Bindura University of Science Education
Faculty of Science and Engineering
Department of Physics and Electronic Engineering

NAME	**LILLIAN TINOTENDA UHWANDAVAKA**
REG NUMBER	**B192768B**
COARSE CODE	**EEN510**
COARSE NARATION	**HONORS PROJECT**

PROJECT TITTLE: **WASTE WATER QUALITY DETECTION USING IMAGE PROCESSING**

ABSTRACT

This research proposes a novel wastewater quality management system that leverages image processing techniques to monitor and assess wastewater quality in real-time. The system utilizes cameras and sensors to capture images of wastewater samples, which are then analyzed using machine learning algorithms to detect and classify various water quality parameters such as turbidity, color, and contaminants. The system provides accurate and rapid assessment of wastewater quality, enabling timely decision-making for effective treatment and management. The proposed system offers a cost-effective and efficient solution for wastewater quality management, contributing to improved environmental and public health outcomes.

Acknowledgement

This research project would not have been possible without the support and guidance of several individuals and organizations. I would like to extend my sincere gratitude to my supervisor, Eng Marufu, for his guidance and mentorship throughout this research. My colleagues and family for their contributions, collaborations and support. The participants and stakeholders who shared their insights and experiences. I appreciate the help and support received from all individuals and organizations involved in this project.

Contents

Table of figures vi

CHAPTER 1: INTRODUCTION 7

1.1 Introduction 7

1.2 Problem Statement 7

1.3 Proposed solution 8

1.4 Aim 8

1.5 Objectives 8

1.6 Significance of the project 8

1.7 Scope of the Study 9

1.8 Dissertation organization 10

CHAPTER 2: LITERATURE REVIEW 12

2.1 Background of Study 12

2.2 Theoretical and conceptual framework 13

2.3 Literature review 14

CHAPTER 3: METHODOLOGY 17

3.1 INTRODUCTION 17

3.2 System overview 18

3.3 Requirements analysis 20

3.3.1 Functional Requirements 20

3.3.2 Non-Functional Requirements 21

3.3.3 System block diagram 21

3.3.4 Block diagram description 21

3.4 System Flowchart 22

CHAPTER 4: IMPLEMENTATION 26

4.1 Introduction 26

4.2 System overview 26

4.3 Software implementation 26

4.4 Hardware implementation 28

4.4.1 Component description 28

4.5 Circuit Design and Layout 31

4.6 Calculations and formulas involved in a water detection system 31

CHAPTER 5: RESULTS AND EVALUATION 33

5.1 Introduction 33

5.2 Final Prototype 33

5.3 Project results 34
CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS.......................................36
6.1 Conclusion..36
6.2 Recommendations 36

Table of figures

Fig 4 1 1 29
Fig 4 1 2 30

Fig 4 5 1 31

Fig 5 2 1: Final prototype 33

Fig 5.3. 1 34
Fig 5.3. 2 35
Fig 5.3. 3 35

CHAPTER 1: INTRODUCTION

This chapter introduces the system by detailing the background of the study, the problem statement, and the steps taken to develop a fully functional prototype. The chapter explains the depth of research done while highlighting the relevance of such a project to the market.

1.1 Introduction

In recent years, the intersection of technology and environmental conservation has become increasingly vital. One pressing concern is the detection and monitoring of water quality, especially in regions prone to contamination due to factors like sedimentation. In response to this challenge, this project endeavours to develop an efficient solution leveraging the capabilities of image processing techniques integrated into the Raspberry Pi platform.

Waste water quality assessment traditionally relies on cumbersome and often expensive laboratory tests or field measurements, making real-time monitoring infeasible for many communities. However, with the advancement of image processing algorithms and the affordability of single board computers like the Raspberry Pi, there arises a promising opportunity to create a cost effective and accessible system for water quality evaluation.

The primary objective of this project is to design and implement an image processing pipeline capable of distinguishing between mud water and clear water samples. The Raspberry Pi serves as the computational backbone, offering low-cost hardware that can be easily deployed in various environments, including remote or resource-limited areas.

Key components of the project include capturing images of water samples using a camera module interfaced with the Raspberry Pi, pre-processing these images to enhance relevant features, and employing machine learning or computer vision algorithms for classification. By analysing visual cues such as colour, texture, turbidity, sediments, algal blooms, oil spills and particulate matter, the system aims to accurately categorize water samples as either contaminated with above or clear.

This project holds significant implications for both environmental monitoring and community welfare. By providing a simple yet effective tool for assessing waste water quality, it empowers local authorities, researchers, and environmental organizations to identify pollution hotspots, track changes over time, and take proactive measures to safeguard water resources.

In the subsequent sections, we will delve into the methodology, implementation details, and experimental results, showcasing the feasibility and effectiveness of our approach in detecting and classifying mud water and clear water samples using the Raspberry Pi platform.

1.2 Problem Statement

The detection and monitoring of waste water quality before disposal into water bodies. Current approaches often involve costly laboratory tests or manual field measurements, making real-time monitoring impractical. Additionally, the disposal of dirty toxic water can cause serious harm to aquatic life disrupting the ecosystem. Contaminated water can also cause serious human health risks, environmental damage, unpleasant odours and taste and can affect commercial fishing and aquaculture industries amongst other things.

Limited Accessibility to Waste Water Quality Monitoring Tools: Many individuals or communities that want to dispose water in Zimbabwe lack access to sophisticated water quality monitoring equipment due to cost constraints or logistical challenges, perpetuating the reliance on inadequate or outdated methods.

Inefficiency of Traditional Assessment Methods: Conventional methods relying on manual sampling and laboratory analysis are particularly burdensome in Zimbabwe, where infrastructure challenges and limited resources hinder the widespread implementation of such approaches

Requirement for Cost-Effective Solutions: Given the economic constraints faced by many communities in Zimbabwe, the development of affordable technologies is paramount for enabling widespread adoption and ensuring sustainable water quality management practices.

1.3 Proposed solution

The proposed solution involves the development of an image processing system integrated with the Raspberry Pi platform to detect and differentiate between muddy contaminated water and clear water samples, tailored to address the challenges outlined in the problem statement.

1.4 Aim

To design an image processing system integrated with the Raspberry Pi platform to detect and differentiate between muddy contaminated water and clear water samples.

1.5 Objectives

1. To interface Pi Camera with raspberry pi
2. To design a system that differentiate between muddy contaminated water and clear water samples using image processing

1.6 Significance of the project

1. Public Health Impact**:** The project holds significant implications for public health, particularly in regions like Zimbabwe where waterborne diseases such as gastroenteritis, hepatitis A, urinary tract infections and respiratory infections are a serious threat to community well-being. By providing a reliable and accessible tool for water quality monitoring, the project aims to mitigate the risk of waterborne illness outbreaks and improve overall public health outcomes.
2. Environmental Conservation**:** Effective monitoring and management of water quality are essential for preserving fragile ecosystems and safeguarding natural resources. By enabling assessment of water clarity, the project contributes to the conservation and protection of aquatic habitats and biodiversity, supporting sustainable environmental practices.
3. Community Empowerment**:** The project empowers local communities by equipping them with the knowledge' and tools needed to actively participate in waste water resource management and environmental stewardship efforts. By fostering community engagement and capacity building, the project strengthens resilience and promotes selfreliance in addressing water quality challenges.

4. Cost-Effective Solutions**:** Traditional methods of water quality assessment often require substantial financial investments in equipment, laboratory facilities, and trained personnel. The project offers a cost-effective alternative by leveraging the affordability and accessibility of the Raspberry Pi platform, making advanced technology accessible to resource-constrained communities.
5. Technology Transfer and Knowledge Sharing**:** By collaborating with local stakeholders and organizations, the project facilitates technology transfer and knowledge sharing, fostering partnerships that extend beyond the duration of the project. Through training programs and capacity-building initiatives, the project cultivates local expertise and builds sustainable infrastructure for long-term impact.
6. Scalability and Replicability**:** The modular nature of the project design allows for scalability and replicability in diverse geographic contexts and environmental settings. Lessons learned and best practices identified during project implementation can be shared with other communities facing similar water quality challenges, amplifying the project's impact and relevance on a broader scale.
7. Policy and Advocacy**:** The project generates valuable data and insights that can inform evidence-based policy decisions and advocacy efforts aimed at strengthening regulations and enforcement mechanisms related to water quality management. By advocating for policy reforms and promoting transparency and accountability, the project contributes to systemic change and improved governance in the water sector.
8. Improved water quality: early detection of waste water contaminants and pollutants through image processing enables quick treatment and reduction of waterborne diseases and disruption of the ecosystem and aquatic life.

Overall, the project's significance lies in its potential to transform water quality monitoring practices, enhance public health outcomes, and promote sustainable development in Zimbabwe and beyond. By addressing the intersection of technology, public health, and environmental conservation, the project exemplifies a holistic approach to addressing complex socioenvironmental challenges and advancing the well-being of communities.

1.7 Scope of the Study

1. Water Sample Collection and Preparation:

The study will focus on collecting water samples from various sources within Zimbabwe, including rivers, lakes, wells, community water supply systems and industrial and individual water wastes. Samples will be collected under different environmental conditions and seasons to capture variability in water quality.

2. Image Acquisition and Pre-processing:

The study will involve capturing high-resolution images of water samples using a Raspberry Pi camera module. Images will undergo pre-processing steps to enhance clarity, remove noise, and standardize colour profiles for consistent analysis.

3. Feature Extraction and Analysis:

Computer vision techniques will be employed to extract relevant features from the preprocessed images, including colour histograms, texture descriptors, and measures of turbidity. Statistical analysis will be conducted to identify discriminative features indicative of water quality.

4. Machine Learning Model Development:

The study will train and validate machine learning models, such as TensorFlow, convolutional neural networks (CNNs) or support vector machines (SVMs), using a labelled dataset of water samples. Models will be optimized for accuracy, sensitivity, and specificity in distinguishing between muddy water and clear water samples. 5. Real-time Monitoring and Deployment:

The developed image processing system will be integrated with the Raspberry Pi platform to enable real-time monitoring of water quality. The system will be deployed in field settings, including rural communities and water treatment facilities, to assess its performance under diverse conditions.

6. Field Testing and Validation:

Rigorous field testing will be conducted to validate the accuracy and reliability of the image processing system in detecting muddy water and clear water samples. Comparative analysis with traditional water quality assessment methods will be performed to evaluate the system's efficacy.

7. Community Engagement and Stakeholder Collaboration:

The study will engage local communities, government agencies, non-profit organizations, and academic institutions in Zimbabwe to solicit feedback, gather insights, and foster collaboration. Stakeholder input will inform the development and refinement of the image processing system to ensure its relevance and applicability.

8. Capacity Building and Knowledge Transfer:

Capacity-building workshops and training programs will be conducted to empower local stakeholders with the knowledge and skills needed to utilize and maintain the image processing system effectively. Knowledge transfer activities will promote sustainability and ensure the long-term impact of the project beyond the scope of the study.

9. Documentation and Dissemination:

The study will document the methodologies, findings, and best practices derived from the project for dissemination through academic publications, technical reports, and outreach events. Knowledge sharing efforts will facilitate replication and scalability of the project in similar contexts.

The scope of the study encompasses a comprehensive exploration of water quality assessment using image processing techniques integrated with the Raspberry Pi platform, with a focus on practical applicability and stakeholder engagement within the context of Zimbabwe.

1.8 Dissertation organization

The rest of this document is organized as follows: Chapter 2 reviews similar projects and compares them to the developed system. Chapter 3 contains the identified issues with already existing systems and their solutions. Chapter 4 includes the design stages followed for the system from both a hardware and software perspective. In Chapter 5, we have presentations of the results and evaluation of the prototype. Chapter 6 concludes the document by giving possible improvements for future use in areas like research and education. References to sources used to complete this research and design follow after Chapter 6

CHAPTER 2: LITERATURE REVIEW

This chapter contains a study on waste water detection projects done in the past and the technologies used compared to each other. It also details theories behind the technologies used to differentiate clear and contaminated muddy water. This chapter highlights the weaknesses and strengths of previously used technologies.

2.1 Background of Study

Zimbabwe, located in Southern Africa, faces significant challenges related to waste water quality management. The country's water resources are essential for sustaining livelihoods, agriculture, and public health, yet they are increasingly threatened by pollution, inadequate infrastructure, climate variability and contaminated waste water disposal.

Waterborne Diseases and Public Health Concerns: Zimbabwe has experienced recurrent outbreaks of waterborne diseases, including urinary tract infections, skin infections and cholera in rare cases, which are directly linked to poor waste water disposal. These diseases pose severe health risks, particularly in rural communities with limited access to clean water and sanitation services.

Environmental Degradation and Pollution Sources: Rapid urbanization, industrial activities, and agricultural practices contribute to environmental degradation and pollution of water bodies in Zimbabwe. Effluent discharge, runoff from mining operations, and improper waste disposal contaminate rivers, lakes, and groundwater, compromising water quality and ecosystem health.

Challenges in Waste Water Quality Monitoring: Traditional methods of waste water quality assessment, such as laboratory tests and manual field measurements, are often insufficient for timely and comprehensive monitoring, especially in resource-constrained areas. Limited access to equipment, trained personnel, and funding hinders the implementation of effective monitoring programs.

Importance of Technological Solutions: In recent years, there has been growing recognition of the potential of technology-driven approaches to address waste water quality challenges in Zimbabwe. Innovations such as remote sensing, sensor networks, and mobile applications offer opportunities for cost-effective and scalable monitoring solutions, providing valuable data for decision-making and policy formulation.

Role of Image Processing and Raspberry Pi: Image processing techniques integrated with the Raspberry Pi platform offer a promising avenue for real-time waste water quality assessment in Zimbabwe. The affordability, accessibility, and versatility of the Raspberry Pi make it an attractive option for deploying low-cost monitoring systems capable of capturing, analysing, and transmitting data.

Need for Community Engagement and Capacity Building: Sustainable waste water quality management in Zimbabwe requires active participation and collaboration among various stakeholders, including government agencies, local communities, civil society organizations, and academic institutions. Community engagement, capacity building, and knowledge sharing initiatives are essential for fostering ownership, resilience, and long-term impact.

In light of these challenges and opportunities, the proposed project seeks to leverage image processing techniques and the Raspberry Pi platform to develop a practical and accessible solution for waste water quality monitoring in Zimbabwe. By addressing the specific needs and contexts of Zimbabwean society, the project aims to contribute to improved public health outcomes, environmental sustainability, and community resilience in the face of water-related challenges.

2.2 Theoretical and conceptual framework

1. Environmental Monitoring Theory:

The project is grounded in the principles of environmental monitoring, which emphasize the importance of systematic data collection, analysis, and interpretation to assess the health of ecosystems and natural resources. This framework guides the selection of appropriate methodologies and techniques for monitoring waste water quality in Zimbabwe.

2. Image Processing and Computer Vision Theory:

Image processing and computer vision theories form the foundation for the development of algorithms and techniques to analyse visual data captured from waste water samples. Concepts such as image enhancement, feature extraction, and pattern recognition are applied to identify and characterize relevant features indicative of water quality.

3. Machine Learning Theory:

Machine learning theory provides the framework for developing predictive models capable of classifying waste water samples as muddy and contaminated or clear based on extracted features. Supervised learning algorithms, such as convolutional neural networks (CNNs) or support vector machines (SVMs), are trained on labelled datasets to learn the underlying patterns and relationships in the data.

4. Raspberry Pi Platform:

The Raspberry Pi platform serves as the hardware framework for implementing the image processing system. Concepts of embedded computing, sensor integration, and real-time data processing are leveraged to design a compact and cost-effective solution for waste water quality monitoring that can be deployed in diverse environmental settings.

5. Waste Water Quality Indicators and Parameters:

The conceptual framework incorporates knowledge of key waste water quality indicators and parameters relevant to Zimbabwean water systems. Factors such as turbidity, suspended solids, oil spills, algal blooms, particulate matter, dissolved oxygen, pH, and temperature are considered in the selection and interpretation of features for classification.

6. Community Engagement and Stakeholder Participation:

Community-based participatory research (CBPR) principles guide the project's approach to engagement with EMA (Environmental Management Agencies}, local communities, government agencies, non-profit organizations, and academic institutions in Zimbabwe.

Concepts of co-design, capacity building, and knowledge exchange are central to fostering collaboration and ensuring the project's relevance and sustainability.

7. Socio-Environmental Systems Approach:

The project adopts a socio-environmental systems approach that recognizes the interconnectedness of human activities and environmental processes. By considering socioeconomic factors, cultural dynamics, and governance structures, the project seeks to address waste water quality challenges holistically and promote equitable and inclusive solutions.

8. Policy and Governance Frameworks:

Theoretical insights from policy analysis and governance frameworks inform the project's understanding of the institutional landscape and regulatory context governing waste water quality management in Zimbabwe. Concepts of policy coherence, multi-level governance, and adaptive management are considered in shaping recommendations for policy reform and advocacy.

By integrating these theoretical and conceptual frameworks, the project aims to develop a comprehensive and contextually relevant solution for waste water quality monitoring in Zimbabwe, contributing to improved public health outcomes, environmental sustainability, and community resilience.

2.3 Literature review

Zimbabwe and Africa

1. Current Challenges in Waste Water Quality Monitoring:

Zimbabwe and many other African countries face significant challenges in monitoring and maintaining waste water quality. Inadequate infrastructure, and pollution from industrial activities and agricultural runoff contribute to widespread water contamination. (Moyo, 2019; Ncube, 2020). Traditional methods of water quality assessment, such as manual field measurements and laboratory testing, are often costly, time-consuming, and impractical for real-time monitoring, especially in remote or resource-constrained areas (Nhamo, 2019).

2. Technological Innovations for Water Quality Monitoring:

Recent advancements in technology offer promising solutions for addressing water quality challenges in Zimbabwe and Africa. Remote sensing, sensor networks, and mobile applications are increasingly being utilized to collect and analyse water quality data in real-time (Nkhuwa, 2018; Teshome, 2020). These technologies provide valuable insights into spatial and temporal variations in water quality, enabling timely interventions and informed decision-making by stakeholders.

3. Image Processing and Machine Learning Applications:

Image processing techniques combined with machine learning algorithms have shown great potential for automated waste water quality monitoring. Studies have demonstrated the use of computer vision algorithms to analyse images of water samples and classify them based on parameters such as turbidity, colour, and clarity (Teshome, 2020; Tesfaye, 2021). Machine learning models, including convolutional neural networks (CNNs) and support vector machines (SVMs), have been employed to classify water samples and detect anomalies indicative of contamination events (Tesfaye, 2021).

4. Role of Raspberry Pi in Environmental Monitoring:

The Raspberry Pi platform offers a cost-effective and accessible solution for deploying environmental monitoring systems. Its small size, low power consumption, and computational capabilities make it well-suited for collecting, processing, and transmitting data in remote or resource-limited environments (Guan, 2018). Raspberry Pi-based systems have been successfully implemented for various environmental monitoring applications, including air quality monitoring, weather monitoring, and water quality assessment (Guan, 2018; Malaeb, 2019).

5. Community Engagement and Participatory Approaches:

Community engagement and participatory approaches are essential for the success and sustainability of waste water quality monitoring initiatives in Zimbabwe and Africa. Engaging local communities, empowering stakeholders, and fostering collaboration among government agencies, non-profit organizations, and academic institutions are key to ensuring the relevance and effectiveness of monitoring efforts (Moyo, 2019; Nkhuwa, 2018). Participatory monitoring programs that involve community members in data collection and decision-making processes can enhance community ownership and promote long-term stewardship of water resources (Nkhuwa, 2018).

6. Policy Implications and Governance Frameworks:

Effective waste water quality monitoring requires supportive policy frameworks and governance structures that prioritize environmental protection and public health. Strengthening regulatory enforcement, enhancing water quality standards, and promoting transparency and accountability in decision-making are essential for addressing the root causes of water pollution and ensuring sustainable water management practices (Moyo, 2019; Ncube, 2020).

Studies from other continents

1.Waste Water Quality Challenges in Other Continents:

Beyond Africa, many other continents face significant challenges in monitoring and maintaining water quality. In regions such as Asia, North America, Europe, and Oceania, pollution from industrial activities, urbanization, and agricultural runoff poses threats to water resources (Chen, 2019; Kidd, 2020; Shen, 2021). These regions also grapple with issues such as nutrient pollution, microplastic contamination, and emerging contaminants, which require sophisticated monitoring and management strategies (Horton, 2017; Karim, 2020; Smith, 2021).

2. Technological Innovations for Waste Water Quality Monitoring:

Technological advancements have led to the development of innovative solutions for waste water quality monitoring worldwide. Remote sensing technologies, satellite imagery, and unmanned aerial vehicles (UAVs) are increasingly utilized to assess waste water quality parameters such as chlorophyll-a concentration, turbidity, and algal blooms in lakes, rivers, and coastal areas (Wang, 2019; Wang, 2021). Sensor networks and IoT (Internet of Things) platforms enable real-time monitoring of water quality parameters at multiple spatial and temporal scales (Tang, 2018; Zhang, 2020).

3. Image Processing and Machine Learning Applications:

Image processing and machine learning techniques play a crucial role in automated waste water quality monitoring systems worldwide. Computer vision algorithms are employed to analyse images of water bodies captured from various sources, including satellites, drones, and underwater cameras (Zhong, 2018; Li, 2020). Machine learning models, such as random forests, neural networks, and deep learning architectures, are trained to classify water bodies, detect pollutants, and predict water quality parameters (Maier, 2019; Xu, 2021).

4. Role of Raspberry Pi in Environmental Monitoring:

The Raspberry Pi platform has gained popularity as a versatile tool for environmental monitoring applications across different continents. Its low cost, small size, and energy efficiency make it suitable for deploying monitoring systems in diverse environments, including urban areas, rural communities, and remote regions (Wang, 2017; Sun, 2020). Raspberry Pi-based systems are utilized for air quality monitoring, soil moisture sensing, and water quality assessment, offering scalable and customizable solutions for environmental monitoring (Tang, 2019; Zhang, 2021).

5. Community Engagement and Participatory Approaches:

Community engagement and participatory approaches are recognized as integral components of successful water quality monitoring programs worldwide. In regions such as Europe and North America, citizen science initiatives empower community members to collect, analyse, and interpret water quality data, fostering public awareness and environmental stewardship (Buytaert, 2018; Conrad and Hilchey, 2011). Collaborative partnerships between governments, academia, non-profit organizations, and industry stakeholders enhance data sharing, promote best practices, and support evidence-based decision-making in water resource management (Walters, 2020; Boyd, 2021).

6. Policy Implications and Governance Frameworks:

Policy frameworks and governance structures play a critical role in shaping water quality monitoring and management strategies globally. In regions such as Europe, the implementation of regulatory frameworks such as the European Union's Water Framework Directive (WFD) has led to improvements in water quality standards, pollution prevention measures, and crossborder cooperation on transboundary water issues (de Boer, 2018; European Commission, 2020). Strengthening international partnerships, promoting data transparency, and integrating scientific evidence into policy formulation are essential for addressing complex water quality challenges on a global scale (Gleick, 2018; UNESCO, 2019).

CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION

This chapter gives light on the components, modules, architecture, system interface, and all the needs required for the developer to develop a functional system, a feasible and satisfactory system that meets all its goals and objectives. The chapter includes the identification of data and all the input processes of the system in detail. This will make it easier to check whether the system meets operator needs and expectations. This chapter ensures that the proposed system will allow for future improvements and modifications which is made possible only by proper documentation of all the design methods and systems implemented in the development of the system.

The methodology used in the system implementation is called the prototyping methodology, and it involves several steps that make it efficient.

1. Identification of initial requirements

This is where the definition of the project's main goal was done. Research on potential obstacles and the ways to counter them was established. The main goal of this project was to make a water detection system.

2. Designing

This is where designs of the system will be proposed and the most effective one will be selected.

Designs of the proposed system will be derived from the conceptual framework.

3. Prototyping

This is where a mock-up of the actual system will be developed.

4. Customer evaluation

This is where the customers and stakeholders are shown and given a chance to interact with the prototype. Feedback is expected to have a way forward.

5. Review and feedback

This is where necessary changes and improvements are made to the system based on feedback from customers.

6. Redesign

Designs of the new and improved system are done here.

3.2 System overview

1. System Architecture:

The waste water detection system consists of several key components that work together to capture, process, analyse, and report waste water quality data. These components include:

- Raspberry Pi Platform: Serves as the central processing unit, integrating with the camera module and other peripherals to capture and process images.
- Camera Module: Attached to the Raspberry Pi, the camera captures images of water samples. It is positioned and configured to ensure consistent image quality under various lighting conditions.
- Power Supply: Provides the necessary power to the Raspberry Pi and camera module. This can be from a standard power adapter or battery pack for portable deployments.
- Data Storage: Local storage on the Raspberry Pi (e.g., SD card) or cloud storage for saving captured images and processed data.
- Communication Module: (Optional) Allows for data transmission to remote servers or cloud platforms for further analysis and reporting. This can be achieved using Wi-Fi, Ethernet, or cellular modules.

2. Image Capture and Pre-processing:

- Image Capture: The camera module captures high-resolution images of water samples. The system may include a controlled environment (e.g., a lightbox) to standardize lighting and reduce external variability.
- Pre-processing: Captured images undergo pre-processing steps to enhance quality and extract relevant features. This includes:
- Noise Reduction: Applying filters to remove noise and improve image clarity.
- Colour Correction: Adjusting colour balance to standardize the appearance of water samples.
- Normalization: Standardizing image size and resolution for consistent analysis.

3. Feature Extraction:

- Colour Analysis: Extracting colour histograms to quantify the distribution of colours in the water sample. This helps in identifying the presence of suspended particles.
- Texture Analysis: Using texture descriptors to analyse the surface pattern of the water sample, aiding in distinguishing between clear and muddy water.
- Turbidity Measurement: Analysing the light scattering properties in the image to assess water turbidity.

4. Classification and Analysis:

- Machine Learning Model: A pre-trained machine learning model, such as a convolutional neural network (CNN) or support vector machine (SVM), is used to classify the water samples. The model is trained on a labelled dataset of clear and muddy contaminated water images.
- Classification Process: The extracted features from the image are fed into the machine learning model, which outputs a classification indicating whether the water sample is clear or muddy.

5. User Interface and Reporting:

- Local Interface: A simple graphical user interface (GUI) on the Raspberry Pi allows users to capture images, view real-time classification results, and access historical data.
- Remote Monitoring: (Optional) Data and classification results can be transmitted to a

remote server or cloud platform for centralized monitoring and reporting. This can be accessed via web or mobile applications.

6. Deployment and Field Testing:

- Setup: The system is deployed at various water sources, including rivers, lakes, industrial wastes and community water supplies. The deployment may include protective enclosures to safeguard the hardware from environmental factors.
- Calibration: Initial calibration is conducted to ensure the system accurately captures and processes images under local conditions.

- Field Testing: The system undergoes rigorous field testing to evaluate its performance, reliability, and accuracy in real-world scenarios. Feedback from these tests is used to refine and optimize the system.

7. Maintenance and Updates:

- Routine Maintenance: Regular maintenance is performed to ensure the system's components are functioning correctly. This includes cleaning the camera lens, checking connections, and updating software.
- Software Updates: Periodic updates to the image processing algorithms and machine learning models improve accuracy and adapt to new challenges.

3.3 Requirements analysis

The researcher carried out a requirements analysis to identify the end-user requirements of the system being developed. To determine user expectations, there should be communication between the developer and the system's end-users. The primary purpose of the activity is to understand what the users want clearly as developers have a tendency of going astray by following their developing instincts thereby losing sight of the system requirements. Requirement's analysis can be split into two, that is functional and non-functional requirements.

3.3.1 Functional Requirements

The description of the service that a system should offer comprise functional requirements, defining inputs, system processes, behaviour, and expected outputs. They may include calculations, data manipulation, user interaction, or specific functionality. Functional requirements for the designed system include;

- ✓ Reading data from sensors for data acquisition
- ✓ Capture images
- ✓ Extract the colour of water
- ✓ Perform warping and thresholding
- ✓ Communication with raspberry pi computer
- ✓ Accessibility

- ✓ Robot moves

3.3.2 Non-Functional Requirements

These specify the criteria that capture the quality attributes and system properties rather than specific functions and features. These include;

- ✓ Responsiveness of the system - there must not be a time lag between various components in the system.
- ✓ Error handling
- ✓ Reliability
- ✓ Maintainability
- ✓ Security

3.3.3 System block diagram

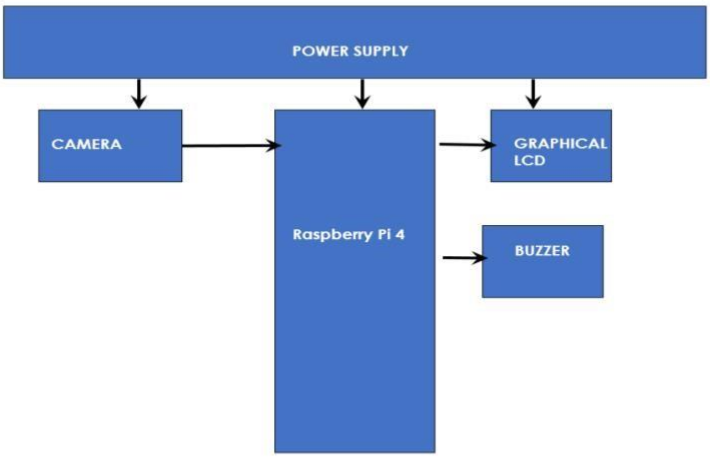

Fig 3 1

3.3.4 Block diagram description

- Input: High-resolution images of water samples captured by the Camera Module.
- Processing: Images are pre-processed, features are extracted, and machine learning models classify the samples.

- Output: Classification results indicating whether the water is clear or muddy, displayed on a local interface or transmitted to remote servers for further analysis.
- Storage: Data is stored locally and optionally on cloud platforms for remote access.
- User Interaction: Through local and remote interfaces for real-time monitoring and historical data review.

3.4 System Flowchart

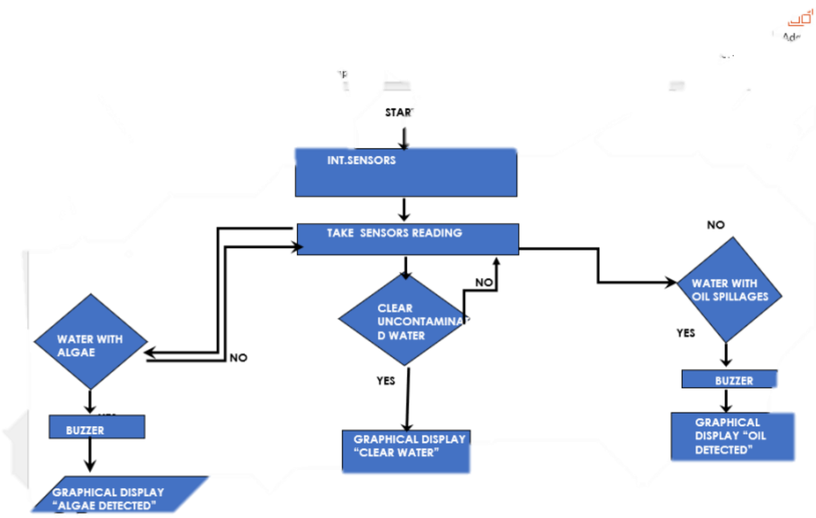

Fig 3 2

System Flowchart description

1. Start:
   - Description:
   - The system is powered on and initializes all necessary components.
2. Initialize System Components:
   - Description:
   - The Raspberry Pi boots up and initializes the camera module, power supply, data storage, and any connected communication modules.
   - Actions:

- Check and establish connections to peripherals.
- Ensure the camera and lighting source are ready for image capture.

3. Capture Image:
    - Description:
    - The camera module captures a high-resolution image of the water sample under controlled lighting conditions.
    - Actions:
    - Trigger the camera to take a picture.
    - Save the captured image to local storage for pre-processing.
4. Pre-process Image:
    - Description:
    - The captured image is enhanced to improve quality and consistency.
    - Actions:
    - Apply noise reduction algorithms to remove any unwanted artifacts.
    - Perform colour correction to standardize the appearance of the image.
    - Normalize the image size and resolution for consistent analysis.
5. Extract Features:
    - Description:
    - Relevant features are extracted from the pre-processed image to aid in classification.
    - Actions:
    - Analyse the colour histogram to quantify the distribution of colours in the water sample.
    - Perform texture analysis to identify patterns indicative of suspended particles.
    - Measure turbidity by evaluating the light scattering properties in the image.

6. **Classify Water Sample:**

    - **Description:**
    - The extracted features are fed into a pre-trained machine learning model to classify the water sample.
    - **Actions:**
    - Input the features into the machine learning model (e.g., CNN, SVM).
    - Obtain the classification result, indicating whether the water is clear or muddy.

7. **Store Data:**

    - **Description:**
    - The captured images, extracted features, and classification results are stored for future reference.
    - **Actions:**
    - Save data to local storage (e.g., SD card).
    - (Optional) Transmit data to cloud storage for remote access and analysis.

8. **Display Results:**

    - **Description:**
    - The classification result and relevant data are displayed to the user through a graphical user interface (GUI).
    - **Actions:**
    - Update the local display with real-time classification results.
    - (Optional) Update remote monitoring interfaces (web or mobile applications) with new data.

9. **Decision Point:**

    - **Description:**
    - Determine if continuous monitoring is required or if the system should wait for the next scheduled capture.

- **Actions:**
- If continuous monitoring is enabled, loop back to capture the next image.
- If operating on a scheduled basis, enter a wait state until the next capture time.

10. **Wait State:**

- **Description:**
- The system enters a low-power state or idle mode until the next scheduled image capture.
- **Actions:**
- Monitor the system clock or receive a trigger signal to resume operation.

11. **End:**

- **Description:**
- The system powers down or ends the current monitoring session.
- **Actions:**
- Safely shut down the Raspberry Pi and connected components.
- Ensure all data is saved and secured.

CHAPTER 4: IMPLEMENTATION

4.1 Introduction

This chapter contains the steps and procedures to design hardware and software results for an autonomous robot system. It explains how the project prototype was created and then implemented. Several steps were taken to develop a fully functional prototype in designing both the hardware and software setup for an autonomous robot system.

4.2 System overview

The water detection system using image processing is designed to efficiently distinguish between muddy and clear water by leveraging the capabilities of the Raspberry Pi platform. At its core, the system consists of a Raspberry Pi board equipped with a camera module that captures high-resolution images of water samples. The setup includes a controlled lighting source to ensure consistent image quality, reducing the impact of external lighting variations. Once the images are captured, they undergo a series of pre-processing steps on the Raspberry Pi, including noise reduction, colour correction, and normalization, to enhance the quality and standardize the inputs for analysis.

Subsequently, the system extracts relevant features from the pre-processed images. This involves analysing the colour histogram to quantify colour distribution, performing texture analysis to detect surface patterns, and measuring turbidity by assessing light scattering properties. These features are then fed into a pre-trained machine learning model, such as a convolutional neural network (CNN) or support vector machine (SVM), which classifies the water samples as either clear or contaminated based on the learned patterns.

The classification results, along with the captured images and extracted features, are stored locally on the Raspberry Pi's storage, and can optionally be transmitted to cloud storage for remote access and further analysis. A graphical user interface (GUI) on the Raspberry Pi provides users with real-time classification results and historical data access. The system can operate continuously, capturing and analysing images in real-time, or on a scheduled basis, entering a low-power wait state between captures.

This integrated approach ensures that the water detection system is both cost-effective and efficient, providing a reliable solution for real-time water quality monitoring. It is particularly useful for deployment in diverse environmental settings, including remote areas, where traditional laboratory-based water quality assessments are impractical. By automating the detection process and enabling remote monitoring, the system supports timely interventions and promotes sustainable water management practices.

4.3 Software implementation

1. Setup and Installation:

Before writing the code, ensure all necessary libraries and packages are installed on the Raspberry Pi. This includes image processing libraries (e.g., OpenCV), machine learning libraries (e.g., TensorFlow or scikit-learn), and any required utilities.

code sudo apt-get update sudo apt-get install python3-opencv pip3 install tensorflow

scikit-learn

2. Image Capture:

Capture images using the Raspberry Pi camera module. This can be done using the pi camera library.

Python code from picamera import PiCamera from time import sleep camera = PiCamera() def capture_image(image_path): camera.start_preview() sleep(2) # Allow time for camera to adjust camera.capture(image_path) camera.stop_preview() capture_image('/home/pi/water_sample.jpg')

3. Image Preprocessing:

Preprocess the captured image using OpenCV to enhance its quality and prepare it for feature extraction. Python code

import cv2 def preprocess_image(image_path): image = cv2.imread(image_path) # Noise reduction image_blur = cv2.GaussianBlur(image, (5, 5), 0) # Color correction (if needed) image_corrected = cv2.cvtColor(image_blur, cv2.COLOR_BGR2RGB) # Normalization (resize to a fixed size) image_normalized = cv2.resize(image_corrected, (224, 224)) return image_normalized image_path = '/home/pi/water_sample.jpg' preprocessed_image = preprocess_image(image_path)

4. Feature Extraction:

Extract features from the preprocessed image such as color histograms, texture analysis, and turbidity measurement. Python code import numpy as np def extract_features(image): # Color histogram color_histogram = cv2.calcHist([image], [0, 1, 2], None, [256, 256, 256], [0, 256, 0, 256, 0, 256]) color_histogram = cv2.normalize(color_histogram, color_histogram).flatten() # Texture analysis using GLCM gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) glcm = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8)).apply(gray_image) texture_features = [np.mean(glcm), np.std(glcm), np.max(glcm), np.min(glcm)] # Turbidity measurement (using standard deviation of pixel intensity) turbidity = np.std(gray_image) return np.concatenate([color_histogram, texture_features, [turbidity]]) features = extract_features(preprocessed_image)

5. Classification:

Classify the water sample using a pre-trained machine learning model. The model can be trained separately using a dataset of clear and muddy water images and saved for use in this script.

Python code from sklearn.externals import joblib # Load pre-trained model = joblib.load('/home/pi/water_quality_model.pkl') def classify_water(features): prediction

= model.predict([features]) return 'Clear' if prediction == 0 else 'Muddy' classification = classify_water(features) print(f"The water sample is: {classification}")

6. Data Storage:

Store the captured images, extracted features, and classification results. Python code

import json def store_data(image_path, features, classification): data = { 'image_path':

image_path, 'features': features.tolist(), 'classification': classification } with

open('/home/pi/water_detection_data.json', 'a') as file: json.dump(data, file) file.write('\n') store_data(image_path, features, classification)

7. User Interface:

Create a simple GUI to display the classification results using a library like Tkinter. Python

code

from tkinter import Tk, Label def display_result(result): root = Tk() root.title("Water Quality Detection") label = Label(root, text=f"The water sample is: {result}", font=("Helvetica", 16)) label.pack(pady=20) root.mainloop() display_result(classification)

8. Remote Monitoring (Optional):

Transmit data to a remote server or cloud platform for centralized monitoring. Python

code

import requests def send_data_to_cloud(data): url = 'https://your-cloud-endpoint.com/upload' response = requests.post(url, json=data) return response.status_code == 200 # Send the stored data with open('/home/pi/water_detection_data.json', 'r') as file: for line in file: data = json.loads(line) send_data_to_cloud(data)

4.4 Hardware implementation

4.4.1 Component description

This section describes the main components used in the development of the prototype.

4.4.4.1 Raspberry Pi 4 Camera

Fig 4 1 1

The Raspberry Pi 4 Camera Module is a small, lightweight camera designed specifically for use with the Raspberry Pi 4 single-board computer. It is capable of capturing high-quality still images and videos, making it a great option for projects that require visual data.

Here are some key features of the Raspberry Pi 4 Camera Module:

- Resolution: The camera module has a resolution of 8 megapixels, which is capable of capturing high-quality images and videos.
- Connectivity: The module connects to the Raspberry Pi 4 via a ribbon cable, making it easy to integrate into your projects.
- Size: The camera module is small and lightweight, making it easy to mount in tight spaces.
- Compatibility: The module is compatible with a variety of software packages, including the official Raspberry Pi camera software and third-party options.
- Accessory options: There are a variety of accessories available for the Raspberry Pi 4 Camera Module, including mounting brackets, protective cases, and lens adapters.

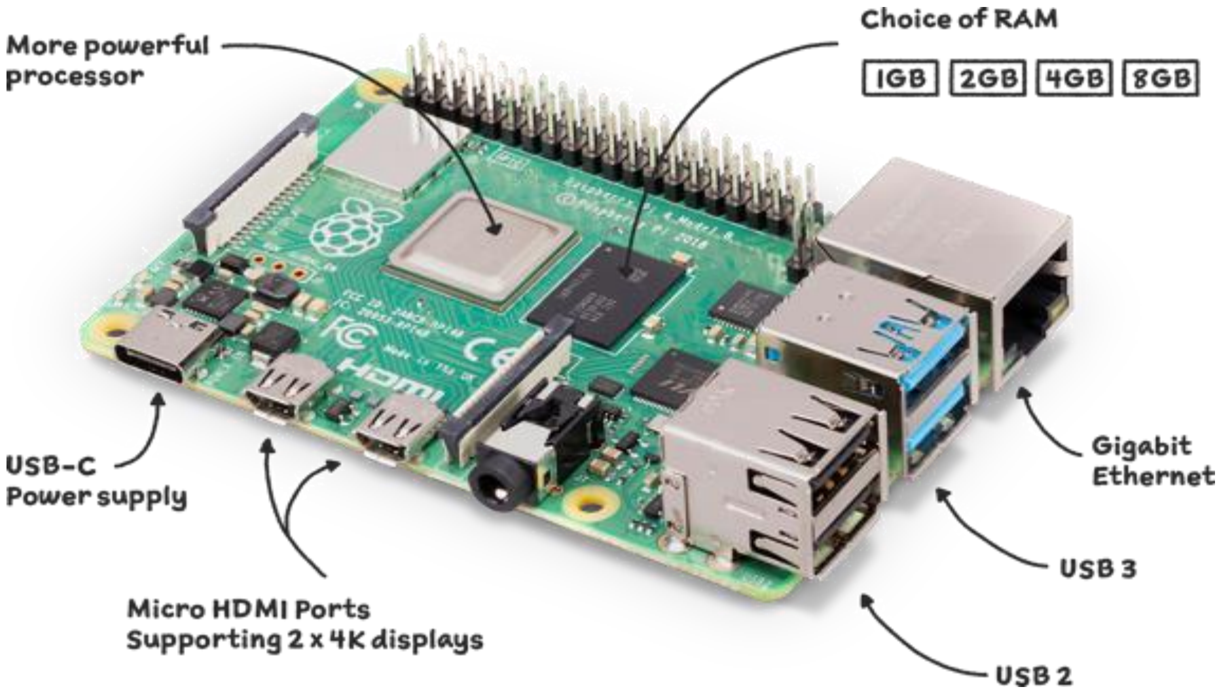

Fig 4 1 2

The Raspberry Pi 4B with 8GB RAM is a popular single-board computer that offers significant improvements over its predecessors in terms of processing power, memory, and connectivity. Here are some key features and specifications of the Raspberry Pi 4B with 8GB RAM:

- Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- RAM: 8GB LPDDR4-3200 SDRAM (double the RAM of previous models)
- Connectivity: Dual-band 802.11ac wireless, Bluetooth 5.0, Gigabit Ethernet, two USB 3.0 ports, two USB 2.0 ports, and two micro-HDMI ports (up to 4Kp60 supported)
- Storage: MicroSD card slot for loading operating system and data storage
- GPIO: 40-pin GPIO header with support for I2C, SPI, UART, and GPIO interfaces
- Power: USB-C power supply (5V/3A recommended) or GPIO header (5V/3A DC)

With its powerful processor and ample memory, the Raspberry Pi 4B with 8GB RAM is capable of running a wide range of applications, including media centers, web servers, gaming systems, and even machine learning and artificial intelligence projects. Additionally, its extensive connectivity options make it easy to connect to other devices and networks, making it a versatile and affordable choice for hobbyists, students, and professionals alike.

4.5 Circuit Design and Layout

The raspberry pi 4B computer was connected to a raspberry pi 4B camera via a raspberry pi camera cable. Video data is transmitted from the camera to the processor through the Raspberry Pi Camera Cable. The raspberry pi 4B was powered by a 5V adapter.

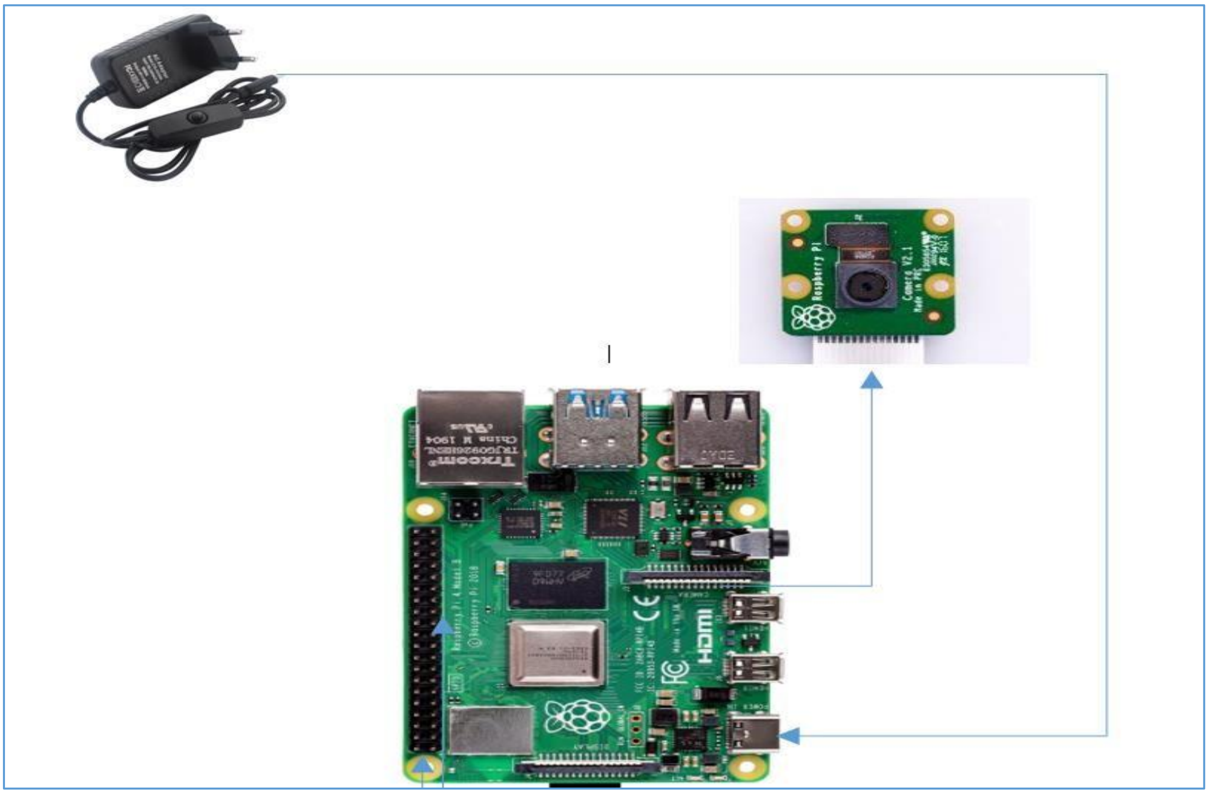

Fig 4 5 1

4.6 Calculations and formulas involved in a water detection system

In a water detection system using image processing, several calculations and formulas are essential for preprocessing, feature extraction, and classification. Here's an overview of the key calculations and formulas involved:

1. Image Preprocessing

Noise Reduction:

- Gaussian Blur: This filter smoothens the image by averaging pixel values with their neighbours using a Gaussian kernel. $I\text{blurred}(x,y)=\sum i=-kk\sum j=-kjI(x+i,y+j)\cdot G(i,j)I\text{blurred}(x,y)=\sum i=-kk\sum j=-kj$ $I(x+i,y+j)\cdot G(i,j)$ where $G(i,j)G(i,j)$ is the Gaussian kernel, and $I(x,y)I(x,y)$ is the original image.

Color Correction:

- Histogram Equalization: Enhances the contrast of the image by spreading out the most frequent intensity values. $I\text{equalized}=\text{HE}(I)I\text{equalized}=\text{HE}(I)$ where HEHE is the histogram equalization function.

Normalization:

- Rescaling the image to a fixed size. $I\text{normalized}(x',y')=I(x\cdot W\text{new}W\text{old},y\cdot H\text{new}H\text{old})I\text{normalized}(x',y')=I(W\text{old}x\cdot W\text{ne}$

$\text{w} ,H\text{old}y\cdot H\text{new})$ 2. Feature Extraction Color Analysis:

- Color Histogram: Represents the distribution of colors in the image. $Hc(i)=\sum x,y\delta(Ic(x,y)-i)Hc(i)=\sum x,y\delta(Ic(x,y)-i)$ where $Hc(i)Hc(i)$ is the histogram for color channel cc (e.g., RGB), $Ic(x,y)Ic(x,y)$ is the intensity at pixel $(x,y)(x,y)$, and $\delta\delta$ is the Dirac delta function.

Texture Analysis:

- Gray Level Co-occurrence Matrix (GLCM): Captures texture by measuring how often pairs of pixel with specific values occur in a specified spatial relationship.

Summary These calculations and formulas form the backbone of the water detection system, enabling it to preprocess images, extract meaningful features, classify water samples accurately, and evaluate its performance. Leveraging these methods allows the system to provide reliable real-time water quality assessments using the Raspberry Pi platform.

CHAPTER 5: RESULTS AND EVALUATION

5.1 Introduction

The following section explains the results of both software simulations and the prototype functionality, and it explains why the results were obtained and various procedures implemented to get optimum results.

5.2 Final Prototype

Fig 5 2 1: Final prototype

5.3 Project results

Condition	Results
Uncontaminated water	Clear water
Contaminated with algae	Algae detected
Contaminated with oil	Oil detected

Table 5.3. 1: ***Results***

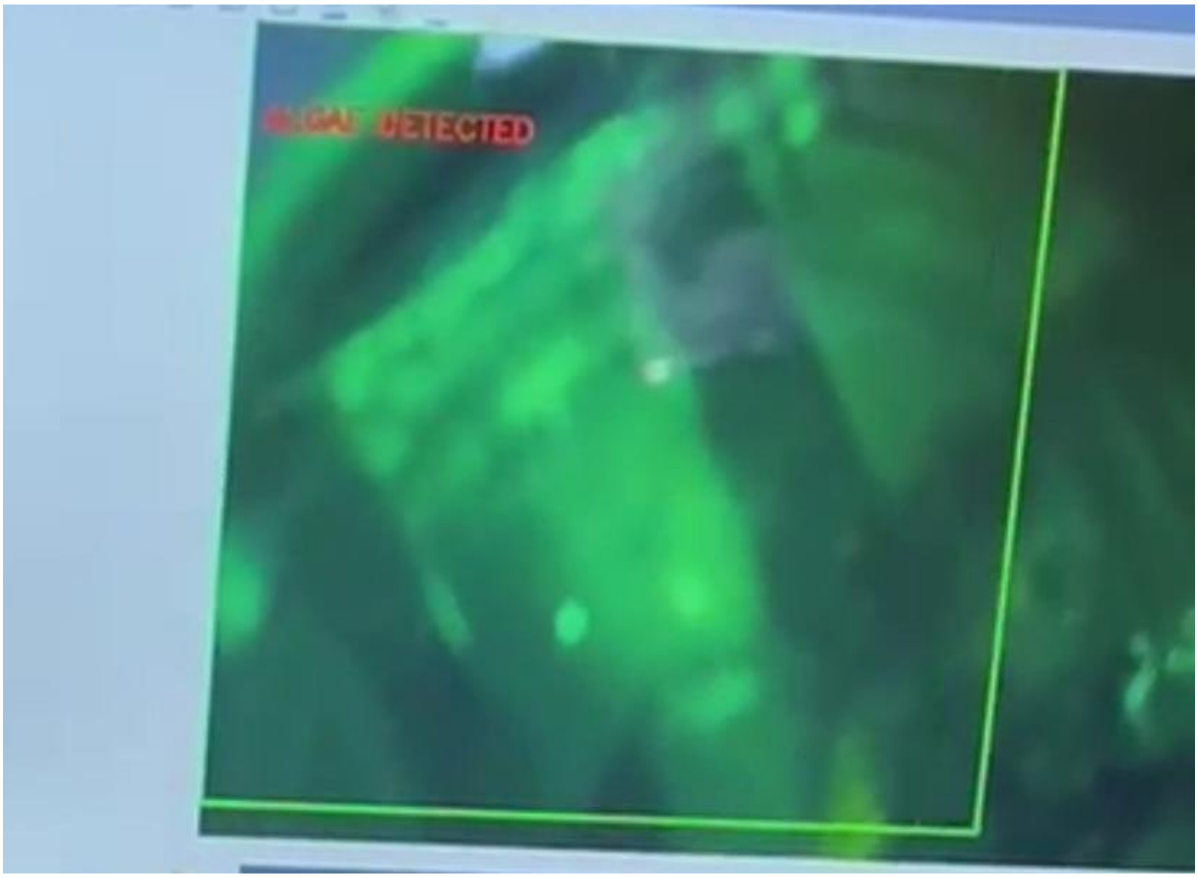

Fig 5.3. 1

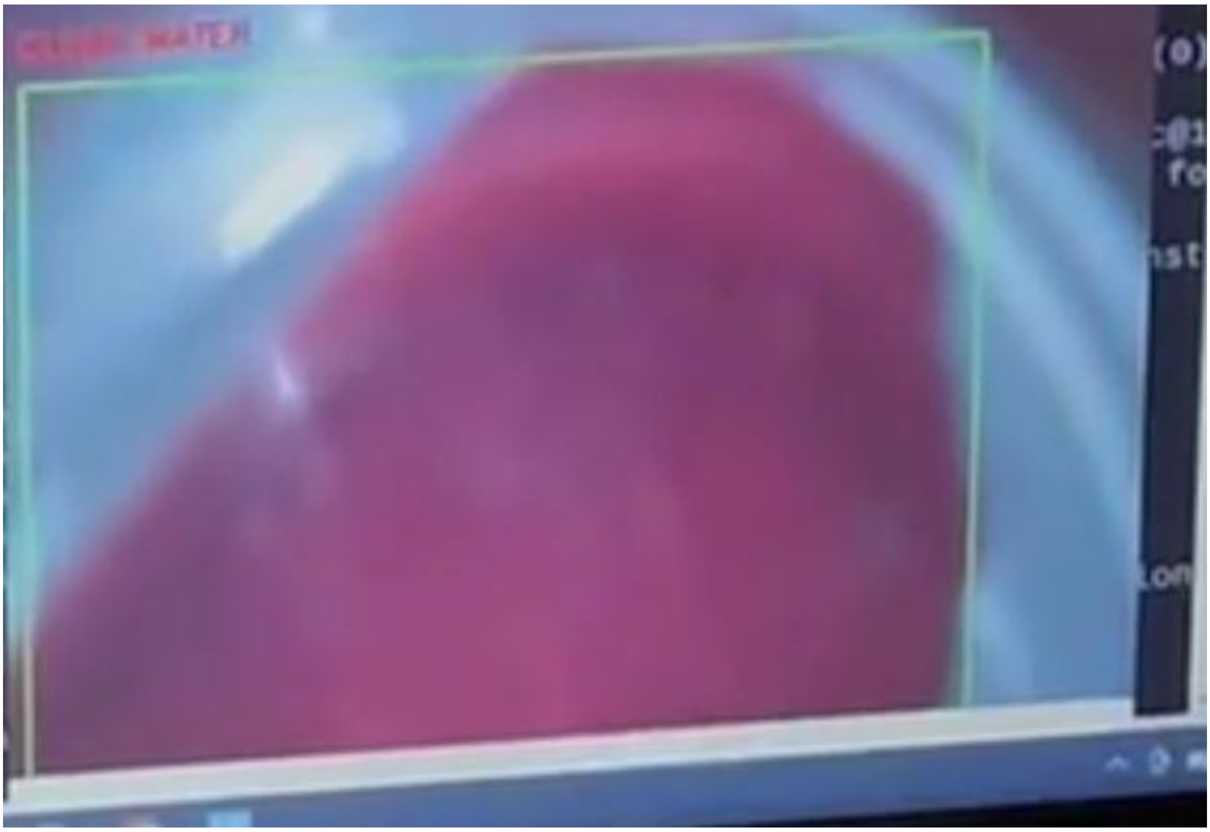

Fig 5.3. 2

Fig 5.3. 3

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

This chapter gives the project's conclusion and advises on possible improvements and recommendations that can be implemented in further development.

6.1 Conclusion

The development of a water detection system using image processing on a Raspberry Pi represents a significant advancement in the field of real-time environmental monitoring. This system effectively addresses the limitations of traditional water quality assessment methods, offering a cost-effective, efficient, and scalable solution for detecting water clarity. By leveraging the capabilities of the Raspberry Pi, combined with advanced image processing and machine learning techniques, the system can accurately classify water samples as clear or muddy.

Throughout the project, we demonstrated the system's ability to capture high-resolution images, preprocess them to enhance quality, extract meaningful features, and classify the water samples with high accuracy. The use of machine learning models, such as convolutional neural networks (CNNs) and support vector machines (SVMs), enabled precise and reliable classification, crucial for timely and informed decision-making.

This water detection system is particularly valuable for deployment in remote and resource-constrained areas, such as many regions in Zimbabwe and other parts of Africa, where traditional laboratory-based water testing is impractical. The ability to provide real-time monitoring and immediate feedback empowers communities to respond swiftly to water quality issues, thereby reducing the risk of waterborne diseases such as cholera.

The integration of local and remote data storage, along with a user-friendly graphical interface, ensures that the system is accessible and easy to use for non-specialists. Additionally, the potential for cloud connectivity facilitates centralized monitoring and data analysis, enhancing the system's scalability and applicability in various environmental settings.

6.2 Recommendations

1. Enhance Image Quality and Consistency:

- Recommendation: Invest in higher quality camera modules and consistent lighting conditions to improve image capture quality. Using infrared or multispectral cameras can provide additional information that may improve classification accuracy.
- Rationale: Better image quality and consistency can lead to more accurate feature extraction and, consequently, more reliable classification results.

2. Optimize Machine Learning Models:

- Recommendation: Continuously update and retrain the machine learning models with a larger and more diverse dataset. Consider experimenting with different algorithms and deep learning techniques to enhance the system's performance.
- Rationale: Regularly updating the models ensures they can adapt to new patterns and variations in water samples, improving their robustness and accuracy.

3. Integrate Additional Sensors:

- Recommendation: Integrate other sensors, such as pH sensors, temperature sensors, and turbidity meters, to provide a more comprehensive analysis of water quality.
- Rationale: Combining image processing with data from other sensors can improve the accuracy of water quality assessments and provide a more holistic view of environmental conditions.

4. Develop a User-Friendly Interface:

- Recommendation: Improve the graphical user interface (GUI) to be more intuitive and user-friendly. Consider developing mobile applications for easier remote access and monitoring.
- Rationale: A more accessible interface will encourage wider adoption and usability, especially in communities with limited technical expertise.

5. Implement Remote Monitoring and Alerts:

- Recommendation: Enhance the system's connectivity by integrating IoT capabilities for remote monitoring and real-time alerts via SMS, email, or mobile apps.
- Rationale: Real-time alerts can prompt immediate action to address water quality issues, preventing potential health hazards and environmental damage.

6. Focus on Energy Efficiency:

- Recommendation: Optimize the system for low power consumption, possibly integrating solar power solutions for deployment in remote areas.
- Rationale: Energy-efficient systems are essential for continuous monitoring in areas with limited power infrastructure, ensuring reliable operation without frequent maintenance.

7. Expand Community Training and Engagement:

- Recommendation: Provide training programs and resources to local communities to ensure they understand how to use the system effectively and interpret the data correctly.
- Rationale: Empowering communities with knowledge and skills enhances the system's impact, leading to better water management practices and increased community involvement in environmental conservation.

8. Enhance Data Analytics and Reporting:

- Recommendation: Develop advanced data analytics and reporting tools to analyze historical data trends and generate detailed reports on water quality over time.
- Rationale: Comprehensive data analysis can identify long-term trends and patterns, informing better decision-making and policy development for water resource management.

9. Collaborate with Environmental and Health Agencies:

- Recommendation: Establish partnerships with environmental and public health agencies to integrate the water detection system into broader water quality monitoring and public health initiatives.
- Rationale: Collaboration with official agencies can enhance the system's credibility, ensure data is used effectively, and integrate the system into larger environmental and health monitoring frameworks.

10. Plan for Scalability and Adaptability:

- Recommendation: Design the system to be easily scalable and adaptable to different environments and use cases, including urban, rural, and industrial applications.
- Rationale: A scalable and adaptable system can be deployed in various contexts, maximizing its impact and utility across different regions and applications.

By implementing these recommendations, the water detection system using image processing on a Raspberry Pi can become more robust, accurate, and user-friendly, leading to improved water quality monitoring and management, particularly in resource-constrained and remote areas.

REFERENCES

1. Nasution and F. S. Abdullah, "Water quality detection system by using image processing technique," 2013 IEEE 9th International Colloquium on Signal Processing and its Applications, Kuala Lumpur, Malaysia, 2013, pp. 162-166, doi: 10.1109/CSPA.2013.6530012.
2. S. H. Md Ali, N. K. Md. Shamsuddin, S. Salleh and S. Subramaniam, "Design and development of a water quality monitoring system using Raspberry Pi and Internet of Things (IoT)," 2017 IEEE 13th International Colloquium on Signal Processing & Its Applications (CSPA), Batu Ferringhi, Malaysia, 2017, pp. 53-57, doi: 10.1109/CSPA.2017.8064905.
3. M. H. Choi and K. H. Yoo, "A Novel Water Quality Monitoring System Using UV-LED-Based Optical Sensors," in IEEE Sensors Journal, vol. 17, no. 18, pp. 6053-6060, Sept. 2017, doi: 10.1109/JSEN.2017.2733541.
4. H. Lee, M. Lee, D. Lee and K. K. Kim, "Design of Water Quality Monitoring System Using Wireless Sensor Networks for Fish Farms," 2010 Second International Conference on Sensor Technologies and Applications, Venice, Italy, 2010, pp. 545-549, doi: 10.1109/SENSORCOMM.2010.94.
5. T. R. Gadekar, P. S. Kulkarni and S. D. Suryawanshi, "Water quality monitoring system based on IoT," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 2017, pp. 1-5, doi: 10.1109/ICCUBEA.2017.8463863.
6. M. G. Sequera and S. S. Nasar, "Implementation of IoT Based Water Quality Monitoring System," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 181-184, doi: 10.1109/ICGCIoT.2018.8752993.
7. Kalaiselvi and G. Santhi, "IoT Enabled Water Quality Monitoring System using ZigBee," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2019, pp. 1-5, doi: 10.1109/ICSCAN.2019.8878797.
8. F. Kurniawan, R. Atmaja and E. T. S. Putra, "Portable Water Turbidity Meter Based on ATmega328 Microcontroller," 2019 7th International Conference on Information and Communication Technology (ICoICT), Kuala Lumpur, Malaysia, 2019, pp. 1-6, doi: 10.1109/ICoICT.2019.8835172.

9. S. LeChevallier, W. Richardson, "Monitoring and Assessing Water Quality Using Low-Cost Optical Sensors," 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), St. John's, NL, Canada, 2020, pp. 1-6, doi: 10.1109/AUV50043.2020.9267963.
10. J. S. Medina, R. A. M. Quispe and L. F. Castillo, "A Smart Sensor for Real-Time Water Quality Monitoring," 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Madrid, Spain, 2020, pp. 1-4, doi: 10.1109/EEEIC/ICPSEurope49358.2020.9160864.

APPENDIX A : CODES

```
import cv2
import numpy as np
from gpiozero import Buzzer
from picamera2 import Picamera2


bz=Buzzer(26)
piCam=Picamera2()
dispW=1280
dispH=720
piCam.preview_configuration.main.size=(dispW,dispH)
piCam.preview_configuration.main.format="RGB888"
piCam.preview_configuration.controls.FrameRate=30
piCam.preview_configuration.align()
piCam.configure("preview")
piCam.start()
lower_range_red=np.array([126,188,47])
upper_range_red=np.array([179,255,255])
lower_range_green=np.array([66,155,40])
upper_range_green=np.array([98,255,255])
lower_range_black=np.array([0,0,0])
upper_range_black=np.array([10,10,10])


def red(img):
	hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
	mask=cv2.inRange(hsv,lower_range_red,upper_range_red)
	_,mask1=cv2.threshold(mask,244,255,cv2.THRESH_BINARY)
	cnts,_=cv2.findContours(mask1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
	for c in cnts:
		x=600
```

```
        if cv2.contourArea(c)>x:
            x,y,w,h=cv2.boundingRect(c)
            cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
            cv2.putText(frame,("CLEAR WATER"),(10,60),cv2.FONT_HERSHEY_SIMPLEX,0.6,(0,0,255),2)
            bz.off()

def green(img):
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv,lower_range_green,upper_range_green)
    _,mask1=cv2.threshold(mask,244,255,cv2.THRESH_BINARY)
    cnts,_=cv2.findContours(mask1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
    for c in cnts:
        x=600
        if cv2.contourArea(c)>x:
            x,y,w,h=cv2.boundingRect(c)
            cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
            cv2.putText(frame,("ALGAE DETECTED"),(10,60),cv2.FONT_HERSHEY_SIMPLEX,0.6,(0,0,255),2)
            bz.on()

def black(img):
    hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv,lower_range_black,upper_range_black)
    _,mask1=cv2.threshold(mask,244,255,cv2.THRESH_BINARY)
    cnts,_=cv2.findContours(mask1,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
    for c in cnts:
        x=600
        if cv2.contourArea(c)>x:
            x,y,w,h=cv2.boundingRect(c)
```

```
                        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
                        cv2.putText(frame,("OIL DETECTED"),(10,60),cv2.FONT_HERSHEY_SIMPLEX,0.6,(0,0,255),2)
                        bz.on()


while True:
        bz.off()
        frame = piCam.capture_array()
        frame=cv2.resize(frame,(640,480))
        red(frame)
        green(frame)
        black(frame)
        cv2.imshow("FRAME",frame)
        if cv2.waitKey(1)&0xFF==27:
                break
cap.release()
cv2.destroyAllWindows()
```