TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASHCHIMANCHAL CAMPUS

LAMACHAUR, POKHARA

[Subject Code: EX654]

A PROJECT REPORT

ON

HOME AUTOMATION USING SPEECH RECOGNITION

SUBMITTED BY:

BINAYAK SHRESTHA	[WRC077BEI012]
PRAKANDA BHANDARI	[WRC077BEI030]
PRATHAM ADHIKARI	[WRC077BEI032]
SANDESH BASHYAL	[WRC077BEI036]

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

March, 2024

TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

PASHCHIMANCHAL CAMPUS

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled **"HOME AUTOMATION USING SPEECH RECOGNITION"**, submitted by Binayak Shrestha, Prakanda Bhandari, Pratham Adhikari, and Sandesh Bashyal in partial fulfilment of the requirements for the Bachelor degree in “Electronics, Communication and Information Engineering" has been accepted as a bonafide record of work independently carried out by team in the department.

Supervisor: Suraj Basant Tulachan

Department of Electronics and Computer Engineering

External Examiner

Khemraj Koirala

Head of the Department

Department of Electronics and Computer Engineering

DATE OF APPROVAL: March, 2024

COPYRIGHT ©

Head
Department of Electronics and Computer Engineering
Pashchimanchal Campus, Institute of Engineering
Lamachaur, Pokhara
Nepal

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our supervisor, Mr. Suraj Basant Tulachan, for all of his help and support during the course of this project. Without a doubt, his priceless blessings and encouragement will help us move forward in life.

Furthermore, we would like to thank the Pashchimanchal Campus, Institute of Engineering, Department of Electronics and Computer Engineering for offering us useful courses and a supportive learning environment. We sincerely appreciate all of the professors who helped and encouraged us to finish the assignment.

Lastly, we give appreciation to the almighty, our parents, friends, and everyone who has consistently supported and encouraged us. This project would not have been possible without their help.

ABSTRACT

Home automation using speech recognition is a project carried out with intention to explore the concept of home automation through integration of technology. It is a hardware and software-based project which focuses on economic implementation of home automation. The goal of this project is to explore various technological aspects involved in achieving automation of tasks to improve the overall convenience. To achieve this goal, real-time audio data from the microphone is transmitted to raspberry pi. The data is then processed by the speech recognition model in raspberry pi which generates text as an output. This text obtained is compared with predefined keywords and incase of a match, the text is converted to signals which is then transferred to appliance(s) through the relay module. The status of appliances are then updated in real-time and shown through a web application.

Keywords: ***Arduino Nano, PyTorch, Raspberry Pi, Speech-to-text, Transformer (Whisper) model***

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
AI	Artificial Intelligence
API	Application Programming Interface
ARM	Advanced RISC Machine
ASR	Automatic Speech Recognition
COM	Common terminal
CUDA	Compute Unified Device Architecture
FAIR	Facebook’s AI Research Lab
GPIO	General Purpose Input/Output
HDMI	High-Definition Multimedia Interface
IDE	Integrated Development Environment
I2C	Inter-Integrated Circuit
LSTM	Long-Short Term Memory
MEAN	MongoDB, Express, Angular, and Node
MERN	MongoDB, Express, React, and Node
MEVN	MongoDB, Express.js, Vue.js, and Node.js
ML	Machine Learning
NC	Normally Closed terminal
NLP	Natural Language Processing
NO	Normally Open terminal
RISC	Reduced Instruction Standard Computer
RNN	Recurrent Neural Network
REST	Representational State Transfer
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
WER	Word Error Rate
Wi-Fi	Wireless Fidelity

CHAPTER 1

INTRODUCTION

1.1 Background

In the contemporary era of smart living, home automation systems have become increasingly popular due to their ability to enhance convenience, efficiency and also catering to individuals with mobility challenges. Integration of speech recognition technology into home automation systems adds an additional layer of user-friendliness, accessibility, leveraging its power to seamlessly interact with various devices within a home environment.

Traditional method with physical switching of appliances, checking if they are on or not brings a level of inconvenience to disabled as well as abled people. This system can accurately interpret the user instructions and generates control signals for appliances. It does so by capturing audio input through a connected microphone.

Upon receiving the audio signals, Raspberry pi utilizes a speech recognition model within it which converts the received voice data into text commands enabling the system to comprehend user instructions accurately. Once the command is interpreted, It then generates necessary control signals and sends them to a 1-channel relay module and other hardware components, allowing a seamless control of different household appliances connected to the system. The status of the appliances are updated in real-time which can be viewed on a dashboard of a web application. This system not only operates the appliances, but also views their status which makes it easier to observe them even if not physically present near them.

1.2 Motivation and Inspiration

Our project's motivation and inspiration stemmed from a genuine desire to simplify and improve daily life within homes. Witnessing the growing integration of technology into our lives, we were inspired to utilize advancements in speech recognition and automation to make household tasks more efficient and accessible. Our goal was to empower users with intuitive control over their environment, aiming for a user-friendly home automation system. Additionally, we were excited about the opportunity to explore the intersection of artificial intelligence, embedded systems, and IoT technologies, seeing it as a chance to deepen our understanding and expertise in these areas.

1.3 Problem Statement

The absence of automated controls requires manual operation of various appliances and thus results in a time consuming routine. Managing multiple devices manually can be troublesome especially for elderly and the disabled. The commercial home automation systems are based mainly on Zigbee, Z-Wave, or Wi-Fi protocols. Such automated systems are either ineffective in terms of cost or power consumption.

Additionally, the complexity of processing speech in real-time poses a significant challenge. Existing systems often struggle with accurately interpreting the intent of the speaker. Thus, the technologies that are being applied are prone to various kinds of failures, for instance an individual's speech can be misinterpreted,in case of multiple commands by more than one person the commands might be neglected.

1.4 Objectives

The objectives of this project is:

- To design and develop a speech recognizing home automation system that can comprehend user instructions accurately and operates accordingly.

1.5 Scope

Our project's scope involves creating a comprehensive home automation system that merges speech recognition technology with hardware automation to improve household functionality. Our main goal is to develop an easy-to-use interface enabling residents to control various home fixtures and devices through voice commands.

We also plan to incorporate features for task scheduling, such as setting timers and activating predefined routines, to streamline daily routines further. The Raspberry Pi 4 Model B will serve as the central controlling unit, facilitating communication between the speech recognition algorithm and connected hardware components through the Arduino Nano. While our initial focus is on essential features, we're designing the system to be scalable, allowing for future expansions and updates to meet changing user needs and technological advancements in home automation.

CHAPTER 2

LITERATURE REVIEW

Paper [1] was concerned with the analysis of the voices from different individuals to extract different properties and intents from the voice to build a door access system. It uses various electronic components to build the voice recognition door access control system. Some of them are Arduino, ESP8266, Wires and Electronic Magnetic Locks.

Paper [2] introduces low-power RF ZigBee wireless communication modules used for establishing home automation system. The system consists of three modules. The modules include Handheld Microphone Module, Central Controller Module and Appliance Control Module. The Handheld Microphone Module incorporates a microphone with RF module which uses ZigBee protocol. The Central Controller Module is PC based.

Paper [3] was concerned with establishing a smart home automation system using reliable, compact, fast and low cost technology. It uses a bluetooth chip with an arduino, thus it eliminates the use of PCs. It incorporates various devices such as lights and DC Servomotors to demonstrate the feasibility and reliability of smart home systems.

Paper [4] discusses the shift in the speech community from using hybrid models to end-to-end (E2E) models for automatic speech recognition (ASR). While E2E models have shown superior accuracy in ASR benchmarks, hybrid models are still widely used in commercial ASR systems. This is because traditional hybrid models, which have been optimized for production for many years, excel in practical factors that affect model deployment. In an E2E ASR system based on Transformer, the model directly takes raw audio waveform as input and produces text transcriptions as output, without the need for intermediate representations or separate components for acoustic modeling, pronunciation modeling, and language modeling.

Paper [5] discusses the advancements in deep learning-based speech recognition systems, highlighting the transition from complex hybrid architectures to simpler end-to-end architectures. While server-based speech recognition systems offer high accuracy, they face challenges related to privacy, security, and reliability due to network dependencies. In contrast, offline speech recognition on client devices, such as Raspberry Pi and Nvidia Jetson Nano, offers increased privacy and reliability but poses challenges due to resource constraints. The performance and efficiency of transformer-based speech recognition systems on edge devices have latency that is significantly improved, with real-time inference achievable on the Raspberry Pi CPU and even better performance on the Nvidia Jetson Nano GPU, compared to server-based systems. Although the word error rate on edge devices is slightly higher than on server-based inference, the results are promising and demonstrate the potential for using transformer-based models on edge devices for speech recognition tasks.

Recurrent neural networks, long short-term memory and gated recurrent neural networks in particular, was firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder architectures [6]. Recurrent Neural Networks (RNNs) and their variant LSTMs, while powerful, suffer from limitations. LSTMs attempt to address the vanishing gradient problem in RNNs, where crucial information from earlier parts of the sequence gets lost. However, LSTMs can still struggle with very long sequences. Additionally, both RNNs and LSTMs process information sequentially, limiting their ability to capture complex relationships within the data. This sequential nature also makes them computationally expensive for training. These drawbacks hinder their performance in tasks requiring long-range dependencies and efficient processing, paving the way for advancements like transformers.

Paper [6] introduces Transformers that solves the drawbacks of these models being inability to generalize to long sequences, unparalizability. They outperform LSTMs in sequence modeling tasks like machine translation due to their powerful attention mechanism. Unlike LSTMs, which process information sequentially, transformers can attend to all parts of the input sequence simultaneously using self-attention. This allows them to capture long-range dependencies more effectively, crucial for understanding complex relationships in language. Additionally, encoders and decoders within transformers are specifically designed to handle input and output sequences, respectively, leading to better focus and efficiency compared to LSTMs' single, recurrent processing. This combination of self-attention and dedicated encoder-decoder architecture grants transformers superior accuracy in various NLP tasks.

Paper [7] uses new speech recognition model using attention-based recurrent network for Automatic Speech Recognition. Unlike traditional methods, the model can focus on important parts of the speech throughout the sequence. This is useful for long and noisy speech recognition tasks. The model was tested on a phoneme recognition task and show that it performs competitively, especially for longer speech samples.

The Transformer model was first introduced in Gaussian mixture hidden Markov modeling for speech recognition [8], in order to reduce sequential calculations and the number of operations for correlating input and output position signals[9].The model performed exceptionally well in compared to state-of-the-art models at the time.

Speech recognition relies on labeled data, limiting its reach. Baevski [10] proposes wav2vec-U, an unsupervised method using self-supervised learning from wav2vec 2.0 representations. k-means clustering segments speech, and adversarial training maps segments to phonemes, even allowing for silence labels. wav2vec-U achieves significant reductions in phone error rates on TIMIT and rivals supervised models on Librispeech, demonstrating its effectiveness across languages. This approach paves the way for speech recognition in low-resource settings. Experiments on the standard Librispeech benchmark show performance close to the state of the art models from only a few years ago, even though these models relied on nearly 1,000 hours of labeled data [10].

Paper [11] introduces the model, Whisper, leverages a massive dataset of audio recordings with corresponding, but imperfect, internet transcripts. This weak supervision approach, alongside multilingual and multitask training, allows Whisper to excel in zero-shot generalization and achieve competitive performance compared to fully supervised models. Furthermore, Whisper simplifies the recognition pipeline by eliminating the text normalization step. These findings highlight the potential of weak supervision for robust and efficient speech recognition.The smallest zero-shot Whisper model, which has only 39 million parameters and a 6.7 WER on LibriSpeech test-clean is roughly competitive with the best supervised LibriSpeech model when evaluated on other datasets [12].Whisper showed high transcription performances (for each language the percentage of correct words transcribed is higher than 90%). For the English and Italian languages, the most committed mistakes were words substitutions, while for Russian there were observed higher percentages of both words substitution and insertion errors.

.

CHAPTER 3

RELATED THEORY

3.1 Microphone

A microphone is a device that translates sound vibrations into electrical signals. It enables users to capture audio and transmit it to various devices such as recording medium or over a loudspeaker. Microphones are used in many applications such as telephones, sound engineering, live and recorded audio engineering, radio and television broadcasting, etc., for communication purposes.

3.2 Arduino Nano

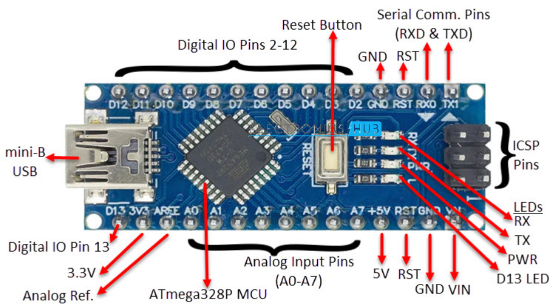

Figure 3.1: Arduino Nano [13]

The Arduino Nano in 3.1 is a small, breadboard-friendly microcontroller board based on the ATmega328. It is equipped with 30 male I/O headers which can be programmed using the Arduino Software IDE (Integrated Development Environment). It has 10 bit ADC. It can communicate with a computer and other microcontrollers.It consists of flash memory of 32KB capacity of which 2 KB is used for Bootloader.It supports Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI) and Universal Asynchronous Receiver / Transmitter (UART) [14].

3.3 Raspberry Pi 4 Model B

The Raspberry Pi in 3.2 is a credit card-sized single-board computer that provides a powerful processing platform in a compact form factor. It is equipped with a range of input/output interfaces, including USB ports, HDMI, Ethernet, GPIO (General Purpose Input/output), and a camera interface. The processor speed of Raspberry Pi 4 Model B is 1.5 GHz. It comes with onboard wireless networking and bluetooth [15].

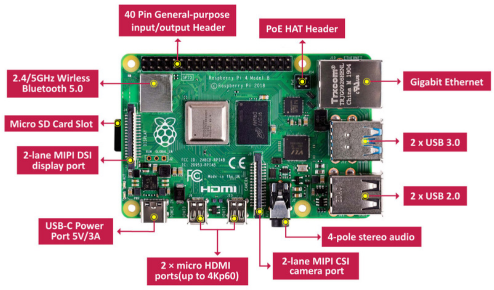

Figure 3.2: Raspberry Pi 4B [16]

3.4 Motor

Figure 3.3: Servo Motor [17]

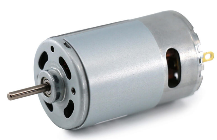

Figure 3.4: DC Motor [18]

Motors in 3.3 and 3.4 also known as actuators are fundamental components in robotics, providing the necessary actuation, precision, and control for robot movement and manipulation. With the help of motors, the system can be moved to different locations using a precise control system in the 2D axis. develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android) [19].

3.5 Arduino IDE

The Arduino IDE is a software application designed specifically for programming Arduino boards. It provides users with a convenient and user-friendly interface for writing, editing, and uploading code to Arduino microcontrollers [20].

3.6 PyTorch

PyTorch is a widely used open-source machine learning framework developed by Facebook's AI Research lab (FAIR). It is known for its flexibility, ease of use, and dynamic computation graph, which makes it particularly well-suited for research and experimentation in deep learning [21]. It's key features include:

- Tensor Computation
- Dynamic Computational Graph
- Automatic Differentiation
- GPU acceleration

3.7 React.js

React is a free and open-source front-end JavaScript library for building user interfaces based on components. It can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js. It allows the creation of interactive and visually appealing user interfaces. Developers can design and implement components that display real-time status of the appliances [22].

3.8 Express.js

Express is a back-end web application framework for building RESTful APIs with Node.js, released as open-source software. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack together with the MongoDb database software and a JavaScript front-end framework [23].

3.9 Recurrent Neural Network

A unique kind of artificial neural network called RNN is designed to cope with time series data or data that contains sequences. RNN are equipped with the idea of ”memory,” which enables them to store the states or details of earlier inputs in order to produce the next output in the sequence.Here, the query data are also the kind of sequence data which needs to be processed to extract feature using the special kind of RNN methods called LSTM. RNNs have been used to generate mathematical proofs and translate human thoughts into words [5][4].

3.10 Transformer

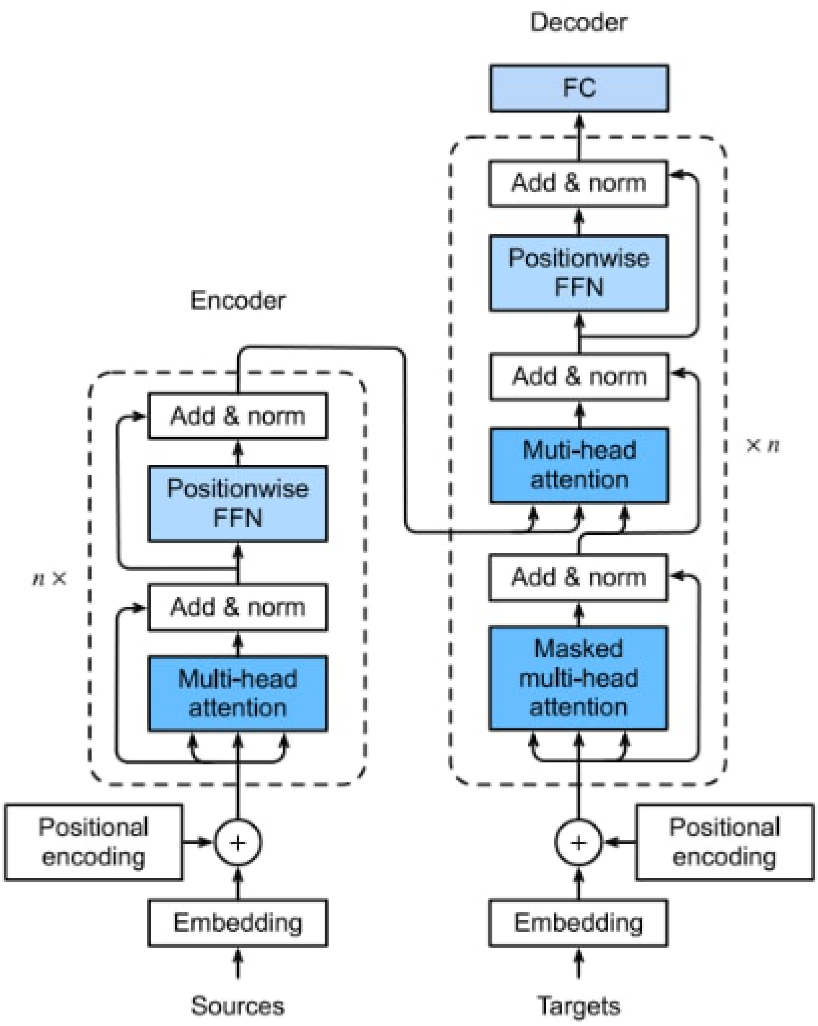

Figure 3.5: Attention Mechanism [24]

The drawbacks of RNN being inability to generalize to long sequences, unparalizability are solved by Transformer. Transformers are used in state of the art in almost all NLP tasks. A transformer is a deep learning model that uses sequential data analysis for understanding and learning the context. It follows encoder-decoder structure and process the entire sequence at a time as shown in 3.5. The encoder of the Transformer processes the input embedding and generate the hidden representations. The input embedding are added with a positional encoding to indicate its position in the sequence before it is fed to the encoder. Self-attention is used to output the attended features which are simply a weighted average of the input features. The attended features signify a combination of most important features related to another feature. Then a normalization and feed forward network is used to transform the attended features to required dimension. The decoder takes in the hidden representation from encoder and performs cross attention to generate the output sequence. The output generated also depends on the previously generated output of decoder which is initially empty. The rest of the architecture is similar to that of an encoder. The token with the highest probability is chosen to generate the output sequence [24].

3.11 Whisper

Whisper is an automatic speech recognition (ASR) system trained on 680,000 hours of multilingual and multitask supervised data collected from the web. We show that the use of such a large and diverse dataset leads to improved robustness to accents, background noise and technical language. The Whisper architecture is a simple end-to-end approach, implemented as an encoder-decoder Transformer. Input audio is split into 30-second chunks, converted into a log-Mel spectrogram, and then passed into an encoder. A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and to-English speech translation.Other existing approaches frequently use smaller, more closely paired audio-text training datasets or use broad but unsupervised audio pretraining. Because Whisper was trained on a large and diverse dataset and was not fine-tuned to any specific one, it does not beat models that specialize in LibriSpeech performance, a famously competitive benchmark in speech recognition. However, when Whisper's zero-shot performance across many diverse datasets was measured , it turns out to much more robust and makes 50% fewer errors than other models [25].

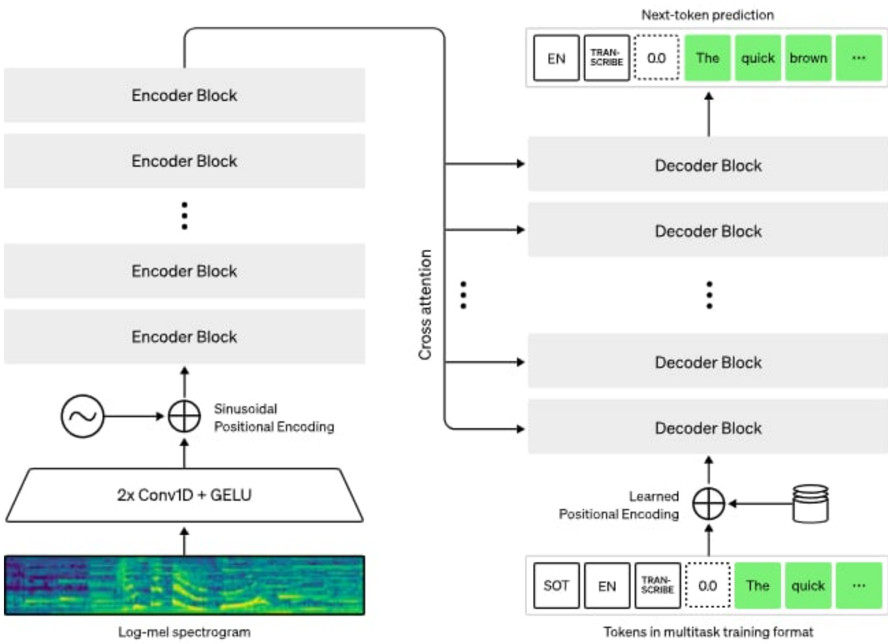

Figure 3.6: Architecture of Whisper [25]

3.12 Relay Module

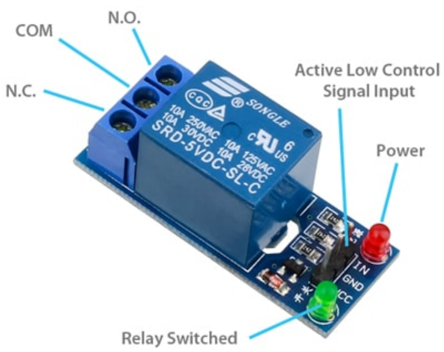

Figure 3.7: 1- Channel Relay [26]

Relay in 3.7 is electrically operated switches that open and close circuits.They permit a small amount of electrical current to control high current loads.They protect the control systems from high voltage and current as they are used to provide electrical isolation between the control and load circuits [27].

CHAPTER 4

METHODOLOGY

4.1 SYSTEM BLOCK DIAGRAM

In layman's terms, the home automation project with speech recognition is: you speak into a machine, and it takes that command to control whether household appliances turn on or off as shown in 4.1.

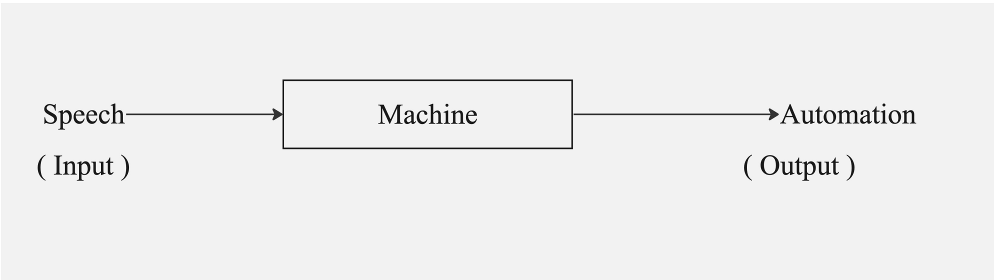

Figure 4.1: Block Diagram of Home Automation

Delving deeper into the workings of the system, the process begins with the user providing speech as input through a portable microphone. This input is then transmitted to a Raspberry Pi. Within the Raspberry Pi, a speech-to-text model interprets the spoken words. Upon successful interpretation, the Raspberry Pi generates a signal that commands the control of the appliance(s). This comprehensive system involves multiple components working in tandem to seamlessly translate spoken commands into tangible actions for home automation. The block diagram of the system hardware is shown in 4.2.

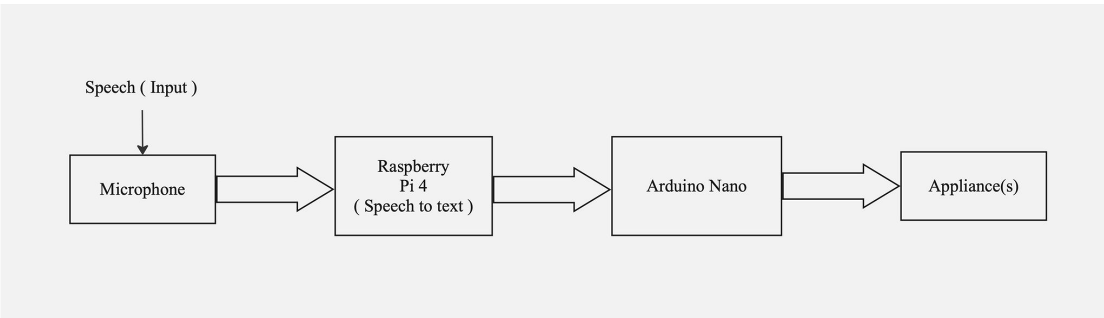

Figure 4.2: Block Diagram of System Hardware

The process unfolds with the speech serving as input for the speech-to-text model. The converted text undergoes comparison with a wake word in the wake word detector model. If the text does not match a wake word, the system remains inactive until a wake word is detected, prompting the process to restart. Upon recognizing the wake word, the converted text that follows is subjected to a set of conditions, and signals are then dispatched to control the appliances. This iterative sequence ensures that appliance control is contingent on the recognition of a specific wake word, enhancing the system's precision and responsiveness. The flowchart is shown in 4.3.

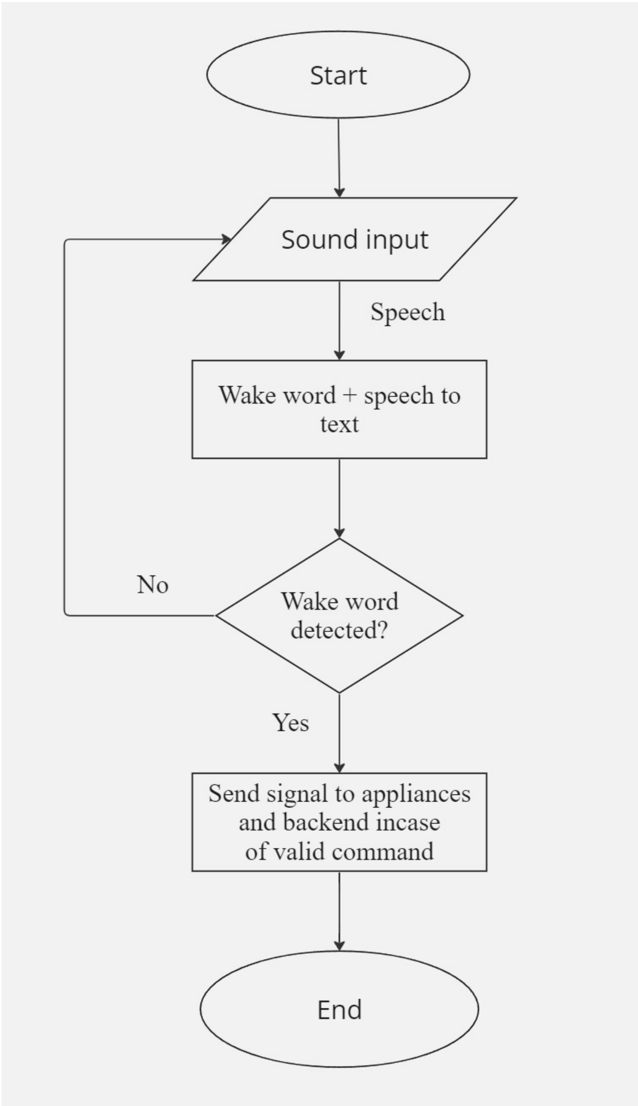

Figure 4.3: Flowchart of Speech Recognition

4.2 Selection of Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B was chosen for our home automation project due to its robust performance, and versatile connectivity. With its quad-core ARM Cortex-A72 processor and adequate memory options, it efficiently executes complex algorithms such as speech recognition. Its built-in Wi-Fi, Bluetooth, USB ports, and GPIO pins facilitate integration with peripheral devices like the Arduino Nano, enabling straightforward hardware control. Overall, the Raspberry Pi 4 Model B serves as the fundamental of our project, offering powerful performance, and reliability to meet the demands of our home automation system.

4.3 Activity diagram

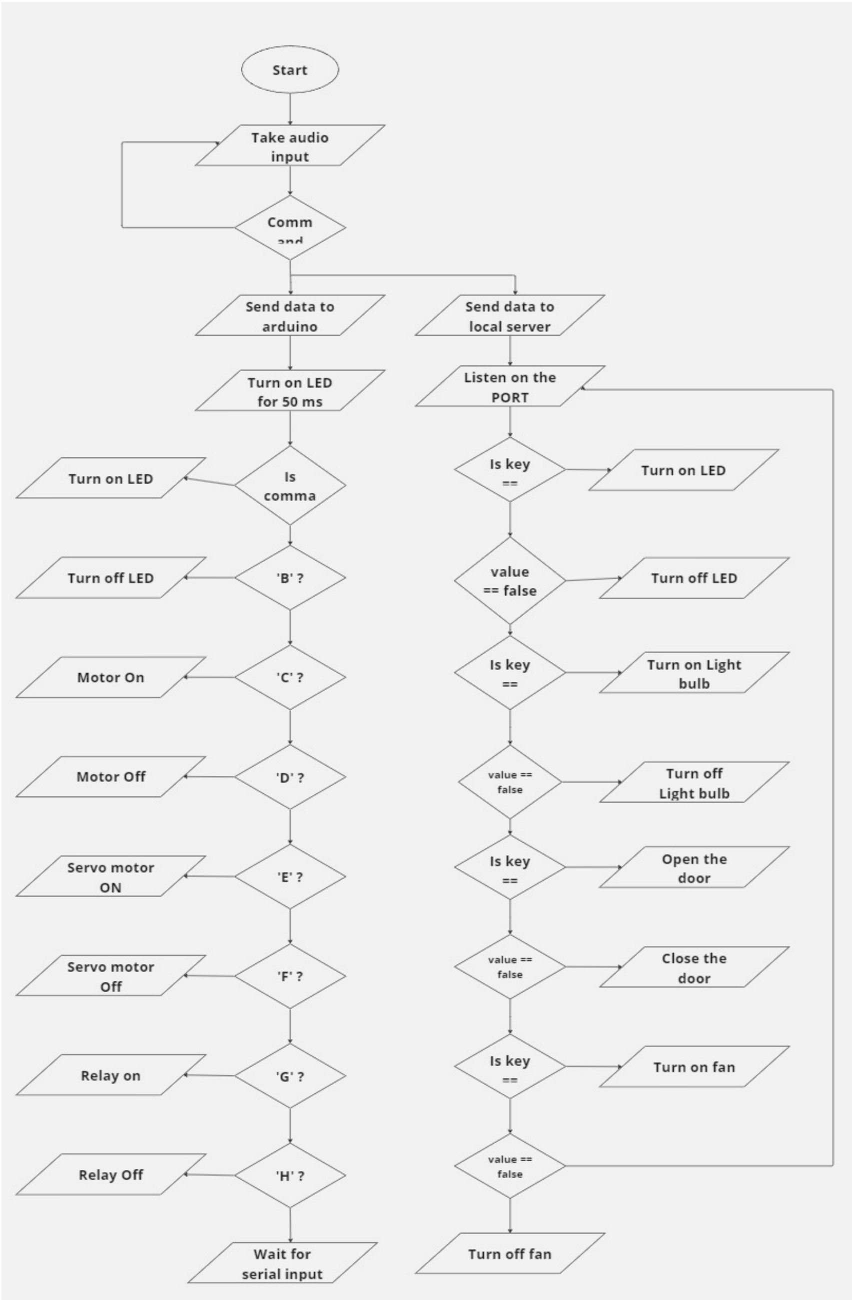

Figure 4.4: Activity Diagram

4.4 Interfacing between Raspberry Pi 4 Model B and components

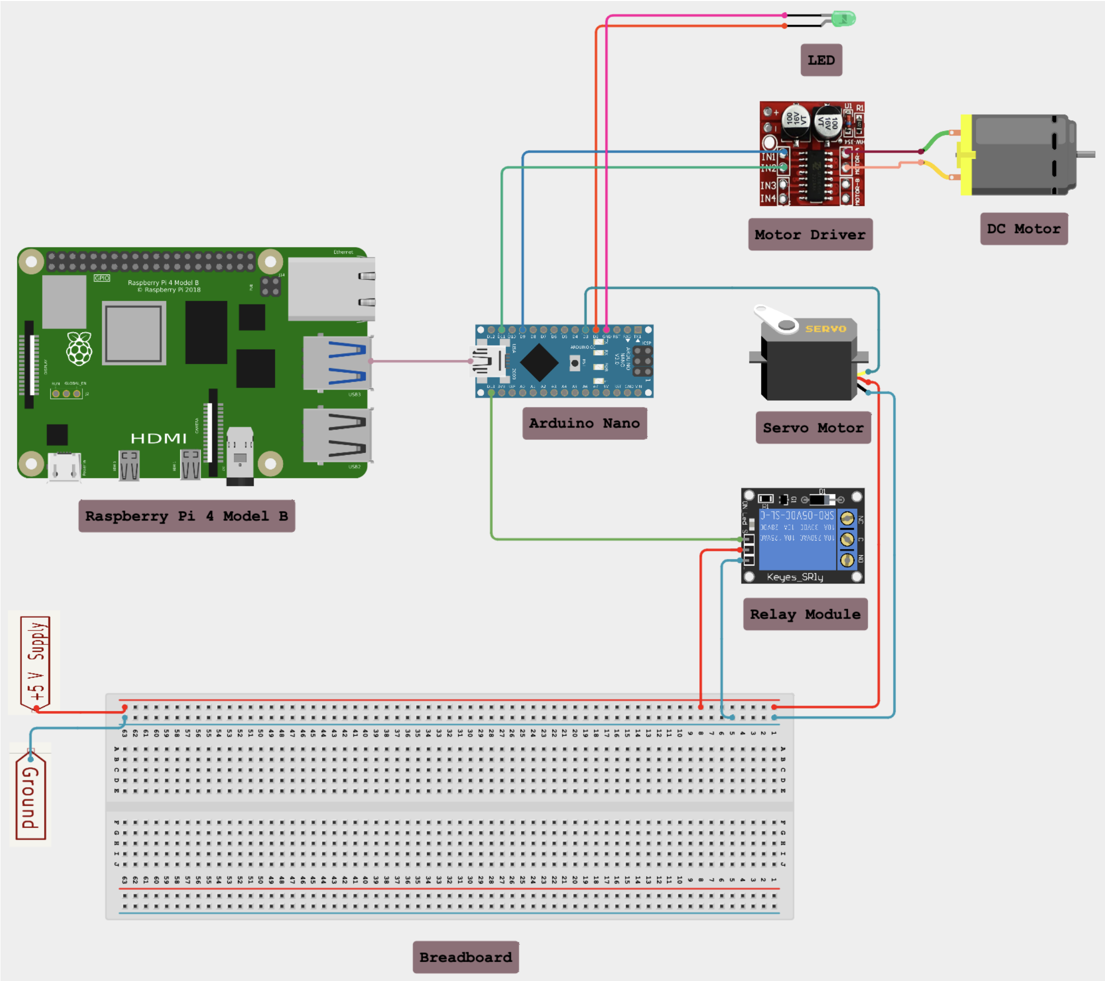

Figure 4.5: Interfacing

The Raspberry Pi 4 Model B serves as our development board, which oversees the functions of different hardware elements that, in turn, regulates the operations of household fixtures. The figure below shows all the interfacing of the home automation system.

From the above diagram, we are illustrating the interfacing between different components present in our system. We are using Raspberry Pi 4 Model B as our development board which uses ARM Cortex-A72 CPU. Control Signal is transmitted from the USB port of the Raspberry Pi to Arduino Nano.The Arduino Nano controls the actions of fixtures. The fixtures are controlled by the signal received from Arduino Nano. External power is provided to the hardware components if needed.
Hence, this is the interfacing between all our components of hardware, i.e Raspberry Pi, Arduino Nano, Relay module, LED, Servo motor, Motor driver and DC motor.

4.5 Interfacing between Hardware and Application

The Raspberry pie contains the inbuilt Wi-Fi module which enables the system to come online so a local server is set up. The local server acts as a bridge between the web application and the Raspberry pie. When a command is given to the system then it sends the string to the Arduino Nano via serial communication along with a '/post' request to send the json data for the change in status of the device. Similarly at the website there is a continuous 'http.get()' request to fetch the status in realtime, which is then used to change the status of devices in the webpage. Users can see the changes in real time via dashboard.

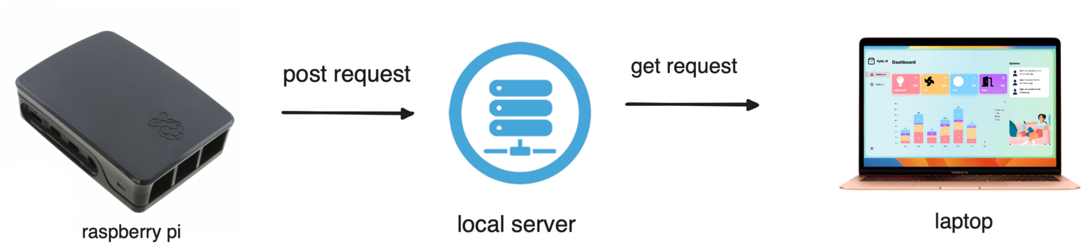

Figure 4.6: Interfacing between Hardware and Application

CHAPTER 5

OUTPUT AND ANALYSIS

This is what our UI of our web application looks like on 'OFF' (5.1) and 'ON' (5.2):

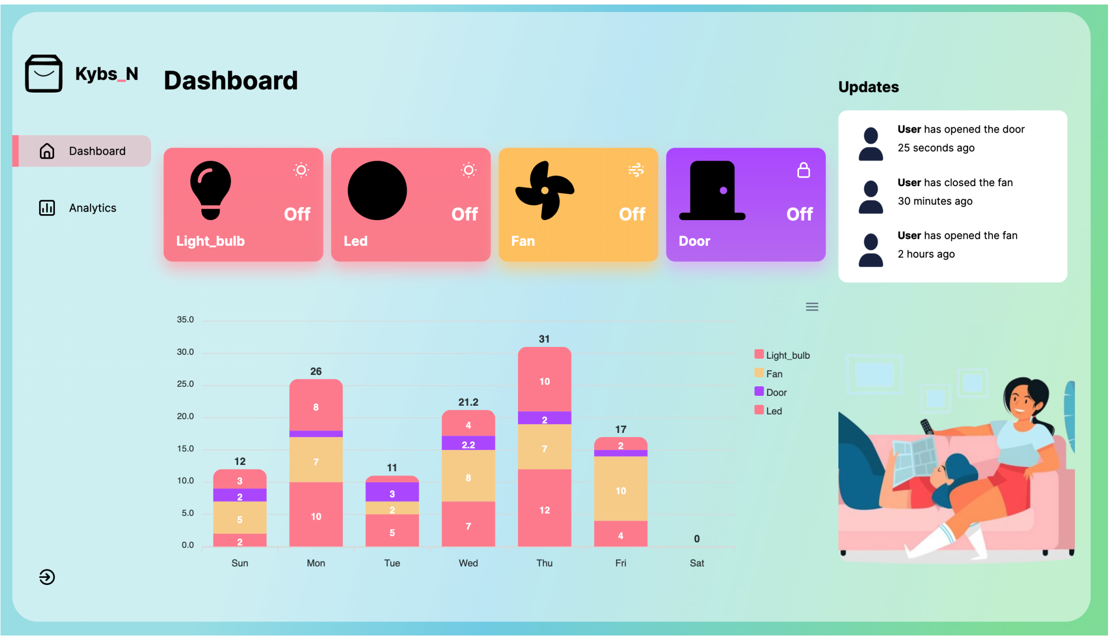

Figure 5.1: Web Application UI when devices are "OFF"

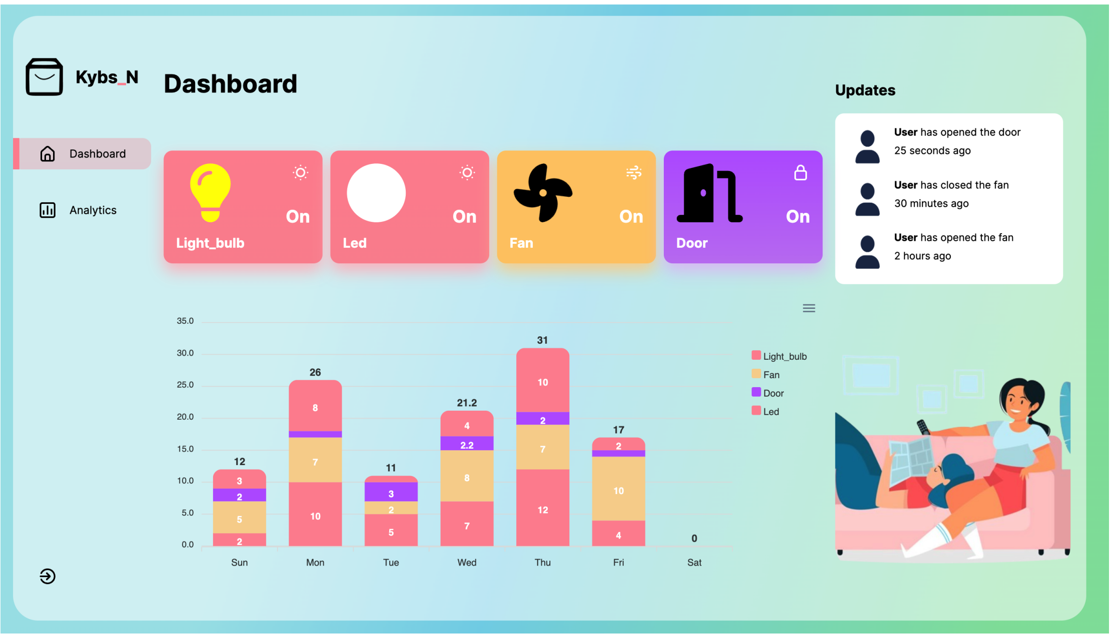

Figure 5.2: Web Application UI when devices are "ON"

The above figures show the UI of our web application we have made in React, when

they are “On” and “Off”. In the landing page of this web page we can see 3 different sections. Left section shows different options of Dashboard and Analytics. Currently there is nothing in the Analytics section.
The main part of this dashboard is in the middle section which shows the status of all the appliances which are controlled by our automation system. It shows if the device is “On” or “Off”, on clicking the device card we can see the analytics. It shows the data of the device on how long (in hour) they were in “On” state. An area plot is given for better visualization of the data. Below it is a bar graph showing the cumulative data of all individual devices.
Right section shows the recent commands given to devices and how long has it been since the command was given.

The complete system is shown in 5.3.

Figure 5.3: House

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

In wrapping up our home automation project that utilizes speech recognition and hardware automation, we've achieved notable success in enhancing household convenience and efficiency. Through the integration of the Raspberry Pi 4 Model B as the project's core controller, we've established a user-friendly interface for controlling various household devices via voice commands.

6.2 FUTURE ENHANCEMENT

Looking ahead, we envision several avenues for further development. These include refining the speech recognition system to enhance accuracy and expanding the compatibility and features of the system. Moreover, incorporating machine learning algorithms could enable the system to adapt to user preferences over time, enhancing the overall user experience. Additionally, exploring options for remote access and integration with popular smart assistants like Amazon Alexa or Google Assistant could extend the system's reach and functionality.

Overall, while our current implementation helps in home automation, there's enough room for ongoing innovation and improvement to meet the evolving demands of users in the dynamic realm of smart home technology.

REFERENCES

[1] O. Adekola, A. Akinsanya, A. Olufowobi, O. Babajide, O. Somefun, and A. Oduroye, "Voice recognition door access control system," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 21, no. 2019, pp. 01–12, 2019.

[2] H. Alshu'Eili, G. S. Gupta, and S. Mukhopadhyay, "Voice recognition based wireless home automation system," in *2011 4th international conference on mechatronics (ICOM)*. IEEE, 2011, pp. 1–6.

[3] S. Das, S. Ganguly, S. Ghosh, R. Sarker, and D. Sengupta, "A bluetooth based sophisticated home automation system using smartphone," in *2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI)*. IEEE, 2016, pp. 236–240.

[4] J. Li *et al.*, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[5] S. Gondi and V. Pratap, "Performance evaluation of offline speech recognition on edge devices," *Electronics*, vol. 10, no. 21, p. 2697, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[8] X. Cui and Y. Gong, "Variable parameter gaussian mixture hidden markov modeling for speech recognition," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 1, 2003, pp. I–I.

[9] M. Orken, O. Dina, A. Keylan, T. Tolganay, and O. Mohamed, "A study of transformer-based end-to-end speech recognition system for kazakh language," *Scientific Reports*, vol. 12, no. 1, p. 8337, 2022.

[10] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 826–27 839, 2021.

[11] T. Amorese, C. Greco, M. Cuciniello, R. Milo, O. Sheveleva, and N. Glackin, "Automatic speech recognition (asr) with whisper: Testing performances in different languages." 2023.

[12] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[13] "Arduino nano pinout, board layout, specifications, pin description," Electronicshub.org, 2024. [Online]. Available: https://www.electronicshub.org/wp-content/uploads/2021/01/Arduino-Nano-Board-Layout.jpg

[14] Arduino.cc, 2024. [Online]. Available: https://docs.arduino.cc/hardware/nano/

[15] R. P. Ltd, "Raspberry pi 4 model b specifications – raspberry pi," Raspberry Pi, 2024. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

[16] "Raspberry pi 4 model b," Robot Advance.com, 2024. [Online]. Available: https://www.robot-advance.com/userfiles/www.robot-advance.com

[17] "Servo motor," Media-amazon.com, 2024. [Online]. Available: https://m.media-amazon.com/images/I/61yfIwAxe0L.jpg

[18] Micro DC Motors, "High speed high torque brushed dc motor," Microdcmotors.com, 2024. [Online]. Available: https://microdcmotors.com/wp-content/

[19] The Pi Hut, 2019. [Online]. Available: https://thepihut.com/blogs/raspberry-pi-tutorials/

[20] Arduino.cc, 2024. [Online]. Available: https://docs.arduino.cc/software/ide-v1/tutorials/Environment/

[21] "Pytorch documentation — pytorch 2.2 documentation," Pytorch.org, 2023. [Online]. Available: https://pytorch.org/docs/stable/index.html

[22] "React," https://react.dev/, 2015.

[23] "Express - node.js web application framework," Expressjs.com, 2017. [Online]. Available: https://expressjs.com/

[24] "11. attention mechanisms and transformers — dive into deep learning 1.0.3 documentation," D2l.ai, 2018. [Online]. Available: https://d2l.ai/chapter$_{a}ttention - mechanisms - and - transformers/index.html$

[25] Openai.com, 2022. [Online]. Available: https://openai.com/research/whisper

[26] Techdelivers.com, 2024. [Online]. Available: https://techdelivers.com/image

[27] "Relay module: Meaning, function, and applications - tongou electrical," TONGOU Electrical, 10 2023. [Online]. Available: https://elcb.net/smart-relay-module/