IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 December 2023, accepted 9 January 2024, date of publication 16 January 2024,
date of current version 25 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355018

RESEARCH ARTICLE

AViTRoN: Advanced Vision Track Routing and Navigation for Autonomous Charging of Electric Vehicles

V. C. MAHAADEVAN[1], R. NARAYANAMOORTHI[1], (Member, IEEE), SAYANTAN PANDA[1], SANKHADDEP DUTTA[1], AND GERARD DOOLY[2], (Member, IEEE)

[1]Electric Vehicle Charging Research Centre, School of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Chennai 603203, India
[2]Centre for Robotics and Intelligent System (CRIS), University of Limerick, Limerick, V94 T9PX Ireland

Corresponding author: R. Narayanamoorthi (narayanamoorthi.r@gmail.com)

This work was supported by the Government of India, Department of Science and Technology (DST) Science and Engineering Research Board (SERB) Core Research Grant CRG/2020/004073.

ABSTRACT The ascent of electric vehicle (EV) technology as a leading solution for green transportation is accompanied by advancements in charging infrastructure and automation. A notable hindrance is the low level of automation in charging procedures. In response to this, Automatic Charging Robots (ACR) have emerged, equipped transitioning from the manual operation to an automated plugging and unplugging operation. However, for this process to be executed flawlessly, these robots necessitate a charging port detection system with a precise navigation system to ensure accurate insertion of the charging gun into the designated charging port. This paper presents a sophisticated system, AViTRoN (Advanced Vision Track Routing and Navigation), which is developed for Automated Charging Robots in the context of Electric Vehicle (EV) charging. AViTRoN integrates advanced technologies to enable efficient charging port detection, navigation, and seamless user interaction. Utilizing the YOLOv8 deep learning model, AViTRoN ensures real-time charging port type detection using the data from a 3D depth sensor and an IR sensor within the Robot Operating System (ROS) framework. The 3D depth sensor provides detailed spatial information, while the IR sensor detects subtle environmental changes, enhancing the system's accuracy during operation. AViTRoN also incorporates a charging completion notification mechanism, sending instant alerts to users via GSM/GPRS communication upon the conclusion of the charging cycle, thereby enhancing user convenience and experience.

INDEX TERMS YOLOv8, ROS, electric vehicles, GSM/GPRS communication, autonomous charging, socket detection.

I. INTRODUCTION

The current electric vehicle (EV) technology, aimed at reducing emissions, has evolved significantly with various charging socket designs [1]. However, manual operation for charging presents challenges, such as parking alignment, heavy lifting, and plug selection. Figure 1 shows various charging port positions of different car models. Automation is crucial due to safety risks and the growing need for intelligent charging services. Research focuses on image recognition-based identification and automated charging control systems [2]. Several automated charging methods have been explored, such as Tesla's charging robot and Volkswagen's E-Smart Connect system. Existing systems often work only with specific plug types and fixed locations, restricting their usability in various settings, requiring end-user involvement. To address this, an advanced navigation system is proposed, which will aid in identification and navigation of ACR for plugging the right charging gun in the right socket. The device is equipped with 3d vision camera and IR sensors for tracking and alignment of the charging gun according to the socket available on the EV. The enhanced camera system captures intricate details of the charging port's structure in real-time, forming a comprehensive visual dataset [3]. Embedded algorithms precisely identify the required port type, comparing with various standards such as IEC62196-3

The associate editor coordinating the review of this manuscript and approving it for publication was Turgay Celik.

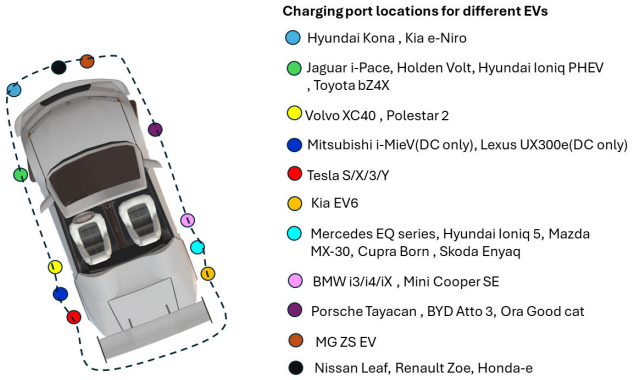


FIGURE 1. Universal charging port placement.

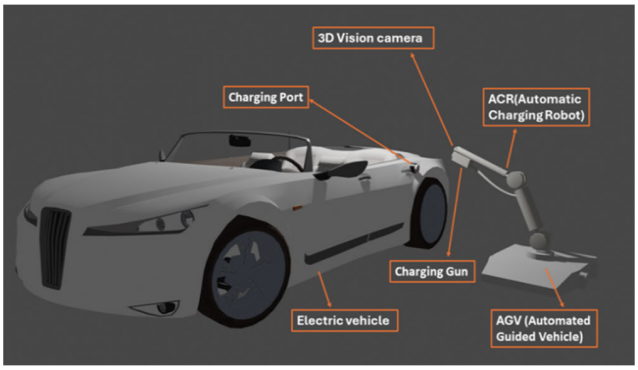


FIGURE 2. Proposed EVs charging through ACR and AGV integration: key components and architecture.

and SAEJ1772/IEC62196-2 [4].Charging electric vehicles requires a localized approach for precise charging port identification and positioning.

This process relies on the calibrated camera sensor and involves YOLOv8 recognition [5], Region of Interest (ROI) extraction [6], feature point detection using Hough circles, and feature matching with the KM (Kuhn-Munkres) algorithm [7]. In YOLOv8 recognition, a pre-trained YOLOv8 model processes the input image, predicting bounding box coordinates, class probabilities, and objectness scores. Non-maximum suppression (NMS) filters [8] confident bounding boxes, ensuring predictions match the charging port class. Feature point extraction identifies essential charging port attributes using Hough circle detection. This yields an array of feature points for precise localization. Feature matching, facilitated by the KM algorithm, determines the degree of correspondence between images, emphasizing proximity. This sophisticated process ensures that electric vehicles can be efficiently and accurately charged, contributing to the convenience and reliability of EV charging solutions. The enhanced plug-in process focuses on achieving precise alignment between the connector plug and the charging port, ultimately enabling a final motion along a single axis. This strategic alignment involves a four-step approach: a rapid initial approach, followed by velocity reduction and IR alignment to ensure exact positioning [9]. Subsequently, a gradual vertical motion is executed to enhance alignment precision. The final step involves a calculated rotation of the charging gun, accurately aligning it with the target plug. This aligned configuration enables the initiation of the charging mechanism, guaranteeing both physical and functional connection between the plug and the charging system [10]. Figure 2 shows the automated charging robot (ACR) seamlessly executes the plug-in maneuver with remarkable adaptability, eliminating the need for precise parking alignment. The system's effectiveness is rooted in the widely recognized ''peg-in-hole assembly'' paradigm, ensuring precise alignment with the electric vehicle's charging port.

A. MOTIVATION

The motivation behind our research lies in the practical integration of established deep learning methods, specifically Convolutional Neural Networks (CNNs) and YOLOv8, within the realm of computer vision and robotics. Historically, computer vision relied on manual feature engineering techniques like HOG and SIFT. However, the advent of CNNs and YOLOv8 signifies a shift towards more data-driven, adaptive solutions. By adopting these proven technologies, our aim is to enhance the efficiency of computer vision tasks, especially in object detection, without the need for complex, handcrafted algorithms. Incorporating CNNs and YOLOv8 into the ROS framework further empowers our robotic systems. ROS provides a reliable platform for integrating various robotic components, enabling seamless communication and collaboration. By integrating these deep learning techniques with ROS, we facilitate the fusion of advanced vision capabilities and real-time decision-making within robotic applications.

Additionally, the utilization of GSM/GPRS communication adds a practical dimension to our system [11]. Enabling charging notifications through this communication channel ensures timely user updates without relying on sophisticated, groundbreaking technologies. This straightforward approach enhances user experience and contributes to the overall efficiency of the charging process. Our motivation is rooted in the pragmatic application of established methodologies, aiming to create functional, user-friendly solutions in the realm of computer vision and robotics. By leveraging these existing technologies, our objective is to build a reliable and efficient charging system that seamlessly integrates into real-world scenarios.

B. CONTRIBUTIONS

The main contribution of this paper includes:

- AViTRoN Device: Engineered AViTRoN, a specialized device equipped with advanced technologies, including a 3D depth camera sensor, LiDAR, and Jetson Nano. This device forms the core of the automated charging system.

- ROS Navigation: Integrated ROS navigation capabilities, enabling the AViTRoN device to autonomously navigate to the vehicle's charging port location with precision and safety.
- Charging Port Detection: Implemented YOLOV8, a deep learning algorithm, to accurately detect various types of charging ports on different car models with. This ensures compatibility and proper connection.
- User Notification System: Implemented a notification mechanism that informs the user when the charging process is complete, enhancing user experience and allowing timely retrieval of the charged vehicle.

II. RELATED WORKS

In In the realm of electric vehicle (EV) charging systems, researchers and engineers have undertaken pioneering initiatives, exploring a multitude of avenues to enhance the efficiency, reliability, and user-friendliness of charging infrastructure. One significant avenue of exploration has been the integration of advanced object detection algorithms, with a focus on the evolution from YOLOv4 [12] to the swifter and more precise YOLOv5. By incorporating these algorithms into charging stations, researchers have achieved remarkable milestones in real-time, automated socket recognition. The fusion of YOLOv4's meticulous object detection capabilities and YOLOv5's exceptional speed has ushered in an era of efficient and seamless autonomous charging processes, marking a significant step towards widespread EV adoption [13].

New developments in research have introduced a cost-efficient, highly accurate technique for identifying and locating charging ports. This method leverages the Scale-Invariant Feature Transform (SIFT) and Semi-Global Block Matching (SGBM). The SIFT approach entails extracting features and creating a scale space through the generation of the Difference of Gaussian (DOG) [14]. Furthermore, a feature matching algorithm, employing nearest-neighbor search, is employed as a machine learning technique to generate a set of matching points forming a map. Concurrently, researchers have delved into the realm of reinforcement learning algorithms [15], reshaping the landscape of charging station optimization. Through the power of machine learning, these algorithms dynamically allocate and relocate charging stations based on intricate variables such as traffic patterns [16], user demand fluctuations, and real-time energy availability [17]. Recent advancements in visual tracking have witnessed the amalgamation of Siamese networks and region proposal networks. The SiamEXTR innovatively combines Siamese networks and region proposal networks for keypoint-guided tracking, bypassing region classification or bounding box regression [18]. Researchers have also delved into enhancing visual tracking methods which include Circular and Structural Operator Tracker (CSOT) integrated with SOSVM discriminative capability with DCF efficiency, using circular and structural operators for top-notch visual tracking for port detection [19]. In visual tracking research, a Siamese lightweight hourglass network is introduced for capturing contextual information at multiple scales. This Network is called SATIN which tackles the challenge of efficient and robust methods, its cross-attentional module boosts discriminative and localization capabilities through channel-wise and spatial intermediate attentional information [20]. Predictive maintenance, facilitated by sophisticated artificial intelligence models, has become a cornerstone in charging infrastructure. By analyzing historical usage data and employing predictive analytics, these models identify subtle patterns and potential issues within charging stations. Proactive maintenance schedules [21], triggered by these predictive insights, have significantly reduced downtime, ensuring uninterrupted charging services and enhancing the overall reliability of the charging network. Moreover, researchers have ventured into sustainable solutions, such as solar-powered charging stations [22] equipped with energy storage systems. These stations operate independently from the grid, harnessing solar energy and minimizing environmental impact. Through intricate energy management algorithms, these stations not only charge vehicles but also store surplus energy [23], transforming charging hubs into microgrids [24], contributing to a cleaner and more sustainable energy future. The evolution of wireless charging technologies, employing intricate techniques like magnetic resonance and inductive coupling [25], has redefined convenience and accessibility in EV charging. Furthermore, the revolutionary concept of vehicle-to-grid (V2G) integration [26], [27], [28] has emerged as a beacon of sustainability. Enabling bidirectional energy flow between the grid and EV batteries, V2G systems empower electric vehicles to supply excess energy [29], [30] back to the grid during peak demand periods This transformative technology not only stabilizes the grid but also promotes the integration of renewable energy sources, shaping a resilient and eco-conscious energy ecosystem. By placing comprehensive information at users' fingertips, these applications empower EV owners to make informed decisions, plan their routes effectively, and seamlessly navigate the charging landscape [31].

Table 1 shows the collective efforts of researchers in integrating advanced object detection techniques, reinforcement learning algorithms, predictive maintenance models, sustainable energy solutions, wireless charging technologies, V2G integration [32], and user-friendly mobile applications exemplify a multifaceted approach toward shaping the future of EV charging. These monumental advancements stand as a testament to human ingenuity, paving the way for a greener, more connected, and technologically advanced transportation ecosystem. As these innovations continue to evolve, they hold the promise of a sustainable future, where electric mobility seamlessly intertwines with our everyday lives.

III. METHODOLOGY

A. YOLOv8 STRUCTURE

Figure 3 shows the architecture of YOLOv8, a significant evolution, building upon the foundation of its predecessors

TABLE 1. List of similar works.

References	Analysis type	Models	Techniques	Reference
Mahaadevan *et al.* 2023	Images	YOLO v5	Swin Transformer, SimAM Attention Mechanism, YOLOv5 Structure	[37]
Guney *et al.* 2022	Images	Based on HSV Color Space Model, YOLO v4	HSV Color Space-Based Method , YOLO v4 Architecture	[38]
Quan *et al.* 2023	Images	YOLO v7	Deep Learning with YOLOV7-tinp, CBAM, Feature recognition, CTMA	[39]
Xu *et al.* 2015	Sensor Data	ROS	SLAM, Navigation Stack, TF (Transform) Configuration	[40]
Miseikis *et al.* 2017	Images	CATIA v5	Region-based CNN	[41]
Shibl *et al.* 2021	Images	Based on LSTM	K-Nearest Neighbors, Deep learning	[42]
ElKashlan *et al.* 2023	Network Traffic Data	Based on CNN	Convolution Neural Network , Deep Learning	[43]
Park *et al.* 2022	Video Frames	DQN,VCE	Deep Reinforcement Learning, RNN	[44]
Shi *et al.* 2018	Images	ROS	SLAM based on Particle Filter	[45]
Lin *et al.* 2022	Images	Based on CNN, LSTM	Deep CNN, KNA	[46]
Jiang *et al.* 2019	Video Frames	YOLO v3	multi-agent deep reinforcement learning	[47]
Barbosa *et al.* 2021	Images	IBVS, ROS	SLAM, IBVS, STA	[48]
Li *et al.* 2022	Images	DUAL-200M-030T160	Fast Library for Approximate Nearest Neighbors	[14]
Bochkovskiy *et al.* 2020	Images, Video Frames	YOLO v4	BoF and BoS, CmBN	[49]

in the YOLO (You Only Look Once) series [33]. This object detection algorithm relies on a convolutional neural network (CNN) [34]with two primary components: the backbone and the head, which collectively contribute to its exceptional performance. The YOLOv8 relies on a customized iteration of the CSPDarknet53 framework [35]. This architecture features an intricate arrangement of 53 convolutional layers, strategically designed to facilitate information flow between various stages of the network. The incorporation of cross-stage partial connections within CSPDarknet53 is pivotal in enhancing the model's capacity to capture and process complex image data. These components serve a pivotal function in the concluding phases of object detection, tasked with forecasting bounding boxes, objectness scores, and class probabilities for the objects discerned within an image [36].

This component essentially serves as the decision-making module, interpreting the features extracted by the backbone and transforming them into meaningful object detection results. This self-attention mechanism enables the model to dynamically focus on distinct regions of the image, adjusting the importance of different features based on their relevance to the detection task. By doing so, YOLOv8 becomes more adaptive and responsive to the intricacies of real-world images, which often contain objects of varying sizes, orientations, and complexities.

Multi-scaled object detection is another pivotal capability of YOLOv8 [50]. The model employs a feature pyramid network to achieve this feat, allowing it to detect objects of different scales and sizes within a single image. Consequently, YOLOv8 can accurately identify both large and small objects within an image, making it highly versatile and applicable to a wide range of scenarios.

In operation, YOLOv8's architecture performs a sequence of transformations, beginning with the backbone's convolutional layers [51], which extract hierarchical features from the input image. These features are then passed to the head, where the self-attention mechanism and additional convolutional layers refine the information and generate predictions regarding the location, class, and confidence of detected objects. The multi-scaled object detection, facilitated by the feature pyramid network, ensures that the model is capable of capturing objects of varying sizes and scales, enhancing its robustness in real-world settings. The accuracy of object detection models is frequently measured using established benchmarks like COCO (Common Objects in Context) [52] and specialized evaluation datasets such as Roboflow, YOLOv8 excels on both fronts.

For instance, the YOLOv8m model, which represents the medium-sized variant of the architecture, achieves a remarkable 50% mean average precision (mAP) [53] when evaluated on the COCO dataset. This result is a testament to the model's ability to accurately identify and localize objects within images, making it a formidable choice for tasks where precision is paramount. YOLOv8 and broadens its applicability across a diverse range of scenarios. Another critical enhancement in YOLOv8 is its utilization of adaptive training [54]. This dynamic training approach optimizes the learning rate and balances the loss function during the training process. The result is a more effective learning procedure that leads to improved model performance. Adaptive training ensures

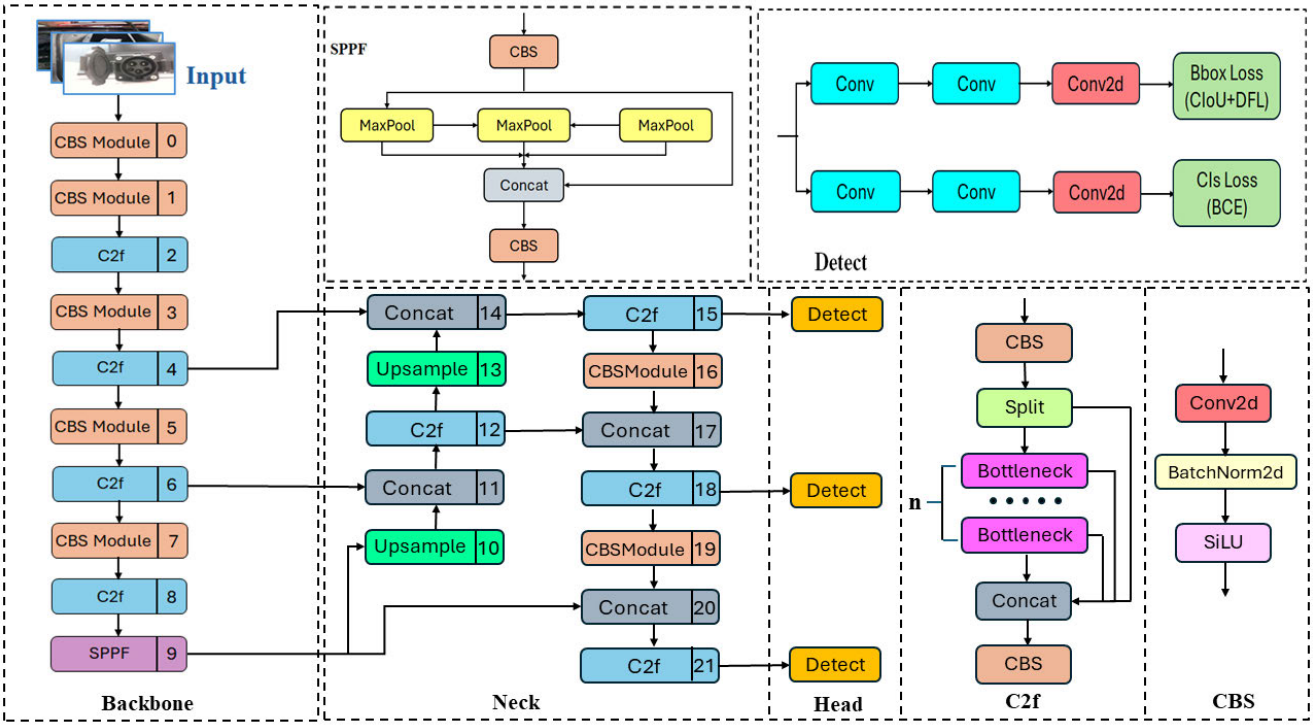


FIGURE 3. Architecture of YOLOv8 (Head, Neck, and Backbone).

that YOLOv8 can adapt to different datasets and tasks, making it a more versatile and capable tool for researchers and practitioners working with varied data sources and objectives.

Lastly, YOLOv8 simplifies the implementation and transfer learning process by providing pre-trained models. These pre-trained models serve as a valuable starting point, allowing users to kick start their projects with a strong foundation.

B. IMPLEMENTING ROBOT OPERATING SYSTEM

ROS stands as a robust open-source framework developed collaboratively by Willow Garage and Stanford University. Although its name suggests otherwise, ROS is not a conventional operating system. Instead, it operates as middleware, running on top of existing host operating systems like Linux. Figure 4 represents the unique architecture of ROS, provides a layer of essential functionalities and services, simplifying the complexities associated with robotics software development. What sets ROS apart is its flexibility and adaptability. It offers a modular platform that allows seamless integration of diverse software components.

This modularity facilitates the development of a wide range of robotic applications, from basic robotic arms to sophisticated autonomous systems. By handling critical tasks such as hardware abstraction, device drivers, and inter-process communication, ROS liberates researchers and developers from the burden of dealing with low-level software intricacies.

A key advantage of ROS is its focus on reusability. Developers can create and share software packages and libraries, fostering a collaborative environment within the robotics community. This collaborative spirit accelerates progress in research and development, as solutions to common challenges are collectively refined and improved. ROS supports a multitude of robotic platforms and hardware, making it versatile and applicable across various robotic systems. It provides a standardized communication framework, enabling seamless interaction between different subsystems of a robot, including sensors, actuators, and control algorithms. This standardized approach ensures interoperability, a crucial aspect in the development of complex robots where different components must work cohesively.

In summary, ROS serves as a catalyst for innovation in robotics. By simplifying software development complexities, offering modularity, and promoting collaboration, it empowers bundling libraries, executables, datasets, and configuration files [55]. They are the fundamental components of ROS applications, enabling collaboration and resource reuse among the ROS community. Topics and Messages facilitate communication between nodes, acting as named data buses where nodes exchange structured data entities called messages. These messages define the type of information shared, such as sensor data or control commands. ROS communication also involves Services, which enable nodes to request specific tasks from others and receive responses, enhancing the flexibility of interactions.

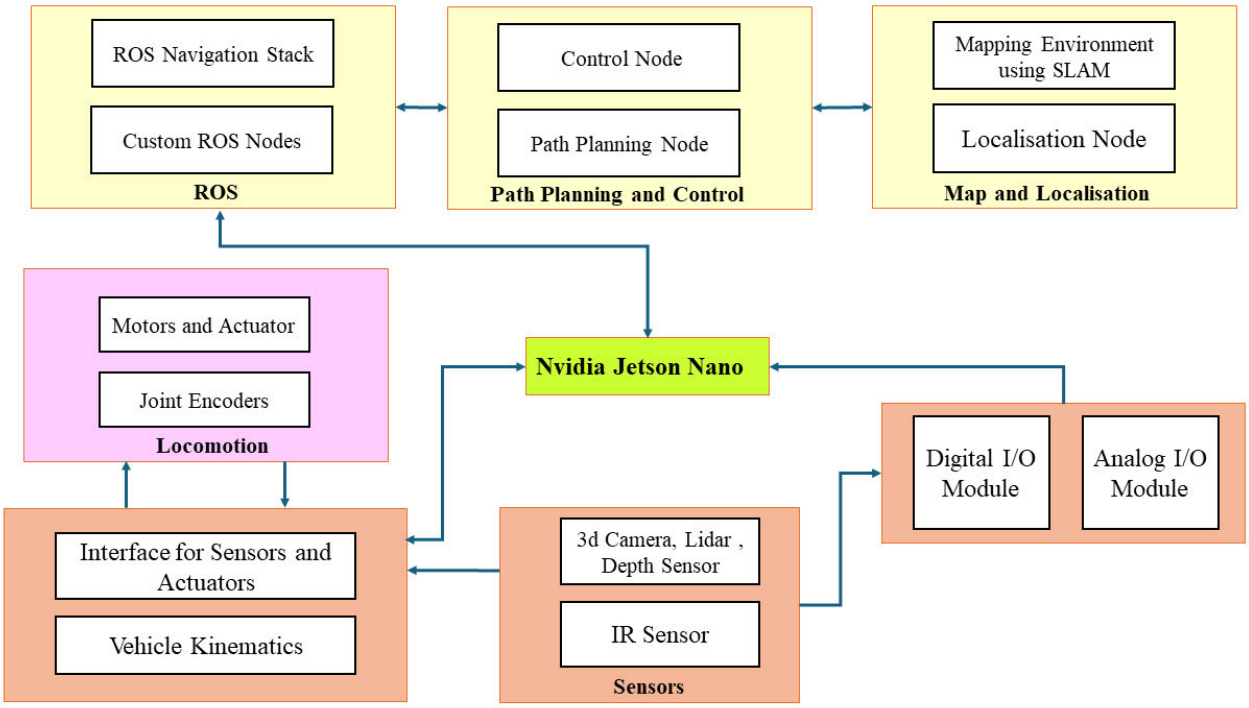


FIGURE 4. Basic architecture of ROS.

The ROS Master plays a pivotal role in orchestrating communication, managing names, and registration services [56]. It ensures seamless node discovery, allowing smooth message passing. Lastly, Catkin, ROS's robust build system, simplifies package organization into workspaces. It expertly handles the intricacies of the build process, easing the management of complex robotic systems. Together, these elements form the foundation of ROS, fostering collaboration, modularity, and efficient communication in robotic software development researchers, engineers, and enthusiasts to create, iterate, and deploy robotic applications effectively. Its open-source nature and collaborative ecosystem continue to drive advancements in the field, ecosystem continue to drive advancements in the field, making it an indispensable tool for roboticists worldwide.

Nodes are the core entities in ROS, representing autonomous modules that perform specialized tasks within a robotic system [57]. They communicate by exchanging messages, creating a modular and distributed framework for software design.

1) PACKAGES OF NOTE

- **Systems and Tools:**

actionlib: This component sets the standard for managing preemptable tasks, ensuring a smooth and efficient flow of operations [58].

nodelet: Enabling the concurrent execution of multiple algorithms within a single process, nodelet optimizes computational resources.

rosbridge: Acting as a bridge to the broader programming world, rosbridge offers a JSON API, enabling seamless integration with non-ROS programs.

- **Mapping and Localization:**

slam toolbox: This tool is a powerhouse, delivering extensive capabilities in 2D SLAM (Simultaneous Localization and Mapping) and localization.

gmapping: Serving as a wrapper for OpenSlam's Gmapping algorithm [59], this package facilitates simultaneous localization and mapping, a cornerstone of autonomous navigation

cartographer: Engineered by Google, cartographer implements real-time 2D and 3D SLAM algorithms, ensuring accurate and dynamic mapping [60].

amcl: The adaptive Monte-Carlo localization [61] implemented by amcl is pivotal, enabling robots to adapt and precisely locate themselves in complex environments.

- **Navigation:**

navigation: With this component, ROS bestows mobile robots with the ability to navigate seamlessly within planar environments. Its obstacle avoidance mechanisms ensure safe and efficient robot movement.

- **Manipulation:**

MoveIt: MoveIt is a robotic manipulation platform. MoveIt is the epitome of precision in motion planning. It equips robot manipulators with advanced capabilities, allowing them to perform intricate tasks such as object manipulation and grasping. Utilizing the Open Motion Planning Library (OMPL) [62], MoveIt ensures optimal planning strategies.

- **Perception:**

vision_opencv: Integrating ROS seamlessly with OpenCV [63], vision_opencv empowers robots with perception abilities. It enables them to interpret visual data from cameras and sensors, enhancing their understanding of the environment.

- **Coordinate Frame Representation:**

tf: Initially serving as the bedrock for representing, tracking, and transforming coordinate frames, tf paved the way until ROS Hydro [64]. In subsequent versions, tf2 emerged, inheriting and advancing the same capabilities, ensuring a smooth transition for ROS users.

- **Simulation:**

gazebo_ros_pkgs: This meta-package facilitates the integration of ROS with Gazebo [65], a sophisticated simulator. Gazebo allows developers to rigorously test and validate their robotic algorithms in a realistic virtual environment, ensuring robust performance in real-world scenarios.

stage: Providing an interface for the 2D Stage simulator, this tool offers a simulated environment for testing and experimentation. These meticulously crafted packages collectively enhance ROS's functionalities, rendering it a remarkably versatile platform for the development of intricate robotic systems. As ROS continues to evolve, it remains at the forefront of robotics innovation, fostering a collaborative ecosystem and driving the field towards new horizons.

C. WORK FLOW EXPLANATION OF THE OVERALL PROPOSED SYSTEM

The system's operational flow begins with the acquisition of images of the charging port via the camera module as per the Figure 5. These images are subjected to a segmentation process, dividing them into segments or grids, facilitating subsequent feature extraction. The feature extraction and bounding box prediction are executed by the YOLOv8 Model.

After the bounding box prediction, the system proceeds to classify the charging port based on predefined types or classes. This classification step involves leveraging a convolutional neural network (CNN) within the YOLOv8 Model, which excels in image classification tasks.

In scenarios where a match is not initially found during classification, additional images are captured for further analysis. This iterative process enhances the system's robustness and adaptability to various environmental conditions and charging port appearances.

Simultaneously, the system integrates a 3D camera for capturing spatial information. This 3D data is crucial for implementing the Simultaneous Localization and Mapping (SLAM) technique. SLAM allows the system to construct a map of its surroundings while concurrently determining its own location within that map. This spatial awareness is vital for effective navigation and positioning of the Automated Charging Robot (ACR). The subsequent step involves path planning, where global and local costmaps are generated and evaluated. Global costmaps consider the entire environment,

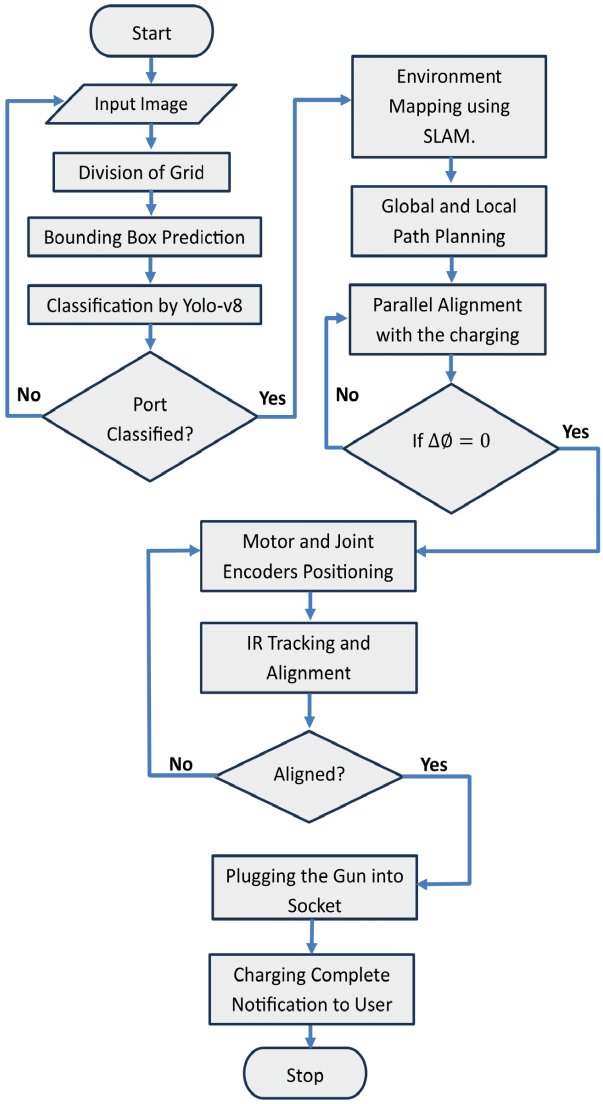


FIGURE 5. Flow chart of the proposed working model.

while local costmaps focus on immediate surroundings. Path planning is often achieved through algorithms like A∗ (A-star) or Dijkstra's algorithm, ensuring the ACR navigates optimally towards the charging port. As the ACR advances, it strategically aligns charging gun parallel to charging socket using feedback control systems. $\Delta\emptyset$ the angle between the socket and charging gun axis, is actively maintained at zero to ensure precise alignment. The motors then drive the ACR forward while simultaneously adjusting its orientation to sustain the parallel alignment. As the ACR approaches the charging port within a specified range (0.15m), it initiates an infrared (IR) ray tracing process. This involves tracking the IR rays emitted by the charging port, enabling the ACR to dynamically adjust the gun's position based on the detected rays.

Upon achieving precise alignment through the coordinated adjustment of x, y, and z coordinates, the ACR inserts the

charging gun into the socket, initiating the charging process. The charging process involves communication between the ACR and the charging infrastructure, ensuring a secure and efficient power transfer.

Upon successful completion of the charging process, a notification is sent to the user through a communication protocol such as GSM/GPRS, informing them of the concluded charging cycle. This notification mechanism enhances user convenience and ensures timely awareness of the charging status.

D. AViTRoN (ADVANCED VISION TRACK ROUTING AND NAVIGATION)—A PROPOSED DEVICE

In response to the evolving landscape of electric vehicle (EV) charging, Figure 6 shows our pioneering initiative, "AViTRoN," seamlessly integrates advanced technologies, including the ROS, for charging port detection and precise navigation of Adaptive Charging Robots (ACRs). At the core of our mission is the optimization of EV charging processes, empowering ACRs to detect charging ports with remarkable accuracy and navigate seamlessly in intricate 3D environments. AViTRoN's adaptability overcomes challenges such as unwieldy cables and low-light conditions, simplifying the charging experience for users and significantly enhancing efficiency. Beyond its immediate application in EV charging, AViTRoN's impact resonates across diverse industries. Its sophisticated 3D vision and detection capabilities, combined with ROS integration, enable precise object identification, seamless navigation, and collision avoidance. This adaptability positions AViTRoN as an invaluable asset in sectors like manufacturing and healthcare robotics, where complex and dynamic environments demand intelligent and efficient solutions. Furthermore, AViTRoN plays a pivotal role in promoting environmental sustainability by optimizing the charging process, encouraging the widespread adoption of eco-friendly transportation solutions.

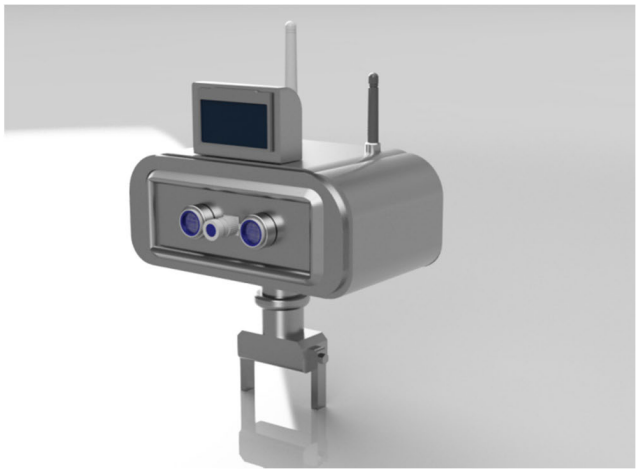


FIGURE 6. Proposed design of AViTRoN.

In essence, AViTRoN signifies a transformative leap in the realms of EV charging and industrial automation. By amalgamating advanced technologies, including ROS, and addressing multifaceted challenges, our initiative streamlines EV charging processes and ushers in a new era of intelligent automation, fostering a sustainable future and technological advancement across various sectors.

1) COMPONENTS OF AViTRoN

In the innovative design of AViTRoN, Figure 7 shows a multitude of components converge to revolutionize the landscape of electric vehicle (EV) charging. The 2.42 Inch OLED Display serves as an intuitive interface, offering real-time charging level updates to users and enhancing the overall user experience. Central to AViTRoN's capabilities is the Camera Depth 3D Vision Sensor, a sophisticated tool capturing intricate 3D images. These images not only facilitate precise mapping and navigation within the charging environment but are also processed through the powerful ROS on the Nvidia Jetson Nano. Designed specifically for AI and deep learning applications, the Jetson Nano is a compact yet high-performance computing platform. Its remarkable computational abilities enable AViTRoN to execute intricate tasks seamlessly, ranging from charging port detection to complex coordination of ROS functions.

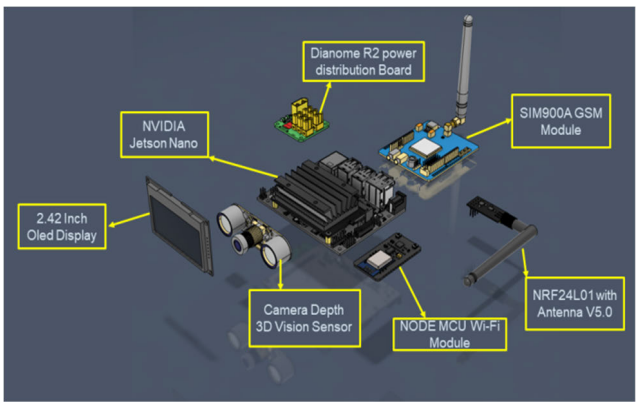


FIGURE 7. Components of AViTRoN.

The Jetson Nano features a quad-core ARM Cortex-A57 CPU [66], combined with a 128-core NVIDIA Maxwell GPU. This amalgamation of processing units delivers lightning-fast processing speeds, ensuring real-time analysis of the charging environment. With 4 GB of LPDDR4 RAM, the Jetson Nano can handle data-intensive tasks with ease, providing a responsive and efficient user experience. Furthermore, the Jetson Nano's compatibility with various neural network frameworks, including TensorFlow and PyTorch, empowers AViTRoN with advanced machine learning capabilities. Its ability to process data from the Camera Depth 3D Vision Sensor, capturing intricate 3D images of the charging environment, enables AViTRoN to create detailed spatial maps for precise navigation.

The integration of ROS with the Jetson Nano is pivotal for AViTRoN's functionality, facilitating accurate charging port detection and precise navigation. AViTRoN also utilizes the SIM900A GSM module for timely charging completion notifications, enhancing user engagement. The NVIDIA Jetson Nano serves as the primary processing unit, managing tasks like charging port detection. For data management, AViTRoN employs the NODE MCU Wi-Fi Module, ensuring seamless data transmission to the cloud for secure storage and analysis. To boost communication capabilities, the NRF24L01 with Antenna V5.0 extends Wi-Fi range significantly.

Figure 8 offers an overview of the AVITRON system. The 3D Camera Module is responsible for capturing intricate visual data, which is subsequently transmitted to the NVIDIA Jetson Nano for real-time analysis. The Jetson Nano processes the data, utilizing machine learning algorithms to discern various charging port types and their respective conditions. Concurrently, the IR Receiver Module embedded within the charging gun communicates with the ACR, facilitating seamless coordination throughout the plug-in procedure.

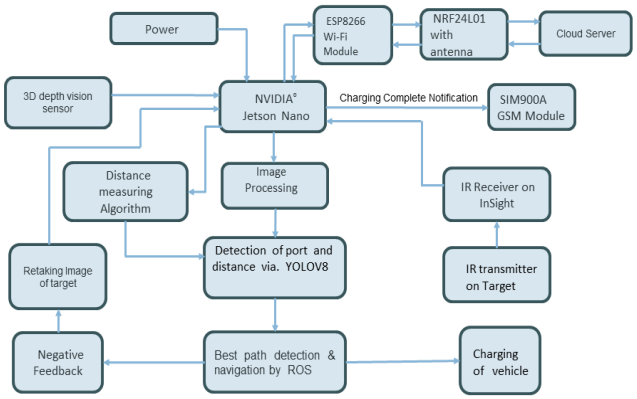


FIGURE 8. Proposed block diagram of AViTRoN.

FIGURE 9. Different types of charging port.

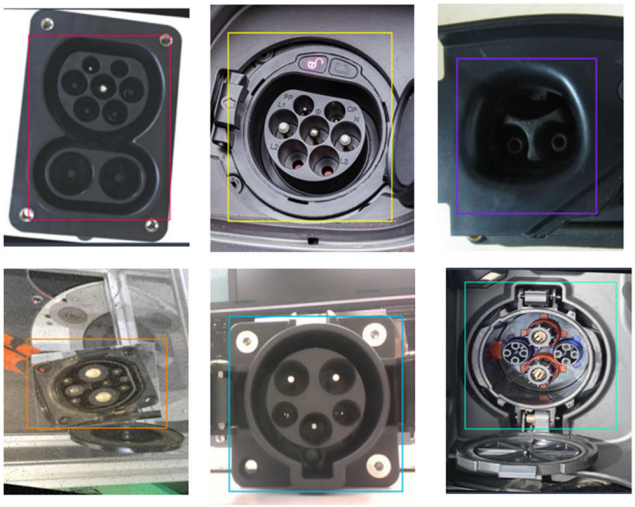


FIGURE 10. Annotation the images into different class.

2) WORKING OF AViTRoN

a: ACQUIRING IMAGES OF CHARGING PORTS

Within the realm of global identification and positioning, Figure 9 shows the preliminary step involves acquiring general positioning data of the charging port. This positioning data serves as a reference for maneuvering the mechanical arm joint, facilitating the movement of the charging gun toward the vicinity of the charging port.

b: PRECISION CHARGING PORT DETECTION THROUGH YOLOv8 RECOGNITION AND ROI EXTRACTION

The process of extracting a Region of Interest (ROI) using the YOLOv8 recognition algorithm [67] for a charging port begins with loading a pre-trained YOLOv8 model and its configuration files. The input image containing the potential charging port is resized and normalized.

A forward pass through the YOLOv8 network predicts bounding box coordinates, class probabilities, and objectness score. Non-maximum suppression (NMS) retains the most confident bounding boxes [68]. The algorithm filters predictions for the charging port class. During ROI extraction, the possibility of bounding boxes extending beyond the port's area is considered. Figure 10 shows that by using bounding box coordinates, the precise charging port region is defined, accommodating potential extensions. The algorithm harnesses YOLOv8's capabilities for accurate identification while handling bounding box variations.

c: MATCHING WITH KM ALGORITHM

The alignment of input image with reference depends on angles and perspectives from which they were captured. Proximity in position directly influences the degree of correspondence; closer positions yield higher correspondence levels. The KM (Kuhn-Munkres) algorithm [69] is harnessed to formalize this relationship, employing the inverse of distance as the algorithm's weight. This weight calculation,

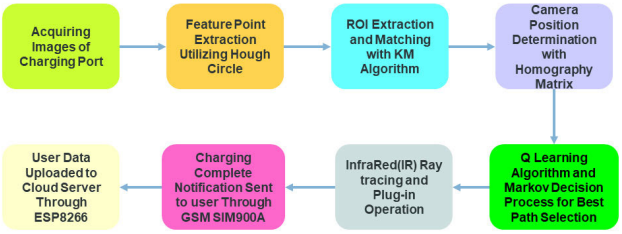


FIGURE 11. Proposed working flow of AViTRoN.

as outlined in equation (1), can be expressed as:

$$Cij = \frac{k}{dist\ (\ Pi - Pj\ )} \tag{1}$$

Within this equation, the variable k is a parameter, *dist* denotes the distance, and $P = (a_n,\ b_n,\ r_n)$ signifies central coordinate. This formulation delineates the interplay between distance, weight, and the inherent characteristics of feature points, underpinning the KM algorithm's fundamental approach to feature matching.

d: ENHANCING CAMERA POSITION DETERMINATION WITH HOMOGRAPHY MATRIX

A homography matrix represents a projection transformation [70] that captures the mapping between images of a planar object acquired from distinct camera viewpoints, assuming an absence of lens distortion. This matrix encapsulates the geometric alignment between these images. In Fig. 12, an object is situated in space, and two photographs are taken from varying camera angles. These images encompass several points denoted as P. These points are characterized by pixel coordinates in the two images: P1 (x1, y1) and P2 (x2, y2) as shown in Figure 12.

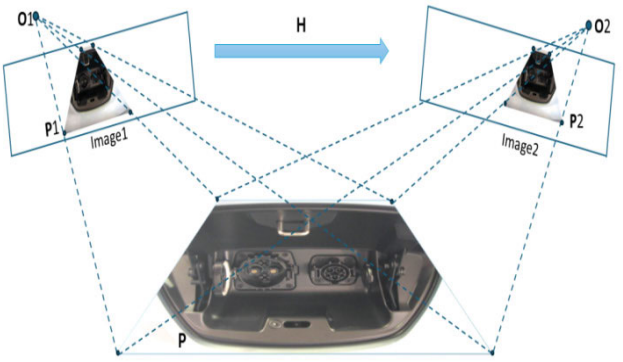


FIGURE 12. Enhancing camera position determination with Homography Matrix.

The connection between these coordinates is established through a homography transformation, mathematically expressed as P1 = H ∗ P2. In this equation, 'H' signifies the homography matrix, illustrating the transformation relationship that governs the mapping between the two sets of coordinates. This framework facilitates the translation of points from one image to another, enabling geometric correspondence between the images despite their distinct perspectives. The approach involves computing based on the matched points and subsequently decomposing it to determine the camera's external parameters, encompassing the rotation and translation matrices.

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{2}$$

Normalize the matrix H by multiplying it with a non-zero factor to ensure that h9 equals 1. This normalization is essential as matrix H contains 8 DOF, expanding formula (2), we get:

$$h_7x_1 + h_8y_1 + h_9 = 1 \tag{3}$$

$$x_2 = \frac{h_1x_1 + h_2y_2 + h_3}{h_7x_1 + h_8y_1 + h_9} \tag{4}$$

$$y_2 = \frac{h_4x_1 + h_5y_2 + h_6}{h_7x_1 + h_8y1 + h_9} \tag{5}$$

After simplification we get

$$x_2 = h_1x_1 + h_2y_2 + h_3 - h_7x_1x_2 - h_8y_1 \tag{6}$$

$$y_2 = h_4x_1 + h_5y_2 + h_6 - h_7x_1y_2 - h_8y1y_2 \tag{7}$$

Two sets of corresponding point pairs, P1 and P2, contribute to the establishment of two constraints. This implies that a homography matrix, possessing 8 degrees of freedom, can be computed using four distinct points.

e: INTEGRATION OF POINT CLOUD PROCESSING, NAVIGATION STACK AND MOTION PLANNING FOR PRECISE PLUGGING

In the context of 3D perception, PCL (Point Cloud Library) and pcl_ros package are crucial components in ROS, facilitating the manipulation and processing of point cloud data. pcl_ros is a ROS wrapper for the Point Cloud Library (PCL), which is an open-source library for 2D/3D image and point cloud processing. pcl_ros acts as a bridge between ROS and PCL, allowing seamless integration of point cloud processing capabilities into ROS-based robotic systems. A point cloud is a collection of data points in a 3D space, often representing the surfaces of objects within an environment. The sensor_msgs/PointCloud2 message type in ROS is a standardized way of representing point cloud data. It provides a flexible and efficient binary serialization format for point cloud information, making it suitable for communication between different ROS nodes and packages. Generation of a 3D point cloud representation of the environment using a 3D depth camera in ROS, a requires a systematic process. First, the 3D depth camera is connected and configured to the robot, ensuring that it publishes depth information in a ROS-compatible format. Subsequently, a ROS node is utilized to acquire and publish depth data from the 3D camera in the form of sensor_msgs/PointCloud2 messages as shown

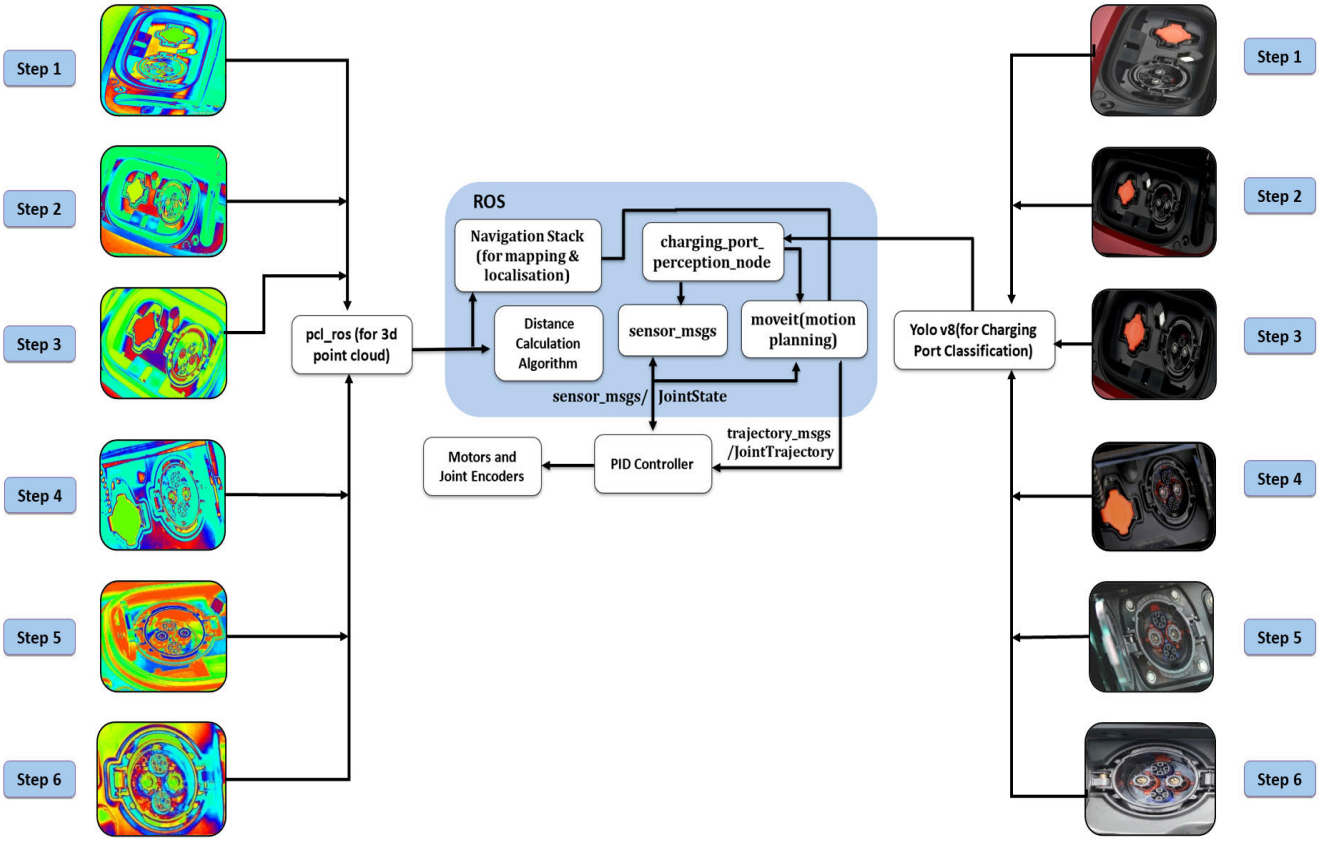


FIGURE 13. Pictorial representation of integration of nodes, point cloud processing, navigation, and motion planning in ROS.

in Figure 13. The integration of the pcl_ros package comes into play with the implementation of a ROS node that subscribes to the sensor_msgs/PointCloud2 topic from the depth camera node. This node then converts the received data to a PCL point cloud using the pcl::fromROSMsg function, facilitating further processing. The point cloud processing stage involves leveraging PCL functionalities for tasks such as filtering, segmentation, feature extraction, and registration. The integration of 3D point cloud data into the ROS Navigation Stack creates a seamless robotic navigation system. Initiated by sensor data integration in the form of sensor_msgs/PointCloud2, the Navigation Stack constructs a costmap that encapsulates both static and dynamic obstacles. Global path planning, powered by algorithms like Dijkstra's or A∗, charts a high-level trajectory, while the Local Planner dynamically recalculates a local trajectory in real-time. the stack continuously updates its map using 3D point cloud data, providing nuanced insights into the environment. After the acquisition and processing of the point cloud data, the next phase integrates this information into the motion planning framework.

MoveIt, a widely used motion planning library in ROS, employs algorithms like RRT (Rapidly-exploring Random Trees) or OMPL (Open Motion Planning Library) to generate collision-free trajectories for the robot's end-effector or joints. The trajectory_msgs/JointTrajectory message type is instrumental in conveying the planned trajectory to the robot's control system. This message encapsulates the desired positions, velocities, and accelerations for each joint over time. The joint trajectory, computed based on the 3D point cloud analysis, serves as a high-level command to drive the motors and encoders of the robot toward the target configuration. the robot's control system interprets the JointTrajectory message and executes the corresponding joint movements. The control loop consists of a PID (Proportional-Integral-Derivative) controller, which adjusts the motor commands based on the difference between the current and joint states. Encoders, which measure the actual joint positions, provide feedback to ensure precise and accurate execution of the planned trajectory. The collision information derived from the 3D point cloud data plays a vital role in ensuring the safety of the robotic system during motion planning. MoveIt utilizes sophisticated collision checking algorithms that leverage the 3D point cloud representation to validate planned trajectories, effectively preventing collisions and enhancing overall safety in the robot's environment. The determination of joint states and alignment positions is facilitated through the utilization of the sensor_msgs package

and, specifically, the sensor_msgs/JointState message type in ROS. The sensor_msgs package in ROS encompasses message types for sensor data communication between nodes in a robotic system. One crucial message type within this package is sensor_msgs/JointState. This message is instrumental in conveying information about the state of joints in a robot, including positions, velocities, and efforts. Nodes involved in perception, control, and planning can subscribe to or publish JointState messages, enabling synchronized awareness of the robot's joint configuration. This data is fed into the motion planning node to formulate a path that is free of errors. The Charging Port Perception Node employs YOLO v8 for charging port detection using the 2d camera system of AViTRoN, receiving visual data as sensor_msgs/Image messages and obtaining motor and joint encoder states through sensor_msgs/JointState messages. Utilizing trajectory_msgs/JointTrajectory, it communicates with a PID controller to drive motors and joint encoders. Trajectory_msgs is a ROS package that defines message types, including JointTrajectory, for representing and transmitting trajectory information. This package is instrumental in conveying planned trajectories to the robot's control system, facilitating precise and coordinated motion execution. This integration of YOLO v8, sensor_msgs, Moveit path planning, and trajectory_msgs/JointTrajectory facilitates dynamic navigation, allowing the robot to detect charging ports, adjust motor alignment, and execute precise paths with finesse.

f: FOUR-STEP STRATEGY FOR ACCURATE PLUG-IN WITH IR ALIGNMENT

Upon determining the charging port's pose, a strategic coordinate system is established, positioning the origin at the plug's center while orienting the Z-axis outward. Similarly, another coordinate system is aligned with the connector plug held by the robot. To further enhance alignment accuracy, an IR transmitter is integrated into the target plug, while an IR receiver is placed on the charging gun, as depicted in Figure 14. The primary objective of this enhanced plug-in procedure is to ensure precise alignment between the connector plug and the charging port, enabling a final motion along a single axis. To achieve this accuracy, a four-step strategy has been devised:

g: RAPID APPROACH

The initial step involves swift motion as the robot guides the plug to the approach position, located within a 0.15-meter radius around the charging port. This initial positioning sets the stage for subsequent adjustments.

h: VELOCITY REDUCTION AND IR ALIGNMENT

In the second step, the robot's velocity is meticulously reduced to 15% of its maximum joint speed. Simultaneously, the IR receiver on the charging gun detects the IR signal from the transmitter on the target plug. This controlled motion and IR alignment ensure exact positioning [71], bringing the connector plug and charging port in perfect agreement along their Z-axes and a few millimeters from the contact point.

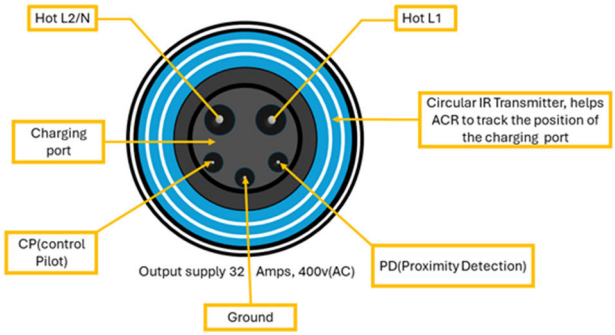


FIGURE 14. Proposed EVs charging port: integrated with IR transmitter.

i: GRADUAL Z-AXIS MOTION

Figure 14 shows the third phase focuses on careful motion along the Z-axis. The robot's speed is reduced to 2% of its maximum, allowing it to approach the final plug-in motion gradually while maintaining IR alignment. This gradual approach enhances alignment precision.

j: ROTATION, ACTIVATION, AND IR CONFIRMATION

Achieving precise alignment, the robot executes a calculated rotation of the charging gun, accurately aligning it with the target plug. This aligned configuration enables the initiation of the charging mechanism, facilitating power transfer. Additionally, the IR receiver confirms the continued alignment during the plug-in process. This step guarantees both physical alignment and functional connection between the plug and the charging mechanism as shown in Figure 15.

k: FORWARD AND INVERSE KINEMATICS OF PROPOSED ACR DRIVEN BY AVITRON

Forward Kinematics:

The forward kinematics of a 6-degree-of-freedom [72] Autonomous Charging Robot (ACR) driven by AViTRoN is a foundational aspect of its operation. This process involves calculating the precise position and orientation of the ACR's charging arm based on its joint angles and arm lengths as shown in Figure 16. Utilizing advanced algorithms and sensor data, AViTRoN accurately computes these kinematic equations, enabling the ACR to autonomously navigate towards an electric vehicle, align its charging arm with the socket, and establish a secure charging connection.

AViTRoN's ability to perform these forward kinematic calculations ensures the ACR's movements are exact and dependable, guaranteeing the efficiency and safety of the charging process. This integration of advanced technologies demonstrates the seamless combination of robotics, artificial intelligence, and motion control, showcasing AViTRoN's potential in real-world applications. Its precise forward

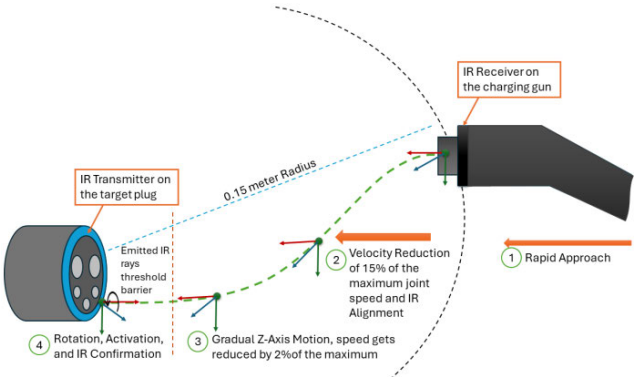


FIGURE 15. Four-Step strategy for accurate plug-in with IR alignment.

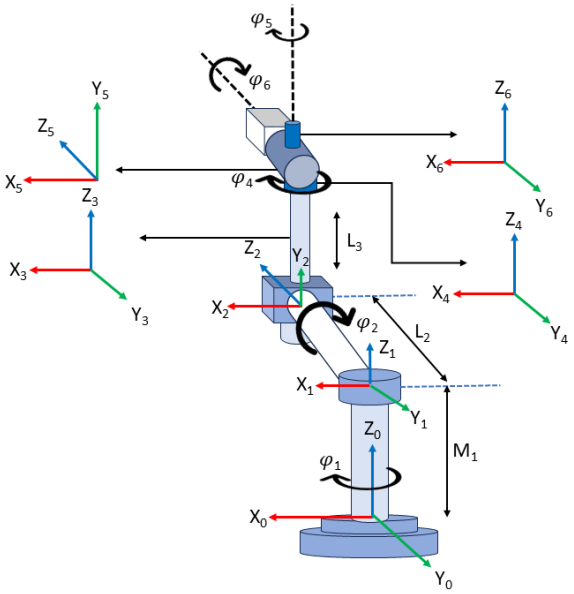


FIGURE 16. Forward kinematics of proposed ACR.

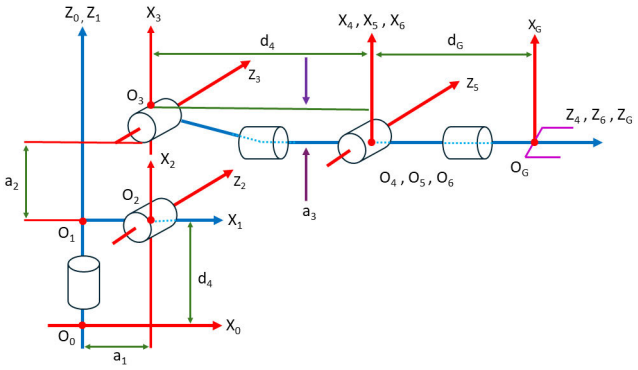


FIGURE 17. Inverse kinematics of proposed ACR.

kinematics capabilities make it a significant advancement in autonomous charging systems, streamlining the charging process for electric vehicles.

TABLE 2. DH Parameter of the ACR.

i	φ_i	β_{i-1}	a_{i-1}	L_i
1	φ_1	0	0	M_1
2	φ_2	90	3.18	L_2
3	0	-90	2.03	L_3
4	φ_4	0	0	0
5	φ_5	90	0	0
6	φ_6	-90	0	0

Each link transformation Matrices will be

$$ {}^{0}_{1}T = \begin{bmatrix} cos\varphi_1 & -sin\varphi_1 & 0 & 0 \\ sin\varphi_1 & cos\varphi_1 & 0 & 0 \\ 0 & 0 & 1 & M_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8} $$

$$ {}^{1}_{2}T = \begin{bmatrix} cos\varphi_2 & -sin\varphi_2 & 0 & 0 \\ 0 & 0 & -1 & -l_2 \\ sin\varphi_2 & cos\varphi_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9} $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10} $$

$$ {}^{3}_{4}T = \begin{bmatrix} cos\varphi_4 & -sin\varphi_4 & 0 & 0 \\ sin\varphi_4 & cos\varphi_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11} $$

$$ {}^{4}_{5}T = \begin{bmatrix} cos\varphi_5 & -sin\varphi_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ sin\varphi_5 & cos\varphi_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12} $$

$$ {}^{5}_{6}T = \begin{bmatrix} cos\varphi_6 & -sin\varphi_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -sin\varphi_6 & -cos\varphi_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13} $$

The calculation of the end-effector's position concerning the base frame is established by the multiplication of all the individual transformation matrices ${}^{i-1}_{i}$ T, where each matrix represents a specific transformation step.

$$ {}^{Base}_{End-Effector}T = {}^{0}_{1}T\ {}^{1}_{2}T\ {}^{2}_{3}T \ldots . {}^{i-1}_{i}T \tag{14} $$

$$ {}^{0}_{6}T = \begin{bmatrix} k_{11} & k_{12} & k_{13} & Q_x \\ k_{21} & k_{22} & k_{23} & Q_y \\ k_{31} & k_{32} & k_{33} & Q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15} $$

where

$$ \begin{aligned} k_{11} =& -sin\varphi_6(cos\varphi_4 sin\varphi_1 + cos\varphi_1 cos\varphi_2 sin\varphi_4) - cos\varphi_6 \\ & (cos\varphi_5(sin\varphi_1 sin\varphi_4 - cos\varphi_1 cos\varphi_2 cos\varphi_4) \\ & + cos\varphi_1 sin\varphi_2 sin\varphi_5) \end{aligned} \tag{16} $$

$$ \begin{aligned} k_{12} =& sin\varphi_6(cos\varphi_5(sin\varphi_1 sin\varphi_4 - cos\varphi_1 cos\varphi_2 cos\varphi_4) \\ & + cos\varphi_1 sin\varphi_2 sin\varphi_5) - cos\varphi_6(cos\varphi_4 sin\varphi_1 \\ & + cos\varphi_1 cos\varphi_2 sin\varphi_4) \end{aligned} \tag{17} $$

$$ \begin{aligned} k_{13} =& sin\varphi_5(sin\varphi_1 sin\varphi_4 - cos\varphi_1 cos\varphi_2 cos\varphi_4) \\ & - cos\varphi_1 sin\varphi_2 cos\varphi_5 \end{aligned} \tag{18} $$

$$ \begin{aligned} k_{21} =& sin\varphi_6(cos\varphi_4 cos\varphi_1 - sin\varphi_1 cos\varphi_2 sin\varphi_4) + cos\varphi_6 \\ & (cos\varphi_5(cos\varphi_1 cos\varphi_4 + sin\varphi_1 cos\varphi_2 cos\varphi_4) \\ & - sin\varphi_1 sin\varphi_2 sin\varphi_5) \end{aligned} \tag{19} $$

$$ \begin{aligned} k_{22} =& cos\varphi_6(cos\varphi_4 cos\varphi_1 - sin\varphi_1 cos\varphi_2 sin\varphi_4) \\ & - sin\varphi_6(cos\varphi_5(cos\varphi_1 cos\varphi_4 + sin\varphi_1 cos\varphi_2 cos\varphi_4) \\ & - sin\varphi_1 sin\varphi_2 sin\varphi_5) \end{aligned} \tag{20} $$

$$ \begin{aligned} k_{23} =& -sin\varphi_5(cos\varphi_1 sin\varphi_4 + sin\varphi_1 cos\varphi_2 cos\varphi_4) \\ & - sin\varphi_1 sin\varphi_2 cos\varphi_5 \end{aligned} \tag{21} $$

$$ \begin{aligned} k_{31} =& cos\varphi_6(cos\varphi_2 sin\varphi_5 + cos\varphi_5 sin\varphi_2 cos\varphi_4) \\ & - sin\varphi_2 sin\varphi_4 sin\varphi_6 \end{aligned} \tag{22} $$

$$ \begin{aligned} k_{32} =& -sin\varphi_6(cos\varphi_2 sin\varphi_5 + sin\varphi_2 cos\varphi_5 cos\varphi_4) \\ & - cos\varphi_6 sin\varphi_2 sin\varphi_4 \end{aligned} \tag{23} $$

$$ k_{33} = cos\varphi_2 cos\varphi_5 - sin\varphi_5 sin\varphi_2 cos\varphi_4 \tag{24} $$

$$ Q_x = l_2 sin\varphi_1 - l_3 cos\varphi_1 sin\varphi_2 \tag{25} $$

$$ Q_y = -l_2 cos\varphi_1 - l_3 sin\varphi_1 sin\varphi_2 \tag{26} $$

$$ Q_z = M_1 + l_3 cos\varphi_2 \tag{27} $$

Inverse Kinematics:

Homogeneous transformation matrix of the 6 DOF ACR is given by [73]

$$ {}^{k-1}_{k}T = \begin{bmatrix} cos\varphi_k & -sin\varphi_k cos\beta_k & sin\varphi_k cos\beta_k & a_k cos\varphi_k \\ sin\varphi_k & cos\varphi_k sin\beta_k & -cos\varphi_k sin\beta_k & a_k sin\varphi_k \\ 0 & sin\beta_k & cos\beta_k & l_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{28} $$

$$ R = \begin{bmatrix} cos\varphi_k & -sin\varphi_k cos\beta_k & sin\varphi_k cos\beta_k \\ sin\varphi_k & cos\varphi_k sin\beta_k & -cos\varphi_k sin\beta_k \\ 0 & sin\beta_k & cos\beta_k \end{bmatrix} \tag{29} $$

Here R represents the rotation matrix of the arm. The Jacobian matrix (J) is computed to establish the relationship between joint angle changes and end effector position changes, enabling precise control of the robotic arm's movements towards a desired goal position.

$$ J = \begin{bmatrix} Ja \\ Jb \end{bmatrix} = \begin{bmatrix} & -1\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \\ R^0_k & & (d^0_k - d^{0-1}_k) \\ & -1\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} & \\ & R^0_k & \end{bmatrix} \tag{30} $$

Here R represents the rotation matrix and d^0_k represents displacement from the origin of the global coordinates Jacobian for 6 DOF or joint robot is calculated as.

$$ Ja = [Ja_1, Ja_2, \ldots Ja_6] \tag{31} $$

where

$$ Ja_n = k_n(U_{Grip} - U_n) \tag{32} $$

K_n represents the 3-element axis.

U_{Grip} is the 3-element relative position vector of the end effector with respect to the base. U_n is the relative position of joints with respect to base.

I: EV BATTERY MONITORING AND NOTIFICATION SYSTEM WITH SOC PREDICTION

During the electric vehicle (EV) charging process, a precise assessment of the State of Charge (SOC) is conducted. Simultaneously, the time required for charging is measured and conveyed to a web-based platform, where the data can be visually presented through graphs, images, and other means. This serves the fundamental purpose of informing the user about the performance of their EV battery [74].

Upon reaching a SOC of 100%, denoting the completion of the charging process, an automatic notification or message is dispatched to the user. This timely notification plays a crucial role in assisting the user in determining when to retrieve their fully charged EV. To implement this system, an ESP8266 module is integrated with an Arduino Uno board. The SOC data, accurately computed during charging, is transmitted to the ESP8266 via UART communication pins. Subsequently, this data is uploaded to a web-based platform over a Wi-Fi connection. When the EV's battery reaches a 100% SOC, signaling the successful completion of the charging process, the system proceeds to calculate the final SOC value. The Arduino Uno integrated into the system, is promptly activated once a full SOC is reached. The Arduino is responsible for managing the notification process. To send a notification to the user, the Arduino interfaces with a GSM SIM 900A module. This module enables the system to connect to a mobile network, allowing for the prompt delivery of messages or alerts. Once the Arduino detects the 100% SOC, it initiates the GSM module to send a notification to the user, alerting them that the charging process is complete, and their EV is ready for use. This real-time notification streamlines the

user's experience, ensuring they are informed at the optimal time to collect their vehicle as shown in Figure 18.

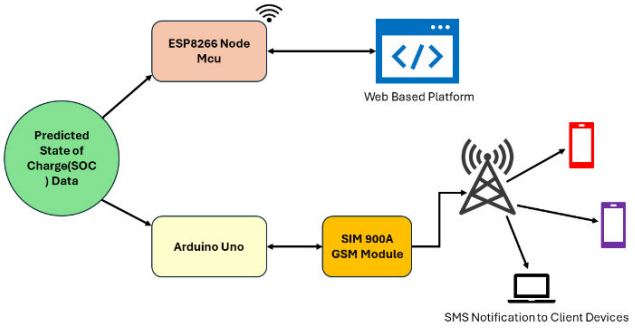


FIGURE 18. Illustration of SOC data transmission and SMS notification through AViTRoN.

IV. EXPERIMENT

In our experimental methodology, we meticulously curated an extensive dataset comprising over 2000 images depicting Electric Vehicles (EVs) charging sockets obtained from Roboflow. These images were not only gathered but also meticulously annotated, providing valuable contextual information to our dataset. The annotations were carefully organized into seven distinct classes: Type1, Type2, CCS1, CCS2, CHAdeMO, GB/T, and TESLA Model(S) as shown in Figure 19. This detailed classification ensured the diversity of our dataset, covering various EV charging socket types encountered in real-world scenarios.

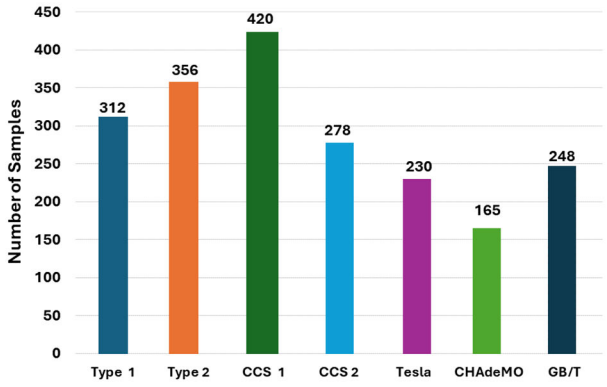


FIGURE 19. Class distribution in the dataset.

To enhance the model's adaptability and robustness, we implemented a range of data augmentation techniques. These techniques involved altering multiple parameters to simulate diverse environmental conditions and scenarios. We applied cropping techniques, varying from 0% minimum zoom to 20% maximum zoom, allowing the model to handle discrepancies in the size and scale of charging sockets. Additionally, shear transformations were applied both horizontally and vertically, with angles of $\pm 15°$, replicating real-world perspectives. Adjustments in hue and brightness, spanning from -25° to +25° and -25% to +25%, respectively, were made to accommodate varying lighting conditions. To simulate environmental challenges, we introduced blur of up to 2.5 pixels and noise up to 5% of pixels, ensuring the model's adaptability in less-than-optimal visual scenarios.

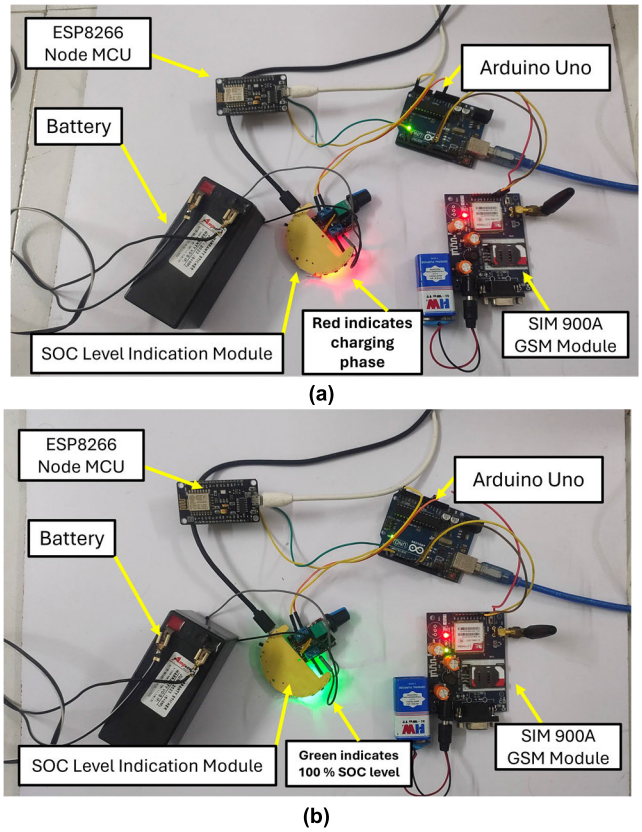


FIGURE 20. Illustration (a) Charging Phase (b) Indication for 100% SOC.

Furthermore, a mosaic effect was applied to the images, enhancing the model's resilience against complex, mosaic-like patterns in the charging environment. By subjecting our dataset to these diverse augmentation techniques, our objective was to equip our model with the capability to perform effectively under a wide array of environmental changes, ensuring its accuracy and reliability in real-world applications. This approach was tailored to ensure the model's robustness and versatility in handling diverse charging scenarios, while maintaining data integrity and originality.

The depicted experimental setup illustrates the framework employed for State of Charge (SOC) data transmission and Short Message Service (SMS) notifications. An Arduino Uno served as the focal point for SOC detection and dispatching SMS notifications upon charging completion. The charging process involved a lead battery with a capacity of 12V and 1.3A, meticulously monitored throughout the experiment. SOC detection was achieved through a dedicated SOC indication module featuring an integrated LED indicator. The LED emitted a red glow during the charging phase as shown

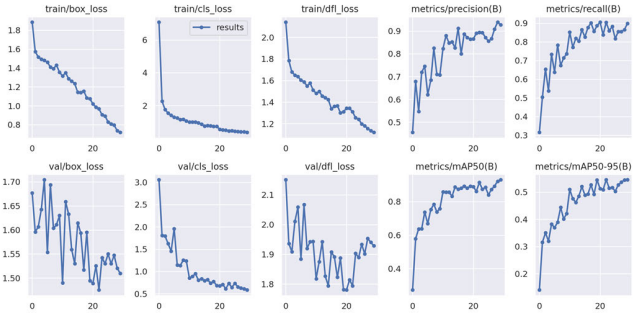


FIGURE 21. Loss function of the proposed YOLOv8 model during training.

TABLE 3. Results and F1% Score for Each Classes of Proposed Model.

Class	Precision	Recall	mAP (0.50)	F1% Score	mAP 0.5-0.95
All	0.927	0.947	0.930	0.912	0.545
Type1	0.988	1.000	0.995	0.993	0.622
Type2	0.861	0.842	0.917	0.851	0.504
CHAdeMO	0.905	0.750	0.830	0.820	0.419
CCS1	0.990	0.875	0.971	0.928	0.423
CCS2	0.991	1.000	0.995	0.995	0.711
Tesla(S)	0.838	0.882	0.879	0.859	0.562
GB/T	0.990	0.933	0.925	0.960	0.577

in Figure 20.(a) and transitioned to green upon reaching a 100 percent SOC as shown in Figure 20.(b), signaling the completion of the charging process.

Simultaneously, for SMS notifications, a SIM900A GSM module was utilized. This module, equipped with an SMA Antenna boasting extensive range and bandwidth, operated at a frequency of 900MHz, ensuring robust communication capabilities.

Furthermore, the experimental setup incorporated an ESP8266 Wi-Fi module to facilitate the upload of SOC data to a web-based platform. This platform enables comprehensive user monitoring and analysis of the charging process. The amalgamation of these components ensured a sophisticated and technologically advanced system for SOC monitoring and user notification in the context of battery charging experimentation. The experimental setup showcases AViTRoN's effective notification system, promptly updating users on charging completion, highlighting the device's advanced features.

The plugging process is carried out in a meticulous six-step procedure for charging port detection and charging gun insertion. In the first step, a 2D camera captures images for charging port detection, utilizing the YOLOv8 classification. The detected data is then relayed to the charging_port_perception_node, which publishes messages containing both charging port detection results and indicative information for other nodes in the system. Moving to step 2, the selection of a charging gun is contingent upon the type of port identified by YOLOv8. In step 3, the chosen charging gun undergoes picking by the ACR (Automated Charging Robot) utilizing its end effector directly from the station. Step 4 involves a meticulous alignment check by the sensor_msgs to ensure precise orientation of the motors. Subsequently the data is transmitted to align the charging gun accordingly, followed by motor adjustments guided by positional and acceleration data from the motion planning node (MoveIt). Upon reaching step 5, a meticulous re-evaluation of the charging gun's alignment occurs, with continuous feedback messages generated. These messages dynamically update during the alignment check, informing the motion planning node for optimal path creation and ACR movement.

In the concluding step 6, as the charging gun approaches within 0.15 meters, the initiation of IR tracking and alignment takes place. Ultimately, the charging gun is seamlessly plugged into the port, commencing the charging process.

V. RESULTS AND DICUSSION

After utilizing Ultralytics YOLOv8.0.20 in conjunction with Python-3.10.12 and torch-2.1.0+cu118 on a Tesla T4 GPU with 15102MiB memory, our charging port detection model has been successfully trained. The model summary indicates its complexity, featuring 168 layers and a total of 11,128,680 parameters. During the evaluation phase, the model demonstrated remarkable accuracy across various charging port classes. The model achieved precision (P) at 0.927 and a recall (R) of 0.947, resulting in a high mean Average Precision (mAP50) score of 0.930 and a mAP50-95 of 0.545.

Furthermore, when analyzing specific charging port types, the model exhibited outstanding performance. For CCS1 charging ports, the precision reached 0.990, with a perfect recall of 0.875, leading to a mAP50 score of 0.971 and a mAP50-95 of 0.423. For Type1 charging ports, the precision reached 0.988, with a perfect recall of 1.000, leading to a mAP50 score of 0.995 and a mAP50-95 of 0.622. Similarly, for CHAdeMO charging ports, the model achieved a precision of 0.905 and a recall of 0.750, resulting in a mAP50 score of 0.830 and a mAP50-95 of 0.419. GB-T charging ports displayed impressive precision at 0.990, with a recall of 0.933, leading to a mAP50 score of 0.925 and a mAP50-95 of 0.577. The Tesla's charging ports exhibited a precision of 0.838 and a recall of 0.882, resulting in a mAP50 score of 0.879 and a mAP50-95 of 0.562. Lastly, Type2 charging ports demonstrated precision at 0.861 and a recall of 0.842, yielding a mAP50 score of 0.917 and a mAP50-95 of 0.504. The model's speed was also noteworthy, with a pre-processing time of 0.4ms, an inference time of 7.8ms, and minimal time spent on loss calculation and post-processing, totaling 1.0ms per image.

Despite having significantly fewer parameters, the YOLOv8model excels in performance and maintains remarkable inference speed, specifically tailored for our dataset consisting of seven classes. Moreover, Figure 22 visually demonstrates our model's detection results for all seven classes, showcasing its effectiveness and accuracy in object

FIGURE 22. Predicted output of our proposed model for different input.

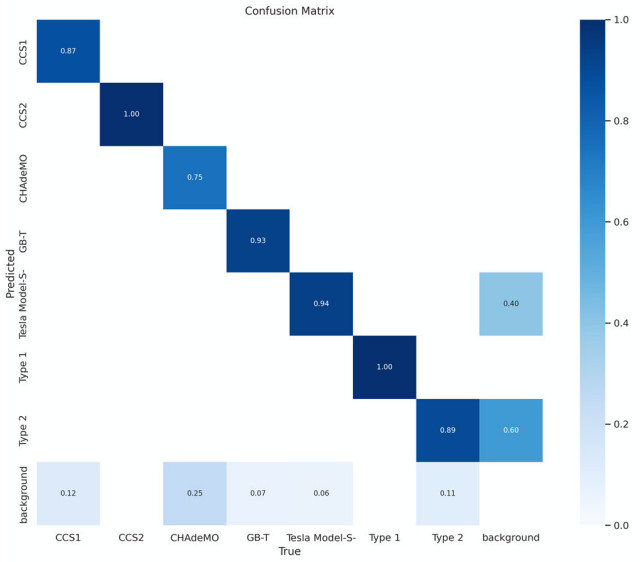


FIGURE 23. Confusion matrix for different classes.

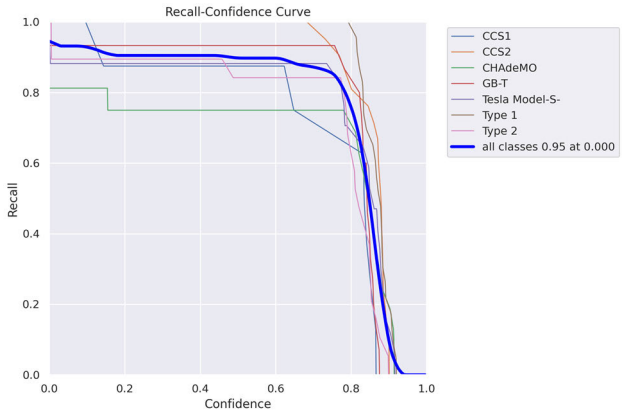


FIGURE 24. Recall-Confidence curve of our proposed model.

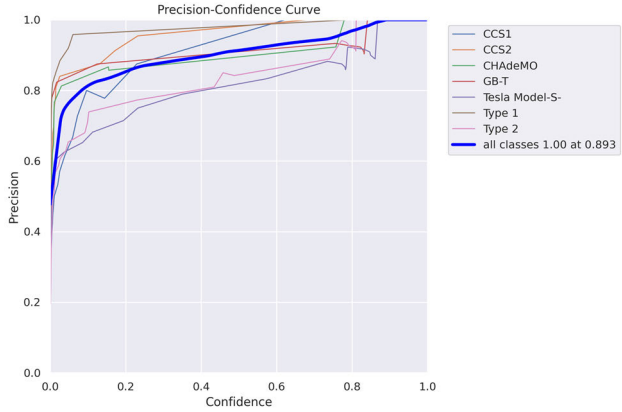


FIGURE 25. Precision-Confidence curve of our proposed model.

recognition without direct comparison to other models. The evaluation metrics, encompassing precision, recall, mAP, and F1 score, offer a comprehensive overview of YOLOv8's exceptional performance in object detection.

A confusion matrix is a comprehensive table that dissects a model's predictions, cross-referencing them with the true classes in a classification task. It provides a detailed breakdown of performance metrics, such as true positives, true negatives, false positives, and false negatives. This matrix is particularly valuable for assessing the model's effectiveness on a class-by-class basis, offering insights into specific strengths and weaknesses. For our proposed model, Figure 23 likely visually presents this matrix, offering a detailed overview of how well the model performs across different classes.

The graph below Figure 24 depicts the recall confidence curve of our model. This curve shows how the model's ability to correctly identify relevant instances changes as the confidence threshold for predictions varies. It helps in finding the right balance between precision and recall by indicating the model's performance at different confidence levels.

Figure 25 represents the precision-confidence curve of our model. This curve demonstrates how the precision rate varies with different confidence thresholds for predictions.

Figure 26 represents the precision-recall curve delineates the compromise between precision and recall across varying thresholds in a classification model, highlighting the nuanced relationship between accurately predicted positive instances and the comprehensive identification of all actual positives.

The YOLOv8 model exhibited robust performance across multiple classes, with notable F1 scores for each category. Specifically, it achieved a F1 score of 0.820 for CHAdeMO class, showcasing a balanced precision and recall. The Type2 class demonstrated exceptional accuracy with an F1 score of 0.851. Impressively, the model excelled in recognizing

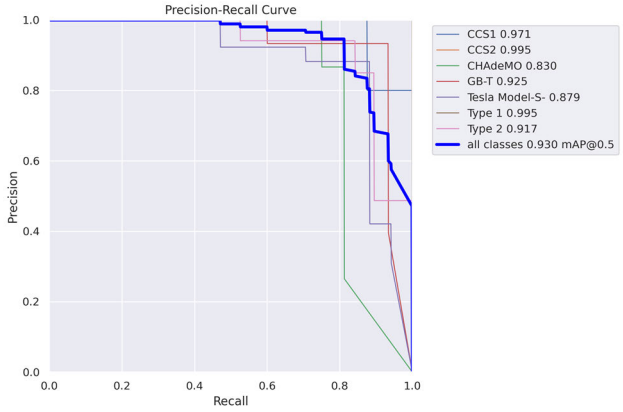


FIGURE 26. Precision-Recall curve of our proposed model.

Type1 instances, attaining an outstanding F1 score of 0.993, highlighting its superior precision and recall rates. Similarly, the CCS1 class was accurately identified with an F1 score of 0.928, underlining the model's reliability in this category. Although slightly lower, the F1 score of 0.995 for CCS2 class still reflected a strong balance between precision and recall. The Tesla Model(S) class showed good performance with an F1 score of 0.859, suggesting room for improvement, possibly by refining the model to reduce false positives or negatives. Lastly, the GB/T class exhibited near-perfect precision and recall, boasting an exceptional F1 score of 0.960, indicating the model's remarkable ability to identify GB/T instances with minimal errors. Overall, the collective F1 score of 0.912 underscores the model's strong overall performance, affirming its reliability and effectiveness in diverse detection tasks as shown in Figure 27.

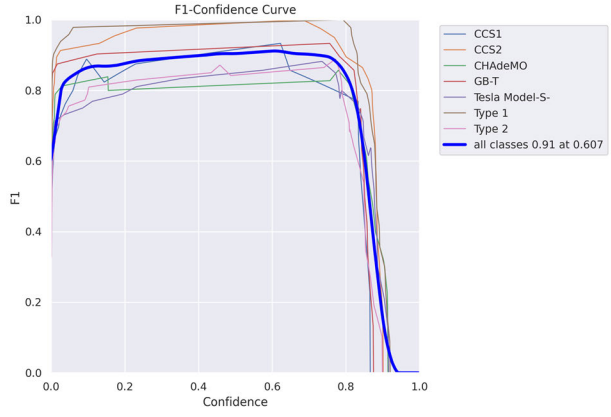


FIGURE 27. F1% score of our proposed model.

SOTA within the realm of Machine Learning denotes "State-of-the-Art." In the context of ML, a State-of-the-Art assessment involves evaluation and benchmarking of a machine learning model or algorithm against the current best-known performance in a specific task or domain. Table 4 provides a comprehensive State-of-the-Art (SOTA) comparison, evaluating the performance of various models

TABLE 4. SOTA analysis.

Models	Precision	Recall	mAP (0.50)	mAP (0.5-0.95)	F1 Score
YOLOv8	0.927	0.898	0.930	0.545	0.912
YOLOv7	0.917	0.829	0.913	0.532	0.870
YOLOv6	0.911	0.799	0.901	0.544	0.851
YOLOv5	0.909	0.783	0.897	0.519	0.841

in charging port detection applications. Notably, our model, built on the YOLOv8 architecture, surpasses the others, showcasing the highest precision, recall, and F1 score. This underscores the efficacy of our model in efficiently detecting charging ports, establishing a clear advantage over alternative architectures. Examining the second row, the YOLOv7 model exhibits commendable performance metrics with a precision of 91.7%, recall of 82.9%, mAP50 of 91.3% and F1 score of 87%. While it demonstrates improved performance compared to its predecessors; it does not attain the level of excellence achieved by our proposed YOLOv8 model. Moving to the third row, the YOLOv6 model presents precision, recall, mAP and F1 values of 91.1%, 79.9%,90.1% and 85.1% respectively. Despite its modest performance, it outperforms the YOLOv5 model across all parameters, showcasing its efficacy in charging port detection. The final row outlines the performance metrics of the YOLOv5 model, recording a precision of 90.9%, recall of 78.3%, mAP of 0.897% and F1 score of 84.1%. Though showing improved performance over its predecessors, the proposed YOLOv8 model still outperforms it significantly.

Each model's performance can be attributed to its architectural intricacies and feature sets. The YOLOv8 architecture, employed in our proposed model, integrates advanced features and optimizations, resulting in superior precision, recall, and F1 score. YOLOv7, while commendable, falls short of the heightened capabilities embedded in YOLOv8. YOLOv6, with its baseline performance, excels compared to YOLOv5, showcasing the incremental advancements in each successive version. A simple bar graph representation of SOTA analysis shows how the YOLOv8 model performed better than in comparison to the other model in Figure 28. The observed performance differences underscore the critical role of architecture and version in influencing the outcomes of charging port detection models.

A. ERROR CLASSIFICATION FOR OUR PROPOSED YOLOv8 MODEL

Upon conducting error analysis for our proposed YOLOv8 model, the results are depicted in Figure 29. The findings reveal insights into various aspects of model performance, including Classification Error (CLS), Localization Error (LOC), both CLS & LOC errors, Duplicate Detection Error (DUP), Background Error (BKG), and Missed Target Error (MISS). These outcomes provide valuable

TABLE 5. Obtained value of Xposition, Yposition, Zposition and Roll, Yaw, Pitch for each step of charging procedure.

Step. No	$X_{position}$	$Y_{position}$	$Z_{position}$	Roll	Pitch	Yaw	q1	q2	q3	q4	q5	q6
Step1	0.387	-0.246	0.695	159.6	38.7	41.0	128.5	103.3	-50.1	89.9	9.9	-48.7
Step2	0.513	-0.405	0.333	122.9	47.7	103.1	128.5	118.5	-19.6	89.9	9.9	-48.7
Step3	0.630	-0.552	-0.025	159.3	18.7	145.1	128.5	164.2	-52.2	170.0	-29.6	-71.2
Step4	0.105	0.711	0.490	8.2	27.2	-94.3	-110.4	137.9	-72.9	166.1	-29.6	-71.2
Step5	0.152	0.839	-0.069	-1.3	-2.0	-35.5	-110.4	171.8	-61.5	166.1	77.5	-71.2
Step6	0.149	0.829	-0.147	79.2	17.2	-157.2	-110.4	177.2	-61.5	166.1	-38.0	-18.7

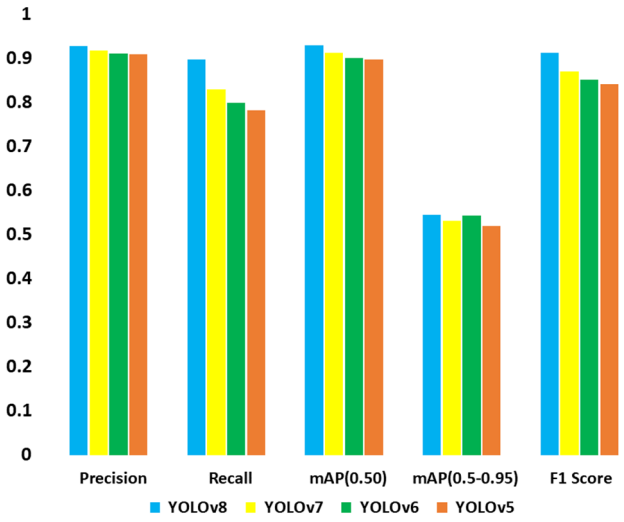


FIGURE 28. Bar graph representation of SOTA analysis.

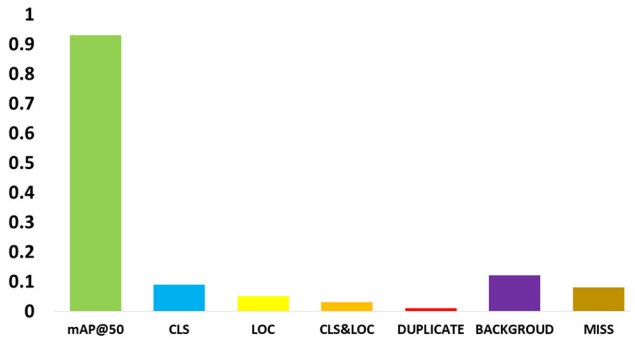


FIGURE 29. mAP value for IoU of 50% (mAP@50) and the impact of each type of error.

information for refining and enhancing the YOLOv8 model's accuracy and robustness in object detection tasks.

B. STIMULATION AND VISUALIZATION OF ACR POSITION DURING CHARGING PROCESS

The arm stimulation in the research experiment was executed utilizing the Gazebo simulation software within the ROS 2 Humbel framework, a significant iteration of ROS 2, which introduces improvements in real-time capabilities, communication middleware, and enhanced support for microcontrollers. Its key features include the DDS (Data Distribution Service) middleware, facilitating efficient communication, and the incorporation of Real-Time Publish-Subscribe (RTPS) protocol, enhancing reliability in robotic systems. ROS 2 was installed and operated on the Ubuntu version 23.04 having a hardware specifications Intel i7 11th generation processor, 16GB RAM, and 1TB HDD memory space. The simulation incorporated crucial ROS 2 packages, including pcl_ros for point cloud processing, sensor_msgs for sensor data communication, moveit for motion planning, and trajectory_msgs for conveying trajectory information. We have also employed RViz, an advanced 3D visualization tool, to dynamically visualize and analyze both robotic sensor data and the performance of our designed system. Figure 30 below vividly illustrates the intricate steps involved in the charging sequence executed by ACR.

In our analysis, we've recorded the XYZ coordinates of the ACR and tabulated the corresponding yaw, pitch, roll, and degrees of freedom (DOF) values in radians. Refer to Table No. 5 for a succinct summary of the ACR's positional and orientational information. Through advanced ROS stimulation and meticulous data analysis, we gained comprehensive insights into the ACR's movements and positioning accuracy throughout the charging procedure. This research not only enhances our understanding of ACR behavior but also contributes significantly to the development of precise and efficient charging technologies in the realm of automated robotics.

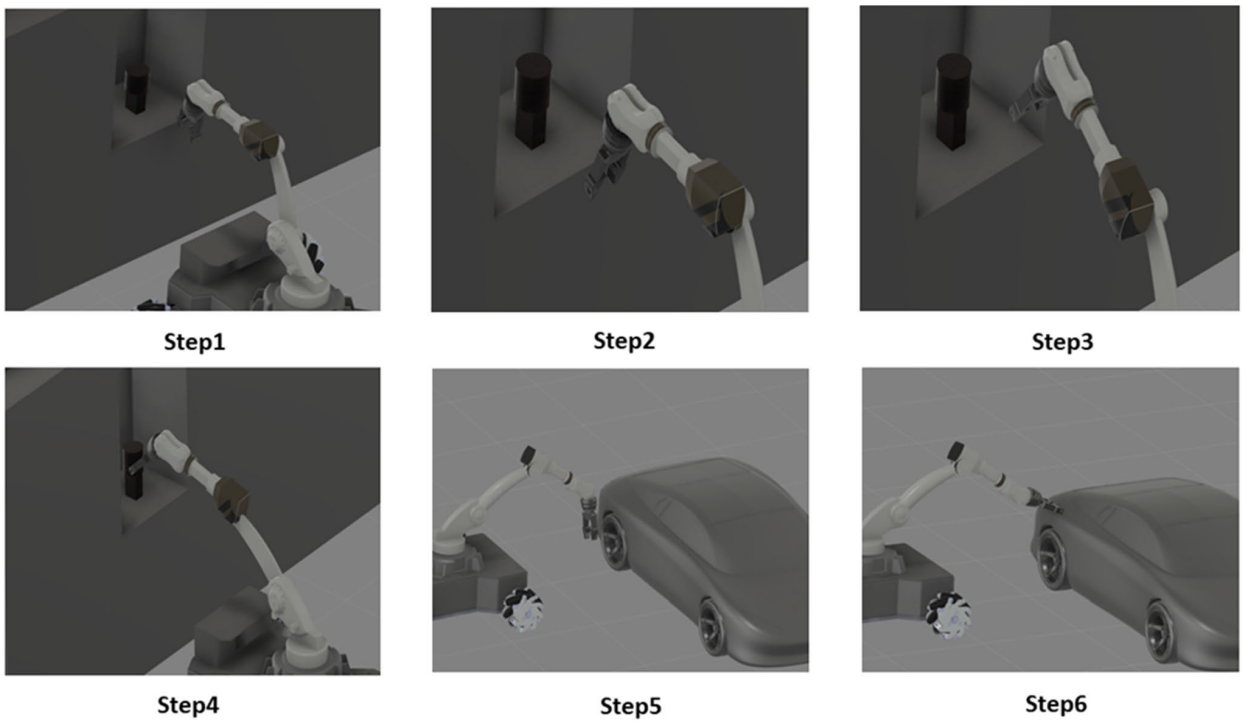


FIGURE 30. Stimulated and analyzed images of our proposed EVs charging method.

VI. CONCLUSION AND FUTURE WORKS

In the course of our research, the development and implementation of the YOLOv8 Model yielded a commendable accuracy rate of 93%, demonstrating its ability to proficiently identify diverse charging port types including CCS2, CCS1, GB/T, Type1, Type2, CHAdeMO, and Tesla. Our investigations extended beyond theoretical algorithms, incorporating practical simulations through ROS in Gazebo. This hands-on approach provided valuable insights into the charging behaviors of the Automated Charging Robot (ACR).

Additionally, we explored the mechanical intricacies of the ACR, calculating both Forward and Inverse Kinematics, with AViTRoN serving as the driving force behind these operations. Notably, AViTRoN showcased its adaptability not only in charging applications but also in controlling various Automated Vehicles and Drones. The significance of improving autonomous Electric Vehicle (EV) charging methods cannot be overstated. As societies globally transition towards sustainable energy solutions, EVs have emerged as a vital component of this eco-conscious shift. The seamless integration of EVs into daily life requires addressing challenges related to charging infrastructure, time efficiency, and user convenience. Empowering EVs to autonomously navigate to charging stations, plug in, and initiate charging without human intervention not only improves the user experience but also optimizes the efficiency of EV usage. Additionally, this technology plays a vital role in facilitating the incorporation of EVs into smart grids, harnessing the potential of renewable energy sources. While our research marks significant progress, the pursuit of perfecting autonomous EV charging methods is ongoing. The integration of AViTRoN into ACR alongside Automated Guided Vehicles (AGVs) remains a priority. This real-world implementation will validate AViTRoN's applicability, enhancing the capabilities of ACR in automated charging scenarios. Further optimization of the YOLOv8 Model, ensuring its reliability across diverse environmental conditions, stands as a key area of focus. In the future work, YOLO-NAS will extend its research beyond charging applications to explore broader horizons, encompassing autonomous navigation and adaptive decision-making, promising impactful advancements in the field.

REFERENCES

[1] A. Wijesekera and P. Binduhewa, "Design of a small electric vehicle charger with multi-ports," in *Proc. 2nd Int. Conf. Electr. Eng. (EECon)*, Sep. 2018, pp. 8–13.

[2] P. H. Gadagi, V. V. Mahadik, and S. N. Hon, "Development of an automated charging solution for electric vehicles," *Eur. Chem. Bull.*, vol. 12, no. S3, pp. 4015–4026, 2023.

[3] V. Tadic, "Study on automatic electric vehicle charging socket detection using ZED 2i depth sensor," *Electronics*, vol. 12, no. 4, p. 912, Feb. 2023.

[4] M. S. Mastoi, S. Zhuang, H. M. Munir, M. Haris, M. Hassan, M. Usman, S. S. H. Bukhari, and J.-S. Ro, "An in-depth analysis of electric vehicle charging station infrastructure, policy implications, and future trends," *Energy Rep.*, vol. 8, pp. 11504–11529, Nov. 2022.

[5] M. Pan, C. Sun, J. Liu, and Y. Wang, "Automatic recognition and location system for electric vehicle charging port in complex environment," *IET Image Process.*, vol. 14, no. 10, pp. 2263–2272, Aug. 2020.

[6] P. Quan, Y. Lou, H. Lin, Z. Liang, D. Wei, and S. Di, "Research on identification and location of charging ports of multiple electric vehicles based on SFLDLC-CBAM-YOLOV7-Tinp-CTMA," *Electronics*, vol. 12, no. 8, p. 1855, Apr. 2023.

[7] P. Quan, Y. Lou, H. Lin, Z. Liang, and S. Di, "Research on fast identification and location of contour features of electric vehicle charging port in complex scenes," *IEEE Access*, vol. 10, pp. 26702–26714, 2022.

[8] R. Rothe, M. Guillaumin, and L. Van Gool, "Non-maximum suppression for object detection by passing messages between windows," in *Computer Vision—ACCV 2014*, vol. 12. Cham, Switzerland: Springer, 2015, pp. 290–306.

[9] V. L. N. S. Raja, P. Selvaraj, G. Adwaita, K. Kalyani, K. V. S. Bhaskar, and J. A. Kamalakar, "Optical alignment procedure for position optimization of an IR detector used in lunar laser range instrument (LLRI)," *Signal*, vol. 1, no. 2, p. 3, 2007.

[10] Y. Wang, Y. Wen, Q. Zhu, J. Luo, Z. Yang, S. Su, X. Wang, L. Hao, J. Tan, H. Yin, and Y. Ge, "Real driving energy consumption and CO_2 & pollutant emission characteristics of a parallel plug-in hybrid electric vehicle under different propulsion modes," *Energy*, vol. 244, Apr. 2022, Art. no. 123076.

[11] V. D. N. Santos, J. P. Trovao, T. P. Branco, and J. M. R. Goncalves, "Information and communication technology solution for the V2G concept implementation," in *Proc. IEEE Vehicle Power Propuls. Conf. (VPPC)*, Oct. 2014, pp. 1–6.

[12] M. Mastrofini, G. Goracci, I. Agostinelli, and F. Curti, "YOLO v4 based algorithm for resident space object detection and tracking," in *Proc. Int. Conf. Appl. Intell. Inform.* Cham, Switzerland: Springer, 2022, pp. 19–33.

[13] M. Behl, J. DuBro, T. Flynt, I. Hameed, G. Lang, and F. Park, "Autonomous electric vehicle charging system," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2019, pp. 1–6.

[14] T. Li, C. Xia, M. Yu, P. Tang, W. Wei, and D. Zhang, "Scale-invariant localization of electric vehicle charging port via semi-global matching of binocular images," *Appl. Sci.*, vol. 12, no. 10, p. 5247, May 2022.

[15] F. Zhang, Q. Yang, and D. An, "CDDPG: A deep-reinforcement-learning-based approach for electric vehicle charging control," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3075–3087, Mar. 2021.

[16] S. M. Asad, J. Ahmad, S. Hussain, A. Zoha, Q. H. Abbasi, and M. A. Imran, "Mobility prediction-based optimisation and encryption of passenger traffic-flows using machine learning," *Sensors*, vol. 20, no. 9, p. 2629, May 2020.

[17] T. Ahmad, R. Madonski, D. Zhang, C. Huang, and A. Mujeeb, "Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm," *Renew. Sustain. Energy Rev.*, vol. 160, May 2022, Art. no. 112128.

[18] P. Gao, X.-Y. Zhang, X.-L. Yang, F. Gao, H. Fujita, and F. Wang, "Robust visual tracking with extreme point graph-guided annotation: Approach and experiment," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122013.

[19] P. Gao, Y. Ma, K. Song, C. Li, F. Wang, L. Xiao, and Y. Zhang, "High performance visual tracking with circular and structural operators," *Knowledge-Based Syst.*, vol. 161, pp. 240–253, Dec. 2018.

[20] P. Gao, R. Yuan, F. Wang, L. Xiao, H. Fujita, and Y. Zhang, "Siamese attentional keypoint network for high performance visual tracking," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105448.

[21] T. Zonta, C. A. da Costa, F. A. Zeiser, G. de Oliveira Ramos, R. Kunst, and R. da Rosa Righi, "A predictive maintenance model for optimizing production schedule using deep neural networks," *J. Manuf. Syst.*, vol. 62, pp. 450–462, Jan. 2022.

[22] S. M. Shariff, M. S. Alam, F. Ahmad, Y. Rafat, M. S. J. Asghar, and S. Khan, "System design and realization of a solar-powered electric vehicle charging station," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2748–2758, Jun. 2020.

[23] O. S. Buchholz, A. G. J. van der Ham, R. Veneman, D. W. F. Brilman, and S. R. A. Kersten, "Power-to-gas: Storing surplus electrical energy. A design study," *Energy Proc.*, vol. 63, pp. 7993–8009, Jan. 2014.

[24] W. Alharbi and K. Bhattacharya, "Electric vehicle charging facility as a smart energy microhub," *IEEE Trans. Sustain. Energy*, vol. 8, no. 2, pp. 616–628, Apr. 2017.

[25] M. A. Yousuf, T. K. Das, M. E. Khallil, N. A. A. Aziz, M. J. Rana, and S. Hossain, "Comparison study of inductive coupling and magnetic resonant coupling method for wireless power transmission of electric vehicles," in *Proc. 2nd Int. Conf. Robot., Electr. Signal Process. Techn. (ICREST)*, Jan. 2021, pp. 737–741.

[26] F. Mwasilu, J. J. Justo, E.-K. Kim, T. D. Do, and J.-W. Jung, "Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration," *Renew. Sustain. Energy Rev.*, vol. 34, pp. 501–516, Jun. 2014.

[27] S.-A. Amamra and J. Marco, "Vehicle-to-Grid aggregator to support power grid and reduce electric vehicle charging cost," *IEEE Access*, vol. 7, pp. 178528–178538, 2019.

[28] S. Das, P. Acharjee, and A. Bhattacharya, "Charging scheduling of electric vehicle incorporating grid-to-vehicle (G2V) and vehicle-to-grid (V2G) technology in smart-grid," in *Proc. IEEE Int. Conf. Power Electron., Smart Grid Renew. Energy (PESGRE)*, Jan. 2020, pp. 1–6.

[29] A. Hashmi and M. T. Gul, "Integrating E-vehicle into the power system by the execution of vehicle-to-grid (V2G) terminology—A review," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Feb. 2018, pp. 1–5.

[30] D. Wang, M. Sechilariu, and F. Locment, "PV-powered charging station for electric vehicles: Power management with integrated V2G," *Appl. Sci.*, vol. 10, no. 18, p. 6500, Sep. 2020.

[31] Y. Cao et al., "A holistic review on E-mobility service optimization: Challenges, recent progress and future directions," *IEEE Trans. Transport. Electrific.*, doi: 10.1109/TTE.2023.3311410.

[32] M. A. Quddus, M. Kabli, and M. Marufuzzaman, "Modeling electric vehicle charging station expansion with an integration of renewable energy and vehicle-to-grid sources," *Transp. Res. E, Logistics Transp. Rev.*, vol. 128, pp. 251–279, Aug. 2019.

[33] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A modified YOLOv8 detection network for UAV aerial image recognition," *Drones*, vol. 7, no. 5, p. 304, May 2023.

[34] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6.

[35] H. Zhao, J. Jin, Y. Liu, Y. Guo, and Y. Shen, "FSDF: A high-performance fire detection framework," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121665.

[36] M. H. F. Afonso, E. H. Teixeira, M. R. Cruz, G. P. Aquino, and E. C. V. Boass, "Vehicle and plate detection for intelligent transport systems: Performance evaluation of models YOLOv5 and YOLOv8," 2023, doi: 10.13140/RG.2.2.11022.95042.

[37] V. C. Mahaadevan, R. Narayanamoorthi, R. Gono, and P. Moldrik, "Automatic identifier of socket for electrical vehicles using SWIN-transformer and SimAM attention mechanism-based EVS YOLO," *IEEE Access*, vol. 11, pp. 111238–111254, 2023.

[38] E. Güney, C. Bayılmış, S. Çakar, E. Erol, and Ö. Atmaca, "Autonomous control of shore robotic charging systems based on computer vision," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122116.

[39] S. Quan, H. He, J. Chen, Z. Zhang, R. Han, and Y.-X. Wang, "Health-aware model predictive energy management for fuel cell electric vehicle based on hybrid modeling method," *Energy*, vol. 278, Sep. 2023, Art. no. 127919.

[40] Z. Xu, W. Su, Z. Hu, Y. Song, and H. Zhang, "A hierarchical framework for coordinated charging of plug-in electric vehicles in China," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 428–438, Jan. 2016.

[41] J. Miseikis, M. Ruther, B. Walzel, M. Hirz, and H. Brunner, "3D vision guided robotic charging station for electric and plug-in hybrid vehicles," 2017, *arXiv:1703.05381*.

[42] M. Shibl, L. Ismail, and A. Massoud, "Electric vehicles charging management using machine learning considering fast charging and vehicle-to-grid operation," *Energies*, vol. 14, no. 19, p. 6199, 2021.

[43] M. ElKashlan, M. S. Elsayed, A. D. Jurcut, and M. Azer, "A machine learning-based intrusion detection system for IoT electric vehicle charging stations (EVCSs)," *Electronics*, vol. 12, no. 4, p. 1044, Feb. 2023.

[44] K. Park and I. Moon, "Multi-agent deep reinforcement learning approach for EV charging scheduling in a smart grid," *Appl. Energy*, vol. 328, Dec. 2022, Art. no. 120111.

[45] L. Shi, Y. Hao, S. Lv, L. Cipcigan, and J. Liang, "A comprehensive charging network planning scheme for promoting EV charging infrastructure considering the chicken-eggs dilemma," *Res. Transp. Econ.*, vol. 88, Sep. 2021, Art. no. 100837.

[46] H. Lin, P. Quan, Z. Liang, Y. Lou, D. Wei, and S. Di, "Collision localization and classification on the end-effector of a cable-driven manipulator applied to EV auto-charging based on DCNN–SVM," *Sensors*, vol. 22, no. 9, p. 3439, Apr. 2022.

[47] M. Jiang, T. Hai, Z. Pan, H. Wang, Y. Jia, and C. Deng, "Multi-agent deep reinforcement learning for multi-object tracker," *IEEE Access*, vol. 7, pp. 32400–32407, 2019.

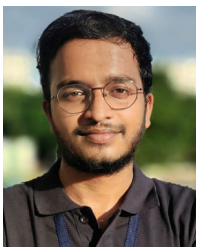
[48] G. B. P. Barbosa, E. C. Da Silva, and A. C. Leite, "Vision-based autonomous crop row navigation for wheeled mobile robots using super-twisting sliding mode control," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Aug. 2021, pp. 1–6.

[49] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[50] K. Ren, X. Chen, Z. Wang, X. Liang, Z. Chen, and X. Miao, "HAM-transformer: A hybrid adaptive multi-scaled transformer net for remote sensing in complex scenes," *Remote Sens.*, vol. 15, no. 19, p. 4817, 2023.
[51] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios," *Sensors*, vol. 23, p. 7190, 2023, doi: 10.3390/s23167190.
[52] S. A. Mahaboobunisa, S. Gayathri, Y. K. Babu, and V. S. Kumar, "A novel approach for using common objects in context dataset (COCO) and real time object detection using ML," *Int. J. Innov. Eng. Manag. Res., Forthcoming*, 2023.
[53] K. Li, Z. Huang, Y.-C. Cheng, and C.-H. Lee, "A maximal figure-of-merit learning approach to maximizing mean average precision with deep neural network based classifiers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4503–4507.
[54] D. Lu, J. Ye, Y. Wang, and Z. Yu, "Plant detection and counting: Enhancing precision agriculture in UAV and general scenes," *IEEE Access*, vol. 11, pp. 116196–116205, 2023.
[55] T. Andre, D. Neuhold, and C. Bettstetter, "Coordinated multi-robot exploration: Out of the box packages for ROS," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 1457–1462.
[56] A. Manzalini and N. Crespi, "An edge operating system enabling anything-as-a-service," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 62–67, Mar. 2016.
[57] A. Santos, A. Cunha, and N. Macedo, "Static-time extraction and analysis of the ROS computation graph," in *Proc. 3rd IEEE Int. Conf. Robotic Comput. (IRC)*, Feb. 2019, pp. 62–69.
[58] B. Andres, P. Obermeier, O. Sabuncu, T. Schaub, and D. Rajaratnam, "ROSoClingo: A ROS package for ASP-based robot control," 2013, *arXiv:1307.7398*.
[59] D. M. Turnage, "Simulation results for localization and mapping algorithms," in *Proc. Winter Simul. Conf. (WSC)*, Dec. 2016, pp. 3040–3051.
[60] L. Sobczak, K. Filus, J. Domanska, and A. Domanski, "Finding the best hardware configuration for 2D SLAM in indoor environments via simulation based on Google Cartographer," *Sci. Rep.*, vol. 12, no. 1, p. 18815, 2022.
[61] I. Wasisto, N. Istiqomah, I. K. N. Trisnawan, and A. N. Jati, "Implementation of mobile sensor navigation system based on adaptive Monte Carlo localization," in *Proc. Int. Conf. Comput., Control, Informat. its Appl. (IC3INA)*, Oct. 2019, pp. 187–192.
[62] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012, doi: 10.1109/MRA.2012.2205651.
[63] D. Portugal, L. Iocchi, and A. Farinelli, "A ROS-based framework for simulation and benchmarking of multi-robot patrolling algorithms," in *Robot Operating System (ROS)*, 2019, pp. 3-28.
[64] G. Deng, Y. Zhou, Y. Xu, T. Zhang, and Y. Liu, "An investigation of Byzantine threats in multi-robot systems," in *Proc. 24th Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 17–32, doi: 10.1145/3471621.3471867.
[65] S. Moon, J. J. Bird, S. Borenstein, and E. W. Frew, "A Gazebo/ROS-based communication-realistic simulator for networked sUAS," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 1819–1827.
[66] *Jetson Nano | NVIDIA Developer*. Accessed: Oct. 27, 2023. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano
[67] A. Vats and D. C. Anastasiu, "Enhancing retail checkout through video inpainting, YOLOv8 detection, and DeepSort tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 5529–5536.
[68] J. Chu, Y. Zhang, S. Li, L. Leng, and J. Miao, "Syncretic-NMS: A merging non-maximum suppression algorithm for instance segmentation," *IEEE Access*, vol. 8, pp. 114705–114714, 2020.
[69] X.-Y. Zhang, J. Zhang, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and Y. Li, "Kuhn–Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 695–710, Oct. 2016.
[70] Y. Wang, M. Yu, G. Jiang, Z. Pan, and J. Lin, "Image registration algorithm based on convolutional neural network and local homography transformation," *Appl. Sci.*, vol. 10, no. 3, p. 732, Jan. 2020.
[71] H. J. Loesch, "Orientation and alignment in reactive beam collisions: Recent progress," *Annu. Rev. Phys. Chem.*, vol. 46, no. 1, pp. 555–594, Oct. 1995.
[72] W. Dewandhana, K. I. Apriandy, B. S. B. Dewantara, and D. Pramadihanto, "Forward kinematics with full-arm analysis on 'T-FLoW' 3.0 humanoid robot," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2021, pp. 356–361.
[73] Y.-Q. Zheng, Q. Lin, J.-P. Wu, and P. Mitrouchev, "Analysis of inverse kinematics and dynamics of a novel 6-degree-of-freedom wire-driven parallel gantry crane robot," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2009, pp. 1786–1791.
[74] M. Surendar and P. Pradeepa, "An IoT-based battery surveillance system for E-vehicles," in *Proc. 5th Int. Conf. I-SMAC*, Nov. 2021, pp. 119–124.

V. C. MAHAADEVAN received the B.E. degree in mechanical engineering from the University of Madras, the M.B.A. degree in marketing from Pondicherry University, and the M.Tech. degree in CAD from SRM University. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Chennai. He is having more than 23 years of experience in automotive industry with quality management system, product development, and supplier chain management. He stared carrier in Tier 2 industry and moved to Hyundai ancillary for interior and seating system and currently working with Renault Nissan as a Global Benchmark Pilot and a Team Leader of the Body in White Costing Team.

R. NARAYANAMOORTHI (Member, IEEE) received the bachelor's degree in electrical engineering and the master's degree in control and instrumentation from Anna University, India, in 2009 and 2011, respectively, and the Ph.D. degree from the SRM Institute of Science and Technology, India, in 2019. He is currently an Associate Professor with the Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology. His research interests include wireless power transfer, electric vehicles, power electronics, artificial intelligence and machine learning in renewable energy systems, and embedded systems for smart sensors.



SAYANTAN PANDA is currently pursuing the bachelor's degree in electronics and communication engineering with the SRM Institute of Science and Technology. His research interests include on practical applications of embedded systems and the IoT, operating system and control mechanisms for robots, embedded Linux and AI development, and FPGA design, aiming to optimize digital circuits for efficient electronic systems.

SANKHADDEP DUTTA is currently pursuing the bachelor's degree in electronics and communication engineering from the SRM Institute of Science and Technology. His research interests include the practical implementation of embedded systems, the IoT, PCB designing, and FPGA design. His principal objective is to elevate circuit efficiency and performance through new and modified designs.

GERARD DOOLY (Member, IEEE) received the B.Eng. degree in electronic engineering from the University of Limerick, in 2003, and the Ph.D. degree from the Optical Fiber Sensors Research Centre, University of Limerick, in 2008. He is currently the Co-Director of the Centre for Robotics and Intelligent Systems, University of Limerick, leading a number of research projects in the area of sensing and perception systems for field robotics. His research interests include computer vision for aerial and subsea domains, with real in field deployments in Ireland and the Atlantic vision for aerial and subsea domains, with real in field deployments in Ireland and the Atlantic.