IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-Driven Campus Assistant Robot Using Computer Vision And RAG Framework

Soji Oommen
Department of Computer Science and Engineering
Musaliar College of Engineering and Technology
Pathanamthitta, Kerala

Abstract:
This paper presents the design and implementation of an AI-driven campus assistant robot that integrates artificial intelligence, computer vision, speech processing, and embedded systems to provide interactive and context-aware assistance in educational environments. The system is developed using Python on a Raspberry Pi 5 and interfaces with an Arduino for navigation control. A Retrieval-Augmented Generation (RAG) framework powered by Google Gemini generates accurate responses from a predefined knowledge base, reducing hallucination and improving reliability. The system includes a graphical user interface built with Tkinter and real-time face recognition using OpenCV and face_recognition for personalized interaction. Voice-based communication is enabled through Speech Recognition and pyttsx3, facilitating natural human–robot interaction. A multi-threaded architecture maintains real-time responsiveness, while PySerial ensures reliable communication between processing and control units. Experimental results demonstrate that the proposed system is scalable, cost-effective, and practical for smart campus applications, including navigation, information retrieval, and personalized assistance.

Keywords— AI Assistant, Retrieval-Augmented Generation, Face Recognition, Smart Campus, Human–Robot Interaction.

INTRODUCTION

The rapid advancement of digital technologies has led to the emergence of smart campuses, where automation and intelligent systems are integrated to enhance the overall academic environment. Educational institutions are increasingly adopting artificial intelligence, Internet of Things (IoT), and robotics to improve operational efficiency, provide better services, and support interactive learning experiences. Among these technologies, AI-driven robotic assistants play a significant role in delivering real-time information, navigation support, and personalized interaction for students, staff, and visitors.

Traditional campus assistance systems, such as static information kiosks and mobile applications, often lack interactivity and real-time adaptability. They are limited in providing context-aware responses and personalized services. In contrast, intelligent robotic systems can combine multiple technologies such as computer vision, natural language processing, and embedded systems to offer dynamic and user-friendly assistance. These systems are capable of understanding user queries, recognizing individuals, and responding in a natural and efficient manner. In this context, the proposed AI-driven campus assistant robot is designed to provide an integrated solution that combines artificial intelligence, speech processing, computer vision, and hardware control. The system utilizes a Raspberry Pi 5 for processing and an Arduino for navigation control, ensuring efficient coordination between software and hardware components. A Retrieval-Augmented Generation (RAG) framework powered by Google Gemini is employed to generate accurate responses based on a predefined knowledge base, thereby minimizing incorrect or misleading outputs.

Furthermore, the system supports real-time face recognition for personalized interaction and incorporates voice-based communication to enhance usability. A graphical user interface is developed to provide an intuitive interaction platform, while a multi-threaded architecture ensures smooth and responsive system performance. Reliable communication between different modules is maintained using serial communication protocols.

The proposed system aims to address the limitations of existing campus assistance solutions by providing a scalable, cost-effective, and intelligent robotic platform. It can be deployed in various smart campus scenarios, including navigation guidance, information dissemination, and personalized user assistance, thereby improving the overall campus experience.

LITERATURE SURVEY

The development of intelligent robotic assistants has gained significant attention in smart environments, particularly in educational institutions. The integration of artificial intelligence, computer vision, and embedded systems enables real-time interaction and automation. An AI-based robotic assistant capable of interacting with users through speech and navigation was proposed by S. Thrun *et al.* [1], where probabilistic algorithms were used for human–robot interaction. G. Bradski introduced the OpenCV library for real-time computer vision applications, widely used for face detection and recognition [2]. R. Szeliski further contributed to computer vision techniques for object detection and recognition in intelligent systems [3]. P. Lewis *et al.* developed the Retrieval-Augmented Generation (RAG) framework to improve response accuracy in conversational AI systems by combining retrieval with generative models [4]. J. Devlin *et al.* introduced BERT, which significantly improved natural to create multileveled equations, it may be necessary to treat the language understanding for conversational interfaces [5]. T. Brown *et al.* presented GPT-based models for advanced language generation tasks [6]. E. Upton and G. Halfacree demonstrated the use of Raspberry Pi for low-cost embedded system development in robotics [7]. M. Banzi introduced Arduino as an open-source platform for hardware control and automation [8]. M. Quigley *et al.* developed the Robot Operating System (ROS) to support scalable robotic applications [9]. D. Jurafsky and J. Martin contributed to speech recognition and natural language processing techniques enabling effective human–machine communication [10]. Systems integrating speech recognition and text-to-speech for interactive robots were further explored in intelligent assistant applications [11]. Face recognition-based systems for personalized interaction and security were proposed in various studies, improving user-specific services in smart environments [12]. Multimodal systems combining graphical user interfaces, voice interaction, and real-time processing were developed to enhance usability and efficiency [13]. Recent research has focused on smart campus assistant robots capable of navigation, information retrieval, and personalized interaction using AI and embedded systems [14]. However, challenges such as scalability, real-time responsiveness, and integration complexity still persist. The proposed system addresses these limitations by integrating AI, RAG-based response generation, computer vision, and embedded platforms into a unified solution [15].

HARDWARE AND SOFTWARE REQUIREMENTS

A. Hardware Components

a) Raspberry Pi 5: Raspberry Pi 5 is the core hardware component used in the proposed AI-driven campus assistant robot. It is a compact single-board computer responsible for executing the main application, including artificial intelligence processing, face recognition, and speech interaction. The device runs on a Linux-based operating system and supports programming in Python. It processes data from various input devices such as the camera and microphone and generates appropriate outputs. The Raspberry Pi also includes built-in Wi-Fi and Bluetooth modules, enabling internet connectivity for accessing cloud-based services and knowledge bases. This allows the system to retrieve real-time information and generate intelligent responses using the implemented RAG framework.

Fig 1. Raspberry Pi 5

b) Arduino Microcontroller: Arduino is used as a secondary controller to handle low-level hardware operations such as motor control and navigation. It receives commands from the Raspberry Pi through serial communication and controls actuators like DC motors and motor drivers. This separation of tasks improves system efficiency and ensures real-time response for movement-related operations. The module also supports interrupt-based programming, enabling immediate response to critical events such as obstacle detection or emergency stops. Power management is another important function, as the Arduino can efficiently handle voltage regulation and ensure stable operation of connected devices.

Overall, the Arduino Microcontroller Module enhances the system's reliability, responsiveness, and efficiency by offloading hardware control tasks from the main processor and ensuring smooth, real-time interaction with physical components.

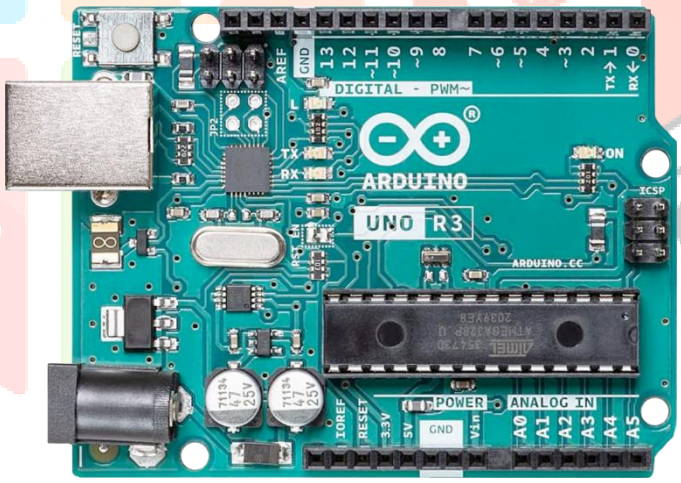

Fig 2. Arduino Microcontroller

c) Camera Module: A camera module is interfaced with the Raspberry Pi to perform real-time face detection and recognition. It captures images and video streams, which are processed using computer vision algorithms. This enables the system to identify users and provide personalized interaction, enhancing the overall user experience.

d) Microphone: A microphone is connected to the Raspberry Pi to capture voice input from the user. It enables speech recognition by converting spoken commands into digital signals. These signals are processed by the system to understand user queries and execute appropriate actions such as providing information or controlling the robot.

e) Speaker: A speaker is used to provide audio output for the system. It enables voice-based responses using text-to-speech technology, allowing natural communication between the user and the robot. This supports two-way interaction, making the system more user-friendly.

f) Motor Driver and DC Motors: Motor drivers and DC motors are used to enable movement and navigation of the robot. The motor driver acts as an interface between the Arduino and motors, controlling direction and speed based on the commands received from the processing unit.

g) Power Supply: A rechargeable battery or power supply unit is used to provide continuous power to all hardware components. Proper power management ensures stable and uninterrupted operation of the system.

B. Software Components

a) Operating System: A Linux-based operating system (such as Raspberry Pi OS) is installed on the Raspberry Pi to support system operations and application execution. It provides a stable platform for running software libraries and managing hardware resources.

b) Python Programming Language: Python is the primary programming language used in this system due to its simplicity and extensive support for artificial intelligence, machine learning, and embedded applications. It is used to implement core functionalities such as data processing, system control, and user interaction.

c) OpenCV and face_recognition: OpenCV is used for image processing and computer vision tasks such as face detection and recognition. The face_recognition library enhances accuracy in identifying users and enables personalized interaction within the system.

d) Speech Recognition: The **Speech Recognition** component forms the foundation of user interaction in the Campus Assistant Robot. By leveraging advanced speech recognition libraries, the system can accurately convert spoken voice input into text. This enables the robot to interpret user commands and queries expressed in natural language, facilitating seamless communication. The system is capable of handling diverse accents and varying speech patterns, ensuring robustness in real-world campus environments.

e) pyttsx3: The **pyttsx3** library is utilized for text-to-speech conversion, allowing the system to vocalize responses generated after processing user queries. This library supports offline TTS, ensuring that the system can operate without dependency on internet connectivity, and provides natural-sounding speech output. By integrating pyttsx3, the Campus Assistant Robot can deliver audio feedback effectively, making interactions more engaging and accessible, including for users with visual impairments.

f) Retrieval-Augmented Generation (RAG): To provide intelligent and context-aware responses, the system employs a **Retrieval-Augmented Generation (RAG)** framework powered by **Google Gemini**. The RAG framework first retrieves relevant information from a pre-defined knowledge base and then uses generative AI techniques to construct accurate, coherent, and contextually appropriate responses. This hybrid approach combines the strengths of knowledge retrieval and AI generation, enabling the system to handle complex queries with high reliability and relevance.

g) Tkinter: Tkinter is used to develop the graphical user interface (GUI) for the system. It provides an interactive platform for displaying information and system status to users.

h) PySerial: PySerial is used for serial communication between the Raspberry Pi and Arduino. It ensures reliable data transfer for coordinated functioning of hardware components.

i)Multi-threading: Multi-threading techniques are implemented to allow simultaneous execution of tasks such as speech processing, image processing, and response generation, ensuring real-time system performance.

SYSTEM ARCHITECTURE

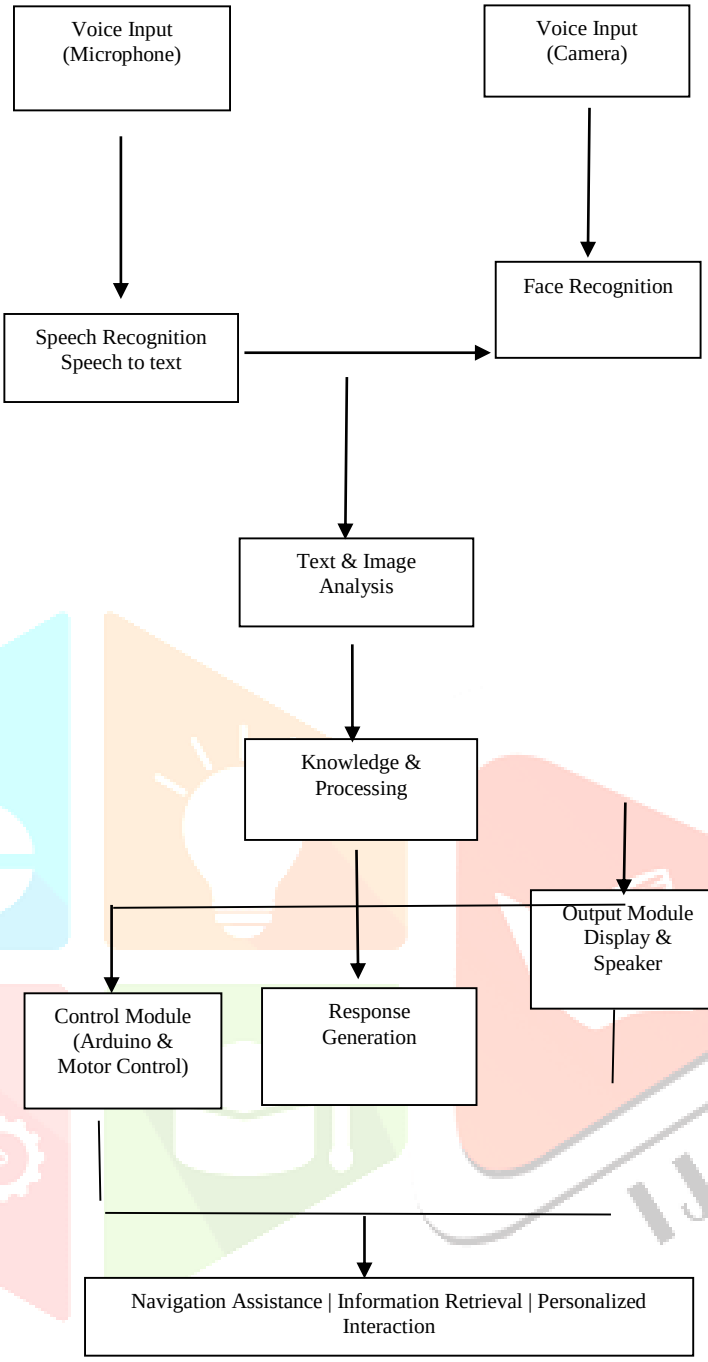

Fig 3: Block diagram of AI-Campus Assistant Robot Working

The working of the proposed system is based on the above block diagram. The system takes both voice and visual inputs from the user and processes them to generate appropriate responses. The overall operation involves multiple stages including input acquisition, processing, analysis, and output generation.

The voice input is captured through a microphone connected to the system. The speech recognition module processes the input audio and converts it into a machine-understandable format using natural language processing techniques. The recognized speech is then converted into text using a speech-to-text conversion process. This text is further analysed, and based on the identified intent, control is transferred to the respective module responsible for executing the requested operation.

Similarly, the visual input is captured using a camera module. The system processes the captured images using computer vision algorithms to detect and recognize faces. Once a user is identified, the system can provide personalized responses based on stored data.

A. Modules Description

Audio Input:
Audio input refers to the voice data provided by the user to interact with the system. It enables users to give commands, ask questions, and request information through natural speech, making the interaction more intuitive and user-friendly.

Speech Recognition:
Speech recognition is the process of converting spoken language into a machine-readable format. It utilizes algorithms based on acoustic and language modelling. The acoustic model analyzes audio signals to identify phonetic units, while the language model predicts sequences of words to form meaningful sentences. Modern speech recognition systems are capable of understanding continuous natural speech with high accuracy, improving overall system efficiency and usability.

Speech-to-Text Conversion: This module converts spoken words into written text. The analog voice signals are first converted into digital signals, which are then processed to extract textual information. The accuracy of conversion depends on sampling rate and processing algorithms.

Text Analysis: The converted text is analyzed to understand user intent. This is achieved by comparing the input text with predefined commands or by using natural language processing techniques. The system identifies keywords and context to determine the appropriate response. Intelligent methods such as machine learning-based classification can also be used for better accuracy.

Face Recognition: The system uses a camera to capture images and detect faces. Using computer vision techniques, the system identifies users and enables personalized interaction. This enhances user experience by providing customized responses.

Knowledge Processing (RAG): The analyzed text is passed to the Retrieval-Augmented Generation (RAG) module. This module retrieves relevant information from a predefined knowledge base and generates accurate, context-aware responses. It reduces incorrect outputs and improves reliability.

Control Module: The processed command is sent to the control module, which communicates with the Arduino using serial communication. Based on the instruction, the Arduino controls motors and other hardware components to perform navigation tasks.

Output Module: The system provides output in both visual and audio forms. The graphical user interface displays information such as navigation guidance and system status. At the same time, text-to-speech conversion generates voice responses for user interaction.

B. Functional Modules

Navigation Assistance:
Provides directions and guidance within the campus environment based on user queries.

Information Retrieval:
Displays or speaks information such as department details, schedules, and general campus-related queries.

Personalized Interaction:
Uses face recognition to identify users and provide customized responses.

Real-Time Interaction:
Ensures continuous communication between user and system using multi-threaded processing.

IMPLEMENTATION

The entire system works based on the voice input given by the user using a switch-case mechanism. Each functionality of the campus assistant robot is divided into separate modules, and each module is activated based on the specific voice command provided by the user. Every module is assigned a unique name, and when the recognized voice input matches the module name, the respective module gets activated and executes its code to perform the required task.

• **Navigation Module:** When the user gives a voice input for navigation, this module gets activated and fetches the code of this module. The speech input is first converted into text and analyzed to identify the destination specified by the user. The system then checks the predefined campus map or stored location data to determine the path. Based on this, control signals are generated and sent to the Arduino through serial communication. The Arduino processes these signals and controls the motor driver and DC motors to move the robot in the required direction. The system also provides voice guidance and visual instructions to assist the user during navigation. Once the destination is reached, the robot stops automatically.

• **Information Retrieval Module:** When the user gives a voice input to retrieve information, this module gets activated and fetches the code of this module. The input text is analyzed and passed to the knowledge processing unit. The system uses a Retrieval-Augmented Generation (RAG) framework to fetch relevant data from a predefined knowledge base. The retrieved information may include details about departments, faculty, schedules, or general campus-related queries. Once the module is executed, the information is displayed on the graphical interface and also delivered through audio output for better interaction.

• **Face Recognition Module:** When a user comes in front of the system, the camera captures the image and activates this module. The captured image is processed using computer vision algorithms to detect the presence of a face. The detected face is then compared with stored datasets to identify the user. If a match is found, the system recognizes the user and enables personalized interaction such as greeting the user by name or providing customized responses. This module continuously runs in the background to ensure real-time detection and recognition.

• **Voice Interaction Module:** This module is responsible for handling communication between the user and the system. When the user provides a voice command, the microphone captures the input audio signal. The speech recognition system processes this signal using acoustic and language models to convert it into text. The converted text is then analyzed to understand the intent of the user. Based on the identified command, the system activates the corresponding module. This module ensures smooth and natural interaction between the user and the robot.

• **Control Module:** When a command related to movement or action is detected, this module gets activated and fetches the control code. The Raspberry Pi sends instructions to the Arduino through PySerial communication. The Arduino interprets these commands and controls the motor driver accordingly. The motor driver regulates the speed and direction of the DC motors, enabling the robot to perform actions such as moving forward, backward, turning left, or turning right. This module ensures precise and efficient control of the robot.

• **Display Module:** This module is responsible for presenting information visually to the user. When a specific command is executed, the corresponding output is displayed on the screen through the graphical user interface developed using Tkinter. The displayed information may include navigation details, system status, or responses to user queries. The layout and position of the display are predefined in the code for better user experience.

•**Audio Output Module:** It is responsible for converting the system-generated text responses into audible speech, enabling effective communication with the user. After processing a user's request and generating a textual response, the module uses **Text-to-Speech (TTS) technology** to produce natural-sounding audio. It supports multiple languages and accents, allowing customization based on user preferences. The audio output is synchronized with visual cues on the display to enhance understanding and provide a more engaging user experience. The module manages speech clarity, volume, and rate, ensuring that

responses are intelligible even in noisy environments such as a campus. By providing real-time audio feedback, the module enables hands-free interaction, making the system more accessible to users with disabilities. Additionally, it can queue multiple responses if several queries are processed simultaneously, ensuring that no information is lost and maintaining smooth interaction.

• **Multi-threading Module:** To ensure real-time performance, the system uses multi-threading techniques. Different tasks such as speech recognition, face detection, response generation, and hardware control are executed simultaneously in separate threads. This reduces delay and improves system efficiency, allowing smooth and continuous operation. The module also improves system scalability and performance. As system complexity increases, new threads can be added for additional functionalities without redesigning the entire architecture. Efficient thread management techniques, such as thread pooling, are used to reduce the overhead of constantly creating and destroying threads. Furthermore, the Multi-threading Module enhances user experience by ensuring smooth and uninterrupted interaction. The system can process multiple user requests concurrently, respond quickly, and maintain continuous operation without freezing or lagging. Proper error handling and thread monitoring mechanisms are also implemented to detect failures and restart threads if necessary, ensuring system reliability.

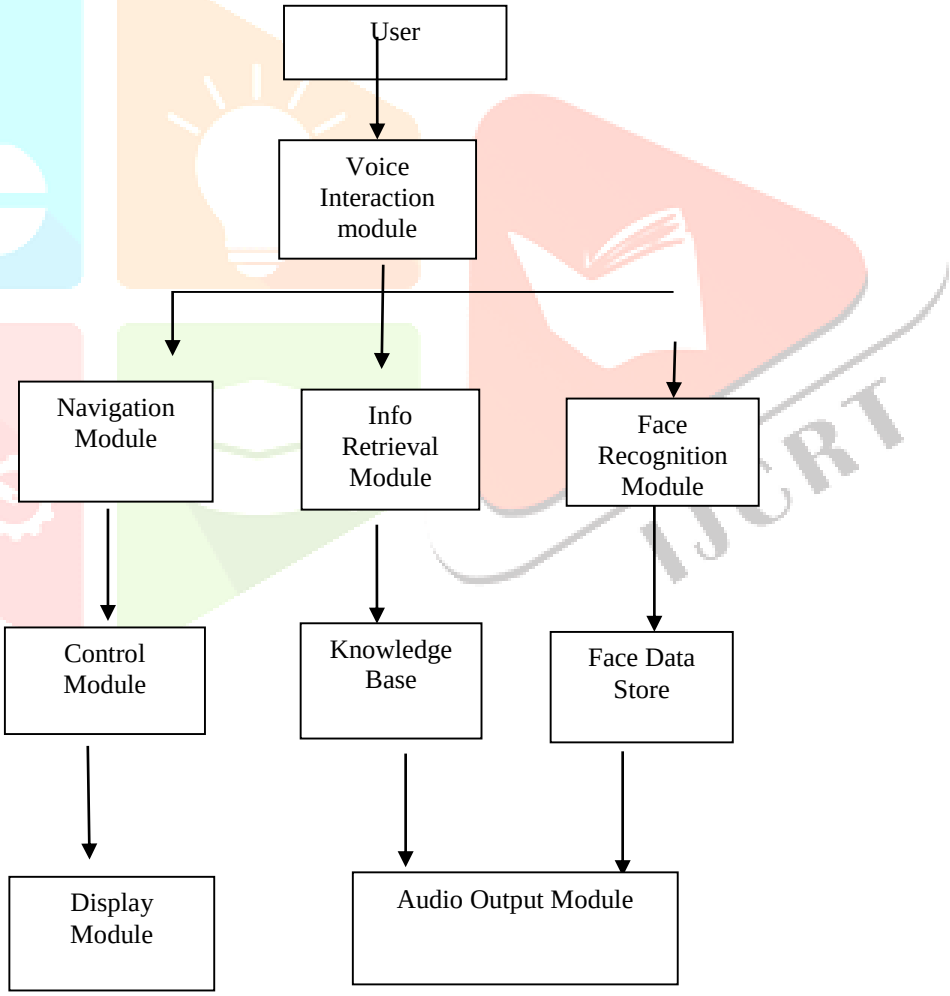

Fig 4. Data Flow Diagram of Campus Assistant Robot

CONCLUSION AND FUTURE WORK

This paper presented an AI-driven campus assistant robot that integrates several advanced features using artificial intelligence, computer vision, and voice recognition. A reliable, efficient, and user-friendly system was designed and implemented using Raspberry Pi and Arduino.

The proposed system provides an interactive platform for users to communicate with the robot through voice commands and receive accurate responses. The integration of speech recognition, face recognition, and Retrieval-Augmented Generation (RAG) enables the system to deliver personalized and context-aware assistance. The robot is capable of performing tasks such as navigation, information retrieval, and real-time interaction, thereby improving the overall user experience in a smart campus environment.

The system not only facilitates communication and information exchange but also demonstrates the potential for integration with larger intelligent systems. It can be extended to support applications such as smart campus management, security monitoring, and automated assistance services. The use of low-cost hardware components makes the system practical and scalable for real-world deployment.

To address user needs, the system successfully provides features such as voice-based interaction, face recognition for personalized responses, and real-time information delivery through both audio and visual outputs. The multi-threaded architecture ensures smooth and efficient system performance.

As part of future work, the system can be further enhanced to operate as a fully autonomous robot with improved navigation capabilities using advanced sensors and mapping techniques. Artificial intelligence can be expanded to include more advanced natural language understanding for better interaction. The system can also be integrated with IoT-based smart campus infrastructure to control devices such as lighting, security systems, and access control.

Additionally, advanced security features such as enhanced face recognition and anomaly detection can be implemented. The system can also be developed into a commercial product with cloud integration, mobile application support, and multilingual capabilities to improve accessibility and usability.

ACKNOWLEDGMENT

I express my sincere gratitude to my project guide and Head of the Department (CSE), Dr. Syam Mohan, for his invaluable guidance, continuous support, and encouragement throughout the course of this project. His insightful suggestions and expert advice significantly contributed to the successful completion of this work. I am also deeply thankful for the facilities, motivation, and opportunities provided by him, which enabled me to carry out this project effectively and successfully.

REFERENCES

1. S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers and D. Fox, “Probabilistic algorithms and the interactive museum tour-guide robot,” *Robotics and Autonomous Systems*, vol. 38, no. 3–4, pp. 171–188, 2002.
2. G. Bradski, “The OpenCV library,” *Dr. Dobb’s Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.
3. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin and N. Goyal, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
4. J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL-HLT*, pp. 4171–4186, 2019.
5. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan and P. Dhariwal, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

6. E. Upton and G. Halfacree, *Raspberry Pi user guide*, 4th ed. Hoboken, NJ, USA: Wiley, 2016.
7. M. Banzi, *Getting started with Arduino*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2011.
8. M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote and J. Leibs, "ROS: an open-source robot operating system," in *Proc. IEEE Int. Conf. Robotics and Automation Workshop*, pp. 1–6, 2009.
9. D. Jurafsky and J. H. Martin, *Speech and language processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2020.
10. L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.
11. W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
12. R. Szeliski, *Computer vision: algorithms and applications*. London, U.K.: Springer, 2011.
13. Dix, J. Finlay, G. Abowd and R. Beale, *Human-computer interaction*, 3rd ed. Harlow, U.K.: Pearson, 2004.
14. Pandey, "AI-based campus assistant robot for smart environments," *International Journal of Robotics Research*, vol. 40, no. 5, pp. 789–802, 2021.
15. K. Goldberg and B. Kehoe, "Cloud robotics and automation: a survey of related work," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2013.