Artificial Intelligence Based Navigation System

Ashika S[1],Dr. R. Suja Mani Malar[2],Dr. G. Kulanthaivel[3], Dr. G. A. Rathy[4], Dr. P. Sivasankar[5]
[1, 2, 3, 4, 5] Dept of Electronics and Communication Engineering
[1, 2, 3, 4, 5] National Institute of Technical Teachers Training and Research, Chennai

Abstract- *In modern educational institutions, students, faculty, staff, and visitors often encounter difficulties in finding the most efficient and timely routes to various destinations across a large campus. The Artificial Intelligence Based Navigation System addresses this issue by providing a real-time solution for campus navigation. This system utilizes artificial intelligence algorithms for environment-aware pathfinding, dynamically calculating optimal routes based on the user's current location and chosen destination. Through an intuitive user interface, users can easily input their desired location and receive clear, step-by-step navigation instructions. Designed for independent use, the platform acts as a self-service tool, helping users locate essential facilities such as classrooms, libraries, administrative offices, and other campus landmarks. By offering smart navigation support, this system enhances the campus experience and contributes to the development of an intelligent, user-friendly educational infrastructure.*

Keywords- Algorithms, Pathfinding, Artificial Intelligence

I. INTRODUCTION

Modern college campuses are vibrant and intricate ecosystems, bustling with activity as they host a diverse array of students, faculty, staff, and visitors. During peak times, such as admissions events, placement drives, symposiums, and academic conferences, the hurdles of navigating an unfamiliar campus can become particularly daunting. This challenge is especially pronounced for freshmen and first-time visitors who may find themselves overwhelmed by the myriads of buildings and pathways. In these bustling contexts, the need for efficient and intelligent navigation support becomes crucial, as it not only alleviates confusion but also saves valuable time, significantly enhancing the overall experience on campus. An innovative AI-based navigation system emerges as a sophisticated, software-driven solution to address these challenges. By harnessing advanced pathfinding algorithms combined with real-time environmental awareness, this system guides users seamlessly across the campus landscape. Utilizing intuitive web or mobile interfaces, users can effortlessly input their desired destination and receive dynamic, step-by-step navigation assistance tailored to their needs. The system is designed to continuously adapt to evolving conditions, such as construction zones, closed pathways, or areas of heavy foot traffic, ensuring that users always benefit from the most efficient routing available. Beyond serving individual navigation needs, this system functions as a centralized digital information hub. It provides users with instant access to event schedules, campus alerts, and location-based announcements, creating a more informed and engaged campus community. Its inclusive design caters to users with varying levels of technical expertise, and features like voice-guided assistance or accessibility modes can be integrated to support individuals with disabilities, ensuring that everyone can navigate the campus with confidence.The implementation of such a navigation system aligns with the broader vision of a "smart campus"—an environment that weaves digital technologies into the very fabric of learning, security, and operational efficiency. By automating routine inquiries and minimizing the reliance on on-ground directional support, institutions can redistribute human resources toward more critical tasks, particularly during high-traffic periods. However, the deployment of AI-driven navigation systems also necessitates careful consideration of challenges such as data privacy, system scalability, and infrastructure limitations. Striking a balance between providing secure, reliable access to various services while maintaining user trust is paramount for fostering long-term adoption.

This paper delves into the design, development, and implementation of an AI-based campus navigation system, showcasing its technical architecture, adaptive routing capabilities, and potential positive impact on accessibility, efficiency, and user satisfaction within educational settings. The proposed system exemplifies how artificial intelligence can be harnessed to cultivate smarter, more navigable campus environments that cater to the diverse needs of both institutions and their varied user bases, with the ultimate aim of enriching the overall campus experience.

II. RELATED WORKS

This section reviews existing studies on AI-based navigation, autonomous robotics, LiDAR technology, and real-time obstacle detection. It examines prior research to establish the knowledge gap and justify the study's significance. the foremost preliminarystep for proceeding with any research work writing.

A. Literature Review

Anca Morar et al., [1] provided an overview of indoor localization techniques that use computer vision to track mobile entities like people or robots in indoor spaces. These techniques fall into two main groups: those that use a network of fixed cameras and those that use cameras on the moving entities themselves. The study covers different aspects of indoor localization, including application areas, commercial tools, existing benchmarks, and related reviews. It also examines research solutions in this field and introduces a new classification system based on various factors, such as the use of known environmental data, the types of sensing devices, the elements detected, and the localization method used. The authors categorize and discuss 70 recent image-based indoor localization methods according to this new system. They emphasize methods that provide orientation information, which is important for many applications, such as augmented reality.

Bimal Paneru et al., [2] presented a study on how probabilistic algorithms assist differential drive robots in navigating autonomously. It focuses on mapping, localization, and path planning. One key area is sensor fusion. Kalman filters combine data from gyroscopes and digital compasses for accurate pose estimation. To improve localization, the study uses Adaptive Monte Carlo Localization (AMCL). For mapping, the authors use occupancy grid mapping with LiDAR, creating a two-dimensional map that shows areas as free, occupied, or unexplored. For path planning, the study combines both global and local strategies. The A* algorithm finds the best global routes, while the Dynamic Window Approach handles local navigation more responsively. The study also integrates depth sensing with Kinect to enhance LiDAR's abilities. This helps detect non-planar obstacles and improve obstacle avoidance. Overall, this study demonstrates how these methods work well together in both simulations and real-world situations, achieving accurate navigation with minimal positional error.

Cristian MODLER et al., [3] discuss human-following algorithms that use proximity data from LIDAR, which is also useful in robot navigation. By combining depth cameras and active infrared markers, researchers have improved the accuracy of controlling indoor environments through sensor fusion. Using data from Kinect and LIDAR highlights the need for different technologies to achieve reliable and accurate autonomous navigation. Processing speed is still a limitation, but new algorithms, like the "Follow Me" navigation strategy, show promise in human-robot interaction. These methods allow robots to track moving targets while keeping a safe distance, showcasing advancements in control systems. In addition, using graphical visualization and dynamic target tracking provides an easy way to watch and improve navigation algorithms.

Daegyu Lee et al., [4] discuss two main ways to determine a robot's location: GPS-based and sensor-based methods. GPS-based methods, like the Kalman filter, can struggle in cities and inside buildings where signals are weak. On the other hand, sensor-based methods, such as Visual Odometry (VO) and LiDAR Odometry and Mapping (LOAM), are effective and accurate, especially when GPS signals are not available. The study also looks at object detection using CNN-based models, like YOLO, which help robots identify objects in real time. It addresses challenges including estimating how far away objects are. This can improve by combining data from LiDAR and cameras and using clustering algorithms to recognize and avoid obstacles in complex settings. Finally, the study highlights motion planning strategies that help robots navigate around obstacles. It mentions algorithms that can plan routes quickly in busy urban and indoor environments, using big data and communication technology to connect vehicles with infrastructure (V2I).

Minghao Liu et al.,[5] demonstrated how robots can navigate urban areas with multi-line LiDAR systems, although extensive sensor calibration is needed. Other methods, like using QR codes for indoor navigation, need prior setup, limiting flexibility. It introduces "teach and repeat," where robots learn and retrace routes from human guidance. LiDAR techniques, such as LOAM (LiDAR Odometry and Mapping), are essential for real-time mapping and help robots detect moving objects in busy environments. The research emphasizes the need for cost-effective and scalable navigation solutions, especially in complex areas like campuses, where robots navigate around pedestrians and obstacles with minimal sensors.

Nicky Zimmerman et al., [6] highlights the essential role of object-based maps in improving our understanding of scenes. These maps merge geometric and semantic information, allowing autonomous robots to navigate and interact with objects more effectively. The main goal is to create metric-semantic maps for long-term object-based localization. The method utilizes 3D object detections from monocular RGB images for two primary tasks: building detailed object-based maps and achieving precise global localization. To adapt to various environments, the study presents a technique for generating 3D annotations that enhances the 3D object detection model. The authors tested their approach in an office building and evaluated long-term localization using challenging sequences over nine months. The results demonstrate the method's effectiveness in creating robust

metric-semantic maps and its resilience to long-term environmental changes. Both the mapping algorithm and localization pipeline operate in real-time on an onboard computer. Additionally, the authors plan to provide an open-source implementation of their method using C++ and ROS.

Rabab Alayham Abbas Helmi et al., [7] created an easy-to-use app that helps new students and visitors get directions to specific places within Management and Science University. Management and Science University plans to use smart campus technologies through the "Find 4 Me Application." This app will help users find indoor locations and improve the experience for new students and visitors. Since the campus is large and lacks a navigation tool, newcomers often have trouble locating classrooms and understanding campus resources. Besides navigation, it will include features like 360-degree views of directions, venue bookings within MSU, a live chat option, and a way to provide feedback for future improvements. The study will use the Spiral Model as a research method to meet the project's goals. Findings show that the system is reliable and trustworthy. It provides a personalized experience for students and visitors while engaging them visually.

Sheryl Sharon et al., [8] addressed navigation issues on university campuses using advanced technology. It combines GPS, user-friendly design, and real-time tracking to help people find their way. Data collection involves mapping buildings and paths with GPS to create a GeoJSON dataset, which is used in a web-based map with Leaflet.js and OpenStreetMap. The Leaflet Routing Machine helps users find optimal routes by entering their start and end points. The interface features a search tool and a feedback system, with MongoDB storing user input to improve the navigation tool. It uses HTML, CSS, JavaScript, and React.js for accessibility. Testing and training encourage adoption, while ongoing maintenance and security ensure it meets users' needs.

Sithara Jeyaraj et al., [9] examined methods to improve positioning accuracy. Zhao's approach combines inertial sensors, magnetometers, and ultrasonic sensors to reduce drift errors. Indoor navigation using Wi-Fi signal strength is also useful for map creation. To enhance position and orientation accuracy, techniques like Kalman filters and sensor fusion are important. The A* (A Star) algorithm efficiently finds optimal paths and works with vision systems or sonar for obstacle detection. The use of accelerometer and magnetometer data aids real-time mapping, while photoelectric encoders help correct errors. The challenges of real-time map building are highlighted in Chen Qiu's work on automatic mobile sensing. The integration of A* algorithms for path planning and ARM Cortex M3 processors for control shows a blend of efficiency and practical use, aligning with trends in indoor navigation and service robotics.

Susovan Jana et al., [10] discussed an Android application made with the Android Software Development Kit (SDK). The app is designed to help users navigate the Jadavpur University campus and stay updated on events. It was tested thoroughly at both the main campus and the Salt Lake campus. Feedback from users showed that the app improves campus navigation and helps them learn about different activities. The application integrates Google Maps, which allows users to track their location in real-time, find the best routes, and get details about ongoing events. This study highlights a practical solution that aims to make it easier for students and visitors at Jadavpur University to navigate the campus and engage with campus events.

Taejin Kim et al., [11] addresses the challenges of autonomous navigation in complex indoor environments where GPS signals are often weak or missing. The proposed system includes several important parts: map management, localization, path planning, perception, and task planning, all designed to work smoothly in buildings with multiple floors. The research highlights major improvements in indoor localization, especially through the use of LiDAR odometry, sensor fusion, and navigation maps that combine 3D point clouds with topography. Wheeled robots face unique challenges when it comes to using elevators. The study tackles this by using perception modules that combine camera and LiDAR data and apply semantic segmentation to detect elevators effectively. A 3D route planning algorithm connects points across floors by utilizing elevator nodes to improve navigation between floors. The study also covers task planning for parcel delivery, which includes driving, operating elevators, making deliveries, and docking. The system's effectiveness was tested in a month-long field trial. It showed that the system could reliably locate itself, avoid obstacles, and complete tasks in a real building during regular business hours.

B. Summary

This literature review explores advancements in navigation technologies, focusing on the integration of robotics and AI in diverse environments, including campuses. Key studies highlight autonomous robots using GPS, LiDAR, sensor fusion, and algorithms like A* and Monte Carlo Localization for accurate navigation and obstacle avoidance. Applications range from campus guidance systems to delivery robots and indoor localization, Zsemantic mapping. Challenges include GPS signal limitations, dynamic environments, and user interaction, while issues like data

privacy and implementation costs persist. Overall, these developments demonstrate the potential for robotics to enhance navigation and user experiences in complex spaces.

III. PROPOSED METHODOLOGY

The proposed methodology, illustrated in Fig. 1, outlines a comprehensive pipeline for intelligent pathfinding on a map, structured as a sequential operational workflow. It begins with Map Preprocessing and Walkability Extraction, where traversable terrain is identified through image processing techniques such as thresholding or segmentation. This stage ensures that the algorithm focuses only on walkable areas. The next step is Graph Construction from the Grid Map, which transforms the processed map into a graph structure by defining nodes (intersections or key points) and edges (possible paths). This transformation facilitates algorithmic navigation. Following this, Coordinate Extraction from the Graph assigns spatial coordinates (in pixel format) to each node, providing a precise spatial reference system for navigation. This is succeeded by Adjacency List Generation, where the relationships between neighboring nodes are formally encoded into a data structure, typically in dictionary or list format. This step is crucial for efficient graph traversal. The core computational process occurs during the Pathfinding Algorithms Execution step, where algorithms such as Dijkstra's, A*, Breadth-First Search (BFS), and Depth-First Search (DFS) are implemented to compute the most efficient or shortest path between nodes. Once a path is determined, it is visually overlaid onto the original map in the Path Visualization Layer, allowing users to see the computed navigation route using various user interface frameworks. The methodology then progresses to Performance Evaluation and Metrics Analysis, where the different algorithms are compared based on metrics such as time complexity, path optimality, and travel distance. This comparison helps identify the most effective strategy under various scenarios. Finally, AI Model Development utilizes the performance data and environmental features to train machine learning models that can predict the optimal pathfinding strategy. This enhancement increases adaptability and automation in complex environments.

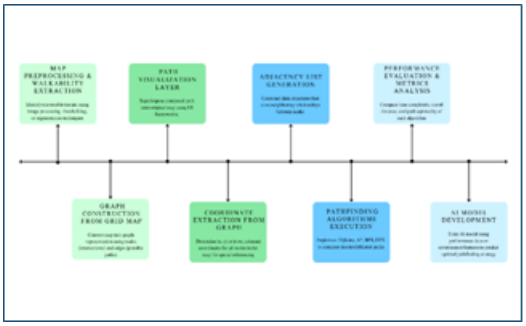

Fig. 1 Operational Workflow of the proposed work

IV. RESULTS AND DISCUSSIONS

HereThis project demonstrates an AI navigation system designed for college campuses. It identifies walkable paths using image processing and creates simple black-and-white representations. By building a network of nodes and edges, it uses the A* algorithm to find the shortest path between points. Users can select destinations on a web page, and after testing over 600 routes, the system has proven to be reliable and user-friendly.

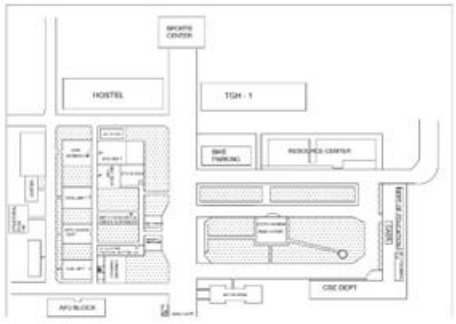

Fig. 2. Campus Map

A. Map preprocessing & Walkability Extraction

The process involves identifying the walkable routes on the campus map, which are usually highlighted in red. The system utilizes image processing techniques to isolate the paths. These red paths are then refined and reduced to single-pixel lines through a method known as skeletonization. The extracted paths are converted into a black-and-white (binary) image, where the white areas represent walkable paths, and the black areas denote obstacles or non-walkable zones, as illustrated in Fig. 3. This transformation simplifies the data for computer processing.

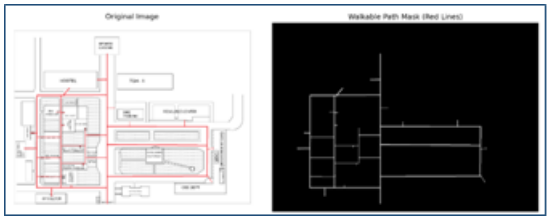

Fig. 3. Path Binarization

B. Graph Construction

The walkable areas are represented as a graph, where the nodes correspond to important points such as turns or intersections, and the edges represent the paths connecting these points. This representation helps to understand how different locations are interconnected. The nodes are identified through an automated edge detection process, while the

grouping of nodes is performed using DBSCAN clustering, with each node labeled as N1, N2, and so on. This process utilizes NetworkX to connect the nodes into a graph, where each edge denotes an actual path between two points and is weighted based on its pixel length. This transformation turns the visual map into a structured network, which can be utilized for navigation or analysis.

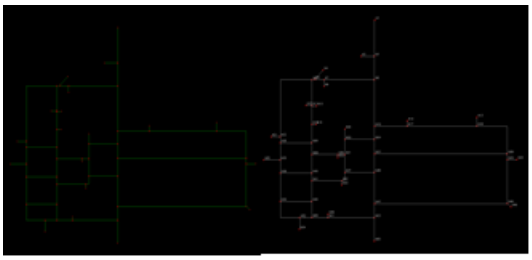

Fig. 4. Graph Construction and Node Annotation

C. Path Visualization Layer

The processed path map is overlayed onto the campus map. First, the node map is resized to match the dimensions of the site image. Both images are then converted to include transparency. Using alpha blending, the two images are combined so that the node map appears semi-transparent over the site plan. This allows both the original layout and the overlaid network of paths and labels to be visible together, enabling the user to see the exact route to follow.

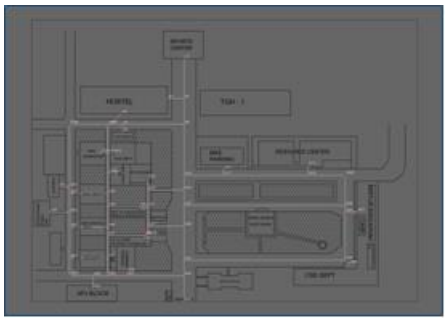

Fig. 5. Path Visualization

D. Coordinate and Adjacency List Extraction from Graph

The process involves identifying nodes using an HSV masking technique and contour detection. For each node, it calculates the centroid (x, y) coordinates. Then, it utilizes a spatial search method, specifically a KDTree, to find nearby nodes for each one. This approach creates a visual representation of how the nodes are positioned and connected on the map.

E. Pathfinding Algorithms

This process visualizes the shortest walking path between two locations on a campus map using pathfinding algorithms. It loads a file that links location names to node identifiers and another file with node coordinates. A binary path map image, where white pixels indicate walkable paths, is used to create a graph connecting walkable nodes. The algorithm retrieves start and end coordinates to find the shortest path, providing an automated visual route between landmarks.

a)Depth-First Search Algorithm: DFS is a graph traversal algorithm that explores nodes and edges by going as deep as possible along each branch before backtracking. It utilizes a stack data structure (either explicitly or through recursion) to track nodes. DFS visits unvisited neighbors until reaching a dead end, then backtracks to explore other paths. It's efficient for tasks like topological sorting, cycle detection, and solving puzzles. Its performance depends on the structure of the graph and the location of the target node.

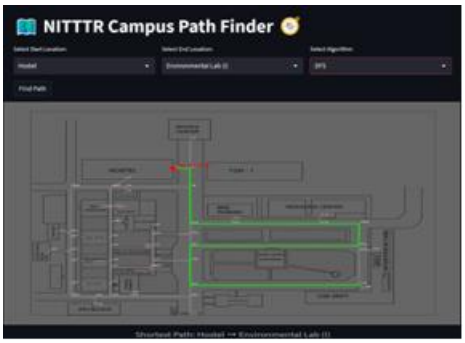

Fig. 6. Pathfinding using DFS algorithm

b)Breadth-First Search Algorithm: BFS is a graph traversal algorithm that explores all vertices in a level-wise manner, starting from a source node. It visits all neighboring nodes before moving to the next level, using a queue to track the nodes being explored. This approach ensures the shortest path (in terms of edges) from the source to any reachable node in an unweighted graph. BFS is commonly used in applications like navigation systems and social networks. However, it may consume more memory than Depth-First Search, particularly in wide graphs.

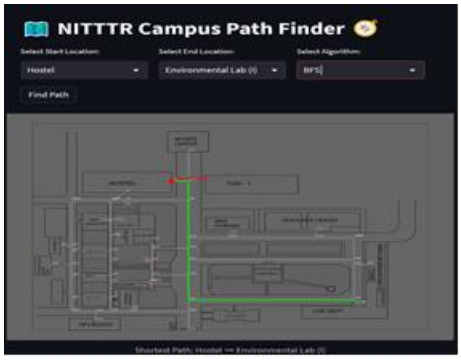

Fig. 7. Pathfinding using BFS Algorithm

c)Dijkstra's Algorithm: Dijkstra's algorithm is a graph search method that finds the shortest path from a source node to all other nodes in a weighted graph with non-negative edge weights. It maintains a set of nodes with known shortest distances, iteratively selecting the node with the smallest tentative distance and updating the distances to its adjacent nodes. This process continues until the shortest paths to all reachable nodes are found. Often utilizing a priority queue (like a min-heap), Dijkstra's algorithm is commonly used in routing and navigation systems, such as GPS and network routing protocols, for its efficiency in finding least-cost paths.

Fig. 8. Pathfinding using Dijkstra's algorithm

d)A-star Algorithm: The A* algorithm is a widely used pathfinding and graph traversal method in navigation, robotics, and game development. It combines Dijkstra's algorithm and Greedy Best-First Search to find the shortest path between two points. A* uses a priority queue to explore nodes based on the total cost function f(n) = g(n) + h(n), where g(n) is the actual cost from the start node and h(n) is a heuristic estimate to the goal. For A* to be optimal, the heuristic must be admissible, ensuring it never overestimates the true cost. This makes A* both complete and optimal, making it ideal for applications that require efficient route planning.

Fig. 9. Pathfinding using A* Algorithm

F. Performance Analysis:

The table compares four pathfinding algorithms—A*, Dijkstra, BFS, and DFS—based on the number of nodes explored and the total path distance (in pixels). A*, Dijkstra, and BFS each visit 10 nodes and find similar distances (around 1384–1386 pixels), indicating near-optimal paths. In contrast, DFS visits 17 nodes and produces a longer path (2026.45 pixels), showcasing its inefficiency for finding the shortest path. While DFS can be useful in some contexts, it is not ideal for optimized pathfinding.

TABLE I. PERFORMANCE ANALYSIS

S.NO	ALGORITHM	NODES	DISTANCE
1.	A*	10	1384.71 px
2.	Dijkstra	10	1384.71 px
3.	BFS	10	1386.7 px
4.	DFS	17	2026.45 px

G. AI model development

The System utilizes Q-learning, a reinforcement learning algorithm, to find optimal paths on a campus graph, where locations are nodes and walkable paths are edges. The Q-learning agent simulates movements from a source to a target node, using an ε-greedy strategy to explore new paths while utilizing learned ones. It receives positive rewards for reaching the goal and incurs penalties for other steps, promoting shorter routes. Over time, Q-values are updated based on the agent's experience, allowing it to determine the optimal path by following actions with the highest Q-values. This process creates a pathfinding matrix for campus landmarks, which can be used for autonomous navigation or virtual tours.

H. User Interface

Users interact with the AI-based navigation system via a clean, intuitive web control panel. This interface allows students, visitors, and faculty to easily select their starting point and destination on campus using input fields, drop-down

menus, and navigation buttons. Once a source and destination are chosen, the system relays the information to the robot's onboard AI module.

Fig. 10. User Interface

I. Navigation Result

Q-learning, a RL algorithm, trained an agent to navigate a graph from a source node to a destination. Initially, the agent explored randomly, facing a penalty for each move to encourage shorter paths and receiving a high reward upon reaching the target. Over time, it learned the most effective sequences of moves, updating its Q-table to reflect the best routes. The red path in the image represents the optimal route, as the agent selects actions with the highest Q-values. This illustrates its ability to find the shortest path, without prior knowledge of the map's layout.

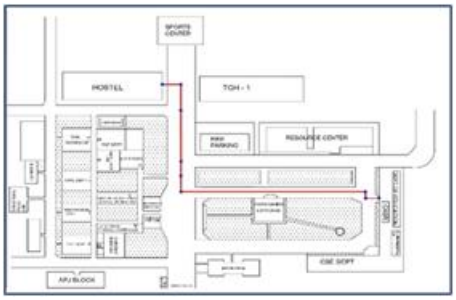

Fig. 11. Navigation Result

V. CONCLUSION

Artificial Intelligence-based navigation systems are emerging as essential tools for enhancing navigation, accessibility, and user experience across university campuses. These systems leverage advanced algorithms and machine learning techniques to deliver efficient navigation solutions, crucial for traversing the intricate layouts typical of such environments. Primarily, their functionality revolves around providing accurate navigation assistance to a diverse user base, including visitors, students, faculty, and guests, thereby facilitating seamless movement within campus grounds. Moreover, these systems have the potential to optimize campus operations by automating routine navigation tasks, leading to enhanced resourcemanagement. As advancements in artificial intelligence continue to unfold, the future of AI-driven navigation systems appears promising, suggesting significant improvements not only within educational contexts but also in other public spaces. This trajectory indicates an increasing integration of innovative technologies to enhance spatial interactions and streamline operational efficiencies.

REFERENCES

[1] Anca Morar, Alin Moldoveanu, Irina Mocanu, Florica Moldoveanu, Ion Emilian Radoi, Victor Asavei, Alexandru Gradinaru and Alex Butean (2020), 'A Comprehensive Survey of Indoor Localization Methods', National Library of Medicine, Sensors (Basel), Vol. 20, pp. 1-36.

[2] Bimal Paneru, Niraj Basnet, Sagar Shrestha, Rabin Giri and Dinesh Baniya Kshatri (2016), 'Autonomous Navigation of a Mobile Robot in Indoor Environments',Zerone Scholar, Vol. 1, No. 1, pp. 3-8.

[3] Cristian MOLDER, Daniel TOMA, and Andrei ȚIGĂU (2019), 'Navigation Algorithms with LIDAR for Mobile Robots', Journal of Military Technology, Vol. 2, No. 1, pp. 5-10.

[4] Daegyu Lee, Gyuree Kang, Boseongkim and D. Hyunchul Shim (2021), 'Assistive Delivery Robot Application for Real-World Postal Services', IEEE Access, Vol. 9, pp. 141981- 141998.

[5] Minghao Liu, Zhixing Hou, Zezhou Sun, Ning Yin, Hang Yang, Ying Wang, Zhiqiang Chu and Hui Kong (2019), 'Campus Guide: A Lidar-based Mobile Robot', European Conference on Mobile Robots (ECMR), pp. 1-6.

[6] Nicky Zimmerman, Matteo Sodano, Elias Marks, Jens Behley and Cyrill Stachniss (2023), 'Constructing Metric-Semantic Maps Using Floor Plan Priors for Long-Term Indoor Localization', IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1366-1372.

[7] Rabab Alayham Abbas Helmi, Harini A.P. Ravichandran, Arshad Jamal and M. N. Mohammed (2022), 'Design and Development of Indoor Campus Navigation Application'. IEEE Conference on Systems, Process & Control (ICSPC), pp. 77-82.

[8] Sheryl Sharon G, Rohit Vikaas P, Chanduru A, Barathkumar S, Harsha Vardhan P. and Dr. M. Mohanapriya (2023), 'Campus Navigation System', International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol. 11, pp. 1121-1127.

[9] Sithara Jeyaraj and Jincy Jose (2018), 'Indoor Navigation for Shopping Robot', International Journal for Research in Engineering Application & Management (IJREAM), pp. 316-322.

[10] Susovan Jana and Matangini Chattopadhyay (2015), 'An event-driven university campus navigation system on android platform', Applications and Innovations in Mobile Computing (AIMoC), pp. 182-187.

[11] Taejin Kim, Daegyu Lee, Gyuree Kang and D. Hyunchul Shim (2024), 'Development of an Indoor Delivery Mobile Robot for a Multi-Floor Environment', IEEE Access, Vol. 12, pp. 45202- 45215