technology has proven particularly useful in predicting heart attack risks in high-risk patients[52], allowing healthcare professionals to take proactive measures and potentially prevent adverse events.

- **Drug Discovery Acceleration:** AI is also accelerating the process of drug discovery by assisting in target identification. By analyzing vast amounts of data[49], AI algorithms can identify potential targets for drug development[111], expediting the search for new treatments for diseases such as cancer or Alzheimer's[35]. this has the potential to revolutionize the pharmaceutical industry and improve patient outcomes.

- **Computer-Assisted Surgery:** AI is transforming the field of surgery by providing computer-assisted guidance to surgeons. this technology enhances precision and safety during complex surgical procedures[85], such as cardiac or neurological surgeries[70]. by leveraging AI, surgeons can improve accuracy, minimize risks, and achieve better patient outcomes.

1.3.2 AI in Finance

The finance industry is another sector where AI is making significant strides. From fraud detection to market prediction, AI is reshaping the way financial institutions operate. Here are some notable applications of AI in finance:

- **Fraud Detection and Prevention:** One of the key applications of AI in finance is fraud detection and prevention. by analyzing real-time transaction [61], AI algorithms can identify fraudulent activities[84], such as credit card fraud or loan application scams. this technology helps financial institutions protect their customers and mitigate financial losses.

- **Market Prediction (with Caution):** AI is also being used in market prediction, where it analyzes vast amounts of market data to predict stock price movements[100] However, it is important to note that market prediction is a complex task, and AI algorithms are not always accurate[92]. While AI can provide valuable insights, it is crucial to exercise caution and consider multiple factors when making investment decisions.

- **Informed Risk Management:** AI plays a crucial role in risk assessment and decision support in the finance industry. by evaluating financial risks, such as company default risks or investment loss risks, AI enables institutions to make informed decisions. This technology helps financial institutions manage risks effectively and optimize their operations[107].

1.3.3 AI in Transportation

The transportation industry is undergoing a significant transformation with the integration of AI technologies. from autonomous vehicles to traffic optimization, AI is reshaping the way we travel. let's explore some key applications of AI in transportation:

- **Autonomous Vehicles (Future Promise):**AI is at the forefront of developing self-driving technology for autonomous vehicles[15]. by leveraging AI algorithms, these vehicles can navigate without human control, potentially revolutionizing transportation systems[57]. however, it is important to note that this technology is still evolving, and there are several hurdles to overcome before widespread adoption

- **Traffic Optimization and Congestion Reduction:** AI plays a vital role in optimizing traffic flow and reducing congestion. by analyzing real-time traffic data, AI algorithms can optimize traffic signals, synchronize traffic lights, and reroute traffic during accidents or road closures. this technology has the potential to improve commuting experiences, reduce travel times, and minimize environmental impact[12].

- **Personalized Route Planning:** AI-powered route planning takes into account various factors such as traffic conditions, weather forecasts, and roadwork. by considering these variables, AI algorithms can find the fastest and most efficient routes to a destination. this personalized route planning ensures a smoother travel experience and helps individuals save time and fuel[40].

1.3.4 AI in Education

In the field of education, AI is transforming the way students learn and teachers educate. from personalized tutoring to adaptive learning platforms, AI is reshaping the educational landscape. let's explore some key applications of AI in education:

- **Intelligent Tutoring and Personalized Learning:** AI enables intelligent tutoring systems that adapt to individual students' needs. by analyzing student performance data and learning patterns, AI algorithms can create personalized learning experiences tailored to each student's requirements. this technology provides extra support to struggling students and challenges advanced learners, enhancing the overall learning experience[21].
- **Adaptive Learning Platforms:** AI-powered adaptive learning platforms are revolutionizing education by delivering content that matches students' learning pace and level. by analyzing individual learning styles and progress, AI algorithms can provide educational materials that are tailored to each student's needs. This ensures that advanced learners are appropriately challenged, while students who require more time receive the necessary support[19].
- **Automated Grading with Caution:** AI has the potential to automate the grading process, freeing up teachers' time for other tasks. by analyzing assignments and exams, AI algorithms can provide automated grading and feedback. however, it is important to exercise caution with automated grading, as it may not always be accurate. Teachers should use it as a tool to support their assessment process rather than relying solely on automated grading[78].

In conclusion, the application of AI across various industries like healthcare, finance, transportation, and education is rapidly transforming our world. While each industry utilizes AI for different purposes and with varying data types, regulations, and ethical considerations, the common threads are data-driven analysis, automation, and personalization.

1.4 Machine learning

Machine learning is a powerful field of study that focuses on developing algorithms and models that enable computers to learn and make predictions or decisions without being explicitly programmed. it has gained significant popularity in recent years due to its ability to analyze large amounts of data and extract valuable insights. Machine learning can be broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning as shown in Figure1.1.

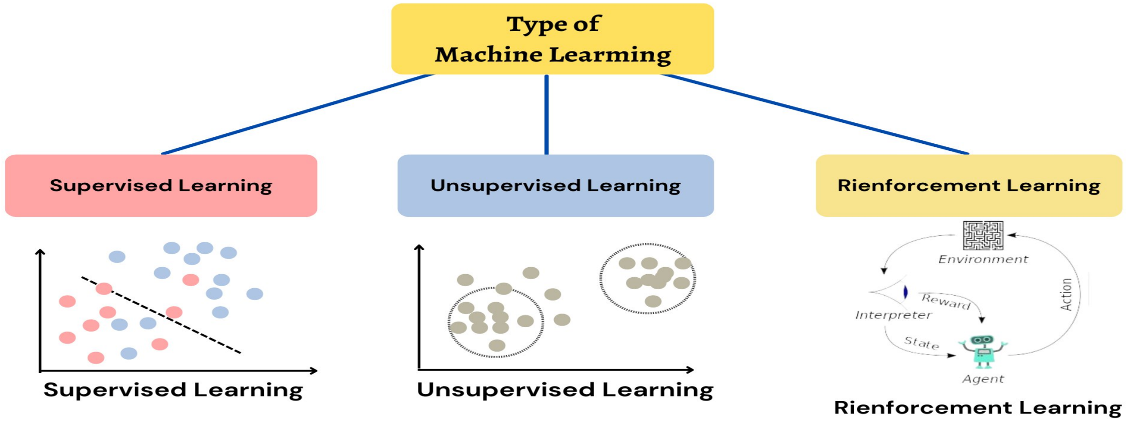


Figure 1.1: The main types of machine learning.

1.4.1 Supervised Learning

The most popular machine learning technique at the moment is supervised learning, which is an effective tool for prediction. Its objective is to develop a function that describes the connections between its input and output data as optimally as possible [66].

This is achieved by first providing labeled data for training (a training dataset) to the machine learning algorithm, which allows it to infer correlations and produce a prediction function f(x)[47]. the classification of new unlabeled input values x is predicted using the prediction function f(x) as a foundation. this is accomplished by using various statistical techniques, such as probability estimation for probabilistic functions, regression for continuous functions, and classification for discrete functions [60]. probability estimation and regression can be used to forecast, detect, predict, and identify connections in numerical data.

Neural networks and decision trees are two methods used in classification that identify patterns in qualitative data. additionally, supervised learning techniques like neural networks and decision trees can be applied to crowdsensing data.

1.4.2 Unsupervised Learning

The unsupervised learning approach is nearly the opposite of supervised learning, as this approach has no information about the output of the data, and the training data is not labeled [77].

The goal of unsupervised learning is to discover structures based on common elements within the input data [91]. cluster analysis is the most popular unsupervised learning approach, including techniques like K-means clustering, hierarchical clustering, and probabilistic clustering. K-means cluster analysis is the most common unsupervised learning technique used for finding groupings or hidden patterns (clusters) in data, which can be used for exploratory data analysis [62]. The cluster analysis algorithm works by iteratively dividing a set of data points (x1, ..,xn) into one of the K (K1,...,Kn) clusters identified. the data points placed in a K-cluster contain similar features identified by the unsupervised learning algorithm [48].

1.4.3 Reinforcement Learning

Reinforcement learning is the study of how an artificial agent learns to maximize a given concept of cumulative rewards through sequential decision-making in an environment [39].

Reinforcement learning allows an artificial agent to learn from its errors and improve at certain activities. It does this through a system of rewards. Therefore, for this strategy to succeed, it is imperative that the reward function and the artificial agent's aim be defined accurately.

Approach	Supervised learning	Unsupervised learning	Reinforcement learning
Objective	Make predictions	Discover structures	Make decisions
Possible techniques	Regression,Probability, Estimation,Classification,Deep Learning	Cluster analysis, Principal Components Analysis,Deep Learning	Markov Decision Processes, Q-Learning,Deep Learning
Training requirements	Labelled data	Unlabeled data	Reward function
Challenges	Human errors, need.for human expertise to correctly train the ML model	Lack of transparency, Computational complexity	Customer segmentation, Market research, Anomaly detection
Application Examples	Sentiment Analysis, Event Detection	Anomaly Detection, Recommendation	Playing a game (e.g. AlphaGo), Natural Language Processing

Table 1.2: Overview machine learning

1.5 Deep Learning (DL)

For more complex domains, deep learning algorithms are particularly useful. deep learning algorithms are based on neural network models, which consist of multiple simple and linked units, or 'neurons' [63]. this type of algorithm attempts to artificially simulate the brain. processes in the brain therefore get used to understanding and explaining concepts of an artificial neural network (ANN), which is a central concept within deep learning. an ANN is a complex, unidirectional network of connections between 'neurons' of different strengths. the ANN contains input and output nodes; these nodes are connected through hidden nodes, which are trained to minimize empirical errors to create more accuracy for the outcome of the task it serves [39].

Based on the Figure1.2 the main differences between ML and DL lie in the complexity of the models, the need for feature engineering, data requirements, performance and scalability, and interpretability. ML models are simpler, more interpretable, and suitable for a wide range of applications.

DL models are more complex, capable of learning hierarchical representations, and excel in domains that require handling large-scale, high-dimensional data.

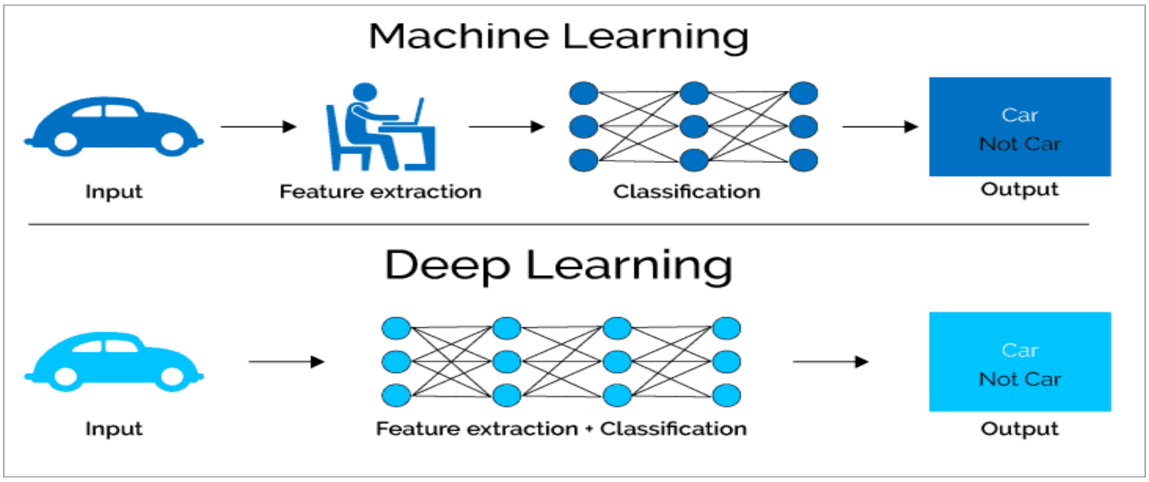


Figure 1.2: Deep Learning VS Machine Learning..

1.6 The Internet of things

The IoT has emerged as a revolutionary concept that is transforming the way we live, work, and interact with technology. with the increasing prominence of IoT in the past decade, it is essential to understand its various aspects, including its definition, characteristics, and implications. in this section, we will delve into the depths of IoT, exploring its multiple definitions, the key characteristics that define it, and its potential to reshape industries and society as a whole.

1.6.1 One Concept, Many Definitions

The Internet of Things (IoT) has become increasingly prominent in the past decade, but its definition varies among different individuals and perspectives. according to Whitmore et al[102], there is no universal definition of IoT.

Two main conceptualizations exist: the technical perspective and the socio-technical perspective. the technical perspective views IoT as a collection of technical artifacts and an ecosystem. it is defined based on these artifacts and their capabilities. for example, Weyrich and Ebert [101] define IoT as a means to achieve innovative functionality and improved productivity by seamlessly connecting devices.

On the other hand, Tarkoma [11] provide a more detailed definition, describing IoT as a global network and service infrastructure with variable density and connectivity. they emphasize self-configuring capabilities, standard and interoperable protocols, and the integration of heterogeneous things into the Internet.

Similarly, whitmore et al[102] define IoT as a paradigm where everyday objects are equipped with identifying, sensing, networking, and processing capabilities. this enables communication between objects, as well as with other devices and services over the Internet, to achieve specific objectives. these technical definitions dominate the literature in the field of computer science. in contrast, the socio-technical perspective of IoT acknowledges not only the technical artifacts but also the actors and processes involved in the IoT ecosystem.

Haller et al[46] recognize the role of connected objects as active participants in business processes. they define IoT as a world where physical objects seamlessly integrate into the information network and become active participants in business processes. services are available to interact with these "smart objects" over the Internet, allowing queries about their state and associated information while considering security and privacy concerns.

Shin [90] argues that IoT is part of wider socio-technical systems, which encompass humans, human activity, spaces, artifacts, tools, and technologies. Shin et al. also notes that in some cases, a biological entity, such as a human with a heart monitor implant or a farm animal with a biochip transponder, can be considered a connected thing within the IoT framework.

It is important to note that the definitions provided here are just a glimpse into the complex and evolving nature of IoT. as the field continues to advance, new perspectives and understandings of IoT will likely emerge.

1.6.2 Characteristics of IoT

The Internet of Things (IoT) is revolutionizing our world by connecting diverse devices and objects to the internet, enabling them to collect and exchange data. this interconnectedness brings about several key characteristics that define the landscape of IoT as shown in Figure1.3:

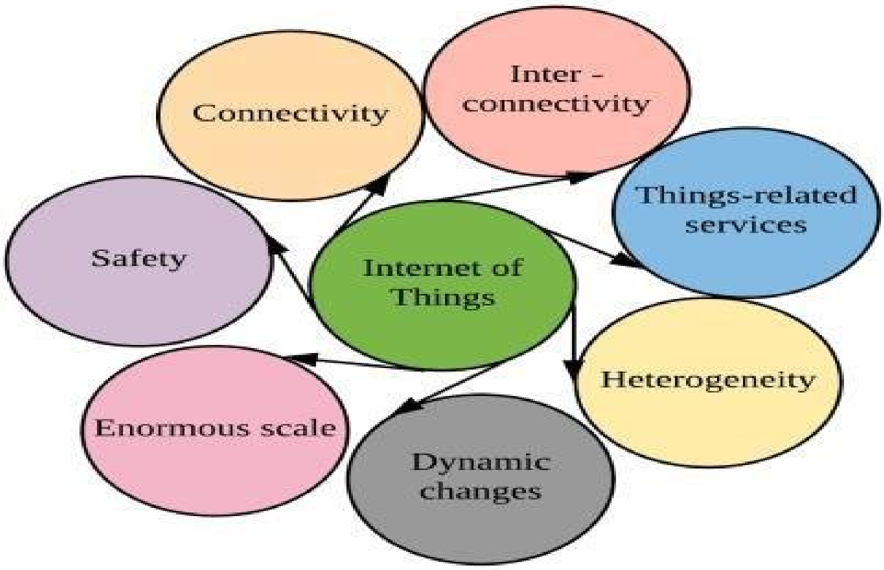


Figure 1.3: Characteristics of IoT.

- **Interconnectivity:** Devices can seamlessly connect to each other and the internet regardless of location, using various networks like Wi-Fi, Bluetooth, and cellular. this enables data exchange, collaboration, and remote monitoring across different domains [94].

- **Heterogeneity:** The IoT ecosystem comprises a wide range of devices with diverse hardware platforms, software, and communication protocols. this necessitates flexible protocols and standards to ensure smooth interaction between devices [96].

- **Dynamic Changes:** The state of devices and the overall network size are constantly in flux. Devices connect and disconnect, change location and functionality, requiring adaptable systems capable of handling these dynamic changes [97].

- **Enormous Scale:** The exponential growth of connected devices generates vast amounts of data. efficient management, processing, and analysis of this data are crucial for extracting valuable insights [41].

- **Things-Related Services:** IoT goes beyond mere connectivity. it offers services related to connected objects, such as privacy protection and ensuring consistency between physical and virtual representations [89]. this necessitates advancements in both physical and information technologies.

- **Safety:** Protecting user privacy, physical well-being, and network security is paramount. robust and scalable security measures are essential to safeguard data and prevent potential harm [45].

- **Connectivity:** Connectivity encompasses both accessibility and compatibility. accessibility refers to joining a network, while compatibility ensures the ability to exchange data seamlessly [29].

While these characteristics open exciting possibilities, it's important to acknowledge potential limitations like security vulnerabilities and ethical considerations regarding data privacy. by understanding these fundamental characteristics and their implications, we can leverage the power of IoT responsibly and effectively.

1.6.3 Architecture of IoT

In the realm of the Internet of Things (IoT), architecture plays a crucial role in defining the structure and functionality of interconnected devices and systems. IoT architecture provides a framework for organizing and managing the various components involved in the IoT ecosystem.

In this section, we will explore different architectures of IoT, including three-layer, four-layer, and five-layer architectures. Each architecture offers unique advantages and caters to specific requirements.

1.6.3.1 Three Layer Architectures

As seen in Figure 1.4, the most fundamental architecture is a three-layer architecture[26]. it was presented in the early phases of this field's investigation. the physical, network, and application layers are its three tiers.

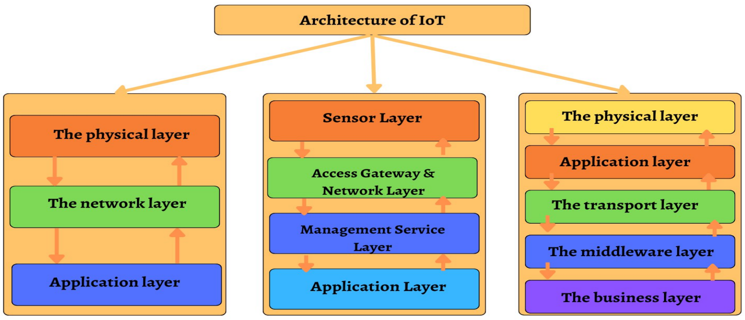


Figure 1.4: Architecture of IoT.

- **The physical layer**, or perception layer, is equipped with sensors to sense and collect data about the surroundings. it detects certain physical characteristics or locates more intelligent items in its surroundings[69, 105].
- **The network layer** is in charge of establishing connections with servers, network devices, and other smart objects. Additionally, sensor processing and transmission are done with its characteristics[69, 105].
- **Application layer** specific services are provided to the user via the application layer. the article outlines several uses for the Internet of Things, such as intelligent homes, intelligent cities, and smart health[69, 105].

1.6.3.2 Four Layer Architectures

In the world of the Internet of Things (IoT), the architecture plays a crucial role in ensuring the seamless integration and efficient functioning of interconnected devices[93]. one popular architecture used in IoT systems is the four-layer architecture

- **Sensor Layer:** This layer is the foundation, consisting of various sensors like RFID tags, embedded systems, and sensor networks deployed in the field[53].

  Each sensor collects data about its environment and may have identification and storage capabilities.

- **Access Gateway & Network Layer:**

  This layer connects sensors to the next layer, transferring collected data.

  it uses scalable, flexible protocols to manage data from diverse devices.

  this layer requires robust networks and supports independent communication between multiple organizations[106].

- **Management Service Layer:**

  This layer acts as a bridge between the network and application layers, managing devices and information.

  it captures and processes raw data, extracting relevant insights, ensuring data security and privacy.

- **Application Layer:**

  This topmost layer provides user interfaces for accessing various IoT applications.

  these applications span diverse sectors like healthcare, transportation, agriculture, supply chain, and more[106].

1.6.3.3 Five Layer Architectures

Perception, transport, processing, application, and business layers are the five layers[42]. the perception and application layers have the same role as the three-layer architecture. we describe how the final three levels work.

- **The transport layer** uses networks like Bluetooth, RFID, NFC, 3G, LAN, WIFI, and others to move sensor data from the perception layer to the processing layer and vice versa[54].

- **The middleware layer** is another name for **the processing layer**. it takes massive volumes of data from the transport layer and stores, processes, and analyzes it. it has the ability to oversee and offer the lower levels a wide range of services.

It makes use of several technologies, including big data processing modules, cloud computing, and databases[106].

- **The business layer** manages the whole IoT system, including applications, business and profit models, and users' privacy. (Architecture of IoT)[106].

Here's a consolidated comparison of the four-layer, three-layer, and five-layer architecture models for IoT, incorporating your previous insights and addressing mentioned limitations shown in table 1.3:

Layer	Four-Layer Architecture	Three-Layer Architecture	Five-Layer Architecture
Focus	Technical aspects	High-level overview	More detailed technical aspects and business considerations
Perception Layer	Sensors, embedded systems, RFID tags	Sensors for data collection	Similar to three-layer
Network Layer	Access Gateway & Network	Network connections, sensor processing & transmission	Transport Layer: Data transfer using various networks
Management Service Layer	Device & information management, data capture	Not explicitly mentioned	Processing Layer: Data storage, processing, analysis, lower-level services
Application Layer	User interface, applications for various sectors	User-specific services	Similar to previous architectures
Additional Layer	None	None	Business Layer: Manages system, applications, business models, user privacy
Strengths	Clear delineation of key technical functions, good for specific technical implementation	Simple and easy to understand	More detailed technical breakdown, includes business layer
Weaknesses	Lacks business perspective, less emphasis on data processing	Limited technical detail, less focus on specific implementation	Can be more complex, requires additional explanation
Suitable for	Technical design, implementation of specific IoT systems	High-level understanding, initial discussions	Detailed analysis, comprehensive IoT project planning

Table 1.3: Comparison of architecture.

1.6.4 Application of IoT

IoT applications range in size from local networks, such as automation for homes, to huge networks, such as cloud-based industrial applications.a succinct overview of the many uses of IoT, including home automation, agribusiness, aquaculture, health care, logistics, and environmental monitoring.

1.6.4.1 Monitoring Environment

Many sensors are available to sense many characteristics, including water pollution, air pollution, temperature, and humidity. temperature is measured with the use of sensors such as thermometers and RTDs. both dust and gas sensors can be used to monitor air pollution. The identification of chemical presence is possible through the utilization of e-nose and e-tongue technologies. These technologies employ software for pattern recognition. these are employed in urban areas to track pollution levels[20].

1.6.4.2 Home Automation

: Domotics is the term for smart buildings or home automation. The different items in the house may be controlled by a single hub, usually a smartphone with sensors like an accelerometer. in other words, NFC, Bluetooth, Zigbee, or any other short-range low-power protocol would be used to link the smart phone to smart appliances, including air conditioners, fans, lighting, and water heaters[58].

1.6.4.3 Agriculture

: In this industry, IoT is sometimes referred to as "smart farming". tractors equipped with GPS allow us to regulate the tractor’s course. Additionally, we can operate the pump (configure the timing for when to give water to the farm) with the aid of a basic embedded IoT board. Soil sensors can also be used to study the state of the land[24].

1.6.4.4 Aquaculture:

The Internet of Things has been utilized to regulate the operation of radiators, which are used to give oxygen to fish tanks. The farmer can obtain information about water temperature by employing chemical sensors[68].

1.6.4.5 Health care :

In the medical field, one may follow a patient's health status by using IoT.medical sensors come in two varieties: wearable and internal to the body. wearable sensors may be used to monitor a number of variables, including body temperature, calories burned, and heart rate. wearables like fitbits and reflex are examples. internally implanted sensors are another kind. these are employed when ongoing patient health monitoring is required. For instance, a sensor implanted in the body allows a physician to track and monitor a patient's heartbeat, pulse rate, and other vital signs[71].

1.6.4.6 Transportation and Logistics:

The IoT plays an important role in transportation and logistics.The industries may track real-time data about cars, such as their position and other details, by installing RFID tags or barcodes on them. Furthermore, one may regulate the vehicle's speed by improving IoT capabilities in the transportation sector. businesses in logistics can monitor the arrival and outflow of goods by using barcodes.We can expand the use of IoT in a number of different industries, including mining, safety, traffic monitoring, etc., by developing appropriate standards[17][51].

1.6.5 Challenges and recent research of IoT

The Internet of Things (IoT) has revolutionized the way we interact with technology and the world around us. it has opened up new possibilities for automation, connectivity, and data collection. However, along with these advancements, there are also various challenges that need to be addressed to ensure the seamless functioning of IoT systems[83].

1.6.5.1 Networking

In general, networking issues are very relevant to the Internet as they involve some of the key components that are used to govern networks. prior to anything else, mobile ad-hoc network traffic and protocols—which greatly influence network behavior—sought to address networking issues. mobile ad hoc networks (MANETs) coupled to fixed networks via various gateways were employed by the authors. It's impossible to foresee where an object will go on the Internet of Things, and it might need to be sent from one network to another. the inability to pinpoint an object's exact position and dynamic gateway changes are the main issues [113].

1.6.5.2 Routing

In order to properly finish the communication process, the routing procedure involves determining the optimum path between the source and the destination. Depending on the kind of communication protocol, there are several methods to choose the optimal path, including considering factors like bandwidth, prices, and hop count. routing protocols may be divided into two primary categories:proactive protocols construct the way first, before the request is made; reactive protocols establish the path after a transmission request is made. suggested the "fault-tolerant routing protocol" for the Internet of Things. the cross-layer idea and learning automate (LA) were used in the construction of this protocol. when tackling optimization challenges, LA must consider cross-layer concepts in order to select the best solutions. in order to select the best solutions for optimization challenges, LA must cross-layer IoT device energy conservation (i.e., RFID)[76].

1.6.5.3 Heterogeneity

Since the Internet of Things (IoT) environment includes a wide variety of devices by nature, it is the most well-known example of a heterogeneity issue. the primary goal of IoT is to develop a common method for abstracting the heterogeneity of these devices and maximizing the utilization of their functionality. in this spirit, regardless of the form of these gadgets, the researchers are always looking for an efficient technique to handle them. the creation of a domain-specific language (DSL), a graphic editor, and the IoT platform Midgar software are examples of the solutions that were attempted to address some of the issues with the Internet of things, such as heterogeneity and interconnection, and to create an application that enables people to interconnect services over the internet[44].

1.6.5.4 Interoperability

The capacity to design devices or systems that cooperate well with one another is known as interoperability. aimed to leverage the "smart-M3" semantic information sharing technologies as the foundation for a ubiquitous computing and Internet of Things semantic level interoperability architecture. the main concept of the suggested design is to simplify the administration process of IoT environments by segmenting them into smaller areas[55].

1.6.5.5 Cloud Computing

IoT and cloud computing have advantages and disadvantages, with cloud computing being seen as a standard foundation for IoT. IoT stands for the real world and little things, but it also has storage issues in addition to standard network issues like privacy and scalability. in order to create systems that can overcome numerous challenges like scalability, storage resource constraints, and virtualization, the integration of cloud computing with the Internet of Things has gained significant attention in recent research. the primary goal of this integration is to leverage cloud computing's processing power, which is needed for sensors and other things[22].

1.6.5.6 Security and privacy

The purpose of the security rule is to safeguard it from threats, which may be divided into two categories: internal threats, which involve misusing the system or information, and external threats, which include attacks on the system by attackers. the three primary components of security are truth, privacy, and data confidentiality. Secrecy, anonymity, and isolation are characteristics of privacy. privacy is described as the ability to regulate access to personal data and to maintain the confidentiality of particular information and data. to provide a trustworthy connection between the real world and the digital realm in the context of the Internet of Things, security and privacy are crucial[82].

1.6.5.7 Quality of Service

When "the amount of time that is taken to deliver the message from the sender and the receiver" is equal to or less than the pre-specified time requirement, the quality of service is considered to have been attained. the ITU redefined quality of service (QoS) as the extent to which a provider complies with an agreement between them when providing a service to a user. dealing with service models is necessary for QoS assurance in order to ascertain the appropriate level of QoS for any Internet service [117].

1.7 Iot Sensing

The Internet of Things (IoT) often overlooks security and privacy concerns due to software and hardware integration [8]. data perturbation and cryptography are common solutions, with a collaborative learning model for privacy protection [6] and an adversarial model-free anonymization technique [9]. homomorphic encryption-based target-finding methods protect privacy [10], while hardware-based techniques like privacy-preserving optical filters [2] and real-time occupancy identification systems [7] ensure security. privacy-preserving parking surveillance systems include real-time sensor-based monitoring [4], smart camera networks [1], and hardware-based vehicle flow identification as shown in Figure1.5[3].

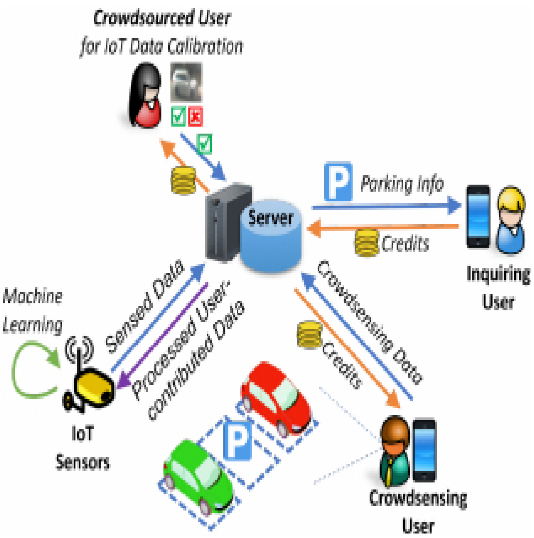


Figure 1.5: Internet of Things Sensing.

1.8 Conclusion

In summary, this chapter has delved into the foundational concepts of artificial intelligence (AI), machine learning, deep learning (DL), and the Internet of Things (IoT). We've explored how these technologies are reshaping industries and transforming various aspects of our lives.

AI, with its capacity to emulate human intelligence, offers remarkable potential for innovation across diverse sectors such as healthcare, finance, transportation, and education. Machine learning, a subset of AI, enables computers to learn from data and make decisions autonomously, while deep learning, a sophisticated form of machine learning, has shown exceptional performance in handling complex tasks like image and speech recognition.

The Internet of Things represents a new era of connectivity, where everyday objects are embedded with sensors and connected to the internet, facilitating real-time data exchange and enabling smart applications in areas like smart cities, healthcare monitoring, and environmental monitoring.

As we move forward, it's crucial to embrace these technologies responsibly, considering their ethical implications and ensuring they benefit society as a whole. By fostering collaboration, transparency, and ethical considerations, we can harness the full potential of AI, machine learning, deep learning, and the Internet of Things to create a more sustainable, efficient, and equitable future.

In the next chapter, we'll explore the complexity and scope of MCS, defining it, its importance, and its life cycle. We'll look at the different stages of the life cycle of MCS: creating tasks, assigning tasks, executing tasks, collecting data, and analyzing data. We'll also look at the various uses of MCS, from environmental monitoring to infrastructure management to social applications, with descripe exesting examples from researchs.

Chapter 2

Crowdsensing Monitoring

Crowdsensing monitoring utilizes the power of user participation. This chapter outlines crowdsensing applications and the lifecycle of mobile crowdsensing systems.

2.1 Introduction

The ubiquity of smart devices has transformed the way we engage with technology, optimizing its sensing capabilities. This innovation has led to the emergence of a new paradigm: mobile crowdsensing(MCS).

In mobile crowdsensing, smart devices are turned into sensing nodes, forming a network that leverages the collective strength of users to accomplish multiple tasks. MCS represents a quantum leap forward in large scale and fine-scale sensing and computing. its rapid expansion in terms of implementation, maintenance, sensing scope and granularity, reusability, and scalability has proven to be advantageous in the age of intelligence, outperforming the typical wireless sensor networks. by bringing crowdsourcing principles together with smart devices with powerful sensing capabilities under the IoT umbrella, MCS can address complex sensing challenges.

In this chapter, contains the definition, meaning, and life cycle of crowdsensing with multiple definitions and descriptions. Also contains how crowdsensing is important in smart cities and how it can be used for urban sensing, with existing examples of applications. Contains the complexity and potential of crowdsensing, including its definition, meaning, life cycle, and more. Contains the different stages of the life cycle, such as creating tasks, assigning tasks, executing tasks, collecting data, and analyzing data.

Furthermore, contains the broad range of applications that can benefit from crowdsensing, with examples of research. These applications can include monitoring environmental conditions, infrastructure management, social applications, air pollution, weather monitoring, traffic congestion, road potholes detection, parking availability, and more.

2.2 Definition of crowdsensing

Crowdsensing has gained a significant research interest in various applications such as personalized recommendations, intelligent transportation, environmental monitoring,epidemiological investigations,etc,numerous definitions related to crowd sensing are provided by research[34]:

- Crowdsensing, also known as mobile crowdsensing (MCS), is a technique where a big number of people using sensing and computing mobile devices (such smartphones, tablets, and wearables), collectively share data and gathering information to measure,map,analyze, estimate, or infer (predict) any processes of common interest.this refers to the crowdsourcing of mobile device sensor data[67].
- Crowdsensing is a subtype of crowdsourcing where sensors are the actual sources of the data gathered[16].
- Crowd sensing (also known as participatory sensing) is a technique for gathering data about people's surroundings using mobile sensing devices. through task division, its highly expressive and powerful sensing skills can complete large-scale sensing projects.

  Increasing participant numbers and collecting higher-quality data is essential for success[86].
- Crowdsensing is one of the most feasible solutions where a large group of volunteers with mobile devices collectively share data and analysis on such data is carried out for extracting insights of common interest[34].

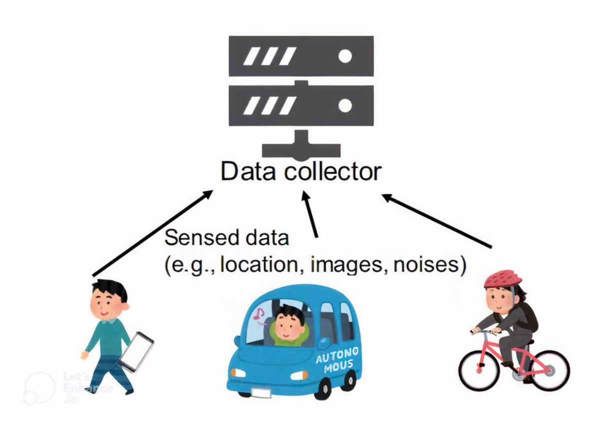


Figure 2.1: Participatory sensing

Figure 2.1: Displays images of several forms of mobile crowdsensing, also known as participatory crowdsensing.

These days, smartphones are capable minicomputers that can gather and analyze data about their users' surroundings,they have an amazing array of sensing components, like accelerometers and cameras,with the ability to collect and analyze users' surrounding information [27][75].

Alot of study reveals that data is also gathered by other modes of transportation, like vehicles, trains, and bicycles, in addition to mobile phones, participatory sensing or mobile crowdsensing are terms used to describe this type of data collection.

On participative sensing, numerous investigations have been carried out :

One approach that uses participatory car sensors to identify anomalous roadway places was proposed by Bridgelall et al[25].

Using information from the accelerometers of participating smartphones, Kozu et al created a danger map of bicycle accidents[59].

2.2.1 Why crowdsensing monitoring is essential?

By embracing the Smart City paradigm, crowdsensing gained significant attention in recent years and has become an appealing paradigm for urban sensing[28].

Crowdsensing becomes a solution able to plays a crucial role in various monitoring since this novel paradigm assumes that a significant number of users perform collaborative sensing tasks,thereby collecting data from different populated locations while doing their daily activities,the collected data is periodically transmitted to a central server (Cloud) for data storage and processing.

Overall, this strategy implies that the sensors used must be cheap and tiny enough for comfortable and easy transportation.

Otherwise, it becomes hard to achieve a widespread distribution and adoption, there must be a communications link for transmitting the acquired data to a cloud-based server, where data are constantly being stored and processed[16].

2.3 Life cycle of mobile crowd sensing

A mobile crowd sensing application's life cycle usually includes creating MCS applications according to the requirements, giving participants sensing tasks to complete, carrying out the task (sensing, computing, and uploading) on each participant's mobile device, and gathering and analyzing sensed results from participants[115].

The life-cycle of MCS can be divided into four stages: task creation, task allocation , task execution and data collection , data processing and analytics.

Motivated by the step-by-step procedure shown in Figure 2.2.[56][119].In the following, we explain the properties of each step.

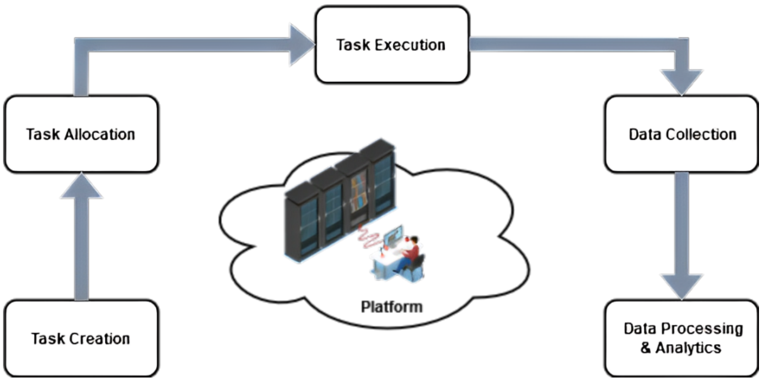


Figure 2.2: Crowd sensing application's life cycle.

2.3.1 Task Creation

By giving the participants access to the matching mobile sensing apps that will eventually be installed on their cellphones.

The MCS organizer establishes an MCS task,the goal of current work at the task creation stage is to make the MCS task and the associated mobile application production process easier for MCS organizers.

Each task has its task property including name, description, reward,and assignments,the content task can be applied to specific activities,such as taking photos etc..

Typically, this is accomplished by providing an easily understood domain specific language (DSL).

Each worker(participants) also has his/her location, and carried device,sensing capability as predefined object property of carried device.

The main research question at this point is how to make MCS task creation more efficient, particularly for those without formal programming experience, shown in Figure 2.3.[115][99][98].

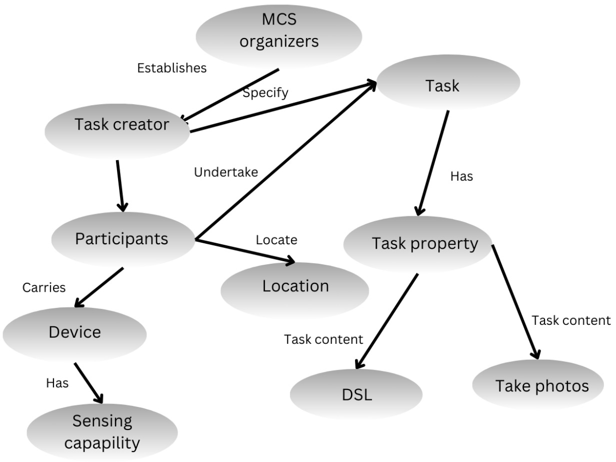


Figure 2.3: Task creation model for crowd-sensing.

2.3.2 Task allocation

It attempts to create location visiting sequences and allocate a set of job locations to a set of workers.

Enlisting participants and giving them personalized sensing tasks to do on their mobile devices.

The main challenge at this point is finding sufficient and suitable crowd sensing participants.

Additionally, previous research has suggested a number of methods for choosing a subset of individuals,for instance, a task's likelihood of success is increased when it is given to a user who frequently spends a lot of time in a particular location for accomplish several optimization objectives, such as guaranteeing coverage of the area, reducing the amount of time needed to complete the work and the number of participants, and selecting participants with a high reputation for performance, shown in Figure 2.4.[115][110][99][118].

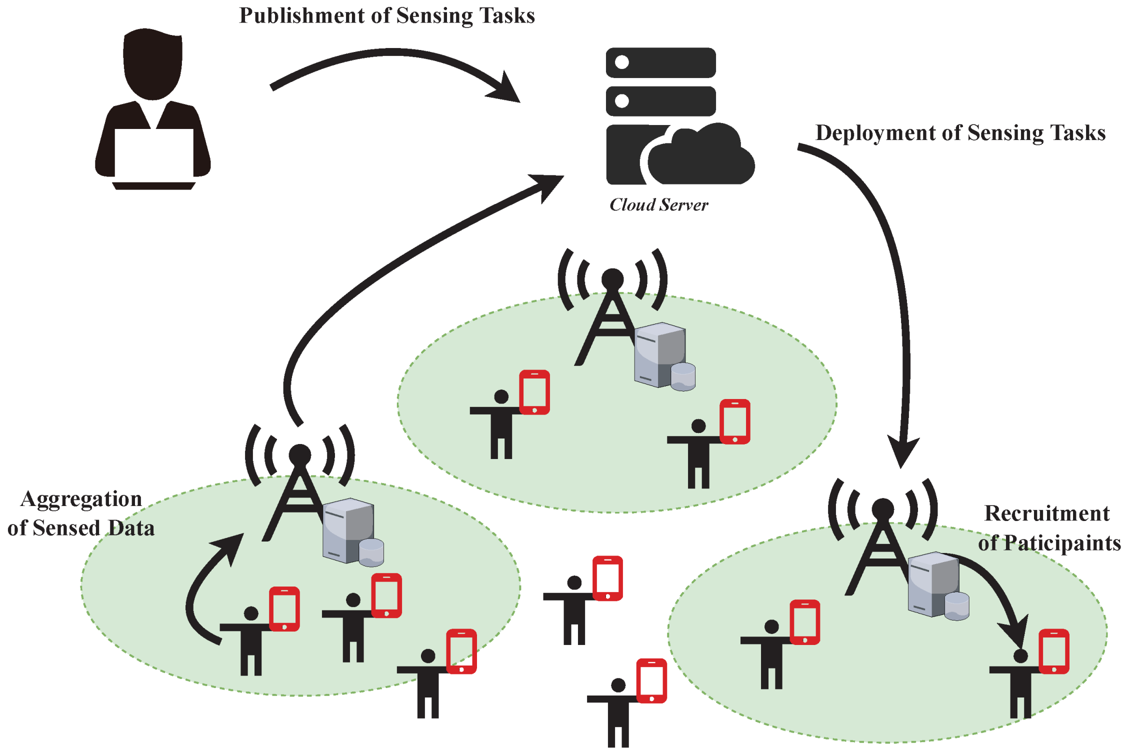


Figure 2.4: Task allocation participants for crowd-sensing.

2.3.3 Task execution

When being given the crowdsensing tasks, the employees finish them within an established spatial-temporal scale (i.e., target region and time duration).

Sensing, computing, and data uploading are all included in this condition[99].

2.3.4 Data collection

Data collecting is the foundation of MCS, is essential to obtaining precise data sensing in the future.

Previous research focused on the security and quality of data collecting to construct mobile crowdsensing systems, data collection including requests from data requesters and data providers is a crucial building piece,data gathering via taking advantage of users' mobility and the sensing capabilities provided by smartphones.

By taking advantage of the forwarding options provided by the connections between nodes, data collecting can be put into practice as required by the task organizers, this stage collects the crowdsourced data reports.

The main challenge at this point is figuring out how to fill in the gaps in the data and present a comprehensive spatial-temporal image of the target event ,with establish a user-friendly and secure platform where participants can register, submit their data, and engage in the data collection process,decide on a fair and motivating compensation mechanism, shown in Figure 2.5.[119][99][65][33].

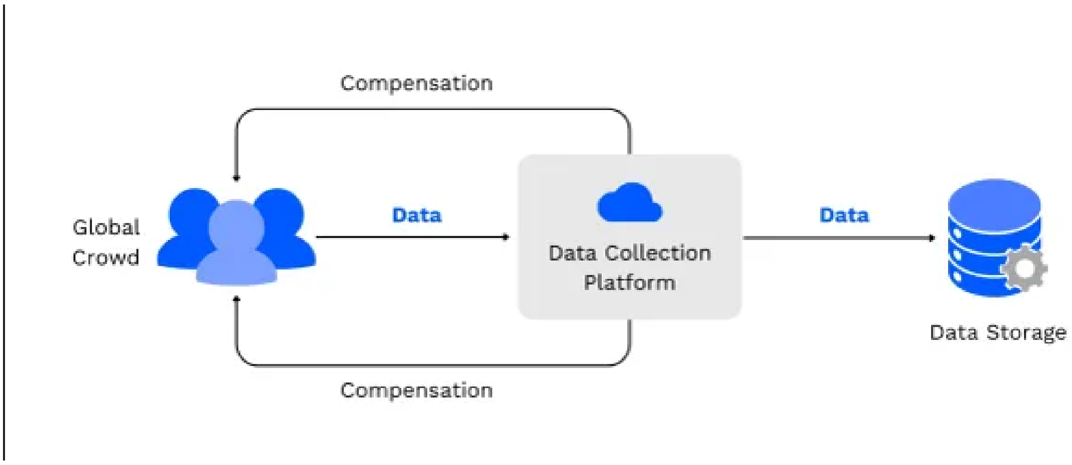


Figure 2.5: Data collection process for crowd-Sensing.

2.3.5 Data processing and analytics

This level is responsible for mining, learning, and data analysis.

Data mining and machine learning techniques are employed to extract and utilize the intended information from the gathered sensory data.

Additionally, filtered extracted data will be ensured of accuracy by removing redundant information,the platform collects data, aggregates it, and performs analysis on it in order to fulfill tasks and provide services, employ complex algorithms to combine data, pivot data, and derive collective intelligence at a high level from massive crowds' raw data.

It is true that the processing jobs may run locally on the mobile device or remotely on the mobile cloud, shown in Figure 2.6. [23][115][56].

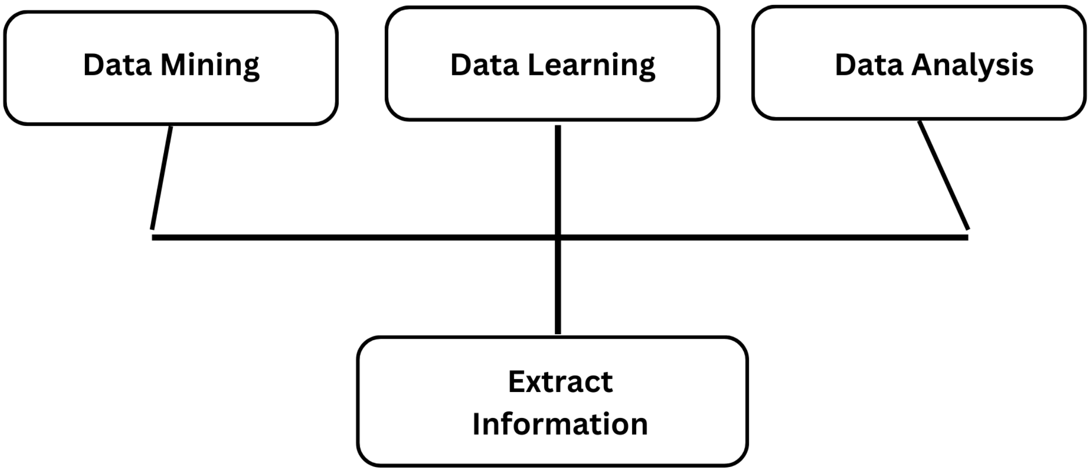


Figure 2.6: Data processing and analytics operation for crowd-sensing.

2.4 Crowdsensing Application

We classify MCS applications into three categories based on the type of phenomenon being measured or mapped,include environmental, infrastructure, and social[43].

We review state-of-the-art crowdsensing applications and projects.see figure2.7.

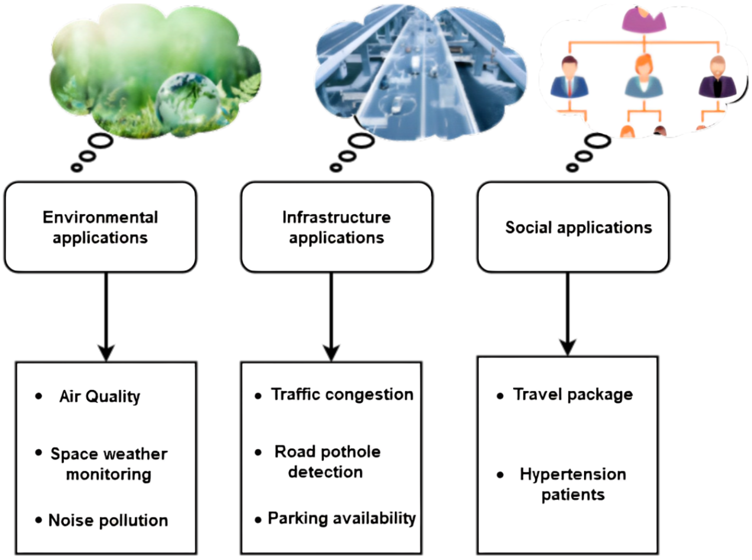


Figure 2.7: Crowdsensing applications' categories.

2.4.1 Environmental applications

Crowdsensing applications in the environmental industry are mostly focused on monitoring air and noise pollution levels and Space weather,protecting natural areas.

Based on the mobile devices of interested individuals, several scientific studies have been carried out in the past years[43][23].

2.4.1.1 Air Quality

One of the most serious environmental challenges facing today's world is air pollution,like we see maintaining good air quality is important for public health[79].

We will see many examples from our proposed research, such as:

- An exemple study presents new advancements in monitoring particulate matter (PM) in urban areas using a participatory vehicle sensor network (VSN).

  The network uses mobile low-cost IoT devices to send geolocated PM measurements to an IT infrastructure, allowing real-time reconstruction of the spatial and temporal distribution of pollutants in a web-based environment.

  The data was integrated with independent reference measurements from governmental environmental agencies.

  The infrastructure was deployed in trieste, italy, since early 2021, with the help of volunteers and the local transportation authority (Trieste Trasporti).

  By analysing the data, we delineate areas with lower air quality and identify the possible causes of these anomalies.

  We were able to define a belt outside the urban center where an enhanced concentration of pollutants occurs due to a higher flux of vehicular traffic that tends to jam there.a web based interactive geographic map of the distribution of pollutants in the designated area (Figure 2.8)[36].

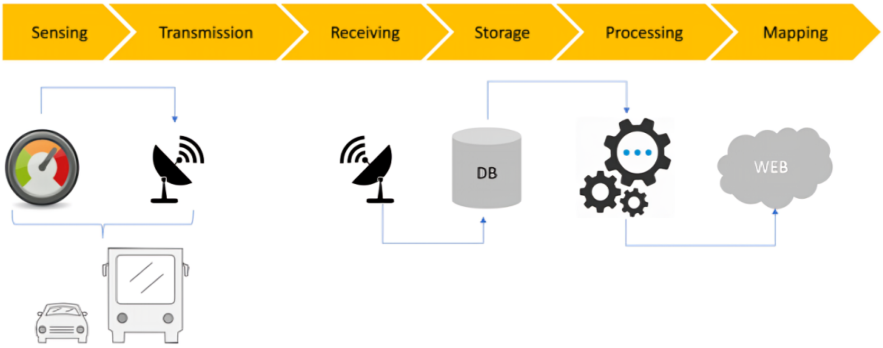


Figure 2.8: Map of the distribution of pollutants.

2.4.1.2 Space weather monitoring

The challenge in atmospheric research lies in providing accurate urban air temperature data with high spatial and temporal resolution over long periods of time[74].

See many examples from proposed research such as:

- An exemple using cloud-based microservice enabling individual users to have access to local air pressure measurements and a proof-of-concept web application providing short-term weather forecasts based only on local crowdsensed data.

  The microservice has been developed using state-of-the-art web technologies and cutting-edge technologies related to cloud computing our microservice, outlined in Fig 2.9[87].

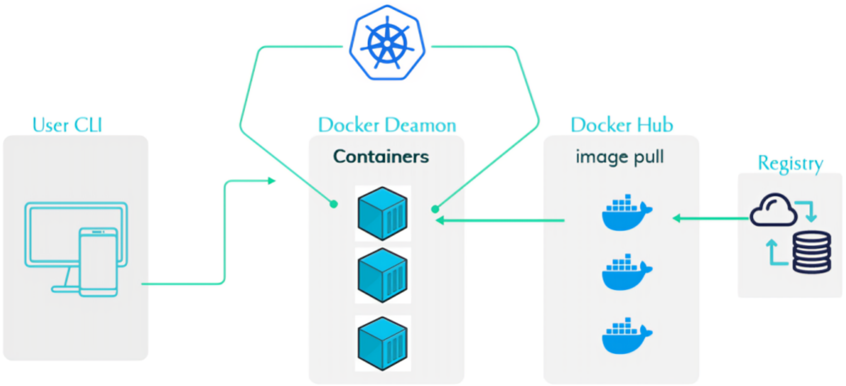


Figure 2.9: Microservice for air pressure measurements .

- Another example is The Ionosphere, a valuable and versatile sensor, has been discovered to be linked to tsunamis and earthquakes, to monitor ionospheric electron density, a dense wide-area sensor mesh is needed.

  Crowdsourcing can help increase the number of sensors and expand data transport capabilities.

  The Mahali project is exploring these promising techniques, using GPS signals to penetrate the ionosphere for science.

  Ground-based sensors will feed data through mobile devices into a cloud-based processing environment, enabling tomographic analysis of the global ionosphere at unprecedented resolution and coverage.

  This novel approach brings the ionosphere as a global earth system sensor technologically and economically within reach(Fig2.10)[80].

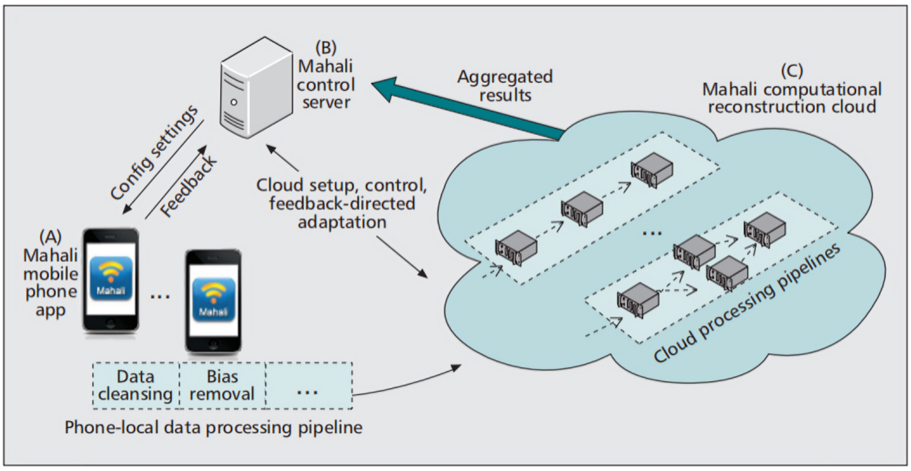


Figure 2.10: Mahali system architecture.

2.4.1.3 Noise pollution

MCS provides new opportunities for noise monitoring because he has negative influence on residents' health and well-being[23].

Will see many examples from our proposed research, such as:

- An example about Noise Map it's a crowdsensing technology has provided a promising possibility for monitoring noise pollution in large-scale areas.

  Constructing noise map by using mobile smart phones in a cost-effective manner is being widely used in the city and industrial plants.

  By using the microphone sensor in mobile devices, noise maps can be constructed by collecting sound samples from multiple individuals across different locations.

  The data is aggregated and analyzed to determine noise levels and create visual representations of noise distribution in a given area.(Figure2.11) [116].

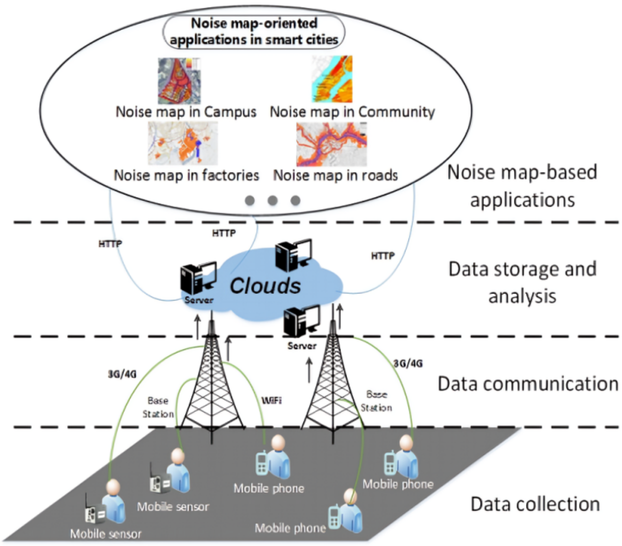


Figure 2.11: Noise map based on crowdsensing in smart cities.

- Another example is MCS-based system exploiting fiware middleware platform and allowing users to gather noise measurements (both opportunistically and participatory) in order to perform large-scale, low-cost and sufficiently accurate urban noise monitoring campaigns.

  The logical architecture of the proposed platform has a three-layer structure.

  Collected measurements are then aggregated, filtered and interpolated in order to provide city managers with an overview of the actual noise pollution levels in their cities.

Specific noise abatement measures are suggested to city managers (Fig 2.12)[114].

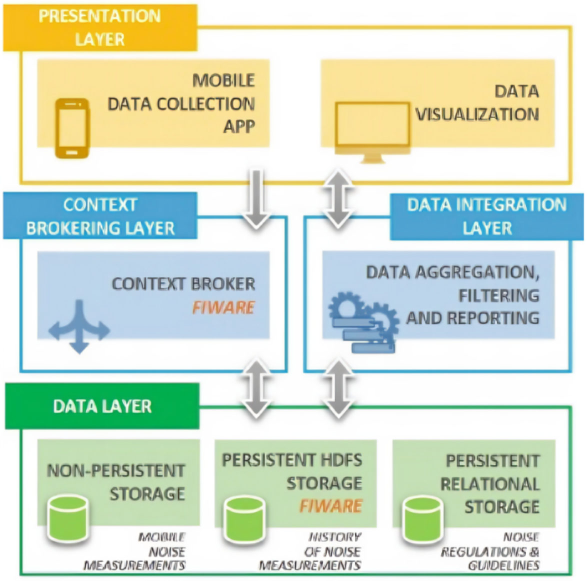


Figure 2.12: Logical architecture.

2.4.2 Infrastructure applications

Infrastructure monitoring represents a growing application field for MCS, which consists of large-scale measurement of phenomena related to public infrastructure, including traffic congestion, road pothole detection, and parking availability[23].

2.4.2.1 Traffic congestion

Traffic congestion levels in cities were measured by early MCS deployments[43].

We will see many examples from our proposed research, such as:

- Examples of which include research based on the IoT using a mobile crowd sensing traffic congestion control model provides a taxonomy of different traffic management schemes to avoid traffic congestion.

  A taxonomy was created to categorize and classify traffic congestion in urban areas, based on model type, sensor technology, data gathering techniques, selected road infrastructure, traffic flow model, and result verification approaches.

Mobile crowdsensing has been attracting more attention in traffic prediction.

In mobile crowdsensing, the vehicular traffic data are collected at a Very low cost without any special sensor network infrastructure deployment because it can transmit information faster, collect vehicle traffic data at a very low cost by using motorists' smartphone or GPS vehicular embedded sensor,Figure 2.13 shows the layered functions of the MCS-based urban traffic congestion management system[14].

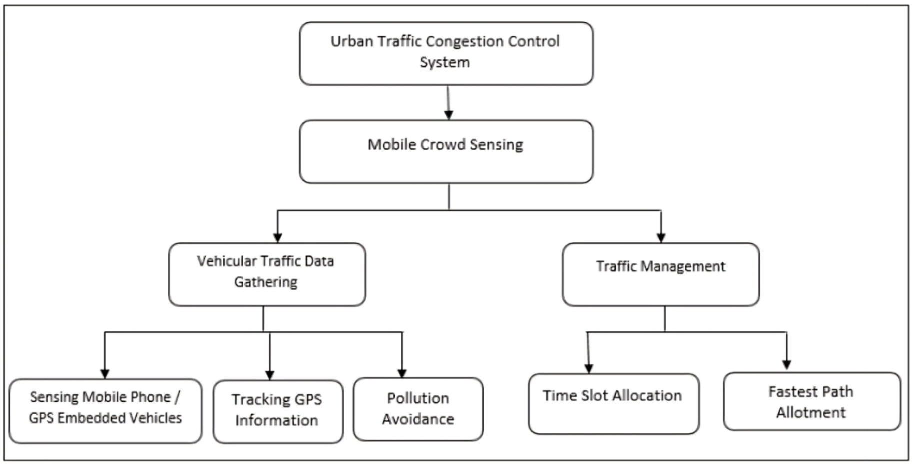


Figure 2.13: Hierarchal functionality of MCS.

- Another example is a system for traffic and road monitoring system(TRMS) which exploits the power of participatory sensing and cloud messaging is proposed.

  Crowd intelligence Which is used to estimate traffic congestion levels, arrival times, while average road speed is harvested from The crowd sensed data.

  Traffic congestion control at route level is implemented with a route guidance system, proactive warnings or recommendations for drivers in the vicinity of, or on the route to, reported events are provided.

  Drivers have the option to report short-term traffic events and physical road conditions for road monitoring,the system architecture of TRMS shown in Fig 2.14.

  It is a cloud-assisted MCS architecture that consists of mobile application components[109].

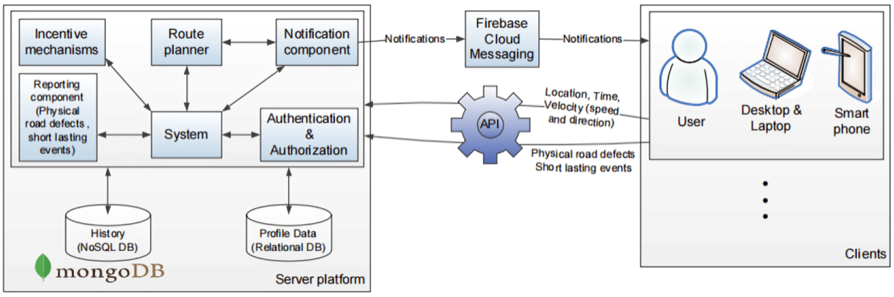


Figure 2.14: System architecture of TRMS.

2.4.2.2 Road pothole detection

Road surface monitoring and maintenance are essential for driving comfort and transportation,preserving infrastructure integrity[108].

see many examples from proposed research, such as:

- Proposes an automatic pothole detection system using the integrated vibration sensor and global positioning system receiver in a smartphone.

  We collected data on road conditions in city using dedicated vehicles and smartphones with mobile applications specifically designed for this study.

  A series of processing methods were applied to the collected data and features from different frequency domains were extracted, along with different machine learning classifiers.as shown in Figure 2.15.

  For the purpose of gathering the location data and vibration signal, the user's phone must have an application installed.

  After applying the necessary processing (such as resampling, reorientation, filtering, etc.) to the gathered data, the program splits the continuous signal into sliding window-based segments.

  In the meantime, portions that may be affected by potholes are found using straightforward thresholds and relayed to the server via the mobile communication network, along with their GPS location.

After signal modification, the server back-end gathers features from the supplied data and uses pretrained machine-learning classifiers to identify actual potholes, using the clustering method.

Potholes found from data from numerous vehicles are grouped to identify the final pothole.

A unique database is kept with the final potholes that can be used by a road maintenance department,it is possible to use this app to provide information on approaching potholes to remind drivers to slow down and be careful, as a reward for using the app.

In this conceptual framework, we prove the feasibility of the system through an offline simulation, include all steps of data processing and pothole identification (except for clustering by location)[104]

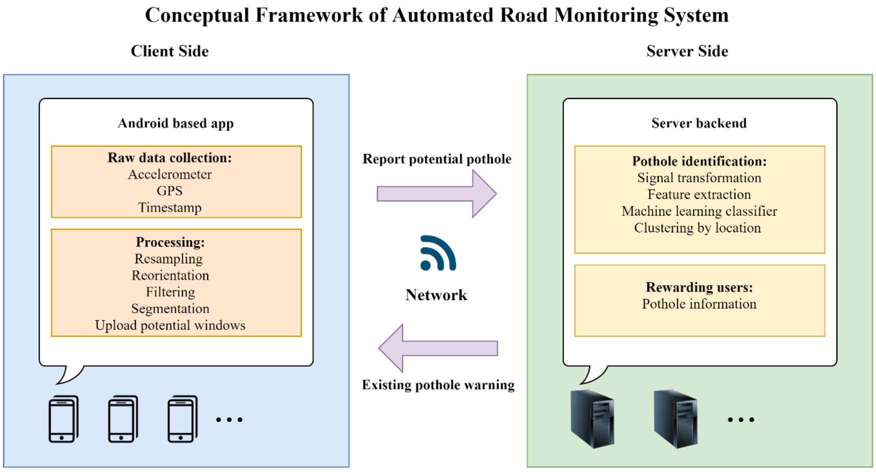


Figure 2.15: System architecture for pothole.

- Another example is proposes a road quality evaluation system capable of detecting potholes and evaluating the severity.

  The system contains a stereo camera and a laser diode that is capable of projecting multiple laser dots.

The stereo camera collects images while the laser diode projects easily identifiable marks on the pavement.

The collected images without laser dots are fed into pothole detection neural network (PDNN), which is based on You Only Look Once (YOLO) network, to detect potholes.

The detected potholes are outlined with label boxes,the laser dots are projected onto the potholes' surface and the corresponding images with the laser dots are taken by the stereo camera.

The laser dots on each picture are identified through image processing and used to calculate the distance from each mark on the potholes' surface to the camera by matching the marks on the left and right images.a rough 3D model of the pothole is constructed using the distances and the volume of each damaged area are calculated as shown in Figure 2.16.[64]

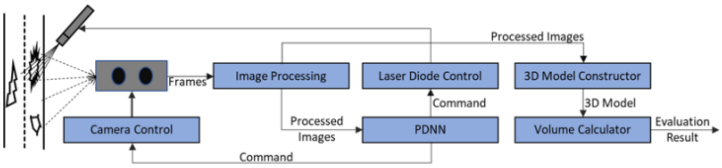


Figure 2.16: Road quality evaluation system.

2.4.2.3 Parking availability

Parking availability statistics on city streets are difficult to come by.Obtaining real-time data would have a major impact on traffic congestion in society[72].

We will see many examples from our proposed research, such as:

- We propose a crowdsensed parking system, namely ParkCrowd, to aggregate on-street and roadside parking space information reliably, and to disseminate this information to drivers in a timely manner.

  Our system not only collects and disseminates basic information, such as parking hours and price, but also provides drivers with information on the real time and future availability of parking spaces based on aggregated crowd knowledge.

To improve the reliability of the information being disseminated,by using logistic regression-based method to evaluate the reliability of crowd knowledge for real-time parking space information as shown in Figure 2.17.[88].

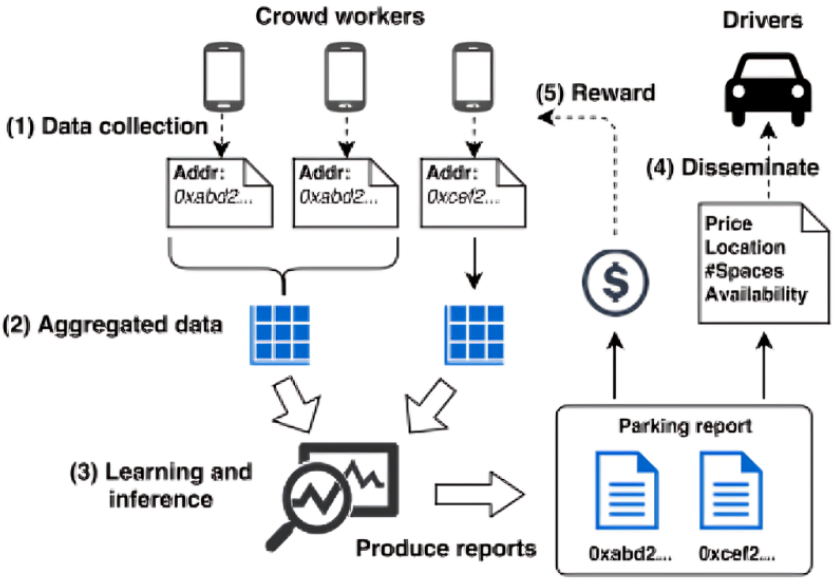


Figure 2.17: The ParkCrowd system.

- Exemple to develop an IoT-based mobile crowdsensing application to efficient tracking of parking slot status so as to notify drivers of parking availability thus leading to less driving time.

  The mobile application, SenseAPP, collects data from registered participants' smartphone sensors, and microcontrollers, gas and temperature sensors placed at different city locations, thus leveraging both crowdsensing and IoT techniques.

  The data stored on a cloud database are processed to provide data visualisation on free parking slots with customised markers.

  Exploits in-built smartphone sensors (GPS - Global Positioning System, accelerometer, gyroscope, microphone) to gather data which is stored in both cloud firestore and firebase realtime database.

Cloud functions listen to data changes to perform actions afterwards,the system relies on google APIs namely geolocation, activity recognition, places, and geofencing APIs and maps SDK (Software Development Kit).

The IoT sensors' data are stored in the firebase realtime database. Fig2.18. depicts the system components and their interactions[81].

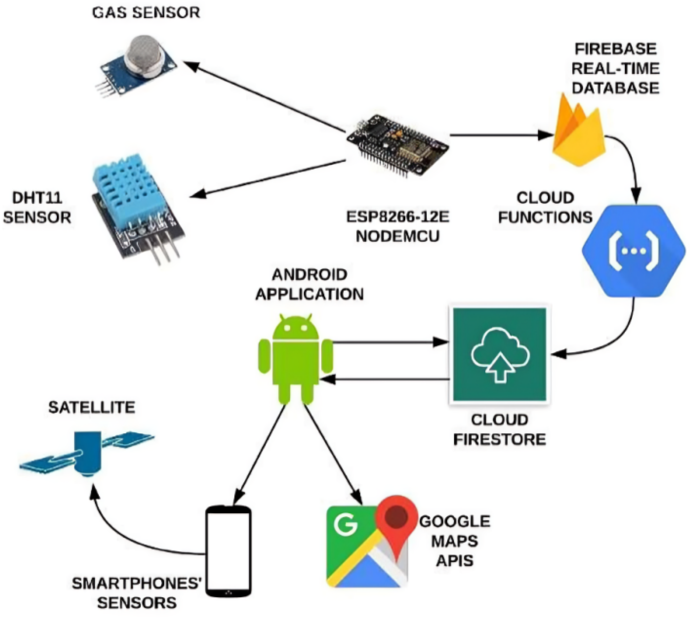


Figure 2.18: SenseAPP system architecture.

2.4.3 Social applications

The collective human power of MCS applications reveals many aspects, for example, individuals share perceived information with each other regarding recommendations and opinions, life experiences and suggestions about services/activities[23].

see many examples from our proposed research, such as:

- Exemple is Mobile data collected by location-based social networking (LBSN) services provides information about individuals' location preferences.

provide a travel package recommendation system to help users plan their trips by leveraging crowdsourced mobile data.

extract user preferences, discover points of interest (POIs), and determine location correlations from check-in records.

then create personalized travel packages that take into account user preferences, POI characteristics, and spatio-temporal constraints such as travel time and departure location as shown in Figure 2.19.[112]

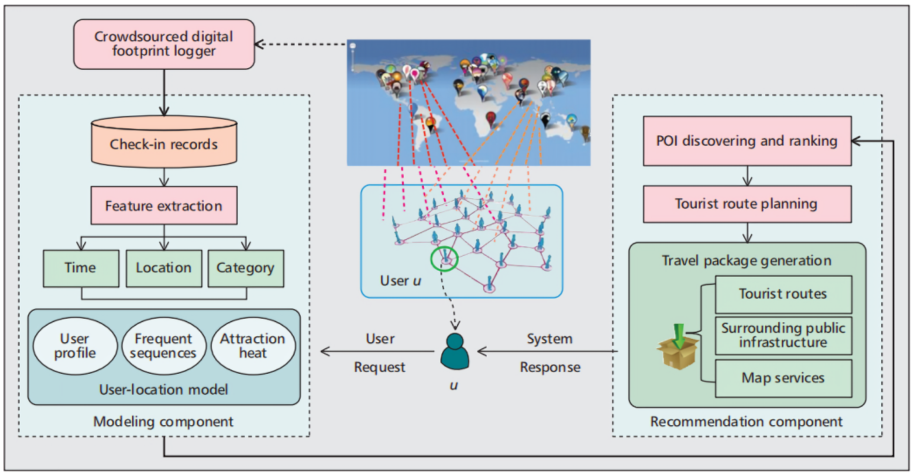


Figure 2.19: System architecture.

- Another example is a method that uses the cloud service Platform as a Service (PaaS) to gather data from hypertension patients and create an experimental database.

  Performing the study without the blood pressure, which is the primary attribute used to diagnose hypertension, represents a challenge.

  The remaining issues focus on reliability of data in a setting with a lot of artifacts, constrained battery life, and bandwidth.

  Machine learning (ML) techniques are utilized to provide feedback on the patient's present status, with the aim of motivating MCS volunteers.

  The random forest algorithm produced results out of the two approaches that examined. with minor adjustments, the suggested platform may be tailored to people with different cardiovascular issues.

The flow diagram of the procedure is shown in Figure 2.20.[50]

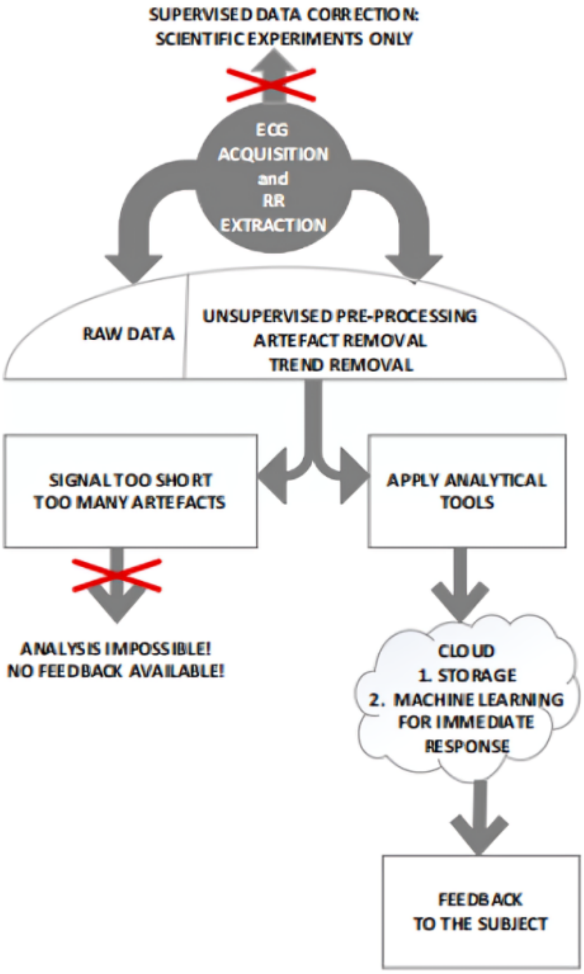


Figure 2.20: Flow diagram of the procedure.

2.5 Conclusion

In this chapter delves into the concept of crowdsensing with multiple definitions and descriptions, and how it is essential in smart cities and useful for urban sensing, with studies that have explored existing examples of applications.

Delves into the intricacies and potential of crowdsensing, exploring its definition, significance, and life cycle.

Examines the various stages of the crowdsensing life cycle, including task creation, task allocation, task execution, data collection, and data processing and analytics.

Furthermore, explores the wide range of applications that benefit from crowdsensing, such as environmental monitoring, infrastructure management, and social applications with examples of research, and how it is helpful for different phenomena and challenges such as air pollution, weather monitoring, traffic congestion, road pothole detection, and parking availability in urban environments.

In the upcoming chapter, explores obstacle detection research, focusing on its motivation and objectives.

Presents a Kinect sensor-based methodology and computer vision for road obstacle detection systems, covering the developmental stages from initial lane detection to advanced sensing with Kinect.

The chapter will discuss algorithms, hardware integration, and experimental results, including a comparison of performance metrics and in-depth analysis of outcomes.

Chapter 3

Leveraging Depth Vision for Safer Roads

Road safety is paramount for transportation systems. This chapter evaluates various computer vision techniques like Kinect, YOLO, and OpenCV for real-time obstacle detection from depth and image sensors. The effectiveness of each method is assessed through experimental results.

3.1 Introduction

The increasing emphasis on road safety and transportation efficiency has highlighted the need for more effective methods of detecting road obstacles. Traditional approaches have limitations in terms of time, cost, and spatial coverage, leading to delayed responses and increased safety risks. This chapter explores the potential of depth vision technology and computer vision, specifically the Microsoft Kinect sensor, for developing innovative and efficient solutions for road obstacle detection.

The first approach introduces a comprehensive self-driving car system equipped with sophisticated sensors and algorithms, capable of detecting both obstacles and lane markings.

The second method focuses on leveraging the capabilities of the Microsoft Kinect technology to enhance sensing abilities.

The third method delves deeper into the development of obstacle detection systems solely based on Kinect sensors.

The fourth method explores the integration of the YOLO (You Only Look Once) algorithm to advance obstacle detection accuracy.

The chapter concludes with a comparative study of the four methods for road obstacle detection and discusses future research in this area.

3.2 Motivation

Road safety has been a paramount concern globally, with millions of lives affected by accidents and collisions each year. Among the various factors contributing to road accidents, obstacles on roads pose significant challenges, especially for autonomous vehicles and individuals with visual impairments. Addressing this challenge requires innovative solutions that leverage advanced technologies to detect obstacles in real-time and mitigate potential hazards. The implementation of computer vision techniques, coupled with sensor technologies, has emerged as a promising approach to tackle obstacle detection on roads effectively. In this context, the integration of Microsoft Kinect technology offers unique capabilities, enabling precise depth perception and object recognition in dynamic environments. By harnessing Kinect's depth and color data, systems can accurately detect obstacles and enhance situational awareness for both autonomous vehicles and pedestrians. This paper presents a comprehensive methodology for developing a system for obstacle detection on roads using Kinect technology. The methodology encompasses multiple stages of research and implementation, starting from initial experimentation with self-driving car systems to the integration of advanced object detection algorithms like YOLO (You Only Look Once). Each stage of development builds upon the previous one, leveraging insights gained from experimental evaluations and user feedback to refine and enhance the system's performance. Furthermore, the proposed system goes beyond traditional obstacle detection. it aims to provide a holistic solution for enhancing road safety and navigation efficiency for all road users. Through a combination of theoretical discussions, practical implementation details, and experimental results, this paper aims to contribute to the ongoing efforts in developing intelligent systems for road safety. By leveraging the capabilities of Kinect technology and advanced computer vision algorithms, the proposed system holds the potential to significantly reduce accidents and improve overall road safety standards.

3.3 Methods

The methods proposed for obstacle detection in road scenarios offer varied approaches to address the challenges of ensuring road safety. The first method introduces the concept of a self-driving car equipped with sophisticated sensors and algorithms for both obstacle and lane detection, aiming to provide comprehensive environmental perception. In contrast, the second method focuses on leveraging Microsoft Kinect technology to enhance sensing capabilities, promising precise obstacle detection through depth and color information. Method three emphasizes the development of obstacle detection systems solely using Kinect sensors, highlighting algorithmic intricacies and real-time visualization techniques. Finally, method four advances obstacle detection accuracy by integrating the YOLO algorithm, offering heightened efficiency and performance through rigorous testing and algorithmic training. Each method represents a step forward in augmenting road safety through innovative approaches to obstacle detection.

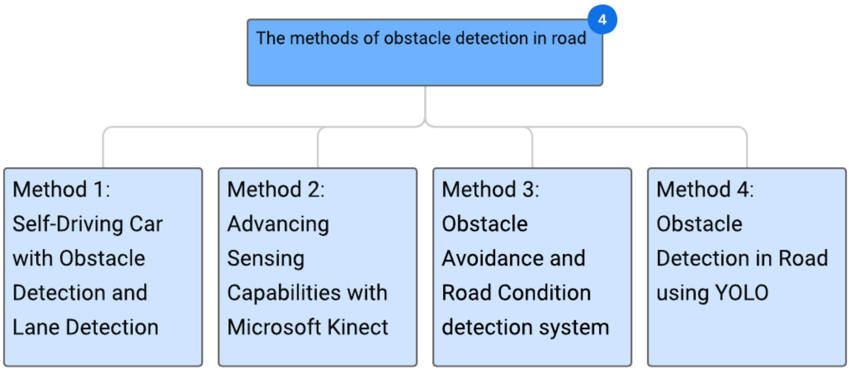


Figure 3.1: The methods proposed for obstacle detection.

3.3.1 Method 1: Self Driving Car with Obstacle Detection and Lane Detection

As self-driving technology advances, ensuring the vehicle's ability to detect and respond to obstacles while staying within lane boundaries is crucial for safe navigation. This method involves integrating obstacle detection and lane detection functionalities into an existing self-driving car system. By leveraging computer vision techniques with OpenCV and Python, the system can better interpret and react to its environment, enhancing overall driving safety and accuracy.

3.3.1.1 Background

Autonomous vehicles represent a transformative advancement in transportation, necessitating sophisticated technologies to ensure safe and efficient operation. The integration of robust obstacle detection and precise lane detection systems is fundamental to the reliability of self-driving cars. These vehicles depend on a variety of sensors, including cameras and lidar, along with advanced algorithms to interpret their surroundings and make real-time decisions. This project leverages computer vision techniques, primarily using OpenCV and Python, to enhance an autonomous vehicle's capabilities in detecting obstacles and maintaining lane discipline. By processing video input in real-time, the system can navigate safely, effectively identifying and avoiding potential hazards while adhering to lane boundaries.

3.3.1.2 Algorithm

The self-driving car algorithm follows a systematic approach to navigate safely and autonomously:

Algorithm 1 Self-Driving Car Algorithm

```
Input: Video stream from the car's camera
Output: Steering commands for the vehicle,Lane detection
procedure LANE DETECTION
    while True do
        #Histogram Analysis:
        $histogram \leftarrow \textsc{ComputeHistogram}(image)$
        $peak\_left, peak\_right \leftarrow \textsc{FindPeaks}(histogram)$
        #Sliding Windows:
        $left\_fit, right\_fit \leftarrow \textsc{SlidingWindows}(image, peak\_left, peak\_right)$
        #Curvature Calculation:
        $left\_curvature, right\_curvature \leftarrow \textsc{CalculateCurvature}(left\_fit, right\_fit)$
        #Vehicle Position:
        $vehicle\_position \leftarrow \textsc{CalculateVehiclePosition}(left\_fit, right\_fit)$
        #Output Steering Commands:
        $\textsc{SteerVehicle}(left\_curvature, right\_curvature, vehicle\_position)$
    end while
end procedure
```

The structured workflow followed in implementing the self-driving car algorithm is given below in Figure 3.2

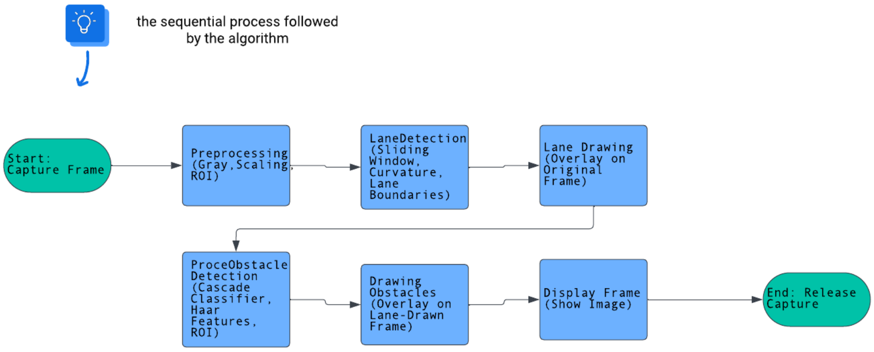


Figure 3.2: The sequential steps involved in self-driving car system.

The self-driving car algorithm initiates by capturing a frame from the video stream, which serves as the starting point for all subsequent analysis. Once captured, the frame undergoes preprocessing steps, including grayscale conversion and defining the region of interest (ROI), ensuring that only relevant portions of the image are analyzed.

Next, the algorithm focuses on detecting lane boundaries using advanced techniques like sliding window analysis and curvature calculation. Detected lanes are then visually overlaid onto the original frame, providing a clear representation of the car's perception of the road.

Simultaneously, the algorithm analyzes the preprocessed frame to identify potential obstacles using methods such as Cascade Classifier and haar Features within the predefined ROI. Detected obstacles are also visually overlaid onto the frame, enhancing visualization and aiding in understanding the car's environment.

To provide a comprehensive view of the scene, the final frame with lane markings and detected obstacles is displayed, enabling real-time monitoring of the car's surroundings. Finally, the algorithm concludes by releasing capture resources, ensuring efficient resource management for subsequent frames. This stepwise process ensures that the self-driving car algorithm effectively perceives and navigates its environment in real-time.

3.3.1.3 Explanation

Here's a breakdown of each step in the algorithm with a brief explanation:

1. **Camera Calibration**:
    - **Input**: Images of a chessboard pattern taken from different angles.
    - **Output**: Intrinsic parameters and distortion coefficients of the camera.
    - **Explanation**: Calibration corrects for distortion caused by the camera lens, ensuring accurate image representation.
2. **Image Undistortion**:
    - **Input**: Distorted images captured by the camera.
    - **Output**: Undistorted images using the camera's calibration parameters.
    - **Explanation**: Applying the calibration parameters removes distortion from the captured images.

3. **Pipeline Image Processing**:
    - **Input**: Undistorted images.
    - **Output**: Processed images ready for feature extraction.
    - **Explanation**: Preprocessing steps such as color space conversion and gradient thresholding prepare the images for further analysis.
4. **Circular Object Detection**:
    - **Input**: Processed images.
    - **Output**: Detected circular objects (e.g., traffic signs, obstacles).
    - **Explanation**: Identifying circular objects using techniques like Hough Circle Transform to enhance obstacle detection.
5. **Color and Gradient Thresholding**:
    - **Input**: Processed images.
    - **Output**: Binary images highlighting lane lines.
    - **Explanation**: Applying thresholds based on color and gradient information to isolate lane features.
6. **Perspective Transform**:
    - **Input**: Binary images of lane lines.
    - **Output**: Warped images with a top-down view.
    - **Explanation**: Transforming the perspective of the road images simplifies lane detection by making lines parallel.
7. **Inverse Perspective Transform**:
    - **Input**: Detected lane lines in the top-down view.
    - **Output**: Overlay of lane lines on the original road view.

- **Explanation**: Converting the detected lane lines back to the original perspective for visualization purposes.

8. **Lane Detection**:
    - **Input**: Warped binary images.
    - **Output**: Detected lane lines.
    - **Explanation**: Identifying lane lines using techniques like sliding windows and polynomial fitting.
9. **Histogram Analysis**:
    - **Input**: Detected lane lines.
    - **Output**: Histogram of lane pixel distribution.
    - **Explanation**: Analyzing the histogram to determine the base points of the lane lines.
10. **Sliding Windows**:
    - **Input**: Histogram analysis results.
    - **Output**: Fitted polynomials representing lane boundaries.
    - **Explanation**: Using a sliding window approach to detect and track lane pixels for polynomial fitting.
11. **Curvature Calculation**:
    - **Input**: Fitted polynomials of lane lines.
    - **Output**: Curvature of the detected lanes.
    - **Explanation**: Computing the curvature of the lane lines to assess road curvature and navigation requirements.
12. **Vehicle Position**:

  - **Input**: Fitted polynomials of lane lines.
  - **Output**: Position of the vehicle relative to the lane center.
  - **Explanation**: Determining the vehicle's lateral position within the lane for navigation control.

13. **Obstacle Detection**:
    - **Input**: Processed images.
    - **Output**: Detected obstacles (e.g., cars, pedestrians).
    - **Explanation**: Using classifiers to identify and locate obstacles in the environment.

14. **Steering Control**:
    - **Input**: Curvature, vehicle position, and obstacle detection results.
    - **Output**: Steering commands for the vehicle.
    - **Explanation**: Generating steering commands based on lane curvature, vehicle position, and obstacle presence for autonomous navigation.

The self-driving car algorithm is a system that uses sensor data from a vehicle to identify lane markings and obstacles. The driver provides input to the system, and the system provides output in the form of lane markings and obstacle information. The steering control system uses this information to adjust the vehicle's steering.

3.3.1.4 Experimental Results

To assess the performance of the developed system, a comprehensive experimental evaluation was conducted.

The evaluation focused on several key aspects, including lane detection accuracy, obstacle detection accuracy, robustness in various conditions, processing speed, control command accuracy, and usability.

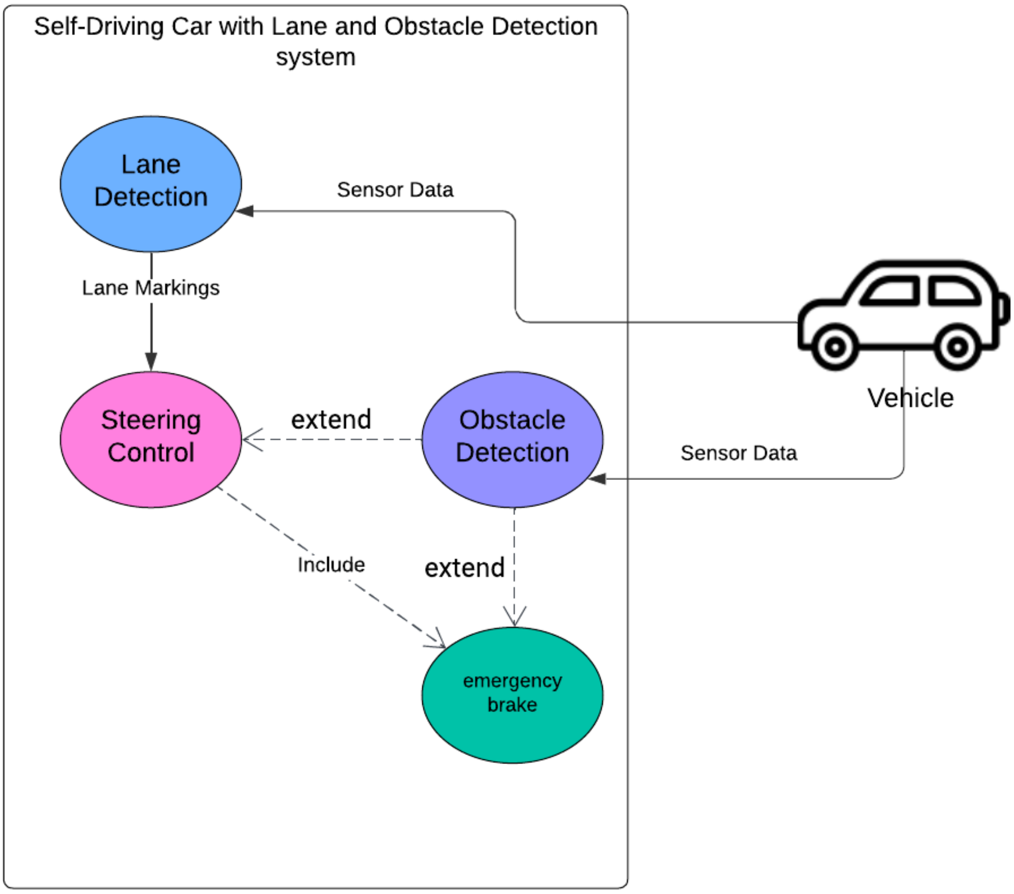


Figure 3.3: Use Case Diagram for Self-Driving Car with Lane and Obstacle Detection

The results of the evaluation are summarized in table 3.1 below:

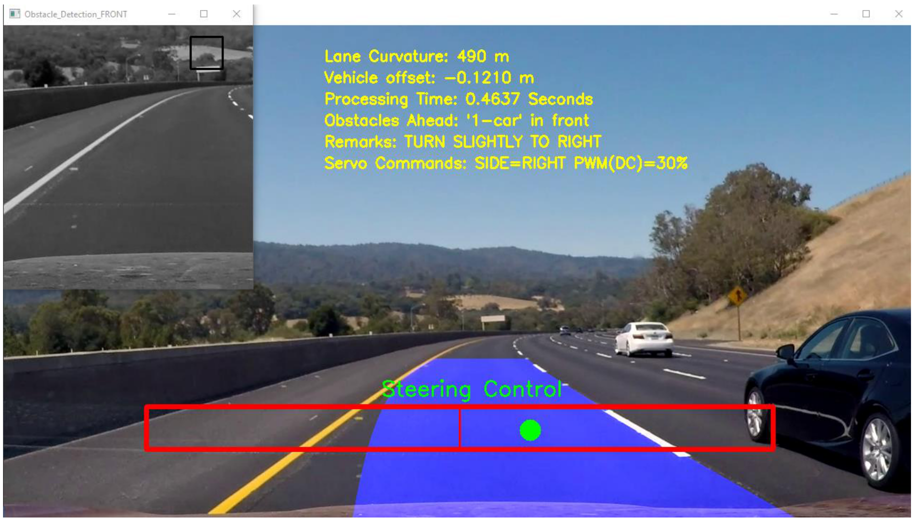
Aspect	**Evaluation**
Lane Detection Accuracy	The lane detection algorithm exhibited high accuracy in identifying lane lines under varying road conditions, with an average detection accuracy of over 80
Obstacle Detection Accuracy	The obstacle detection module exhibited inconsistent performance in identifying various types of obstacles, with an accuracy rate of approximately 40
Robustness in Various Conditions	The system's performance was inconsistent across different lighting, weather, and road conditions, exhibiting significant degradation in accuracy and reliability.
Processing Speed	On average, the system processed each frame of the video stream in approximately 50 milliseconds, ensuring real-time responsiveness, but it took approximately 1 minute to detect obstacles in the video stream.
Control Command Accuracy	Control commands generated based on lane position and obstacle detection were highly accurate, with an error margin of less than 15
Usability	its usability in practical scenarios.
Challenges and Limitations Challenges	Despite its overall effectiveness, the system faced challenges in accurately detecting lane lines under extreme weather conditions such as heavy rain or fog. Further research is needed to enhance performance in such scenarios

Table 3.1: Summary of Experimental Results

To visually complement the summarized experimental findings, the following images showcases the execution results of the developed system. This snapshot provides a real-world glimpse into how the code performs under various conditions and offers insights into its practical implications. By juxtaposing these execution outcomes with the experimental evaluation, a comprehensive understanding of the system's capabilities and limitations emerges.

```
Lane Curvature: 1252 m
Vehicle offset: 0.9593 m
Processing Time: 0.5307 Seconds
Obstacles Ahead: No Obstacles in Front
Remarks: TURN HARDLY TO RIGHT
Servo Commands: SIDE=RIGHT PWM(DC)=75%
```

Figure 3.4: self-driving car system.



```
Lane Curvature: 490 m
Vehicle offset: -0.1210 m
Processing Time: 0.4637 Seconds
Obstacles Ahead: '1-car' in front
Remarks: TURN SLIGHTLY TO RIGHT
Servo Commands: SIDE=RIGHT PWM(DC)=30%
```

Figure 3.5: Self-driving car system with obstacle detection .

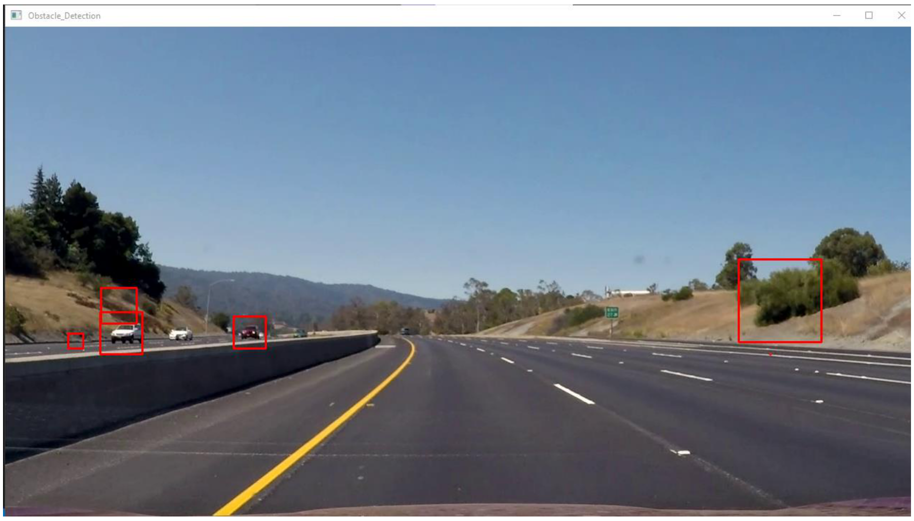


Figure 3.6: Obstacle detection window .

In the pursuit of effective crowdsensing monitoring, our endeavor led us to develop a code framework aimed at real-time obstacle detection and lane tracking, crucial elements in ensuring road safety and navigation efficiency. Despite the absence of sensor integration in this code implementation, our objective was to harness computer vision techniques to achieve comparable results. However, upon rigorous testing, we encountered a limitation in the real-time detection capability, a critical aspect for timely decision-making in dynamic environments.

While exploring potential enhancements, we explored integration possibilities with Kinect sensors but found the compatibility to be unsatisfactory, leading us to transition to the next method of development. This pivotal phase will involve further refinement and innovation to align our system with the stringent requirements of crowdsensing monitoring. Through meticulous iteration and exploration, we remain committed to advancing towards a robust and dependable solution that meets the demands of real-world applications.

3.3.2 Method 2: Advancing Sensing Capabilities with Microsoft Kinect

This section focuses on the application of the Microsoft Kinect sensor for obstacle detection in road safety scenarios. Building upon its initial development for enhancing safety the system leverages Kinect's depth and color data to identify and track obstacles on roads.

3.3.2.1 Background

Road obstacle detection is a critical component in autonomous driving and advanced driver assistance systems (ADAS). Efficient and accurate detection of obstacles such as vehicles, pedestrians, traffic signs, and other hazards is essential for safe navigation. Traditional methods often rely on camera-based vision systems, LIDAR, or RADAR. However, integrating depth-sensing technology like Microsoft Kinect can significantly enhance the system's ability to perceive and understand the environment. The Kinect sensor provides both color and depth information, enabling a more robust and detailed analysis of the road scene. This method leverages Kinect's capabilities to detect and track obstacles, enhancing the safety and reliability of autonomous systems.

3.3.2.2 Algorithm

The algorithm involves several key steps:

Algorithm 2 Road Obstacle Detection Using Microsoft Kinect

```
Input: Video stream.
Output: Obstacle detection with notification.
procedure newSensor_AllFramesReady(sender, e)
    if e.OpenDepthImageFrame() is not null then
        image1 ← new Image⟨Bgr, byte⟩(e.OpenDepthImageFrame().ToBitmap())
        image2, image3 ← copy(image1)
        makeMaskOnce(image1.Width, image1.Height)
        mog2.Apply(image2, mask)
        show_image("Mask", mask)
        contours_detected ← new VectorOfVectorOfPoint()
        find_contours(mask, contours_detected)
        contours_array ← new List⟨VectorOfPoint⟩()
        for all $i \in [0, \text{contours\_detected.Size} - 1]$ do
            contours_array.Add(contours_detected[i])
        end for
        max_contour ← contours_array[0]
        for all $i \in [0, \text{contours\_array.Count} - 1]$ do
            if contour_area(contours_array[i]) > 400 then
                if contour_area(max_contour) < contour_area(contours_array[i]) then
                    max_contour ← contours_array[i]
                    print(contour_area(max_contour))
                end if
            end if
        end for
        show_image("Image", image3)
    end if
end procedure
```

3.3.2.3 Explanation

This algorithm utilizes the Microsoft Kinect sensor to monitor roads continuously, detecting obstacles and potential falls to enhance driver awareness and prevent accidents. It starts by initializing the Kinect sensor and enabling streams for depth, color, and skeleton data capture. With each frame, background subtraction isolates moving objects, and contours are detected to identify potential obstacles. Concurrently, the algorithm tracks the movement of the first detected skeleton to detect falls based on predefined criteria.

Detected obstacles and falls are visualized by overlaying bounding rectangles on captured frames. Overall, this algorithm offers real-time detection and response to hazards, improving road safety and empowering drivers to react promptly to avoid accidents.

Step	Description
Initialization	Set up the Kinect sensor
Frame Processing	Capture and analyze frames to identify moving objects
Obstacle Detection	Detect and classify obstacles based on their size and shape
Skeleton Tracking	Track the movement of a skeleton
Fall Detection	Analyze skeleton data to detect potential falls
Visualization	Display the results of obstacle and fall detection

Table 3.2: Summary of Algorithm 2

Kinect Sensor Overview

Kinect, a motion-sensing input device developed by Microsoft, originated from the Project Natal initiative. Initially conceived as an accessory for the Xbox 360 gaming console, Kinect enables users to interact with games through body motion, eliminating the need for controllers like Sony's PlayStation Move or Nintendo's Wiimote. Microsoft released the first version of the Kinect Software Development Kit (SDK v1) on February 1, 2012, alongside an upgraded version of the device itself [13].

Figure 3.7: Microsoft Kinect input device (Microsoft, 2012)

Physical Device

The Microsoft Kinect is a multifunctional device equipped with a range of sensors designed for use with either the Xbox 360 gaming console or Windows PC applications. Capable of detecting movements, identifying faces, and recognizing speech, the Kinect utilizes a suite of sensors to capture image, audio, and depth information. One notable feature is its ability to operate without requiring players to wear any additional equipment. The device, depicted in Figure 3.8, consists of three types of sensors housed within a horizontal bar.

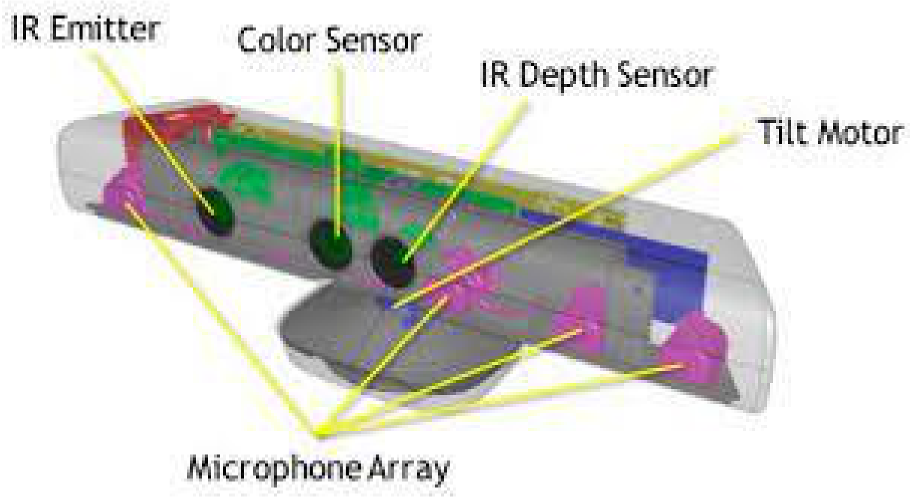


Figure 3.8: Internal components of Kinect sensors (Microsoft, 2012).

The Kinect sensors can be categorized into three main components, as outlined by Crawford [32]:

- **Colour VGA video camera:** This camera facilitates facial recognition and other features by detecting three primary colors: red, green, and blue.
- **Depth sensor:** Comprising an infrared laser projector and a monochrome sensor positioned around the RGB camera, this sensor pair generates depth information. One emitter emits infrared rays while the other detects them. The monochrome CMOS sensor captures this data, allowing the device to provide 3D motion capture capabilities.
- **Multi-microphone array:** Positioned horizontally along the Kinect device, this array of microphones isolates the user's voice from ambient noise, enhancing sound quality and enabling features like noise cancellation, echo cancellation, and sound source localization.

Additionally, the Kinect device features an electric motor controlled by software to adjust the visual field's inclination. This feature, outlined in Table 1, allows for dynamic control over the device's viewing angle. Furthermore, the Kinect utilizes sensors to recognize and track the skeleton and its joints. By detecting the reflection of infrared light, the device measures depth, creating a 3D environment. This information is then processed by software to monitor the skeleton and joints, as explained by LaBelle (2011). Table 3.3 presents the specifications of the Kinect device, as provided by Microsoft (2012).

Kinect Sensor	**Array Specifications**
Sensor item	Specification range
Viewing angle	43° vertical by 57° horizontal field of view
Mechanized tilt range (vertical)	±27°
Frame rate (depth and colour stream)	30 frames per second (FPS)
Default resolution, depth stream	VGA (640 x 480)
Default resolution, colour stream	VGA (640 x 480)
Audio format	16-kHz, 16-bit mono pulse code modulation (PCM)
Audio input characteristics	A four-microphone array with 24-bit analogto-digital converter (ADC) and Kinect-resident signal processing, such as acoustic echo cancellation and noise suppression

Table 3.3: System Performance Metrics and User Feedback

Kinect Drivers and SDK

The content in this sub-paragraph refers to the official documentation titled "Kinect for Windows SDK," found within the "Programming Guide" section and the "Natural User Interface" subsection (Microsoft, 2012). This documentation is accessible offline upon installing the official SDK. Within the Kinect for Windows SDK, users can access comprehensive documentation, APIs, and drivers necessary for utilizing the Kinect device. Figure 3.9visually represents the interaction between the Kinect device, the software library, and various applications. The NUI (Natural User Interface) library, an integral component of the SDK, equips developers with tools and APIs enabling them to harness real-world movements detected by the Kinect device.

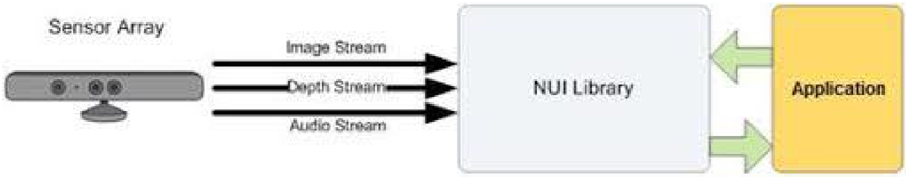


Figure 3.9: Hardware and software interaction with an application (Microsoft, 2012).

The components of the SDK are shown in the Figure 3.10. The highlighted red rectangles are the components of Figure 6. In addition, Figure 3.10 shows more details on the interactions between several components.

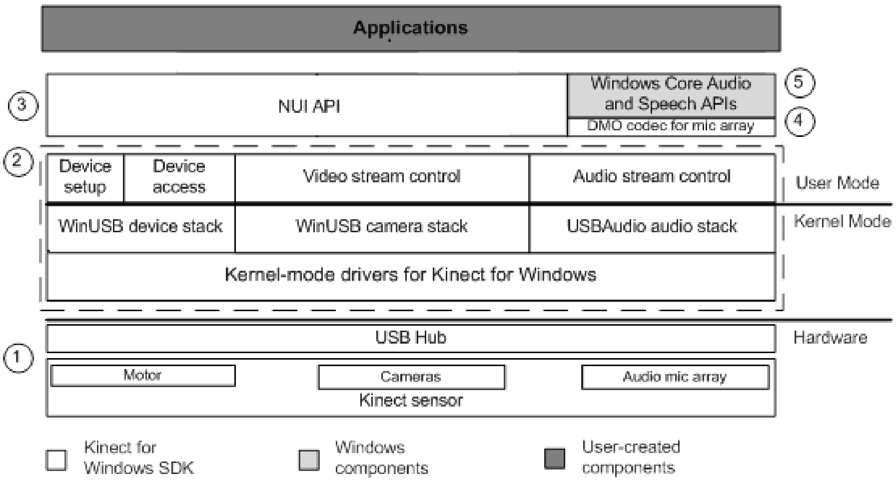


Figure 3.10: SDK Architecture (Microsoft, 2012).

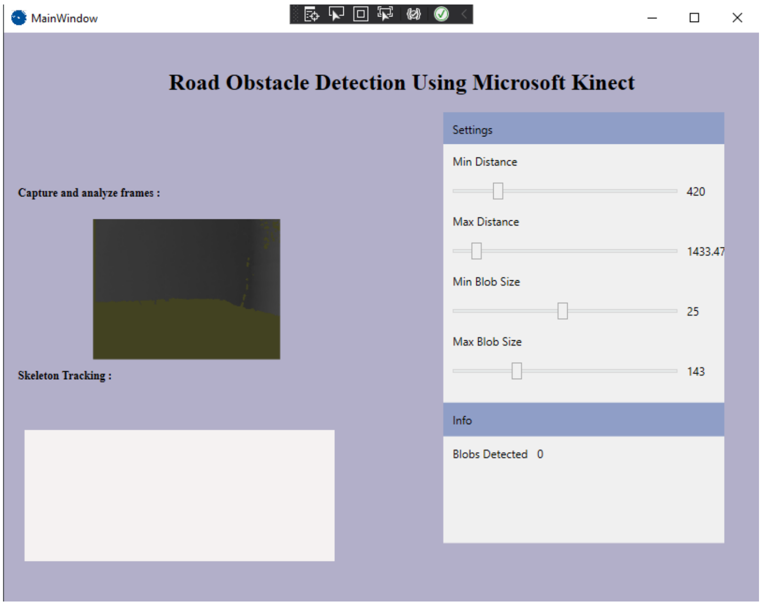
3.3.2.4 Experimental Results

The experimental results of the system employing the Microsoft Kinect sensor and Emgu.CV library demonstrate its potential for enhancing safety by detecting obstacles on roads. The system exhibited commendable performance in identifying obstacles, particularly those with simple shapes and clear visibility. However, certain limitations were observed when dealing with complex shapes or partially obscured objects.

Metrics	**Performance**
Object Detection Accuracy	Occasional inaccuracies in complex shapes
Fall Detection Accuracy	Commendable
Processing Speed	Efficient
Memory Usage	Moderate
Robustness	Resilient under various conditions
User Feedback	Positive
Areas for Improvement	Object detection precision

Table 3.4: System Performance Metrics and User Feedback

In addition to textual descriptions, images of the system's results (Figure 3.11, Figure 3.12) offer a visual representation of its performance. These images can effectively convey key metrics, such as object detection accuracy and user feedback, in a clear and intuitive manner.



MainWindow

Road Obstacle Detection Using Microsoft Kinect

Capture and analyze frames :

Skeleton Tracking :

Settings

Min Distance 420

Max Distance 1433.47

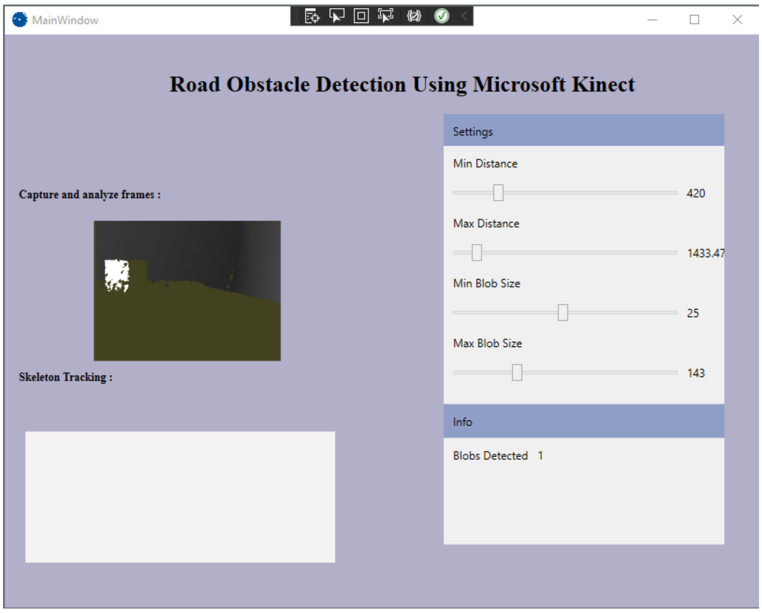
Min Blob Size 25

Max Blob Size 143

Info

Blobs Detected 0

Figure 3.11: Obstacle detection system.



MainWindow

Road Obstacle Detection Using Microsoft Kinect

Capture and analyze frames :

Skeleton Tracking :

Settings

Min Distance 420

Max Distance 1433.47

Min Blob Size 25

Max Blob Size 143

Info

Blobs Detected 1

Figure 3.12: Obstacle detection system.

- **Main Window:** It likely contains controls and displays for interacting with the Kinect sensor and visualizing the output, such as buttons, labels, or images.

- **Mask Window:** This window displays the mask generated by the obstacle detection algorithm. The mask represents areas of interest in the captured image where potential obstacles are detected. It helps visualize the regions that are being analyzed for obstacle detection.

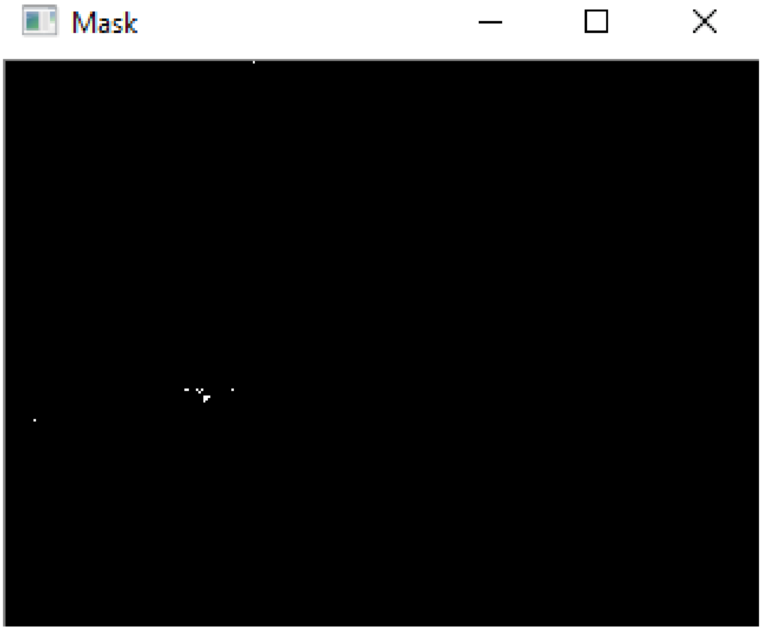


Figure 3.13: The mask window.

- **Image Window:** This window displays the processed image with detected obstacles and fall events overlaid. It provides a visual representation of the environment captured by the Kinect sensor, highlighting areas where obstacles are detected and indicating whether a fall event has occurred based on the analysis of skeletal data.

Figure 3.14: The image window.

By leveraging the advanced sensing capabilities of the Microsoft Kinect, this method enhances the detection and monitoring capabilities of autonomous systems, contributing to improved safety and reliability.

3.3.3 Method 3: obstacle avoidance and road crossing system using the Kinect sensor

This method aims to design and develop the Obstacle Avoidance and Road Crossing System (OARC System) that leverages the capabilities of the Kinect sensor. The system employs computer vision technology to detect obstacles on roads, aiding individuals with visual impairments in navigating safely. Moreover, speech recognition technology is integrated to enable users to interact with the system through voice commands, thereby enhancing usability and accessibility.

3.3.3.1 Background

The Kinect sensor, originally developed for gaming, offers advanced depth sensing and audio capabilities, making it a suitable tool for enhancing mobility aids. Its ability to capture detailed 3D information about the environment allows for more accurate and comprehensive detection of obstacles. By leveraging this technology, the OARC system aims to provide a more effective and user-friendly solution for individuals with visual impairments.

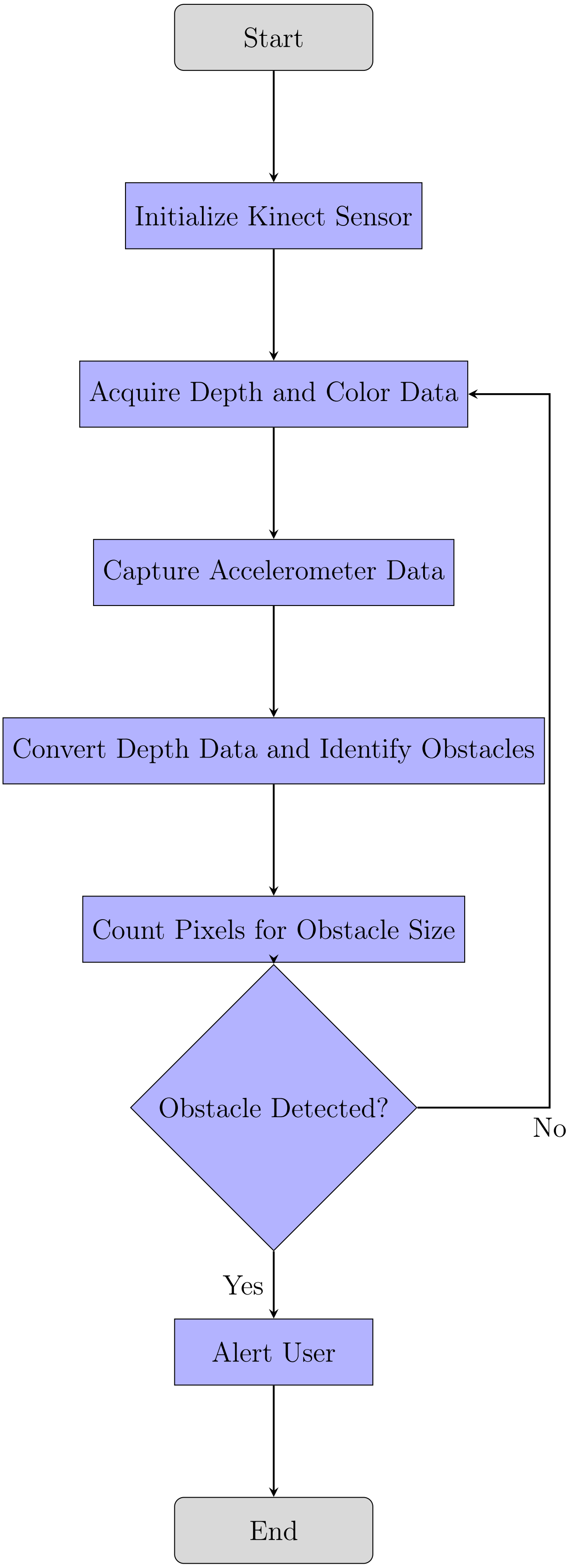
3.3.3.2 Algorithm

Here is the algorithm of OARC System using the Kinect sensor:

Algorithm 3 Obstacle Avoidance and Road Crossing System (OARC System) Using Kinect Sensor

```
Input:Capture frame
Output: Obstacle detection with notification.
procedure SensorDepthFrameReady(sender, e)
    let aDepthSliceFrame $\leftarrow$ e(OpenDepthImageFrame)
    if aDepthSliceFrame $\neq$ null then
        let minDepth $\leftarrow$ aDepthSliceFrame.MinDepth
        let maxDepth $\leftarrow$ aDepthSliceFrame.MaxDepth
        let pixCount $\leftarrow 0$
        let colorPixelIndex $\leftarrow 0$
        for each i in 0 to length(kinectDepthPixels) step 1 do
            let depth $\leftarrow$ kinectDepthPixels[i].Depth
            if depth $>$ nearRangeValue and depth $<$ farRangeValue then
                let pixCount $\leftarrow$ pixCount $+ 1$
            end if
            let colourIntensity $\leftarrow$ if depth $\geq$ minDepth and depth $\leq$ maxDepth then depth else 0
            let kinectColorPixels[colorPixelIndex] $\leftarrow$ colourIntensity
            let kinectColorPixels[colorPixelIndex $+ 1$] $\leftarrow$ colourIntensity
            let kinectColorPixels[colorPixelIndex $+ 2$] $\leftarrow$ colourIntensity
            let colorPixelIndex $\leftarrow$ colorPixelIndex $+ 4$
        end for
        if pixCount $< 200$ then
            let voiceWaitTimer $\leftarrow 8$
        end if
        colorBitmap(WritePixels)(new Int32Rect(0, 0, colorBitmap.PixelWidth, colorBitmap.PixelHeight), kinectColorPixels, colorBitmap.PixelWidth * sizeof(int), 0)
    end if
end procedure
```

This simplified workflow diagram includes the key steps of the algorithm process:



Here is a brief explanation of each step in the workflow diagram for the Obstacle Avoidance and Road Crossing System (OARC System) using the Kinect sensor:

1. **Start**: This is the initial point where the system begins its operation.

2. **Initialize Kinect**: At this step, the Kinect sensor is set up and initialized. This involves configuring the sensor to start capturing depth and RGB data from the environment.

3. **Acquire Data**: The system continuously captures data from the Kinect sensor, including depth information and color images. This raw data is essential for subsequent processing.

4. **Identify Obstacles**: The captured data is processed using computer vision algorithms to detect obstacles in the environment. This involves analyzing the depth information to identify objects that are within the path of the user.

5. **Decision - Path Clear?**: The system evaluates whether the path ahead is clear of obstacles. This is a decision point in the workflow.

    - **Yes**: If the path is clear, the system proceeds to the next step.
    - **No**: If obstacles are detected, the system provides feedback to the user, indicating the presence of obstacles and potentially suggesting alternative paths.

6. **Navigate/Cross Road**: If the path is clear, the system assists the user in navigating through the environment or crossing the road. This step may include giving directional cues or instructions based on the user's location and intended destination.

7. **Provide Feedback**: Throughout the process, the system provides real-time feedback to the user via audio cues or vibrations. This feedback helps users make informed decisions about their movements.

8. **End**: The process completes once the user has successfully navigated the environment or crossed the road. The system can loop back to acquiring data for continuous operation.

3.3.3.3 Explanation

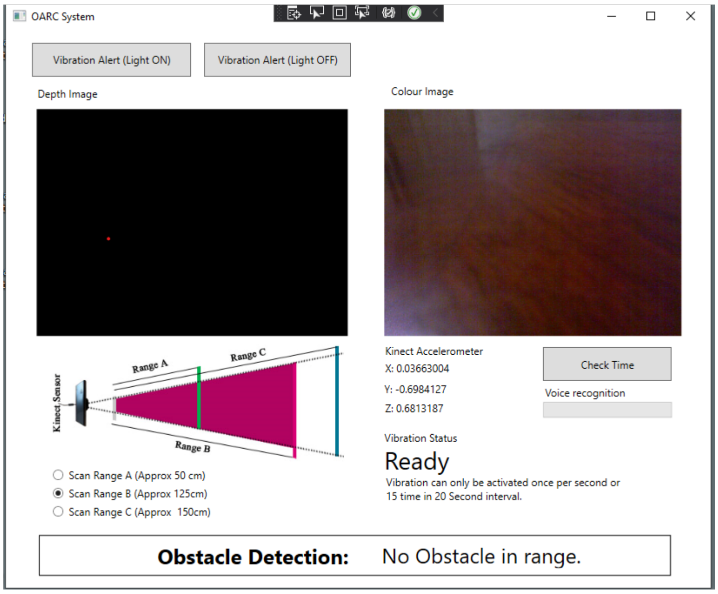
In the OARC System's obstacle detection algorithm, the Kinect sensor first gathers depth data and color images of the surroundings. This data undergoes computer vision processing, where algorithms analyze depth information to pinpoint objects in the user's vicinity. By detecting significant changes in distance, potential obstacles are identified. Through spatial analysis, objects within a predefined range are classified as obstacles, signaling potential hazards to the user. Real-time feedback, such as audio cues or vibrations, alerts users to obstacles, empowering them to navigate safely. This algorithm fosters informed decision-making, enabling individuals with visual impairments to negotiate their environment with greater confidence.

Step	**Description**
1. Data Acquisition	Kinect sensor captures depth data and color images.
2. Preprocessing	Raw data undergoes preprocessing for analysis.
3. Depth Analysis	Computer vision algorithms analyze depth data.
4. Obstacle Detection	Objects within a predefined range are identified.
5. Feedback	Real-time feedback alerts users to obstacles.
6. Decision Making	Users receive information to make informed decisions.

Table 3.5: Obstacle Detection Algorithm

3.3.3.4 Experimental Results

In our investigation of the Obstacle Avoidance and Road Crossing (OARC) System using Kinect sensor technology for visually impaired individuals, we conducted a rigorous evaluation to assess its performance and impact. Through comprehensive analysis, we uncovered valuable insights into the system's effectiveness in enhancing mobility and safety. This section presents a concise overview of our experimental findings, including key performance metrics , shedding light on the practical implications of the OARC System in real-world scenarios. Accompanying our experimental findings are visual representations that vividly illustrate the real-time functionality and impact of the OARC System.



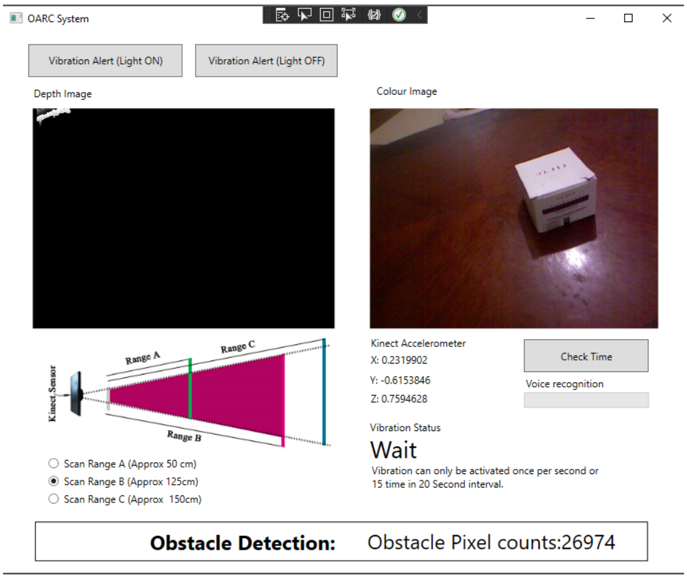
Kinect Accelerometer
X: 0.03663004
Y: -0.6984127
Z: 0.6813187

○ Scan Range A (Approx 50 cm)
● Scan Range B (Approx 125cm)
○ Scan Range C (Approx 150cm)

Vibration Status
Ready
Vibration can only be activated once per second or 15 time in 20 Second interval.

Obstacle Detection: No Obstacle in range.

Figure 3.15: : OARC system.



Kinect Accelerometer
X: 0.2319902
Y: -0.6153846
Z: 0.7594628

○ Scan Range A (Approx 50 cm)
● Scan Range B (Approx 125cm)
○ Scan Range C (Approx 150cm)

Vibration Status
Wait
Vibration can only be activated once per second or 15 time in 20 Second interval.

Obstacle Detection: Obstacle Pixel counts:26974

Figure 3.16: : OARC system with obstacle detection.

This table 3.6 provides a structured overview of the expected outcomes of the OARC System

Expected Outcomes	**Description**
Improved Safety and Independence	- Accurate and reliable information about surroundings, enabling confident navigation. - Reduction in reliance on others for assistance due to increased autonomy.
Increased User Confidence	- Boosted user confidence in the system's capabilities. - Experience of comfortable and less stressful navigation.
Enhanced Usability and Accessibility	- Introduction of a more intuitive user interface. - Personalized settings catering to diverse user needs and abilities.
Advancement in Assistive Technology	- Contribution to the development of more sophisticated and effective solutions for individuals with disabilities. - Push towards a broader range of assistive technologies.

Table 3.6: A structured overview of the expected outcomes of the OARC System

3.3.4 Method 4:Training our model for obstacle detection using YOLO

This method focuses on training a YOLOv8 model within a notebook environment, utilizing the power of a graphics processing unit (GPU) for efficient and accurate obstacle detection.

3.3.4.1 Background

YOLO (You Only Look Once) is a state-of-the-art, real-time object detection system. Unlike traditional object detection models that apply the model to an image at multiple locations and scales, YOLO applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region simultaneously. The latest iteration, YOLOv8, brings significant improvements in speed and accuracy, making it highly suitable for applications requiring real-time processing, such as obstacle detection in road environments.

For our project, we chose YOLOv8 to detect two specific types of obstacles: potholes and cracks on the road. The model's ability to quickly and accurately identify these hazards can significantly enhance safety for individuals with visual impairments, aiding in their navigation and reducing the risk of accidents.

3.3.4.2 Algorithm

Below is the algorithm for training a YOLOv8 model for obstacle detection:

Algorithm 4 Training YOLOv8 Model for Obstacle Detection

```
Input: dataset, input-video, model
Output: video file with the predictions based on the custom dataset
procedure GENERATE-PREDICTION-VIDEO
    # Initialize video capture and writer objects
    cap = cv2.VideoCapture(input-video)
    fourcc = cv2.VideoWriter-fourcc(*'mp4v')
    # Process each frame
    while true do
        ret, frame = cap.read()
        if not ret: then
            break
        end if
    end while
    # Generate predictions using the YOLOv8 model
    results = model(frame)
    annotated-frame = results[0].plot()
    # Write the annotated frame to the output video
    out.write(annotated-frame)
    # Release the video capture and writer objects
    cap.release()
    out.release()
end procedure
```

3.3.4.3 Explanation

We generated our dataset using Roboflow and uploaded the images for annotation. During the annotation process, we manually drew bounding boxes around the objects we wanted to detect. In our case, we focused on detecting two objects: potholes and cracks in the road. With each iteration, we aimed to improve the quality of the predictions.

Once all the images were annotated, we generated the initial version of our dataset. This dataset contained the annotated images with corresponding bounding box coordinates shown in figure3.17,figure3.18.

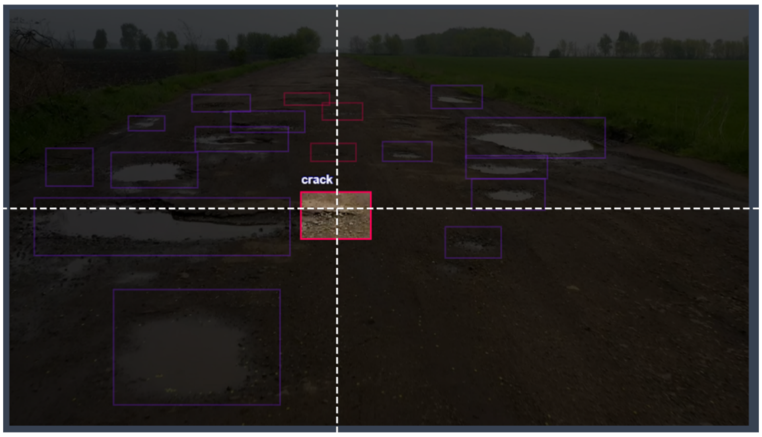


Figure 3.17: The annotation process on crack

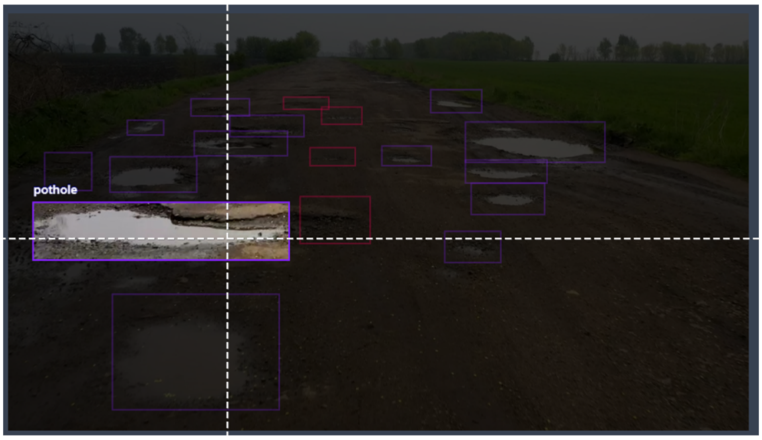


Figure 3.18: The annotation process on pothole

Next, we proceeded to generate a code snippet that included our API key. This key allowed us to download the created dataset in YOLOv8 format, which we would use for training.

Finally, we executed the training process using the prepared training dataset. This involved feeding the dataset into the YOLOv8 model and iteratively updating the model's weights to improve its performance in detecting potholes and cracks in the road. The training process aimed to optimize the model's ability to accurately predict the presence and location of these objects in new, unseen images.

3.3.4.4 Training and validation results

The output provided represents the result obtained by running inference on the validation set using the trained YOLOv8 model. Review the training results as a chart produced by YOLOV8 that will tell us about the training process and the expected accuracy of our model. Let's take a look at the confusion matrix shown in figure3.19.

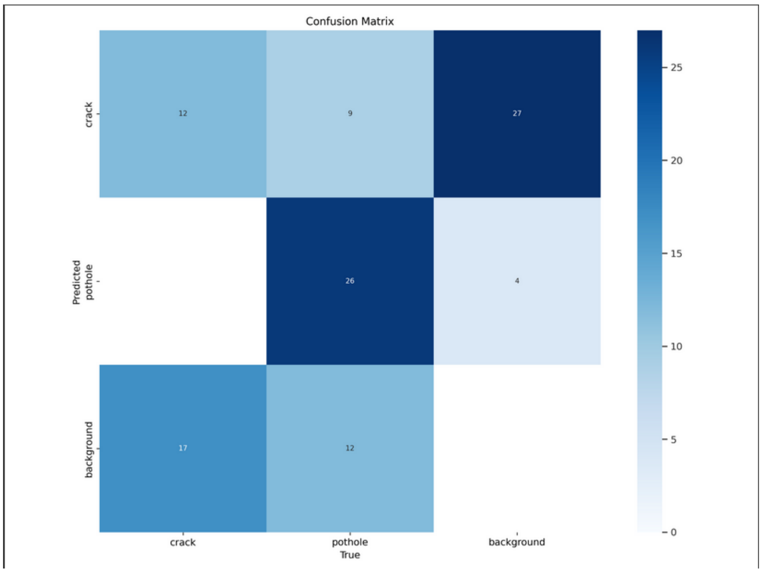


Figure 3.19: The confusion matrix

The chart visualizes the performance of our model across different classes. In the case of cracks, the model correctly detects and classifies them 12 times, but fails to detect them correctly 17 times. Additionally, there are instances where the model provides a bounding box, but incorrectly classifies the object as a pothole, which occurs 9 times.

For the pothole class, the model correctly detects and classifies them in 26 instances. However, there are 12 cases where the model fails to detect potholes correctly, and 4 cases where potholes are not detected at all.

Regarding the speed and processing times of the object detection algorithm, the inference time is reported to be 15.7 milliseconds. This represents the duration taken for the actual inference or prediction step, where the object detection model analyzes pre-processed images and makes predictions about the presence and location of objects, such as cracks and potholes.

These insights from the chart provide an understanding of the model's performance in detecting and classifying different objects. It highlights areas of improvement, such as the model's ability to accurately detect cracks and potholes, as well as instances where misclassifications or missed detections occur. The reported inference time helps evaluate the efficiency of the object detection algorithm in processing video frames.

In the output we provided, it states that there are no background images in the validation set,this means that all the images in the validation set contain objects or instances that the model is expected to detect,this can be considered a positive aspect, as the validation set includes relevant images for evaluating the model's performance on the target objects or classes.

We examine the results that show the key metrics that were tracked by YOLOV8 object detection.

The figure 3.20 appears to display various loss and performance metrics for a machine learning model, likely an object detection model based on the terminology used (box loss, class loss, dfl-loss(distribution focal loss), precision, recall). The metrics are shown for both the training and validation/test datasets, indicating the model's performance on the training data as well as its ability to generalize to unseen data we'll explain each terminology used.

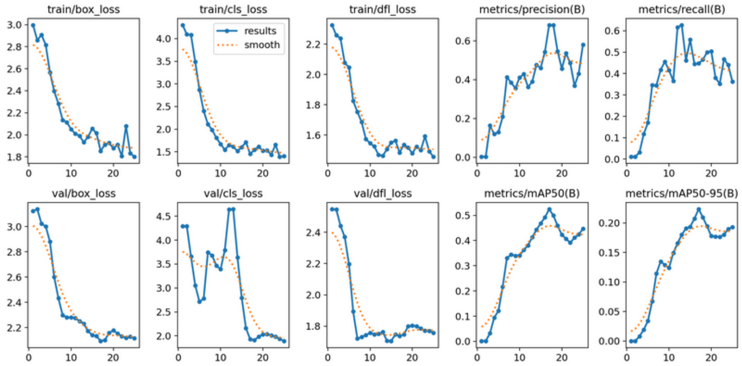


Figure 3.20: Key metrics tracked by YOLOv8

1. train/box-loss and val/box-loss:

- These plots show the box loss, which measures the error in the predicted bounding box coordinates compared to the ground truth.

- The blue line represents the training box loss, and the orange line represents the validation box loss.

- The decreasing trend in both lines indicates the model is learning to predict more accurate bounding boxes over the training iterations.

2. train/cls-loss and val/cls-loss:

- These plots show the classification loss, which measures the error in the object classification predictions.

- The blue line represents the training classification loss, and the orange line represents the validation classification loss.

- The decreasing trend suggests the model is improving at correctly classifying the objects.

3. train/dfl-loss:

- This plot shows the distribution focal loss, which is a specialized loss function used to address class imbalance in the dataset.

- The decreasing trend indicates the model is learning to better detect the rare object classes.

4. metrics/precision(B) and metrics/recall(B):

- These plots show the precision and recall metrics for the object detection model.

- Precision measures the model's ability to correctly identify positive detections, while recall measures its ability to detect all the relevant objects.

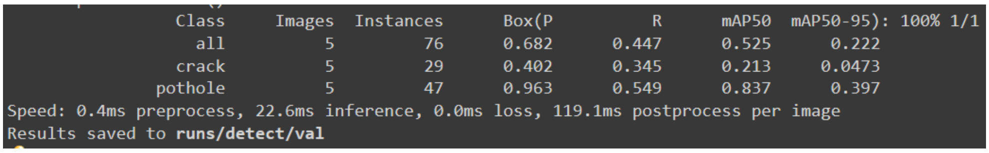
- The fluctuating but stabilizing trends suggest the model is achieving a balance between precision and recall.

5. metrics/mAP50(B) and metrics/mAP50-95(B):

- These plots show the mean Average Precision (mAP) metrics, which are commonly used to evaluate object detection models.
- mAP50 measures the average precision at an Intersection over Union (IoU) threshold of 0.5, while mAP50-95 averages the precision across multiple IoU thresholds.
- The increasing trends indicate the model's ability to detect objects with higher accuracy.

3.3.4.5 Experimental Results

The figure 3.21 shows the evaluation results for a YOLOv8 object detection model on a validation dataset.



```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 1/1
                   all          5         76      0.682      0.447      0.525      0.222
                 crack          5         29      0.402      0.345      0.213     0.0473
               pothole          5         47      0.963      0.549      0.837      0.397
Speed: 0.4ms preprocess, 22.6ms inference, 0.0ms loss, 119.1ms postprocess per image
Results saved to runs/detect/val
```

Figure 3.21: YOLOv8 model evaluation results

The key metrics displayed are:

Metric	**Description**
Class	The object classes the model is trained to detect, including "all", "crack", and "pothole".
Images	The number of images in the validation set for each class,we have 5 crack and 5 pothole.
Instances	The total number of object instances in the validation set for each class,we have 29 crack and 47 pothole.
Box(P)	The average precision (AP) for the bounding box predictions.
R	The average recall (AR) for the bounding box predictions.
mAP50	The mean average precision (mAP) at an intersection-over-union (IoU) threshold of 50%.
mAP50-95	The mAP across IoU thresholds from 50% to 95% (in 5% increments).

Table 3.7: Model Performance Metrics

The results indicate that the overall mAP50 is 52.5% and the mAP50-95 is 22.2%, which provides a high-level assessment of the model's performance on the validation dataset.

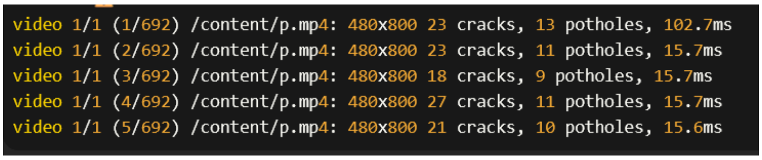
The model also achieves inference speeds of 22.6ms per image, which is a relatively fast inference time.

Overall, these evaluation metrics give a quantitative measure of the YOLOv8 model's object detection capabilities on the specified validation data, which can be used to assess the model's readiness for deployment or identify areas for further improvement.
Using the trained weights from our model, we can now apply them to validate the model's performance on the validation batch of data.

The figure 3.22 we shared provides some examples of the model's predictions on this validation data.

Figure 3.22: Example YOLOv8 inference on a validation batch

Based on the strong performance of the YOLOv8 model on the validation dataset, particularly in detecting potholes and cracks, it sounds like a good next step to run the inference on a 2 videos.



```
video 1/1 (1/692) /content/p.mp4: 480x800 23 cracks, 13 potholes, 102.7ms
video 1/1 (2/692) /content/p.mp4: 480x800 23 cracks, 11 potholes, 15.7ms
video 1/1 (3/692) /content/p.mp4: 480x800 18 cracks, 9 potholes, 15.7ms
video 1/1 (4/692) /content/p.mp4: 480x800 27 cracks, 11 potholes, 15.7ms
video 1/1 (5/692) /content/p.mp4: 480x800 21 cracks, 10 potholes, 15.6ms
```

Figure 3.23: Processed a series of video 1 frames.

```
video 2/2 (1/324) /content/16925161-preview.mp4: 480x800 4 cracks, 7 potholes, 101.7ms
video 2/2 (2/324) /content/16925161-preview.mp4: 480x800 4 cracks, 7 potholes, 15.6ms
video 2/2 (3/324) /content/16925161-preview.mp4: 480x800 3 cracks, 6 potholes, 15.5ms
video 2/2 (4/324) /content/16925161-preview.mp4: 480x800 3 cracks, 6 potholes, 15.5ms
video 2/2 (5/324) /content/16925161-preview.mp4: 480x800 4 cracks, 8 potholes, 15.5ms
```

Figure 3.24: Processed a series of video 2 frames.

Figure 3.23,3.24 display output related to the analysis of a videos file. Each line represents the analysis results for a specific frame of the video,based on the information provided let's analyze comparison of accuracy between the two videos showns in table 3.8:

Metric	Video 1/1	Video 2/2
Ground Truth:		
Potholes	13	13
Cracks	25	4
Detected:		
Potholes	13 (100%)	7 (53.8%)
Cracks	23 (92%)	4 (100%)
Overall Accuracy	96%	76.9%

Table 3.8: Comparison of model accuracy between the two video samples.

Based on the results,the model seems to have performed better on the first video, with an overall accuracy of 96%, compared to 76.9 % on the second video. The pothole detection accuracy was significantly lower in the second video at 53.8%, while the crack detection was perfect.

This suggests the model may have some difficulty consistently detecting potholes across different video samples. Further analysis and model tuning may be needed to improve the overall performance and robustness.

3.4 Comparative Study of Four Methods for Road Obstacle Detection

This section provides a comparative study of the four methods for road obstacle detection discussed in Chapter 3:

Method	**Strengths**	**Weaknesses**
Self-Driving Car with Obstacle Detection and Lane Detection	High accuracy, Real-time processing, Integration with other systems	High cost, Limited availability, Ethical considerations
Advancing Sensing Capabilities with Microsoft Kinect	Cost-effective, Versatile, Ease of use	Limited range, Sensitivity to environmental conditions, Limited resolution
Obstacle Avoidance and Road Crossing System using the Kinect Sensor	Real-time obstacle detection, Improved safety, Low cost	Limited range, Limited functionality, Accuracy limitations
Training our model for obstacle detection using YOLO	High accuracy, Real-time processing, Adaptability	Training data requirement, Computational resources, Generalizability

Table 3.9: Comparative Study of Road Obstacle Detection Methods

3.5 Future research

Future research in road obstacle detection methods will focus on advanced sensor fusion, improved decision-making algorithms, ethical considerations and regulations for self-driving cars, depth image enhancement, object tracking and classification, integration with other systems, expanding functionality, improving accuracy and robustness, integration with wearable devices, transfer learning, adapting to different environments, exploring other deep learning architectures, and integrating YOLO with Kinect.

These research directions aim to address the limitations of existing methods and develop more robust, efficient, and cost-effective solutions for road obstacle detection, ultimately enhancing road safety and improving transportation efficiency for all users.

3.6 Conclusion

The methods presented in this chapter offer a range of innovative approaches to the critical challenge of road obstacle detection, each with its own unique strengths and focus areas.

The first method explores the capabilities of self-driving car technology, leveraging advanced sensors and algorithms to provide comprehensive environmental perception.

In contrast, the second and third methods leverage Microsoft Kinect technology to enhance sensing and detection, demonstrating the versatility of this platform.

The fourth method takes a more specialized approach, integrating the YOLO algorithm to achieve heightened detection accuracy and efficiency.

Collectively, these diverse methodologies represent significant advancements in the field of road safety, paving the way for the development of a robust, integrated system that can accurately identify and respond to obstacles in real-time.

By drawing upon the strengths of each approach, future work can synthesis these methods to create a comprehensive solution that enhances the safety and reliability of road transportation for all users.

General conclusion

In conclusion, this dissertation has comprehensively explored the application of crowdsensing and advanced sensing technologies to enhance urban monitoring and road safety. Through three primary methodologies Kinect sensors for obstacle avoidance and road crossing, a self-driving car system with real-time object detection, and deep learning models for hazard identification this research has demonstrated significant advancements in smart city technologies.

The initial chapters provided a robust overview of AI and IoT, emphasizing their transformative potential in creating intelligent systems that enhance data collection and application in urban infrastructure and environmental monitoring.

Subsequent chapters delved into the specifics of crowdsensing, illustrating how collective data gathering from numerous smart devices facilitates extensive and cost-effective environmental monitoring, particularly for air quality and traffic patterns.

The development of innovative road safety solutions was a focal point, with detailed discussions on the Obstacle Avoidance and Road Crossing (OARC) System utilizing Kinect sensors, a self-driving car system leveraging structured algorithms for lane and obstacle detection, and the application of the YOLOv8 deep learning model for accurate real-time obstacle detection.

These methodologies collectively underscore the potential for real-time, reliable, and scalable solutions that enhance safety for drivers and pedestrians alike, contributing to the development of smarter, more resilient urban environments.

The integration of these systems into broader urban infrastructure networks is crucial for improving interoperability and data-sharing capabilities. Moreover, the ethical considerations and societal impact of these technologies must be addressed to ensure their responsible deployment.

Ultimately, this research highlights the importance of collaborative efforts among technologists, policymakers, and communities to fully realize the potential of these advancements, paving the way for safer and smarter urban environments through continuous learning and innovation.

Bibliography

[1] Heikki Ailisto et al. *Internet of Things Strategic Research Agenda (IoT-SRA).* Sept. 2011.

[2] Yazeed Al Moaiad. *The Internet of Things as A Revolution to Enhance the Technology of the Future.* Dec. 2022. DOI: `10.13140/RG.2.2.19815.11686`.

[3] ER Alberts. “The Kinect Effect: Tracing the Constituents of a Technical Artefact”. MA thesis. 2012.

[4] A Ali et al. *Traffic Efficiency Models for Urban Traffic Management Using Mobile Crowd Sensing: A Survey. Sustainability 2021, 13, 13068.* Accessed: March 05 2024. 2021.

[5] Areen Alsaid et al. “The Effect of Vehicle Automation Styles on Drivers’ Emotional State”. In: *IEEE Transactions on Intelligent Transportation Systems* 24.4 (2023), pp. 3963–3973. DOI: `10.1109/TITS.2023.3239880`.

[6] Oscar Alvear et al. “Crowdsensing in smart cities: Overview, platforms, and environment sensing issues”. In: *Sensors* 18.2 (2018). Accessed: February 12 2024, p. 460.

[7] Giuseppe Amato et al. “A wireless smart camera network for parking monitoring”. In: *2018 IEEE Globecom Workshops (GC Wkshps).* Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018.

[8] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The internet of things: A survey”. In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[9] Kudrat-E Babu. “Artificial Intelligence, Its Applications in Different Sectors and Challenges: Bangladesh Context”. In: Jan. 2021, pp. 103–119. ISBN: 978-3-030-88039-2. DOI: `10.1007/978-3-030-88040-8_4`.

[10] Behram Beldagli and Tufan Adiguzel. “Illustrating an ideal adaptive e-learning: A conceptual framework”. In: *Procedia-Social and Behavioral Sciences* 2.2 (2010), pp. 5755–5761.

[11] Nabarun Bhattacharyya and Rajib Bandhopadhyay. “Electronic nose and electronic tongue”. In: *Nondestructive Evaluation of Food Quality: Theory and Practice.* Springer, 2010, pp. 73–100.

[12] Milind Bhushan et al. *INTELLIGENT TUTORING SYSTEM: PERSONALISED LEARNING PLANS WITH AI.* Oct. 2023. DOI: `10.13140/RG.2.2.36573.59369`.

[13] Alessio Botta et al. “Integration of Cloud computing and Internet of Things: A survey”. In: *Future Generation Computer Systems* 56 (Oct. 2015). DOI: `10.1016/j.future.2015.09.021`.

[14] Djallel Eddine Boubiche et al. "Mobile crowd sensing–taxonomy, applications, challenges, and solutions". In: *Computers in Human Behavior* 101 (2019). Accessed: February 23 2024, pp. 352–370.

[15] Christopher Brewster et al. "IoT in agriculture: Designing a Europe-wide large-scale pilot". In: *IEEE communications magazine* 55.9 (2017), pp. 26–33.

[16] Raj Bridgelall and Denver Tolliver. "Accuracy enhancement of anomaly localization with participatory sensing vehicles". In: *Sensors* 20.2 (2020). Accessed: February 10 2024, p. 409.

[17] Muhammad Burhan et al. "IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey". In: *Sensors* 18.9 (2018). ISSN: 1424-8220. DOI: `10.3390/s18092796`. URL: `https://www.mdpi.com/1424-8220/18/9/2796`.

[18] Andrew T Campbell et al. "Transforming the social networking experience with sensing presence from mobile phones". In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. Accessed: February 16 2024. 2008, pp. 367–368.

[19] Andrea Capponi et al. "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities". In: *IEEE communications surveys & tutorials* 21.3 (2019). Accessed: February 21 2024, pp. 2419–2465.

[20] Fran Casino, Thomas K Dasaklis, and Constantinos Patsakis. "A systematic literature review of blockchain-based applications: Current status, classification and open issues". In: *Telematics and informatics* 36 (2019), pp. 55–81.

[21] Po-Hsuan Cameron Chen et al. "An augmented reality microscope with real-time artificial intelligence integration for cancer diagnosis". In: *Nature medicine* 25.9 (2019), pp. 1453–1457.

[22] Jian Chen et al. "Current status of artificial intelligence applications in urology and their potential to influence clinical practice". In: *BJU international* 124.4 (2019), pp. 567–577.

[23] Stephanie Crawford. "How microsoft kinect works". In: *Howstuffworks [online] Retrieved from the Internet on Aug* 19 (2010), pp. 1–5.

[24] *Crowdsourced Data Collection Benefits Best Practices in 2024*. Shehmir Javaid. Accessed: March 29 2024. 2024. URL: `https://research.aimultiple.com/crowdsourced-data/`.

[25] Debashis De, Shreya Ghosh, and Anwesha Mukherjee. "SocialSense: Mobile crowd sensing-based physical distance monitoring model leveraging federated learning for pandemic". In: *Internet of Things* 23 (2023). Accessed: February 21 2024, p. 100872.

[26] G Dhamodharan and C Gopi Mohan. "Machine learning models for predicting the activity of AChE and BACE1 dual inhibitors for the treatment of Alzheimer's disease". In: *Molecular Diversity* (2022), pp. 1–17.

[27] Paolo Diviacco et al. "Monitoring air quality in urban areas using a vehicle sensor network (VSN) crowdsensing paradigm". In: *Remote Sensing* 14.21 (2022). Accessed: February 15 2024, p. 5576.

[28] Luciano Floridi. *The ethics of information*. Oxford University Press, USA, 2013.

[29] Luciano Floridi and Jeff W Sanders. “On the morality of artificial agents”. In: *Minds and machines* 14 (2004), pp. 349–379.

[30] Pittaluga Francesco and Koppal Sanjeev. “Privacy preserving optics for miniature vision sensors”. In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. Google Scholar, 2015, pp. 314–324.

[31] Vincent François-Lavet et al. “An Introduction to Deep Reinforcement Learning”. In: *Foundations and Trends® in Machine Learning* 11.3–4 (2018), pp. 219–354. ISSN: 1935-8245. DOI: `10.1561/2200000071`. URL: `http://dx.doi.org/10.1561/2200000071`.

[32] Stefan Funke and Sabine Storandt. “Personalized route planning in road networks”. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2015, pp. 1–10.

[33] Ala Al-Fuqaha et al. “Internet of Things: A Survey on Enabling Technologies, Protocols and Applications”. In: *IEEE Communications Surveys amp Tutorials* 17 (Nov. 2015), Fourthquarter 2015. DOI: `10.1109/COMST.2015.2444095`.

[34] Ala Al-Fuqaha et al. “Internet of things: A survey on enabling technologies, protocols, and applications”. In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[35] Raghu K Ganti, Fan Ye, and Hui Lei. “Mobile crowdsensing: current state and future challenges”. In: *IEEE communications Magazine* 49.11 (2011). Accessed: February 28 2024, pp. 32–39.

[36] Cristian González García et al. “Midgar: Generation of heterogeneous objects interconnecting applications. A Domain Specific Language proposal for Internet of Things scenarios”. In: *Computer Networks* 64 (May 2014). DOI: `10.1016/j.comnet.2014.02.010`.

[37] Jayavardhana Gubbi et al. “Internet of Things (IoT): A vision, architectural elements, and future directions”. In: *ArXiv* abs/1207.0203 (2012). URL: `https://api.semanticscholar.org/CorpusID:204982032`.

[38] Stephan Haller, Stamatis Karnouskos, and Christoph Schroth. “The Internet of Things in an Enterprise Context”. In: vol.)5468. Sept. 2008, pp. 14–28. ISBN: 978-3-642-00984-6. DOI: `10.1007/978-3-642-00985-3_2`.

[39] Zhu Hanwei et al. “Privacy-preserving camera-based monitoring and tracking system for parking spaces”. In: *Proceedings of the ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 2020, pp. 346–347.

[40] S.S. Haykin. *Neural Networks: A Comprehensive Foundation*. International edition. Prentice Hall, 1999. ISBN: 9780132733502. URL: `https://books.google.dz/books?id=bX4pAQAAMAAJ`.

[41] Abiodun M. Ikotun et al. “K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data”. In: *Information Sciences* 622 (2023), pp. 178–210. ISSN: 0020-0255. DOI: `https://doi.org/10.1016/j.ins.2022.11.139`. URL: `https://www.sciencedirect.com/science/article/pii/S0020025522014633`.

[42] Ha Young Jang et al. “Machine learning-based quantitative prediction of drug exposure in drug-drug interactions using drug label information”. In: *NPJ Digital Medicine* 5.1 (2022), p. 88.

[43] Slađana Jovanović et al. “A mobile crowd sensing application for hypertensive patients”. In: *Sensors* 19.2 (2019). Accessed: February 29 2024, p. 400.

[44] Riad Kanan and Houssam Arbess. “An IoT-based intelligent system for real-time parking monitoring and automatic billing”. In: *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. Doha, Qatar: IEEE, Feb. 2020.

[45] Bill Karakostas. “A DNS architecture for the internet of things: A case study in transport logistics”. In: *Procedia Computer Science* 19 (2013), pp. 594–601.

[46] Harleen Kaur and Vinita Kumari. “Predictive modelling and analytics for diabetes using a machine learning approach”. In: *Applied computing and informatics* 18.1/2 (2020), pp. 90–100.

[47] Tiarnan D L Keenan et al. “Retinal specialist versus artificial intelligence detection of retinal fluid from OCT”. en. In: *Ophthalmology* 128.1 (Jan. 2021), pp. 100–109.

[48] Rafiullah Khan et al. “Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges”. In: *2012 10th International Conference on Frontiers of Information Technology*. 2012, pp. 257–260. DOI: `10.1109/FIT.2012.53`.

[49] Rafiullah Khan et al. “Future internet: the internet of things architecture, possible applications and key challenges”. In: *2012 10th international conference on frontiers of information technology*. IEEE. 2012, pp. 257–260.

[50] Jussi Kiljander et al. “Semantic interoperability architecture for pervasive computing and Internet of Things”. English. In: *IEEE Access* 2 (2014). Project code: 81532, pp. 856–873. ISSN: 2169-3536. DOI: `10.1109/ACCESS.2014.2347992`.

[51] Xiangjie Kong et al. “Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges”. In: *IEEE Internet of Things Journal* 6.5 (2019). Accessed: February 14 2024, pp. 8095–8113.

[52] Pantelis Kopelias et al. “Connected autonomous vehicles – Environmental impacts – A review”. In: *Science of The Total Environment* 712 (2020), p. 135237. ISSN: 0048-9697. DOI: `https://doi.org/10.1016/j.scitotenv.2019.135237`. URL: `https://www.sciencedirect.com/science/article/pii/S0048969719352295`.

[53] Harika Devi Kotha and V Mnssvkr Gupta. “IoT application: a survey”. In: *Int. J. Eng. Technol* 7.2.7 (2018), pp. 891–896.

[54] Ryohei Kozu et al. “User participatory construction of open hazard data for preventing bicycle accidents”. In: *Semantic Technology: 7th Joint International Conference, JIST 2017, Gold Coast, QLD, Australia, November 10-12, 2017, Proceedings 7*. Accessed: February 21 2024. Springer. 2017, pp. 289–303.

[55] Jochen Kruppa et al. “Probability estimation with machine learning methods for dichotomous and multicategory outcome: Applications: Probability estimation with machine learning methods: Applications”. In: *Biometrical journal. Biometrische Zeitschrift* 56 (July 2014). DOI: `10.1002/bimj.201300068`.

[56] Adarsh Kulkarni, Priya Mani, and Carlotta Domeniconi. "Network-based Anomaly Detection for Insider Trading". In: *CoRR* abs/1702.05809 (2017). arXiv: `1702.05809`. URL: `http://arxiv.org/abs/1702.05809`.

[57] Quoc Le et al. "Building high-level features using large scale unsupervised learning". In: *Proceedings of ICML* 1 (Dec. 2011).

[58] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: `10.1038/nature14539`.

[59] Rongbang Li and Carolyn Liu. "Road Damage Evaluation via Stereo Camera and Deep Learning Neural Network". In: *2021 IEEE Aerospace Conference (50100)*. Accessed: March 06 2024. IEEE. 2021, pp. 1–7.

[60] Jiang Linshan et al. "On lightweight privacy-preserving collaborative learning for internet of things by independent random projections". In: *ACM Transactions on Internet of Things* 2.2 (2021).

[61] Jieyan Liu et al. "Data collection for mobile crowdsensing in the presence of selfishness". In: *EURASIP journal on wireless communications and networking* 2016 (2016). Accessed: March 24 2024, pp. 1–10.

[62] Qiong Liu and Ying Wu. "Supervised Learning". In: (Jan. 2012). DOI: `10.1007/978-1-4419-1428-6_451`.

[63] Yang Liu. "Mobile Crowdsensing: Issues and Challenges". In: *Encyclopedia of Wireless Networks* (2020). Accessed: February 15 2024, pp. 861–865.

[64] Chaitanya Vijaykumar Mahamuni and Chalamala Srinivas Goud. "Unveiling the Internet of Things (IoT) Applications in Aquaculture: A Survey and Prototype Design with ThingSpeak Analytics". In: *Journal of Ubiquitous Computing and Communication Technologies* 5.2 (2023), pp. 152–174.

[65] Ibrahim Mashal et al. "Choices for interaction with things on Internet and underlying issues". In: *Ad Hoc Networks* 28 (2015), pp. 68–90.

[66] Aidana Massalimova et al. "Intraoperative tissue classification methods in orthopedic and neurological surgeries: A systematic review". In: *Frontiers in Surgery* 9 (2022), p. 952539.

[67] Marcela E. Mata-Romero et al. "A Low-Cost Wearable Device to Estimate Body Temperature Based on Wrist Temperature". In: *Sensors* 24.6 (2024). ISSN: 1424-8220. DOI: `10.3390/s24061944`. URL: `https://www.mdpi.com/1424-8220/24/6/1944`.

[68] Suhas Mathur et al. "Parknet: drive-by sensing of road-side parking statistics". In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. Accessed: February 22 2024. 2010, pp. 123–136.

[69] John McCarthy et al. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955". In: *AI Magazine* 27.4 (2006), p. 12. DOI: `10.1609/aimag.v27i4.1904`. URL: `https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904`.

[70] Fred Meier et al. "Crowdsourcing air temperature from citizen weather stations for urban climate research". In: *Urban Climate* 19 (2017). Accessed: February 25 2024, pp. 170–191.

[71] Emiliano Miluzzo et al. “Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application”. In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. Accessed: February 18 2024. 2008, pp. 337–350.

[72] Sudip Misra et al. “An adaptive learning approach for fault-tolerant routing in Internet of Things”. In: *IEEE Wireless Communications and Networking Conference, WCNC* (Apr. 2012), pp. 815–819. DOI: `10.1109/WCNC.2012.6214484`.

[73] Aftab Muhammad et al. “Automatic HVAC control with real-time occupancy recognition and simulation-guided model predictive control in low-cost embedded system”. In: *Energy and Buildings* 154 (2017), pp. 141–156.

[74] Alhirabi Nada, Rana Omer, and Perera Charith. “Security and privacy requirements for the internet of things: A survey”. In: *ACM Transactions on Internet of Things* 2.1 (2021).

[75] Samreen Naeem et al. “An Unsupervised Machine Learning Algorithms: Comprehensive Review”. In: *IJCDS Journal* 13 (Apr. 2023), pp. 911–921. DOI: `10.12785/ijcds/130172`.

[76] Hajihassani Omid, Ardakanian Omid, and Khazaei Hamzeh. “Anonymizing sensor data on the edge: A representation learning and transformation approach”. In: *ACM Transactions on Internet of Things* 3.1 (2021).

[77] Valentine Owan et al. “Exploring the potential of artificial intelligence tools in educational measurement and assessment”. In: *Eurasia Journal of Mathematics, Science and Technology Education* 19 (June 2023), Article ID em2307. DOI: `10.29333/ejmste/13428`.

[78] Zhengxiang Pan et al. “Crowdsensing air quality with camera-enabled mobile devices”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 2. Accessed: February 13 2024. 2017, pp. 4728–4733.

[79] Victor Pankratius et al. “Mobile crowd sensing in space weather monitoring: the mahali project”. In: *IEEE communications magazine* 52.8 (2014). Accessed: March 01 2024, pp. 22–28.

[80] Zuber Purahoo and Sudha Cheerkoot-Jalim. “SenseAPP: An IoT-based mobile crowdsensing application for smart cities”. In: *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*. Accessed: February 10 2024. IEEE. 2020, pp. 47–52.

[81] Biplob Ray, Jemal Abawajy, and Morshed Chowdhury. “Scalable RFID Security Framework and Protocol Supporting Internet of Things”. In: *Computer Networks* 67 (July 2014). DOI: `10.1016/j.comnet.2014.03.023`.

[82] Abolfazl Razi et al. *Deep Learning Serves Traffic Safety Analysis: A Forward-looking Review*. Mar. 2022.

[83] Nick F. Ryman-Tubb, Paul Krause, and Wolfgang Garn. “How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark”. In: *Engineering Applications of Artificial Intelligence* 76 (2018), pp. 130–157. ISSN: 0952-1976. DOI: `https://doi.org/10.1016/j.engappai.2018.07.008`. URL: `https://www.sciencedirect.com/science/article/pii/S0952197618301520`.

[84] Nina Schwalbe and Brian Wahl. “Artificial intelligence and the future of global health”. In: *The Lancet* 395.10236 (2020), pp. 1579–1586.

[85] Yuichi Sei and Akihiko Ohsuga. “Differentially private mobile crowd sensing considering sensing errors”. In: *Sensors* 20.10 (2020). Accessed: February 17 2024, p. 2785.

[86] Dimitris Selis et al. “A crowdsensing cloud-based microservice for weather forecast”. In: (2022). Accessed: February 29 2024.

[87] Fengrui Shi et al. “ParkCrowd: Reliable crowdsensing for aggregation and dissemination of parking space information”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.11 (2018). Accessed: February 12 2024, pp. 4032–4044.

[88] Jung Shim et al. “The Transformative Effect of the Internet of Things on Business and Society”. In: *Communications of the Association for Information Systems* 44 (Jan. 2019), pp. 129–140. DOI: `10.17705/1CAIS.04405`.

[89] Donghee Shin. “A Socio-Technical Framework for Internet-of-Things Design: A Human-Centered Design for the Internet of Things”. In: *Telematics and Informatics* 31 (Nov. 2014). DOI: `10.1016/j.tele.2014.02.003`.

[90] “Front Matter”. In: *Knowledge Discovery in Big Data from Astronomy and Earth Observation*. Ed. by Petr Škoda and Fathalrahman Adam. Elsevier, 2020, pp. i–iii. ISBN: 978-0-12-819154-5. DOI: `https://doi.org/10.1016/B978-0-12-819154-5.00002-3`. URL: `https://www.sciencedirect.com/science/article/pii/B9780128191545000023`.

[91] Sharmila Subudhi and Suvasini Panigrahi. “Effect of Class Imbalanceness in Detecting Automobile Insurance Fraud”. In: Sept. 2018, pp. 528–531. DOI: `10.1109/ICDSBA.2018.00104`.

[92] Hui Suo et al. “Security in the internet of things: a review”. In: *2012 international conference on computer science and electronics engineering*. Vol. 3. IEEE. 2012, pp. 648–651.

[93] Bedir Tekinerdogan, Ömer Köksal, and Turgay Çelik. “System architecture design of IoT-based smart cities”. In: *Applied Sciences* 13.7 (2023), p. 4173.

[94] Muhammad Ali Javed Tengnah, Raginee Sooklall, and Soulakshmee Devi Nagowah. “A predictive model for hypertension diagnosis using machine learning techniques”. In: *Telemedicine technologies*. Elsevier, 2019, pp. 139–152.

[95] Wattana Viriyasitavat, Tharwon Anuphaptrirong, and Danupol Hoonsopon. “When Blockchain Meets Internet of Things: Characteristics, Challenges, and Business Opportunities”. In: *Journal of Industrial Information Integration* 15 (May 2019). DOI: `10.1016/j.jii.2019.05.002`.

[96] Chen Wang, Travis Atkison, and Hana Park. “Dynamic adaptive vehicle re-routing strategy for traffic congestion mitigation of grid network”. In: *International Journal of Transportation Science and Technology* (2023). ISSN: 2046-0430. DOI: `https://doi.org/10.1016/j.ijtst.2023.04.003`. URL: `https://www.sciencedirect.com/science/article/pii/S2046043023000321`.

[97] Jiangtao Wang et al. “A context-driven worker selection framework for crowdsensing”. In: *International Journal of Distributed Sensor Networks* 12.3 (2016). Accessed: March 23 2024, p. 6958710.

[98] Jiangtao Wang et al. “Task allocation in mobile crowd sensing: State-of-the-art and future opportunities”. In: *IEEE Internet of Things journal* 5.5 (2018). Accessed: March 18 2024, pp. 3747–3757.

[99] Qili Wang et al. “Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning”. In: *Neurocomputing* 347 (2019), pp. 46–58. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2019.03.006`. URL: `https://www.sciencedirect.com/science/article/pii/S0925231219303005`.

[100] Michael Weyrich and Christof Ebert. “Reference Architectures for the Internet of Things”. In: *IEEE Software* 33.1 (2016), pp. 112–116. DOI: `10.1109/MS.2016.20`.

[101] Andrew Whitmore, Anurag Agarwal, and Li Xu. “The Internet of Things—A survey of topics and trends”. In: *Information Systems Frontiers* 17 (Apr. 2014). DOI: `10.1007/s10796-014-9489-2`.

[102] Michael Wooldridge and Nicholas R. Jennings. “Intelligent agents: theory and practice”. In: *The Knowledge Engineering Review* 10.2 (1995), pp. 115–152. DOI: `10.1017/S0269888900008122`.

[103] Chao Wu et al. “An automated machine-learning approach for road pothole detection using smartphone sensor data”. In: *Sensors* 20.19 (2020). Accessed: March 10 2024, p. 5564.

[104] Miao Wu et al. “Research on the architecture of Internet of Things”. In: *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*. Vol. 5. IEEE. 2010, pp. V5–484.

[105] Miao Wu et al. “Research on the architecture of Internet of Things”. In: *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. Vol. 5. 2010, pp. V5-484-V5–487. DOI: `10.1109/ICACTE.2010.5579493`.

[106] Wang Xiaoyun and Liu Danyue. “Hybrid outlier mining algorithm based evaluation of client moral risk in insurance company”. In: *2010 2nd IEEE International Conference on Information Management and Engineering*. 2010, pp. 585–589. DOI: `10.1109/ICIME.2010.5478070`.

[107] Hanyu Xin et al. “Sustainable Road Pothole Detection: A Crowdsourcing Based Multi-Sensors Fusion Approach”. In: *Sustainability* 15.8 (2023). Accessed: March 02 2024, p. 6610.

[108] Özgün Yilmaz et al. “Cloud-assisted mobile crowd sensing for route and congestion monitoring”. In: *Ieee Access* 9 (2021). Accessed: March 08 2024, pp. 157984–157996.

[109] Bo Yin, Jiaqi Li, and Xuetao Wei. “Rational task assignment and path planning based on location and task characteristics in mobile crowdsensing”. In: *IEEE Transactions on Computational Social Systems* 9.3 (2021). Accessed: March 10 2024, pp. 781–793.

[110] Jiaying You, Robert D McLeod, and Pingzhao Hu. “Predicting drug-target interaction network using deep learning model”. In: *Computational biology and chemistry* 80 (2019), pp. 90–101.

[111] Khazbak Youssef et al. “TargetFinder: A privacy preserving system for locating targets through IoT cameras”. In: *ACM Transactions on Internet of Things* 1.3 (2020).

[112] Zhiwen Yu et al. “Recommending travel packages based on mobile crowdsourced data”. In: *IEEE Communications Magazine* 52.8 (2014). Accessed: February 25 2024, pp. 56–62.

[113] Mahmoud M. Badawy Zainab H. Ali Hesham A. Ali. “Internet of Things (IoT): Definitions, Challenges and Recent Research Directions”. In: *International Journal of Computer Applications* 128.1 (2015), pp. 37–47. ISSN: 0975-8887. DOI: `10.5120/ijca2015906430`. URL: `https://ijcaonline.org/archives/volume128/number1/22841-2015906430/`.

[114] Marco Zappatore et al. “Improving urban noise monitoring opportunities via mobile crowd-sensing”. In: *Smart City 360°: First EAI International Summit, Smart City 360°, Bratislava, Slovakia and Toronto, Canada, October 13-16, 2015. Revised Selected Papers 1.* Accessed: March 05 2024. Springer. 2016, pp. 885–897.

[115] Daqing Zhang et al. “4W1H in mobile crowd sensing”. In: *IEEE Communications Magazine* 52.8 (2014). Accessed: February 27 2024, pp. 42–48.

[116] Xuhui Zhang et al. “A short review of constructing noise map using crowdsensing technology”. In: *Collaborative Computing: Networking, Applications and Worksharing: 13th International Conference, CollaborateCom 2017, Edinburgh, UK, December 11–13, 2017, Proceedings 13.* Accessed: March 04 2024. Springer. 2018, pp. 37–43.

[117] Ming ZHOU and Yan Ma. “QoS-aware computational method for IoT composite service”. In: *The Journal of China Universities of Posts and Telecommunications* 20 (Aug. 2013), pp. 35–39. DOI: `10.1016/S1005-8885(13)60252-6`.

[118] Weiping Zhu, Wenzhong Guo, and Zhiyong Yu. “Social-aware task allocation in mobile crowd sensing”. In: *Wireless Communications and Mobile Computing* 2020 (2020). Accessed: March 22 2024, pp. 1–15.

[119] Xia Zhuoyue, Azween Abdullah, and SH Kok. “A Survey on the Mobile Crowdsensing System life cycle: Task Allocation, Data Collection, and Data Aggregation”. In: *International Journal of Computer Science & Network Security* 23.3 (2023). Accessed: March 02 2024, pp. 31–48.

Abstract:

The collaborative of crowdsensing and advanced sensing technologies can vastly improve environmental and infrastructure monitoring in cities through creative applications. Novel use cases tracked important metrics like air quality, noise levels and traffic patterns to provide communities with valuable data. In this work, propose a Kinect sensor that leverage computer vision for detecte obstacles in the roads.with adopt YOLO-v8 as obstacle detector,added a series of methods proposed for road obstacle detection, like a self-driving car system and Kinect approach that effectively mapped hazards in 3D for navigation safety. Based on those methodes we are going to integrated to generate system in the futures. Results indicated addressing limitations of traditional solutions by enabling timely, comprehensive data collection, bringing significant benefits in safety, costs, transportation resilience and urban management. Overall,this work is helping advance the potential of collaborative digital tools to revolutionize smarter, safer mobility for all through continuous learning.

Key Words: Road obstacle detection, Self-Driving cars, Microsoft kinect sensor, Obstacle avoidance systems, Crowdsensing, YOLOv8, Advanced sensor fusion, 3D mapping, Real-time detection accuracy, Depth image enhancement.

ملخص:

يمكن للتعاون بين الاستشعار الجماعي وتقنيات الاستشعار المتقدمة أن يحسن بشكل كبير من مراقبة البيئة والبنية التحتية في المدن من خلال تطبيقات مبتكرة. أتبعت حالات الاستخدام الجديدة مقاييس مهمة مثل جودة الهواء ومستويات الضوضاء وأنماط المرور لتزويد المجتمعات ببيانات قيّمة. في هذا العمل، نقترح استخدام أجهزة استشعار kinect التي تستفيد من رؤية الكمبيوتر لكشف العوائق على الطرق. نعتمد على YOLO-v8 كمكشف للعوائق، وأضفنا سلسلة من الطرق المقترحة لكشف العوائق على الطرق، مثل نظام السيارة ذاتية القيادة ونهج kinect الذي خرّط المخاطر بفعالية في ثلاثة أبعاد من أجل سلامة الملاحة. بناءً على هذه الطرق، سنقوم بتكامل لإنشاء نظام في المستقبل. أشارت النتائج إلى معالجة قيود الحلول التقليدية من خلال تمكين جمع البيانات في الوقت المناسب والشامل، مما يجلب فوائد كبيرة في مجال السلامة والتكاليف ومرونة النقل وإدارة الحضرية. بشكل عام، يساعد هذا العمل في تعزيز إمكانات الأدوات الرقمية التعاونية لإحداث ثورة في وسائل النقل الأذكى والأكثر أمانًا للجميع من خلال التعلم المستمر .

الكلمات المفتاحية: ،اكتشاف العقبات على الطرق ، السيارات ذاتية القيادة ، Microsoft kinect sensor ، أنظمة تجنب العوائق ، استشعار الجماعي ، YOLOv8 ، اندماج المستشعرات المتطورة ، التخطيط ثلاثي الأبعاد ، دقة الكشف في الوقت الحقيقي تحسين صور العمق .