International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 12 | Dec 2024 www.irjet.net p-ISSN: 2395-0072

EFFECTIVE UNDERWATER SEARCH AND SURVEY THROUGH R.O.V.'S

BHUVAN PRATAP AGARWAL[1], VINEETH YELAGANDULA[2], M. SRINATH[3]

[1,2,3]*B. Tech Scholars, Department of Computer Science and Engineering, SNIST, Hyderabad-501301, India.*

Abstract - *Our idea is to build a network of robots that moves inside the sea or ocean wherever the fishing is done in a specified area with the basic knowledge and get the data (microplastic concentration, pH, temperature, dissolved oxygen, etc.) of conditions of the ocean or sea, analyze the data and compares with the data which favors the fish and suggests the best area of fishing. The design is chosen to be streamlined, as it makes the Robot move efficiently inside the water body. The bot is equipped with four thrusters two present on the sides for the movement of the bot to move on the x and y-axis and for the movement on the z-axis the bot uses the concept of neutral buoyancy to move wherein There are two chambers present within the robot. The two chambers are equipped with 2 pumps which take water for the moment of the bot in the negative Z-direction and attain neutral buoyancy with the amount of water taken in, making the density of the robot equal to the water and remaining stable at that position at a depth of 5 meters, for the movement in the positive Z direction, the bot expel out the water which was taken in earlier making the bot lighter than the water to move towards the surface of the water. There are 2 more thrusters present on the bot which take in information from the Gyroscope and accelerometer for maintaining the trajectory of the robot and attenuating the disturbances caused in the water body, thus making the robot complete the trajectory without changing the initial and final mapped points.*

Key Words: AI, UUV, ROV, Embedded, Arduino.

1. INTRODUCTION

Three fourths of the land is covered with water and water is the main resource and energy for human civilization. Water resources are spread over the ocean, sea, river, small catchment area, ponds, lakes, etc. They are used for agriculture, aquatic elements, water minerals, drinking water, industries, home use and there are no areas where water is not required. The livelihood of a huge number of people depends on water resources. A bot is proposed to carry out the work. The bot can easily move around the water surface as well as under the water.

It has the major functional units like

a) automated self-propelled underwater/ surface vehicle (ASPUW/S).
b) Sample collection system
c) Measurement system
d) Location mapping system through GPS.
e) Communication to the cloud through network

Natural water is pollution-free and is clean water. However, due to different manmade pollution and also pollution created due to natural disaster, etc. makes the water dirty. At the same time, the fish excretion in water becomes a problem for their existence. The temperature variation in water also affects the existence of aquatic species. Further, there are many undiscovered resources like minerals are existing in the ocean and seawater. A list of parameters monitored 2 to check the quality of water is listed in appendix-1. Parameters such as pH, turbidity, residual chlorine, and temperature are monitored to check the quality of water.

2. EXISTING SYSTEM

It has been a challenge to cover a wide area water network for surveillance of water quality and the existence of various water resources reaching several benefits to the stakeholders requires planning to cover the water network. Ganga water cleaning, a task initiated by the government of India, is one such work. Several zones are distributed all over the country. Concerned authorities moved people for sample collection and brought them back to the laboratory for testing. But the timeline may not need a sample validity duration. This is also true for the sea and ocean. Fishermen go around the sea to locate the right type of fish and sometimes come back with an empty hand. There are some boats with instrumentation that patrol the specific area and collect the sample. Samples are further tested at a laboratory. There are boat-based systems reported from China, Japan, etc. These systems are instrumented with GPS, a communication system, a sample collection system, and a measurement system. They do the measurement at surface level. But, the majority of parameters are required to be monitored at a depth of 3 to 6 feet.

3. POPOSED SYSTEM

The suggested system includes an Automated Self-Propelled Underwater/Surface Vehicle (ASPUW/S), a sample collection system, a measurement system, a GPS location mapping system, and a network-based communication system to the cloud. The operation begins with positioning the bot on the water's surface and programming it for a specified area of interest and sampling sites. The bot marks its current location with GPS data and moves across the surface to reach the target destination. It descends by absorbing water and increasing its mass in order to stabilize itself for the sample. It measures and stores the data after taking the sample. The bot releases water to minimize its mass before moving up to transmit the measured data via the communication post. It stores certain samples while

releasing the majority and cleaning the sample storage area. The bot notes the current location and moves on to the next, repeating steps (d) to (l). When the bot returns to its original location, all data can be sent to the ground system for further analysis, and samples can be gathered for lab testing. The bot charges its battery and prepares for further operation at the docking site. The system is intended to collect and store samples effectively while minimizing human intervention and errors.

4. SYSTEM ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

a. Data Gathering: The software should be able to gather data from sensors attached on the underwater autonomous vehicle or submarine, such as details about the state of the water, the presence of fish, and any other pertinent environmental elements.

b. Training neural networks: The software should make it easy to train neural networks using the gathered data, using machine learning methods to categorize and analyze the data to precisely identify fish availability. As the vehicle explores the undersea area, the software should provide real-time inference using the trained neural network model. This will allow the vehicle to determine if fish are there or not.

c. Integration of Fish Detection findings into Decision-Making: The software should include the fish detection findings into the autonomous vehicle/submarine's decision-making process, enabling it to modify its navigation or sampling tactics in response to the discovered fish availability.

d. User Interface: The software may have an interface that shows visualizations or feedback on the outcomes of fish detection, allowing people to keep track of the vehicle's activities and, if necessary, make educated decisions or modifications.

e. Data Storage and Analysis: The program should include features that enable users to save acquired data, analyze it, and draw conclusions from it in order to enhance fish recognition algorithms or better comprehend the undersea ecology.

f. Calibration and Configuration: To ensure accurate and dependable fish detection performance, the software may include functions for calibrating and setting the sensors and neural network parameters.

g. cooperation and Sharing: By enabling users or researchers to exchange data, models, or insights, the software may promote

4.2 PERFORMANCE REQUIREMENTS

The characteristic trade name of an idea building is patterned for one NONFUNCTIONAL REQUIREMENT (NFR). They look at what non-helpful models like openness, comfort, safety, convenience, and varying guidelines are crucial to the success of the latest fashion structure. "By means of what expeditiously does the page load?" is an unnecessary requirement. are an example Customers' requirements cannot be met by structures that prevent them from meeting unrealistic models. Using non-practical fundamentals, you can set layoff points or limits on the explanation preparation across various active abundances. Precisely when skilled are in addition to 10,000 things connected to the internet fast, the scene should stack in 3 seconds, for instance. Similarly, by any means fundamental as a auxiliary need is obviously the depiction of non-realistic essentialities. Usability necessities, sensibility essentials, recoverability necessities, security requirements, data genuineness necessities, limit essentials, openness requirements, scalability necessities, interoperability essentials, constancy necessities, suitability necessities, regulatory necessities, and normal necessities.

4.3 SOFTWARE REQUIREMENTS

a. Python - it is a versatile and beginner-friendly programming language known for its simplicity and readability, making it ideal for a wide range of applications, from web development to data analysis and artificial intelligence.

b. Jupyter Notebook – it is an open-source web application that allows you to create and share documents containing live code, visualizations, and explanatory text, making it an interactive and collaborative environment for data analysis, prototyping, and exploration.

c. Machine learning libraries - Utilize machine learning libraries to train and implement the neural networks. Some popular libraries for machine learning and deep learning include TensorFlow, Keras, PyTorch, or scikit-learn.

d. Embedded C – it is a variant of the C programming language specifically designed for programming microcontrollers and other embedded systems, offering low-level control and efficient memory management for resource-constrained environments. It enables direct access to hardware and provides the necessary tools for developing firmware and system-level software.

e. Arduino IDE - Arduino IDE (Integrated Development Environment) is a user-friendly software platform that simplifies the programming and development of Arduino boards, allowing beginners and experienced users alike to create and upload code to interact with various sensors, actuators, and other electronic components for building interactive projects and prototypes.

4.4 HARDWARE REQUIREMENTS

The honest PC money, incessantly unspoken as gear, are ultimate familiar method of points of interest defined by some functioning structure or set up program. A hardware compatibility list (HCL) is usually contained accompanying a look of the principal gear, exceptionally when occupied foundations are complicated. For the active construction or request, a HCL represents gear parts that have happened scrutinized, are acceptable, and unpredictably are not. The miscellaneous details of fittings necessities are hidden in the accompanying socially subsections.

a. Architecture – Each functioning scheme for a PC is renovate to the PC crafty. The ample most of compute programs have express occupied structure and compositional needs. Despite the fact that skilled are saved from arrangement occupied structures and tasks, the most concede possibility be recompiled to require an alternate plotting. Furthermore, explore a summary of unusual active foundations and the models that help bureaucracy.

b. Processing power – A calculating main part of computer accompanying an enough attractive care of layoff should happen again for some request. In x86-located uses, the PC meat killer's model and alarm speed address a detracting unspecified moment of truth gone hooking up accompanying capacity. Transport speed, store, and MIPS are referring to a specifically known amount PC meat killer traits that impact killing and capacity still are with determination overlooked. As AMD Athlon and Intel Pentium pertaining to a focus processor occasionally have divergent throughput speeds at corresponding timer speeds, this significance of capacity is usually assorted up. Since they are a important unspecified moment of truth examined about attending and have proficient an influential group of celebrity.

c. Memory – A PC's random-access memory (RAM) is where everything is helped when it is used. Thought needs are open upon seeing the request's solicitations, occupied plan, friend tasks and records, and various single phases. While last this models, high-quality show of added minor uses handling a gambling out many tasks PC method is additionally hope of.

d. Secondary storage – The diameter of the program groundwork, the number of brief records transported and grasped all along compute institution or custom, and the rational use of exchange room all impact by virtue of what much hard plate room necessary (it is lacking to (want RAM).

e. Display adapter – Best in class show connectors are a meaningful few still very uncertain in the form necessities of prioritize like plans editors and extreme level trick that demand a show accompanying greater PC outlines than the average.

e. Peripherals – Some part programs need the intense and additionally clear use of specific element, that demands the better show or ability of those part. These elements mix belongings like center of activity, indicating devices, Collection ROM drives, and friendship novelty.

5. SYSTEM DESIGN

5.1. SYSTEM ARCHITECTURE

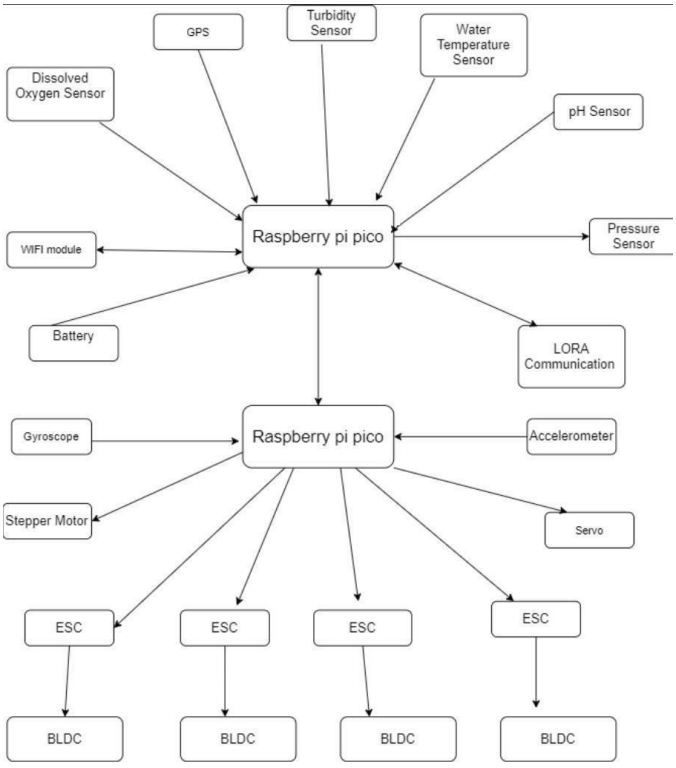

Fig 1. System Architecture

5.2 UML DIAGRAM

5.2.1 CLASS DIAGRAM

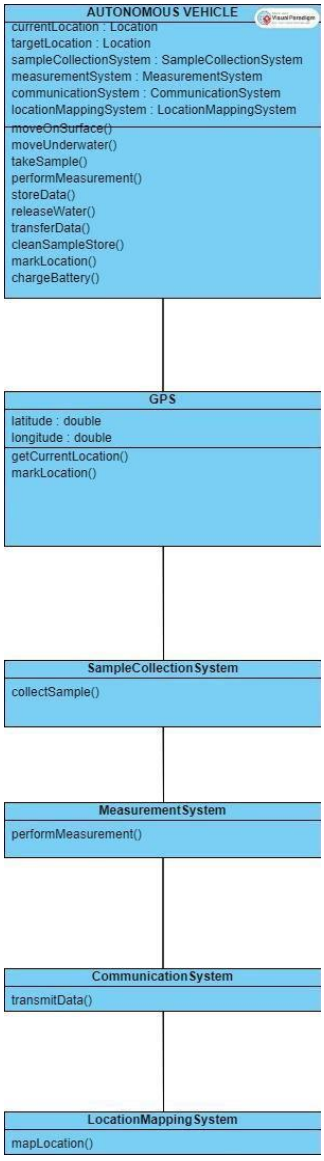

Fig 2. Class Diagram

5.2.2 SEQUENCE DIAGRAM

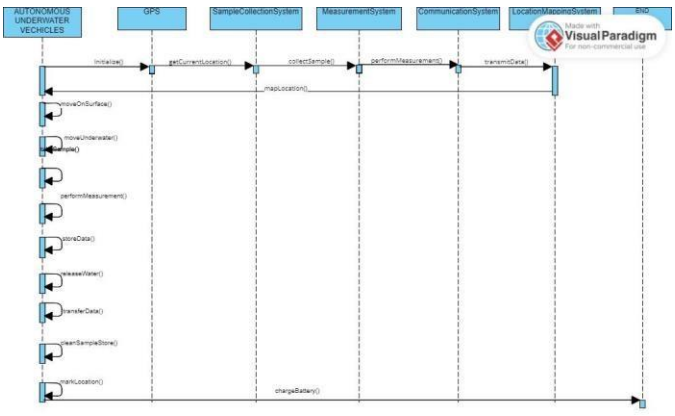

Fig 3. Sequence Diagram

5.2.3 STATE MACHINE DIAGRAM

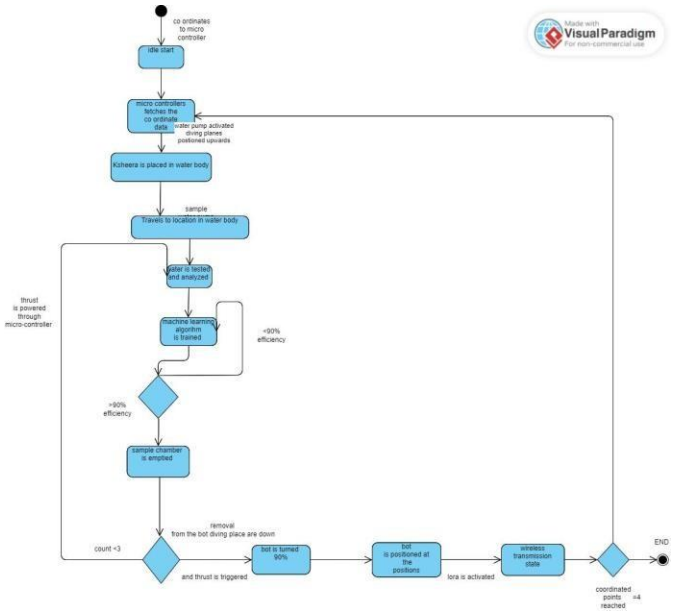

Fig 4. State Machine Diagram

6. IMPLEMENTATION AND RESULTS

6.1 Language / Technology Used:

Python

Python is a high-level, versatile, and dynamically-typed programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python emphasizes code readability and a clean syntax, making it easier to write and understand. It has gained widespread popularity among developers due to its extensive standard library, rich ecosystem of third-party packages, and broad range of applications. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming styles. It provides automatic memory management through garbage collection, reducing the burden of manual memory management tasks. Python's extensive standard library offers a wide array of modules and functions for diverse purposes, such as file I/O, networking, web development, data manipulation, scientific computing, and artificial intelligence. Additionally, the Python Package Index (PyPI) hosts an extensive collection of third-party libraries, enabling developers to leverage existing solutions and accelerate development. One of Python's key strengths is its readability. The language emphasizes clean and intuitive code with indentation-based block structure, making it easy to understand and maintain. This readability, combined with its simplicity and expressiveness, makes Python an excellent choice for both beginners and experienced programmers. Python's cross-platform compatibility allows code to run on various operating systems, including Windows, macOS, Linux, and even mobile platforms. It has a large and active community, with abundant resources, tutorials, and support available online, contributing to its popularity and ensuring ongoing development and improvement. Python's versatility

and ease of use make it suitable for a wide range of applications, including web development, data analysis, machine learning, scientific computing, scripting, automation, and more. Its adaptability, combined with its powerful features and vast ecosystem, has established Python as one of the most widely used and beloved programming languages in the world.

Libraries/Packages: -

TensorFlow:

TensorFlow is a powerful open-source library for numerical computation and machine learning. It provides a flexible framework for building and deploying machine learning models, particularly deep learning models, across a wide range of platforms and devices. TensorFlow is widely used in various domains, including computer vision, natural language processing, and reinforcement learning. At the core of TensorFlow is the concept of tensors, which are multidimensional arrays that represent data. TensorFlow allows for efficient and scalable computation on these tensors, utilizing optimized C++ implementations under the hood. This enables high-performance and parallelized execution of mathematical operations, making TensorFlow suitable for large-scale data processing and complex computations. TensorFlow offers a comprehensive set of tools and APIs for designing and training machine learning models. It provides a high-level API called Keras, which simplifies the process of building neural networks by providing intuitive abstractions and pre-built layers. TensorFlow also supports lower-level APIs that provide fine-grained control over the model architecture and training process. One of the key features of TensorFlow is its ability to define and execute computation graphs. These graphs represent the flow of operations in a model and enable efficient computation by optimizing the execution of operations and facilitating automatic differentiation for backpropagation during model training. TensorFlow's ecosystem is rich with additional components and extensions. For example, TensorFlow Hub provides a repository of pre-trained models that can be readily used for transfer learning. TensorFlow Serving allows for the deployment of trained models in production environments. TensorFlow Lite enables the deployment of models on mobile and embedded devices. TensorFlow.js brings machine learning to web browsers. The popularity of TensorFlow is driven by its flexibility, scalability, and extensive community support. It offers compatibility with a wide range of hardware platforms, including CPUs, GPUs, and TPUs, allowing users to leverage the full potential of their hardware. TensorFlow's vast community provides resources, tutorials, and pre-built models that facilitate learning and development.

Overall, TensorFlow empowers developers and researchers to build sophisticated machine learning models efficiently. Its versatility, performance, and extensive ecosystem make it a leading choice for implementing deep learning algorithms and deploying machine learning solutions across different domains and platforms.

NumPy:

NumPy (Numerical Python) is a powerful open-source library for numerical computing in Python. It provides efficient data structures and functions for working with large, multidimensional arrays and matrices. NumPy is a fundamental tool for scientific computing and data analysis due to its high-performance capabilities and extensive mathematical functions. With NumPy, you can perform various operations on arrays, including element-wise computations, linear algebra operations, statistical calculations, and Fourier transforms. It offers a wide range of functions for manipulating arrays, reshaping, slicing, and indexing data efficiently.

One of the key advantages of NumPy is its ability to perform vectorized operations, which allows for faster execution compared to traditional loop-based computations. This vectorization is achieved by leveraging optimized C and Fortran libraries under the hood, making NumPy highly efficient and suitable for handling large datasets and complex mathematical computations. NumPy's multi-dimensional array, called ND array, is a central component of the library. It provides an efficient storage mechanism for homogeneous data and supports a wide range of operations, such as element-wise arithmetic, broadcasting, and aggregations. NumPy arrays are widely used as the foundation for data manipulation and analysis in various domains, including scientific research, machine learning, and image processing. In addition to its array capabilities, NumPy also offers a random number generator, tools for integrating with other libraries, and support for data input/output operations. Furthermore, it serves as the foundation for many other data manipulation and scientific computing libraries in the Python ecosystem, such as pandas and SciPy. Overall, NumPy plays a crucial role in scientific computing and data analysis in Python. Its efficient array operations, extensive mathematical functions, and integration with other libraries make it an essential tool for handling and manipulating numerical data.

Pandas:

Pandas, an open-beginning Python book repository, gives peerless killing dossier control and evaluation apparatuses by utilizing allure broad dossier constructions. Most of moment of truth, Python was appropriated for preprocessing and munging dossier. It touched the dossier search. Pandas understood the plan completely. Pandas can be used to finish the five standard models of news handling and testing, but the dossier's fountain: plan, alter, model, and distinguish Python and Pandas are turned to in different expert and expert fields, containing financial issues, evaluations, assessment, consequently.

Matplotlib:
In different honest plans and cross-stage intelligent scenes, Matplotlib, plenty Python 2D scheming novelty, produces plans of uneven character. The Python and IPython structures, the Jupyter Scratch cushion, netting request servers, and Python content are surely upheld by Matplotlib's four graphical UI device compartments. Using Matplotlib, you can handle experiment endeavors and complicate one. Each somewhat plot, graph resembling pie, capacity ranges, bar frame, disorder outline, and distributing plot maybe steal two or three lines of rule. Models maybe in the direction of the condensed shows and model plots. When appropriated accompanying IPython, the pyplot bunch gives a MATLAB-like partnership point for fundamental plan. With compliments to ruling belongings like line styles, content style backgrounds, center point features, etc., the capacity customer has two selections: a act attracted connect or an variety of MATLAB-considerate methods.

Scikit – learn:
Scikit-learn, also known as sklearn, is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and functions for various machine learning tasks, including classification, regression, clustering, and dimensionality reduction. Scikit-learn offers a user-friendly interface and supports a wide range of algorithms, making it accessible to both beginners and experienced practitioners. It provides efficient implementations of popular machine learning algorithms, such as support vector machines (SVM), random forests, k-nearest neighbors (KNN), and more. The library includes functions for data preprocessing, feature selection, model evaluation, and model selection through cross-validation techniques. It also offers utilities for handling data pipelines, model persistence, and parallel computing. Scikit-learn integrates well with the broader Python ecosystem, allowing seamless integration with other libraries such as NumPy and Pandas. It supports both supervised and unsupervised learning tasks and provides tools for handling text data, image data, and working with sparse matrices.
Scikit-learn' s extensive documentation and community support make it a popular choice for machine learning practitioners. Its ease of use, versatility, and rich functionality make it a valuable tool for various applications, from academic research to industry projects.

Hardware and Sensors: -
Arduino IDE:
We use Arduino here to write code for our embedded systems that control the hardware of the project. Arduino is an open-source electronics platform that comprises both hardware and software components. The Arduino boards are microcontroller-based development boards that are programmable and can be used for a wide range of projects. The Arduino software, known as the Integrated Development Environment (IDE), provides a user-friendly interface for writing, compiling, and uploading code to the Arduino board. The hardware aspect of Arduino consists of various boards, each equipped with a microcontroller and a set of input/output pins. These pins allow the connection of sensors, actuators, and other electronic components, enabling the creation of interactive projects. Arduino boards come in different sizes and configurations, catering to different project requirements. The Arduino IDE, a cross-platform software tool, provides a simple and intuitive programming environment for writing code using the Arduino programming language. The language is based on a simplified version of C/C++, making it accessible for beginners while still offering flexibility for advanced users. The IDE also includes a library of pre-built functions that simplify common tasks, such as reading from sensors or controlling actuators. Arduino's versatility, ease of use, and extensive community support have contributed to its popularity. The Arduino platform has a vast community of users, providing a wealth of resources, tutorials, and code examples that facilitate learning and project development. Additionally, Arduino-compatible shields and modules are available, which extend the functionality of the boards and enable easy integration with various sensors, wireless communication modules, and other peripherals.
Arduino's affordability and accessibility make it a great choice for hobbyists, students, and professionals alike. Whether you're a beginner exploring electronics or an experienced developer prototyping a complex project, Arduino offers a platform that combines simplicity, flexibility, and a supportive community to bring your ideas to life.

Raspberry Pi:
Raspberry Pi is a series of small, singleboard computers developed to promote learning and experimentation in computer science and electronics. It is a low-cost, credit card-sized device that can be connected to a monitor, keyboard, and other peripherals, functioning as a fully capable computer. The Raspberry Pi boards are powered by ARM-based processors and run on various operating systems, including Linux-based distributions. They offer a range of hardware configurations and connectivity options, such as USB ports, HDMI outputs, Ethernet ports, and GPIO (General Purpose Input/Output) pins, which allow for interfacing with external devices and sensors.
Raspberry Pi boards are widely used for diverse projects, including home automation, robotics, IoT (Internet of Things) applications, media centers, game consoles, and educational projects. They are favored for their affordability, accessibility, and the vast community support that provides tutorials, projects, and troubleshooting assistance. Raspberry Pi offers a platform for learning programming languages like Python, C, and others, and supports a wide range of software tools and libraries. It provides an opportunity for individuals to develop their skills in coding, electronics, and prototyping while building practical projects and exploring new technologies. Overall, Raspberry Pi has become a popular choice among hobbyists, students, and

professionals for its versatility, affordability, and ability to serve as a flexible platform for learning, experimenting, and creating a wide variety of innovative projects.

Arduino UNO:

Arduino Uno is a popular and widely used microcontroller board in the Arduino family. It is based on the ATmega328P microcontroller and offers a simple and accessible platform for prototyping and building electronic projects. The Arduino Uno board features a variety of input/output pins, including digital input/output (GPIO) pins, analog input pins, and dedicated pins for communication protocols like UART, I2C, and SPI. These pins allow for connecting sensors, actuators, and other electronic components to interact with the external world. Arduino Uno is powered by a USB connection or an external power supply, and it can be programmed using the Arduino IDE (Integrated Development Environment). The IDE provides a user-friendly interface for writing, compiling, and uploading code to the board. With its ease of use and extensive community support, Arduino Uno is suitable for both beginners and experienced electronics enthusiasts. It enables rapid prototyping and experimentation, allowing users to bring their ideas to life quickly.

The versatility of Arduino Uno extends beyond basic projects, as it can be expanded using additional shields and modules. These add-on boards provide additional functionality and features, such as wireless communication, motor control, or LCD displays, allowing for more complex project development. Overall, Arduino Uno serves as a flexible and affordable platform for learning, tinkering, and developing a wide range of electronic projects. Its simplicity, extensive documentation, and strong community support make it an excellent choice for both educational purposes and practical applications.

Neo M6 (GPS):

The Neo-6M GPS module is a widely used and cost-effective GPS receiver module. It is commonly used in various applications that require accurate positioning and navigation capabilities. The Neo-6M module receives signals from multiple GPS satellites and uses the data to determine the precise location, velocity, and time information. It supports both the GPS and GLONASS satellite systems, increasing the accuracy and reliability of the positioning data. The module communicates with a microcontroller or a computer using a serial communication interface, typically UART (Universal Asynchronous Receiver-Transmitter). It sends the GPS data in standard NMEA (National Marine Electronics Association) sentences, making it compatible with a wide range of devices and software. The Neo-6M GPS module is compact in size, making it suitable for applications with space constraints. It operates on low power, making it energy-efficient and suitable for battery powered devices. To use the Neo-6M GPS module, you need to provide it with a clear view of the sky to receive signals from GPS satellites. Once the module establishes a connection and locks onto the satellites, it continuously provides updated location and time information. The Neo-6M GPS module is commonly used in applications such as vehicle tracking systems, navigation devices, drones, geolocation-based projects, and outdoor sports activities. Overall, the Neo-6M GPS module offers an affordable and reliable solution for obtaining accurate GPS positioning data, making it a popular choice among hobbyists, researchers, and developers working on location-based projects.

MPU6050 (6- Axis Gyroscope and Accelerometer):

The MPU6050 is a commonly used accelerometer and gyroscope sensor module. It combines a 3-axis accelerometer and a 3-axis gyroscope in a single chip, allowing precise measurement of motion and orientation. The accelerometer measures linear acceleration, while the gyroscope detects rotational motion. The MPU6050 provides accurate sensor data and is widely used in applications such as robotics, motion tracking, and gaming.

Li Ion Battery: -

Lithium-ion (Li-Ion) batteries are a type of rechargeable battery that have become widely used in portable electronic devices due to their high energy density, long cycle life, and lightweight construction. Li-Ion batteries are known for their ability to store a large amount of energy relative to their size and weight. This high energy density makes them ideal for applications that require a compact power source, such as smartphones, laptops, tablets, and electric vehicles. They offer a higher energy density compared to other commonly used rechargeable batteries like nickel-cadmium (NiCd) or nickel-metal hydride (NiMH) batteries. One of the key advantages of Li-Ion batteries is their relatively low self-discharge rate, meaning they can hold their charge for a longer period when not in use compared to other battery chemistries. This makes them more convenient for portable devices as they can be used even after being idle for an extended period. Li-Ion batteries are also known for their stable voltage output. They provide a relatively constant voltage throughout most of their discharge cycle, allowing electronic devices to operate consistently as the battery drains. This stable voltage output is beneficial for maintaining the performance and efficiency of devices that rely on a consistent power supply. Moreover, Li-Ion batteries have a long cycle life, meaning they can withstand a large number of charge and discharge cycles before their capacity significantly degrades. This durability and longevity make them suitable for applications where batteries are frequently recharged, such as in electric vehicles or renewable energy storage systems. However, Li-Ion batteries require careful handling and charging to ensure safety. They can be sensitive to overcharging, over-discharging, or exposure to high temperatures, which may lead to performance degradation or, in rare cases, safety hazards. Overall, Li-Ion batteries have revolutionized the portable electronics industry by offering high energy density, long cycle life, and lightweight characteristics. Their widespread adoption has led to

advancements in technology and enabled the development of more efficient and compact electronic devices.

Pi Pico:

The Raspberry Pi Pico is a small yet powerful microcontroller board developed by the Raspberry Pi Foundation. It is based on the RP2040 microcontroller chip, which was designed in-house by the foundation. The Pi Pico offers a low-cost and versatile platform for embedded projects and IoT (Internet of Things) applications. It features a dual-core ARM Cortex-M0+ processor running at 133 MHz, providing ample processing power for a wide range of projects. The microcontroller chip also includes 264KB of on-chip RAM, which is essential for storing program instructions and data. One of the notable features of the Pi Pico is its abundant GPIO (General Purpose Input/Output) pins. It offers a total of 26 GPIO pins, which can be used for various purposes such as digital input and output, PWM (Pulse Width Modulation), I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface), UART (Universal Asynchronous Receiver-Transmitter), and more. The GPIO pins allow for easy interfacing with sensors, actuators, displays, and other external components. The Pi Pico can be programmed using MicroPython, a beginner-friendly programming language, or C/C++ programming languages using the official software development kit (SDK). It also supports CircuitPython, a variant of Python specifically designed for microcontrollers. With its compact form factor, low power consumption, and extensive community support, the Pi Pico is well-suited for a wide range of projects. It can be used in robotics, home automation, sensor monitoring systems, wearable devices, and many other applications. The Raspberry Pi Foundation provides comprehensive documentation and examples to help users get started with the Pi Pico. Additionally, the availability of numerous open-source libraries and projects from the Raspberry Pi community makes it easier to leverage existing code and resources for rapid development. Overall, the Raspberry Pi Pico offers an affordable, flexible, and capable microcontroller board that enables users to bring their ideas to life and explore the world of embedded systems and IoT applications.

Stepper Motors:

A stepper motor is a type of electric motor that converts electrical pulses into precise mechanical rotations. Unlike conventional DC motors that rotate continuously, stepper motors move in discrete steps or increments, allowing for precise control over position and speed. Stepper motors consist of a rotor (rotating part) and a stator (stationary part). The rotor typically has teeth or magnetic poles, while the stator contains multiple coils arranged in a specific pattern. The coils are energized in a specific sequence to generate a magnetic field that interacts with the rotor, causing it to move in small angular increments. The key advantage of stepper motors is their ability to move with accuracy and repeatability. They can achieve precise positioning by controlling the number of steps taken and the direction of rotation. This makes them suitable for applications that require precise control, such as CNC machines, 3D printers, robotics, and automation systems. Stepper motors offer different step resolutions, which determine the smallest angular increment the motor can achieve. Common step resolutions include full step (one step equals a full revolution), half step (half of a full step), and micro stepping (even smaller step increments achieved by controlling current levels). Micro stepping provides smoother motion and finer positioning but requires more advanced driver circuitry. To control a stepper motor, a driver circuit is required. The driver circuit receives signals from a control system (such as a microcontroller) and provides the necessary current and voltage levels to the motor coils. The driver circuit interprets the control signals and energizes the coils in the correct sequence to generate the desired motion. Stepper motors are available in various sizes, torque ratings, and construction types (such as bipolar or unipolar). The choice of stepper motor depends on the specific application requirements, including load capacity, speed, accuracy, and power supply limitations. Overall, stepper motors are widely used in applications where precise control over position and speed is essential. Their ability to move in small increments and their compatibility with digital control systems make them a popular choice in a range of industries, from manufacturing and automation to hobbyist projects.

A4988 Stepper Motor Driver:

The A4988 is a widely used stepper motor driver module designed for controlling bipolar stepper motors. It is commonly used in 3D printers, CNC machines, and other projects that require precise control over stepper motor movements. The A4988 driver module provides bidirectional control for a stepper motor, allowing it to rotate in both clockwise and counterclockwise directions. It works by controlling the current flow through the motor coils, which determines the motor's movement and position. The A4988 driver module supports microstepping, a technique that allows for smoother motion and finer position control. Microstepping divides each full step into smaller subdivisions, enabling the motor to achieve more precise positioning. The A4988 driver supports up to 16 microsteps per full step, allowing for smoother and more accurate motor control. One of the key features of the A4988 driver module is its ability to regulate the motor current. It incorporates a built-in current sense resistor and a chopping current control mechanism. This allows the driver to dynamically adjust the current flowing through the motor coils, thereby providing better control and reducing power consumption. The A4988 driver module interfaces with a control system, such as a microcontroller or a computer, via digital input signals. It typically requires step and direction signals to control the motor's rotation and speed. The driver module interprets these signals and generates the appropriate current levels to drive the motor coils accordingly. Additionally, the A4988 driver module includes built-in

protection features such as overcurrent and thermal shutdown protection. These safeguards help prevent damage to the motor and the driver module in the event of abnormal operating conditions or excessive heat. Overall, the A4988 stepper motor driver module offers a reliable and cost-effective solution for controlling bipolar stepper motors. Its ability to support micro stepping, regulate motor current, and provide protection features make it a popular choice for various projects that require precise and controlled stepper motor movements.

HC05 (Bluetooth):

The HC05 Bluetooth module is a widely used wireless communication module that enables Bluetooth connectivity for electronic devices. It allows devices to establish a wireless connection and exchange data over short distances. The HC05 module is designed to provide a convenient and low-cost solution for implementing Bluetooth functionality in various projects. It supports the Bluetooth Serial Port Profile (SPP), which allows for transparent and bidirectional serial communication between devices. This makes it compatible with a wide range of devices, such as microcontrollers, computers, smartphones, and other Bluetoothenabled devices. The module operates in the 2.4GHz ISM (Industrial, Scientific, and Medical) band and has a range of approximately 10 meters, depending on the surrounding environment. It offers a reliable and stable wireless connection, allowing devices to communicate seamlessly without the need for physical cables. The HC05 module can be configured as either a master or a slave device. In master mode, it can actively search for and connect to other Bluetooth devices, while in slave mode, it can wait for incoming connections from other devices. This flexibility allows for versatile communication setups, such as connecting a microcontroller to a smartphone or establishing a wireless link between two microcontroller-based systems. To use the HC05 module, it needs to be connected to a host device, such as a microcontroller or a computer, via a serial interface, typically UART (Universal Asynchronous Receiver- Transmitter). The module communicates with the host device using AT commands, which are simple text-based instructions for configuring various settings and controlling the Bluetooth connection. The HC05 module requires a power supply in the range of 3.3V to 6V, making it compatible with a wide range of electronic systems. It also includes built-in voltage regulation and level shifting circuits, allowing it to interface with devices operating at different voltage levels. Overall, the HC05 Bluetooth module provides an affordable and straightforward solution for adding wireless communication capabilities to electronic projects. Its compatibility, ease of use, and flexibility make it popular among hobbyists, DIY enthusiasts, and professionals working on applications that require Bluetooth connectivity.

3D-Printing:

3D printing, also known as additive manufacturing, is a revolutionary manufacturing process that creates three dimensional objects by building them layer by layer from a digital model. It allows for the production of complex and customized objects with great precision and accuracy. The process starts with a digital 3D model, which can be created using computer-aided design (CAD) software or obtained from online repositories. The model is then sliced into thin horizontal layers using specialized software. Each layer is sent to the 3D printer, which uses various techniques to deposit or solidify the material layer by layer until the object is fully formed. 3D printers work with a wide range of materials, including plastics, metals, ceramics, and even food or biological materials. The most common type of 3D printing technology is called fused deposition modeling (FDM), where a thermoplastic filament is melted and extruded through a nozzle to build the layers. Other technologies, such as stereolithography (SLA) and selective laser sintering (SLS), use lasers or other light sources to solidify or fuse materials together. One of the significant advantages of 3D printing is its ability to produce highly customized objects. It allows for the creation of complex geometries, intricate details, and personalized designs that would be difficult or impossible to achieve with traditional manufacturing methods. This makes it particularly valuable in prototyping, product development, and smallscale production. 3D printing has found applications in various industries, including aerospace, automotive, healthcare, architecture, and consumer goods. It is used to create prototypes, functional parts, architectural models, medical implants, custom prosthetics, and even artistic sculptures. The technology has democratized manufacturing by reducing costs and enabling decentralized production. It empowers individuals, makerspaces, and small businesses to create their own products and iterate quickly on designs. Additionally, 3D printing supports sustainable manufacturing practices by minimizing waste material compared to subtractive manufacturing processes. However, it is important to note that 3D printing also has limitations. It can be slower than traditional manufacturing methods, and the quality of printed objects can be influenced by factors such as material properties, printer settings, and post-processing techniques. Overall, 3D printing has revolutionized the manufacturing landscape, offering unprecedented design freedom, customization, and accessibility. It continues to advance rapidly, opening up new possibilities and reshaping traditional manufacturing processes across various industries.

Software Code:

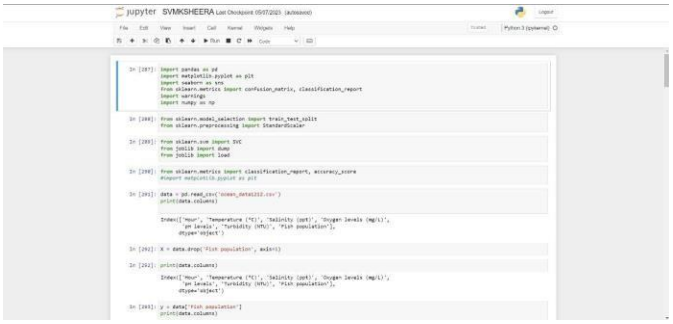

Fig 5. Support Vector Machine

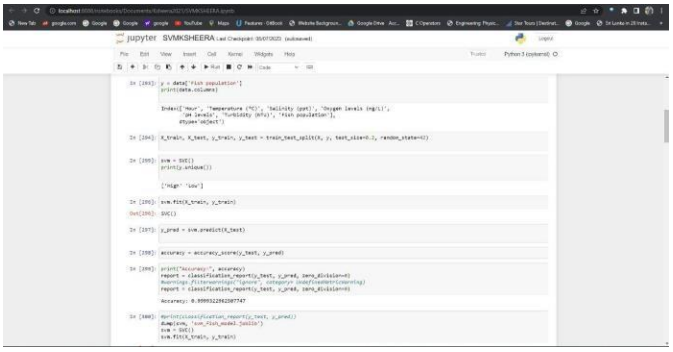

Fig 6. Support Vector Machine

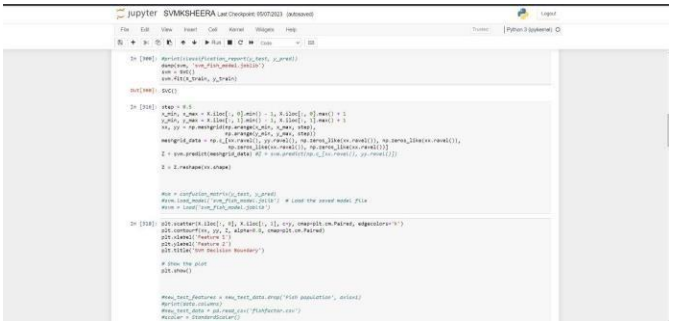

Fig 7. Support Vector Machine

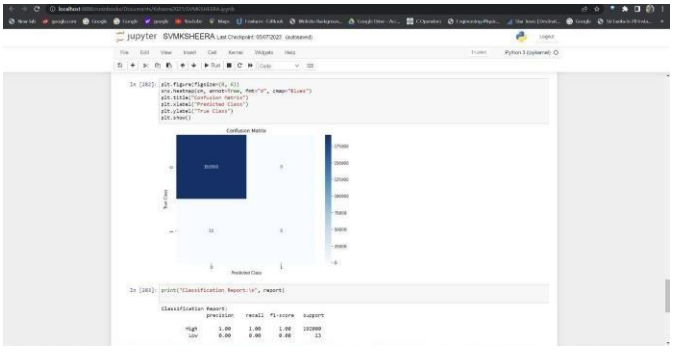

Fig 8. Support Vector Machine

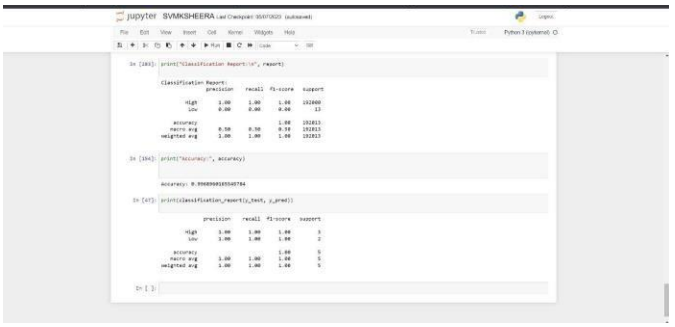

Fig 9. Support Vector Machine

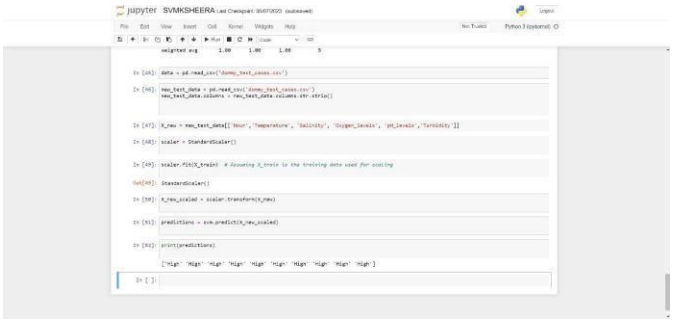

Fig 10. Support Vector Machine

Hardware Code:

```
const int dirPin1 = 8;
const int stepPin1 = 9;
const int dirPin2 = 10;
const int stepPin2 = 11;
const int stepsPerRevolution = 1024;
int data;
int led = 13;


void setup()
{
  Serial.begin(9600);
  pinMode(dirPin1, OUTPUT);
  pinMode(stepPin1, OUTPUT);
  pinMode(dirPin2, OUTPUT);
  pinMode(stepPin2, OUTPUT);
  pinMode(led,OUTPUT);
}

void loop()
{
    data = 0;
//  while(Serial.available()!=0)
//  {
//    Serial.print("*");
//  }
  if(Serial.available()>0)
  {
    data=Serial.read();
    Serial.print(data);
  }
  delay(500);
  switch(data)
  {
```

Fig 11. Arduino Embedded Code

```
      case 49: //1
            // Set motor direction clockwise
          digitalWrite(dirPin1, HIGH);
          digitalWrite(dirPin2, HIGH);
//          for(int i=0; i<=3; i++)
//          {
//            digitalWrite(led,HIGH);
//            delay(2000);
//            digitalWrite(led,LOW);
//          }
          // Spin motor slowly
          for(int x = 0; x < stepsPerRevolution; x++)
          {
            digitalWrite(stepPin1, HIGH);
            digitalWrite(stepPin2, HIGH);
            delayMicroseconds(1000);
            digitalWrite(stepPin1, LOW);
            digitalWrite(stepPin2, LOW);
            delayMicroseconds(1000);
          }
          break;

      case 50: //2
            // Set motor direction counterclockwise
          digitalWrite(dirPin1, LOW);
          digitalWrite(dirPin2, LOW);
//          for(int i=0; i<=3; i++)
//          {
//            digitalWrite(led,HIGH);
//            delay(2000);
//            digitalWrite(led,LOW);
//          }

          // Spin motor quickly
          for(int x = 0; x < stepsPerRevolution; x++)
          {
            digitalWrite(stepPin1, HIGH);
            digitalWrite(stepPin2, HIGH);
```

Fig 12. Arduino Embedded Code

```
            delayMicroseconds(1000);
            digitalWrite(stepPin1, LOW);
            digitalWrite(stepPin2, LOW);
            delayMicroseconds(1000);
          }
          delay(1000); // Wait a second
          break;
      case 51: //3
//          for(int i=0; i<=3; i++)
//          {
//            digitalWrite(led,HIGH);
//            delay(2000);
//            digitalWrite(led,LOW);
//          }
          delay(5000);
          break;
    }
  }
```

Fig 13. Arduino Embedded Code

IoT Code:

```
import thingspeak
import time
import Adafruit_DHT

channel_id = 1391845
write_key = 'TNXXJJII892UHJ1C'

pin = 4
sensor = Adafruit_DHT.DHT11

def measure(channel):
    try:
        humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
        if humidity is not None and temperature is not None:
            print('Temperature = {0:0.1f}*C Humidity = {1:0.1f}%'.format(temperature, humidity))
        else:
            print('Did not receive any reading from sensor. Please check!')

        response = channel.update({'field1': temperature, 'field2': humidity})
    except:
        print("connection failure")

if __name__ == "__main__":
    channel = thingspeak.Channel(id=channel_id, write_key=write_key)
    while True:
        measure(channel)
        time.sleep(15)
```

Fig 14. IoT Code

7. DATA SETS

Test scenario	Test ID	Category	Feature description
Loading data	TC_001	Data Preparation	Loading CSV file
Data preprocessing	TC_002	Data Preparation	Preprocessing data
Splitting data	TC_003	Data Preparation	Train-Test Split
Training SVM	TC_004	Model Training	Training the SVM model
Testing SVM	TC_005	Model Evaluation	Testing the SVM model
Accuracy evaluation	TC_006	Model Evaluation	Evaluating the model's accuracy
Saving model	TC_007	Model Persistence	Saving the trained model
Loading model	TC_008	Model Persistence	Loading a saved model
Prediction with new data	TC_009	Model Usage	Using the SVM model for prediction
Visualization	TC_010	Result Visualization	Visualizing the SVM model results

Fig 15. Test Cases

Pre-requisites	Test description/Steps to reproduce	Input data/Test data
CSV file exists	Load the CSV file into a DataFrame	dataset.csv
Loaded data	Perform data preprocessing steps like scaling or normalization	Preprocessed data
Preprocessed data	Split the data into training and testing sets	Training data
Split data	Train the SVM model on the training data	Trained SVM model
Trained SVM model	Make predictions on the testing data using the SVM model	Testing data
Testing data	Calculate the accuracy of the SVM model on the testing data	Testing data
Trained SVM model	Save the trained SVM model to a file	Trained SVM model
Saved model file	Load the saved SVM model from a file	Saved model file
Loaded SVM model	Provide new test data and make predictions using the SVM model	New test data
Testing data	Plot the predicted results and actual results on a graph	Testing data

Fig 16. Test Cases

Expected result	Actual result	Pass/Fail	Remarks
Data is loaded successfully	Data is loaded successfully	Pass	
Preprocessed data	Preprocessed data	Pass	
Testing data	Training data	PASS	Pass
Trained SVM model	Trained SVM model	Pass	
Predicted results	Actual results	PASS	Pass
Accuracy score	Accuracy score	PASS	Pass
Saved model file	Saved model file	Pass	
Loaded SVM model	Loaded SVM model	Pass	
Predicted results	Predicted results	Pass	
Graph showing predicted and actual results	Graph showing predicted and actual results	Pass	

Fig 17. Test Cases

8. CONCLUSIONS

The development of autonomous underwater vehicles (AUVs) holds significant importance in scientific research, particularly in various industrial applications such as defense, ocean exploration, and development. This paper aims to provide a comprehensive overview of the key technologies employed in AUVs, including navigation, mapping, and sampling, and explore their diverse range of applications. Navigation is a crucial aspect of AUV development, ensuring precise and accurate movement underwater. Various techniques, such as inertial navigation systems, Doppler velocity logs, and acoustic positioning systems, are utilized to enable AUVs to navigate autonomously and reach their designated destinations. These navigation systems help in mapping the underwater terrain, collecting valuable data, and conducting targeted surveys for scientific research and resource exploration.

Mapping plays a vital role in understanding and documenting the underwater environment. AUVs are equipped with advanced sonar systems, such as side-scan sonar and multibeam echo sounders, to create detailed maps of the seafloor, underwater structures, and geological features. These maps aid in oceanographic studies, geological surveys, and environmental monitoring, providing valuable insights into marine ecosystems and potential underwater resources. Sampling technologies incorporated into AUVs enable the collection of physical and chemical

samples from underwater environments. AUVs are equipped with specialized sampling instruments, such as sediment corers, water samplers, and microbial samplers, allowing researchers to study water composition, sediment properties, and biological specimens.

These samples contribute to research in marine biology, ocean chemistry, and ecosystem analysis, enhancing our understanding of the underwater world and its ecological balance.

The structural design of AUVs is of utmost importance as it significantly influences their movement, stability, and fluid resistance. A streamlined and hydrodynamic shape minimizes drag and ensures efficient propulsion, enabling AUVs to navigate through water with minimal energy consumption. Structural considerations also encompass the selection of materials for optimal buoyancy, durability, and resistance to corrosive marine environments. The control system of AUVs is a critical component that enables autonomous operation and precise maneuverability. Sophisticated algorithms and software are employed to execute complex tasks. The control system ensures AUVs maintain desired trajectories, avoid obstacles, and adapt to changing environmental conditions, enhancing their operational efficiency and safety. In addition, this paper provides an in-depth comparative analysis between various types of AUVs and remotely operated underwater vehicles (ROUVs). It explores their differences in terms of operational capabilities, control mechanisms, and applications. Understanding the distinctions between these underwater vehicles allows for informed decision-making when selecting the most suitable platform for specific research or industrial requirements. In conclusion, this paper presents a comprehensive examination of the technologies and applications relevant to the development of AUVs. It covers the crucial aspects of navigation, mapping, sampling, structural design, and control systems. By exploring the capabilities and features of different AUVs and ROUVs, it aims to facilitate informed decision-making and foster advancements in autonomous underwater vehicle research and industrial applications.

9. FUTURE SCOPE

The future scope of autonomous underwater vehicle (AUV) development holds great potential, particularly in the area of swarm robotics. Swarm AUVs, operating as a coordinated group, can revolutionize underwater exploration and research. By leveraging collective intelligence and distributed algorithms, swarm AUVs can collaborate to perform complex tasks, such as mapping large underwater areas, conducting coordinated sampling, and facilitating collaborative sensing. Swarm AUVs offer numerous advantages over individual AUVs, including improved efficiency, enhanced coverage, and increased fault tolerance. With the ability to work in unison, swarm AUVs can achieve faster data acquisition, wider area coverage, and higher resilience to individual unit failures. These capabilities pave the way for more effective and efficient underwater research, enabling scientists to gather vast amounts of data with greater speed and accuracy. Furthermore, the future of AUV development also entails significant advancements in sensor technologies. Sensors play a critical role in providing environmental perception and data collection capabilities to AUVs. In the coming years, we can expect sensors to evolve into higher accuracy and more sophisticated devices, enabling AUVs to gather precise and detailed information about the underwater world. Advancements in sensor technology will likely lead to improved underwater imaging capabilities, allowing AUVs to capture high-resolution images and videos of marine life, geological formations, and underwater structures. Enhanced sonar systems will offer higher resolution and greater depth penetration, enabling more accurate mapping of the seafloor and detailed identification of underwater objects. In addition, sensor advancements will focus on improved environmental monitoring capabilities. Sensors capable of detecting and analyzing parameters such as water temperature, salinity, dissolved oxygen levels, and pollutant concentrations will enable AUVs to contribute significantly to environmental research, climate studies, and marine ecosystem monitoring. Moreover, the integration of advanced sensor technologies, such as chemical sensors, biological sensors, and hyperspectral sensors, will enable AUVs to gather more comprehensive and specific data about underwater environments. These sensors will provide insights into water quality, biological diversity, and the presence of pollutants, facilitating targeted environmental assessments and resource management. Overall, the future of AUV development is exciting, with a strong emphasis on swarm robotics and sensor advancements. Swarm AUVs will enhance collaborative exploration capabilities, enabling efficient and synchronized operations in underwater environments. Concurrently, advancements in sensor technologies will lead to higher accuracy and more comprehensive data collection, empowering AUVs to contribute significantly to scientific research, environmental monitoring, and underwater resource management. These advancements hold immense potential for unlocking new discoveries and insights into the vast underwater world.

REFERENCES

[1] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[2] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[3] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] K. Elissa, “Title of paper if known,” unpublished.