Voice-Controlled Smart Lighting System

Final Year Project

Student Name: George Paizakis

Supervisor: Murouj Aljamaeen

Date: 02/05/2025

# Abstract

The advancement of smart home technologies has transformed how individuals interact with their environments, particularly through voice-controlled systems. Enhanced by Artificial Intelligence (AI) and the Internet of Things (IoT), these systems offer greater accessibility, convenience, personalization, and energy efficiency. This project addresses the growing need for intuitive, hands-free smart lighting solutions that assist users with mobility impairments or limited technical expertise.

The objective was to develop a Voice-Controlled Smart Lighting System capable of interpreting simple spoken commands such as "on" and "off." By integrating speech recognition and Natural Language Processing (NLP) within an IoT framework, the project aimed to deliver a scalable, user-friendly solution for smart home environments, designed with future expansion in mind.

Implemented using Python on Google Colab, the system relied on the publicly available Speech Commands dataset to train its speech recognition model. Key methods included audio preprocessing using Mel Frequency Cepstral Coefficients (MFCCs), neural network training, and system evaluation based on accuracy, precision, recall, and confusion matrices to ensure comprehensive performance assessment.

The final model achieved a recognition accuracy of approximately 85%, demonstrating strong performance under controlled conditions. Although integration with a TP-Link Kasa smart bulb was attempted, real time hardware control was limited by network constraints within the cloud-based environment. Despite these limitations, the simulation-based system successfully validated the design and core functionality.

In conclusion, the project resulted in a modular and scalable prototype that showcases the practical potential of voice-controlled smart lighting. It establishes a strong foundation for future work focused on real-world deployment, broader command sets, edge-based implementation, and enhanced ethical considerations such as data privacy, security, and energy sustainability.

# Acknowledgements

First and foremost, I would like to express my deepest gratitude to my project supervisor, Murouj Aljamaeen, whose invaluable guidance, support, and encouragement were instrumental throughout every stage of this final year project. Her expertise, patience, and thoughtful feedback have played a vital role in shaping both the direction and quality of my work. I am sincerely thankful for her belief in my abilities and for consistently pushing me to achieve the best possible outcome.

I would also like to extend heartfelt thanks to my mother, Esther, whose unwavering love, prayers, and support have been the foundation of my academic journey. Her strength and sacrifices continue to inspire me every day.

To my sister, Priscilla, thank you for being my constant source of motivation and encouragement. Your understanding and moral support have helped me persevere during challenging moments.

Lastly, I would like to acknowledge my work manager, Kieran, for his incredible flexibility and encouragement during this project. His understanding and support have allowed me to balance both work and study effectively, making it possible to dedicate the time and energy required for this achievement.

To all of you, thank you for being an essential part of this journey. I could not have done it without you.

Contents

[Abstract 2](#_Toc197053567)

[Acknowledgements 3](#_Toc197053568)

[Abbreviations and Glossary 6](#_Toc197053569)

[CHAPTER 1 INTRODUCTION 8](#_Toc197053570)

[1.1 Introduction 8](#_Toc197053571)

[1.2 Research Background 8](#_Toc197053572)

[1.3 Research Question 9](#_Toc197053573)

[1.4 Aim 9](#_Toc197053574)

[1.5 Objectives 9](#_Toc197053575)

[1.6 Deliverables 10](#_Toc197053576)

[1.7 Structure of the report 10](#_Toc197053577)

[CHAPTER 2 RESEARCH METHODS 12](#_Toc197053578)

[2.1 Research Methodologies - Design Science Research Methodology (DSR) 12](#_Toc197053579)

[2.2 Research Methods 14](#_Toc197053580)

[2.2.1 Dataset Acquisition 14](#_Toc197053581)

[2.2.2 Dataset Preparation and Preprocessing 15](#_Toc197053582)

[2.2.3 Model Architecture and Training 15](#_Toc197053583)

[2.2.4 Evaluation Metrics and Validation 15](#_Toc197053584)

[2.2.5 System Simulation and Integration Attempts 16](#_Toc197053585)

[2.2.6 Modularity and Extensibility 16](#_Toc197053586)

[2.2.7 Ethical, Accessibility, and Sustainability Considerations 16](#_Toc197053587)

[2.2.8 Documentation 17](#_Toc197053588)

[CHAPTER 3 LITERATURE REVIEW 19](#_Toc197053589)

[3.1 Introduction 19](#_Toc197053590)

[3.2 Core Enabling Technologies in Voice Controlled Smart Home Systems 19](#_Toc197053591)

[3.2.1 Voice Recognition 19](#_Toc197053592)

[3.2.2 Natural Language Processing (NLP) 19](#_Toc197053593)

[3.2.3 IoT and Smart Light Technologies 20](#_Toc197053594)

[3.3 Advanced Interaction 20](#_Toc197053595)

[3.3.1 Multimodal Interaction 21](#_Toc197053596)

[3.3.2 Voice Biometrics 21](#_Toc197053597)

[3.3.3 Context Aware Interaction 21](#_Toc197053598)

[3.3.4 personalization techniques 21](#_Toc197053599)

[3.4 System Performance Challenges 21](#_Toc197053600)

[3.4.1 Noise 22](#_Toc197053601)

[3.4.2 Language 22](#_Toc197053602)

[3.4.3 Constraints 22](#_Toc197053603)

[3.5 Privacy, Ethics, and Security 22](#_Toc197053604)

[3.6 Sustainability Considerations 23](#_Toc197053605)

[3.7 Comparative Analysis 24](#_Toc197053606)

[3.8 Research Gaps and Future Directions 24](#_Toc197053607)

[3.9 Conclusion 26](#_Toc197053608)

[CHAPTER 4 DESIGN OF ARTEFACT 27](#_Toc197053609)

[Extended Design Considerations 30](#_Toc197053610)

[Security and Data Privacy 32](#_Toc197053611)

[Environmental Considerations 33](#_Toc197053612)

[CHAPTER 5 IMPLEMENTATION OF ARTEFACT 34](#_Toc197053613)

[CHAPTER 6 EVALUATION 45](#_Toc197053614)

[Evaluation Methodology 45](#_Toc197053615)

[Speech Recognition Model Evaluation 46](#_Toc197053616)

[Preprocessing Pipeline Evaluation 48](#_Toc197053617)

[Simulated Smart Lighting Evaluation 51](#_Toc197053618)

[System Scalability and Performance Considerations 52](#_Toc197053619)

[CHAPTER 7 CONCLUSION AND FUTURE WORK 54](#_Toc197053620)

[REFERENCE LIST 58](#_Toc197053621)

[APPENDICIES 62](#_Toc197053622)

# Abbreviations and Glossary

|  |  |
| --- | --- |
| Abbreviation | Definition |
| AI | Artificial Intelligence – The simulation of human intelligence in machines. |
| API | Application Programming Interface – A set of tools for building software applications. |
| CNN | Convolutional Neural Network – A type of deep learning model primarily used for pattern recognition. |
| CSV | Comma-Separated Values – A file format used to store tabular data. |
| DL | Deep Learning – A subset of machine learning involving neural networks with multiple layers. |
| GUI | Graphical User Interface – A visual interface allowing users to interact with the system. |
| IoT | Internet of Things – A network of physical devices connected to the internet, capable of collecting and exchanging data. |
| KASA | A TP-Link smart device brand that offers remote-controlled smart lighting solutions. |
| Librosa | A Python library for analysing and processing audio signals. |
| MFCC | Mel-Frequency Cepstral Coefficients – A representation of the short-term power spectrum of sound, used for speech recognition. |
| NLP | Natural Language Processing – A branch of AI that enables machines to understand and respond to human language. |
| SR | Sampling Rate – The number of samples of audio carried per second. |
| TF/Keras | TensorFlow/Keras – Open-source libraries used for machine learning and neural network modelling. |
| TP-Link | A manufacturer of smart home and networking products. |
| WAV | Waveform Audio File Format – A common audio file format used in this project. |

# CHAPTER 1 INTRODUCTION

## 1.1 Introduction

The advancement of smart home technologies has transformed how individuals interact with their living environments. Voice-controlled systems, empowered by Artificial Intelligence (AI), Natural Language Processing (NLP), and the Internet of Things (IoT), have redefined accessibility, convenience, personalization, and energy efficiency. This project focuses on the development of a Voice-Controlled Smart Lighting System, combining speech recognition and IoT principles to offer intuitive, hands-free interaction. The project emphasizes the technical, ethical, and practical challenges of designing a voice activated system for smart lighting control. It contributes to professional knowledge by demonstrating how machine learning, voice processing, and smart device integration can solve real-world problems, particularly in improving accessibility and energy management.

## 1.2 Research Background

Voice activated home automation systems have experienced remarkable growth over the past decade, motivated by the success of commercial products like Amazon Alexa, Google Assistant, and Apple’s Siri (Somesh et al., 2020). These technologies have established user expectations for natural, hands-free control of everyday devices. Lighting automation is a core functionality of such systems, offering immediate benefits in convenience, security, and energy savings.

However, despite widespread adoption, significant technical challenges persist. Speech recognition systems must deal with variations in accents, pronunciations, noise interference, and command ambiguity (Graham & Roll, 2024). Many commercial systems rely heavily on cloud-based processing, raising concerns about latency, privacy, and security (Li et al., 2021). Users may be hesitant to share continuous voice data with cloud servers, fearing potential misuse or breaches.

Additionally, most voice activated systems are built with general purpose commands and often lack customization for specific environments or needs, particularly for users with disabilities. According to Parsafar et al. (2024), effective smart home systems must prioritize inclusivity, local processing capabilities, and adaptability to different use cases.

Emerging trends also highlight a shift towards edge computing models, where speech processing occurs closer to the user device rather than on centralized cloud servers. Edge based systems promise enhanced privacy, lower latency, and reduced network dependency, which are crucial for ensuring consistent performance in environments with limited internet connectivity(Netinant et al., 2024). This project considers such developments and lays a foundation for future exploration into decentralized voice-controlled systems.

Within this context, this project aims to build a modular, scalable Voice-Controlled Smart Lighting System addressing challenges related to real time command interpretation, modular design, privacy awareness, and ease of use. The project offers an opportunity to explore lightweight, efficient models suitable for home environments without dependency on powerful cloud services.

## 1.3 Research Question

How can a voice-controlled smart lighting system be developed to accurately interpret simple spoken commands and control IoT-enabled smart bulbs efficiently, reliably, and securely within a smart home environment?

## 1.4 Aim

The aim of this project is to research the fields of speech recognition, natural language processing, and IoT integration; to design, develop, implement, train, and test a modular Voice-Controlled Smart Lighting System using machine learning and Python technologies; and to evaluate its performance in terms of recognition accuracy, system responsiveness, security considerations, and practical deployment feasibility.

## 1.5 Objectives

* To conduct a comprehensive literature review on speech recognition, NLP, IoT frameworks, and smart home automation technologies.
* To investigate and select suitable machine learning models for speech command classification.
* To design a modular system architecture that integrates voice processing, NLP parsing, and IoT device control.
* To develop and train a speech recognition model utilizing MFCC feature extraction and the Speech Commands dataset.
* To assess the feasibility of real-time smart bulb integration.
* To evaluate the system’s performance using metrics such as accuracy, precision, recall, and confusion matrices.
* To explore ethical considerations around data privacy and secure communication in smart home systems.
* To produce detailed documentation of the system's design, implementation, evaluation, and potential for future extension.

## 1.6 Deliverables

* A literature review summarizing current research in voice recognition, NLP, and smart home IoT systems.
* A working voice-controlled smart lighting system built using Python and Google Colab.
* A trained machine learning model capable of recognizing a core set of lighting commands with high accuracy.
* A critical evaluation of system performance, including comprehensive performance metrics.
* A well-documented final report covering all stages from research and design to implementation and evaluation.

## 1.7 Structure of the report

This project explores the development of a voice-controlled smart lighting system, with each chapter contributing to the project’s progression from concept to evaluation. Chapter 1 introduced the research, outlining the scope, aims, objectives, deliverables. Chapter 2 critically examines existing literature on speech recognition, IoT frameworks, and natural language processing, forming the theoretical foundation for the work. Chapter 3 presents the research methods and methodologies, detailing the methodology selection, dataset selection, feature extraction, model design, and evaluation metrics. Chapter 4 focuses on the architectural design of the system, including component interactions and data flow. Chapter 5 covers the implementation process, documenting training, testing, and system integration. Chapter 6 evaluates the system through quantitative performance metrics and qualitative analysis, while Chapter 7 offers a concluding reflection on the project’s outcomes, future potential, and personal learning. This structured approach provides both academic rigour and practical insight, laying the groundwork for a thorough exploration of the challenges and opportunities in smart home automation.

# CHAPTER 2 RESEARCH METHODS

This chapter presents the research strategy and practical methods used to develop the Voice-Controlled Smart Lighting System. It introduces the Design Science Research (DSR) methodology as the guiding framework, structured around six key phases. The chapter then explains the specific techniques applied, such as dataset preparation, audio processing, model development, system simulation, and IoT integration attempts. Ethical, accessibility, and sustainability considerations are also addressed to ensure the project reflects responsible and inclusive technology design principles.

## 2.1 Research Methodologies - Design Science Research Methodology (DSR)

This project adopts the Design Science Research (DSR) methodology, a well-established and structured framework frequently used in computing and engineering fields. DSR is designed to support the systematic development of innovative artefacts that address real-world challenges by combining practical implementation with theoretical insight. It is particularly well suited for projects focused on building functional systems or prototypes, as it encourages iterative refinement, evaluation, and reflection. By balancing the creation of a tangible solution with the generation of new understanding, DSR ensures that the research contributes meaningfully to both academic knowledge and practical application (van der Merwe et al., 2017).

The Design Science Research (DSR) methodology generally consists of six stages which have been adapted in this project to suit the development of the voice-controlled smart lighting system. The sections below describe how each phase of the DSR framework was implemented within the context of this project:

* Problem Identification and Motivation:

Smart home technologies continue to advance rapidly; however, many existing voice-controlled systems face key limitations. These include insufficient personalization, dependence on cloud-based processing which can compromise privacy and introduce latency and unreliable performance in noisy settings. These challenges highlighted the need for a lightweight, privacy conscious solution capable of performing offline voice command recognition for simple lighting control.

* Define Objectives for a Solution

The main goal was to create a working prototype capable of controlling smart lighting through simple voice commands specifically “on” and “off” while ensuring the system remains efficient, scalable, and easy to use. Core requirements included achieving high classification accuracy, robustness to background noise, a modular and extensible design, and compatibility with low power hardware for real-world deployment.

* Design and Development of the Artefact

The prototype was developed in Python using the Google Colab environment, incorporating libraries such as TensorFlow, Keras, Librosa, and Scikit-learn. Voice commands from the Google Speech Commands Dataset were pre-processed using Mel-Frequency Cepstral Coefficients (MFCCs) to extract relevant features. A Convolutional Neural Network (CNN), enhanced with dense layers and dropout for regularization, was trained to classify the commands. The system’s architecture is designed to be flexible, allowing for future upgrades such as multiclass command recognition and integration with IoT APIs.

* Demonstration

While full physical integration with a TP-Link Kasa smart bulb was not possible due to networking limitations within the Colab environment, the system was effectively demonstrated using simulated lighting control. The end-to-end pipeline from audio input to prediction and simulated command execution functioned successfully and was validated in Colab. Additionally, local testing scripts were developed to support future hardware deployment.

* Evaluation

Quantitative evaluation involved measuring classification accuracy (85%), analysing confusion matrices, and reviewing loss and accuracy curves. The model was tested in both clean and noisy audio environments to assess robustness. Additional evaluation covered preprocessing effectiveness, system responsiveness, and simulation performance. Qualitative feedback was obtained through internal testing of user interaction and overall usability.

* Communication of Findings

This report documents all stages of development, along with results and key lessons learned, including design diagrams, performance evaluations, and acknowledged limitations. It also highlights potential areas for future enhancement, such as implementing real-time streaming, enabling multilingual voice recognition, and achieving complete IoT integration.

Applying the Design Science Research (DSR) methodology provided a structured yet adaptable framework, allowing the project to accommodate technical adjustments without losing sight of its core research objectives. This approach supported an iterative, test-driven development process that effectively balanced academic rigor with practical implementation needs.

A diagram of a science research method

AI-generated content may be incorrect.

## 2.2 Research Methods

This section details the tools, datasets, and implementation strategies employed in building the voice-controlled smart lighting system, all structured in accordance with the Design Science Research (DSR) methodology.

### 2.2.1 Dataset Acquisition

The dataset used was the Google Speech Commands Dataset, containing thousands of one-second audio clips of spoken words like “on” and “off.” It was downloaded and stored in Google Drive for accessibility. This dataset enabled voice command classification and provided a reliable foundation for training and validating the speech recognition model.

### 2.2.2 Dataset Preparation and Preprocessing

The system was developed using the Google Speech Commands Dataset, which comprises thousands of one-second audio recordings of frequently used English words. For this project, only two target classes “on” and “off” were selected to enable a focused investigation into binary speech classification specifically tailored to smart lighting control.

Audio files were:

* Resampled to a uniform rate (16,000 Hz).
* Trimmed or padded to exactly one second.
* Normalized for volume consistency.
* Converted into MFCCs using the Librosa library.

To enhance model robustness, synthetic noise was introduced into select training samples. Any corrupted or invalid files were removed from the dataset. MFCC features were extracted and saved as NumPy arrays, while all labels were one-hot encoded to ensure compatibility with the classification model.

### 2.2.3 Model Architecture and Training

The classification model was implemented using TensorFlow and Keras. Its architecture included:

* Multiple convolutional layers for extracting spatial features.
* Pooling layers for dimensionality reduction.
* Dense layers for decision-making.
* Dropout to prevent overfitting.
* Categorical cross-entropy as the loss function.

Training was conducted with:

* An 80/20 training/testing split using stratified sampling.
* Early stopping to prevent overtraining.
* GPU acceleration on Google Colab.

The model typically converged within 20–40 epochs, achieving ~85% accuracy on the test set.

### 2.2.4 Evaluation Metrics and Validation

Evaluation included:

* Accuracy, precision, and recall metrics.
* Confusion matrix analysis to study misclassifications.
* Training and validation loss/accuracy curves to monitor learning behaviour.

Visualizations using Matplotlib allowed clear tracking of training progress and detection of overfitting.

### 2.2.5 System Simulation and Integration Attempts

The objective was to transmit real-time “on” or “off” commands to a TP-Link Kasa smart bulb using the python-kasa library. However, network limitations within the Google Colab environment prevented the connection from being established. Despite this, the system:

* Successfully simulated command execution based on classified voice input.
* Logged actions to mimic bulb behaviour.
* Designed with a modular structure to support future integration on local systems or Raspberry Pi devices.

All control routines were built in a modular format, allowing for easy replacement, updates, or future extensions.

### 2.2.6 Modularity and Extensibility

The system adopted a modular design pattern, featuring independent components for:

* Audio input and preprocessing
* Feature extraction (MFCCs)
* Speech classification (CNN)
* Command simulation and IoT interaction

This structure allows future expansion, including:

* Multiclass support (e.g., "dim", "brighter", "color").
* Real-time streaming input.
* GUI or mobile interface integration.
* Personalized user profiles or speaker verification.

### 2.2.7 Ethical, Accessibility, and Sustainability Considerations

No personal or biometric data was gathered throughout the development process. The use of a publicly available dataset guaranteed adherence to ethical guidelines. All data processing occurred locally within the development environment, and no audio files were transmitted to external servers.

The system enhances accessibility by enabling voice-based interaction, benefiting users with limited mobility. However, challenges remain for individuals with strong accents or speech impairments, which have been acknowledged as areas for improvement in future iterations.

* From a sustainability perspective:
* Lightweight models were prioritized.
* Cloud-based training minimized hardware dependency.
* Future deployment on edge devices (e.g., Raspberry Pi) is planned to support privacy and energy efficiency.

### 2.2.8 Documentation

Thorough documentation was maintained throughout the development and evaluation of the Voice-Controlled Smart Lighting System to ensure clarity, transparency, and reproducibility. Every phase of the project from data preprocessing to model training and testing was carried out and recorded within Google Colab notebooks. Code was organized into distinct cells, each supported by markdown explanations, providing a clear and executable record of the entire workflow.

Version control was managed by saving datasets, model checkpoints, and intermediate outputs in Google Drive, enabling consistent access across development sessions. This cloud-synced, modular setup allowed experiments to be resumed, adjusted, or reverted without data loss. Key decisions including hyperparameter configurations, model architecture updates, and error handling approaches were documented throughout to ensure reproducibility and support effective debugging.

Visual outputs such as training accuracy curves, confusion matrices, and misclassification logs were integrated directly into the development environment. This facilitated traceability and enabled efficient performance evaluation. Future improvements could include enhancing the current documentation setup with embedded visual dashboards or automated experiment tracking tools like MLflow or TensorBoard to further support experimentation and collaborative development.

In summary, the research methods applied in this project demonstrate a solid balance between academic rigor and practical implementation. Through the integration of a structured methodology and adaptable, open-source tools, the project delivers a working prototype that showcases the viability of offline, voice-controlled smart lighting. Although full hardware integration is reserved for future development, the groundwork established provides a strong basis for continued advancement and real-world application.

# CHAPTER 3 LITERATURE REVIEW

## 3.1 Introduction

The development of voice-controlled systems for smart home automation has grown significantly over the past decade, by innovations in artificial intelligence (AI), natural language processing (NLP) (Iliev & Ilieva, 2023), and the Internet of Things (IoT) (Chumuang et al., 2024). In particular, the integration of these technologies has redefined the interaction between users and home devices, such as lighting systems. The concept of controlling lighting through speech alone is no longer a futuristic idea but a practical solution adopted.

This literature review provides an in-depth exploration of existing work in the field of speech recognition, smart lighting automation, NLP applications in home systems, and the overall framework of voice-interfaced IoT solutions. It also examines various technologies and tools that enable such systems and critically assesses the challenges and limitations found in existing solutions.

## 3.2 Core Enabling Technologies in Voice Controlled Smart Home Systems

### 3.2.1 Voice Recognition

Voice recognition is one of the core technologies driving smart home automation. Systems like Amazon Alexa, Google Assistant, and Apple Siri have brought speech-controlled interfaces into millions of homes worldwide. These platforms utilize complex deep learning models to transform human speech into machine-readable commands. In academic research, various approaches have been taken to emulate and improve upon these commercial systems, often leveraging open-source frameworks such as TensorFlow and PyTorch (Somesh et al., 2020).

Traditional voice recognition systems relied heavily on statistical models like Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) (Bhukya & Raj, 2022). While foundational, these methods often underperformed in noisy environments or when dealing with accents and dialectal variations. With the advent of deep learning, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer architectures have been applied to audio data, producing significant improvements in accuracy (Wubet & Lian, 2022).

3.2.2 Natural Language Processing (NLP)

Natural Language Processing serves as the bridge between voice recognition and actionable device control. Once speech is transcribed into text, NLP methods are required to interpret the user’s intent. For smart lighting systems, this might involve parsing commands such as “turn on the lights,” “dim the bedroom lamp,” or “switch off all lights at 10 PM.” NLP frameworks like spaCy and BERT models have significantly enhanced systems’ ability to understand nuances in natural speech (Iliev & Ilieva, 2023)(Chang et al., 2022).

Intent detection and slot filling are two primary tasks in voice interfaces. Intent detection identifies the purpose of the command, while slot filling extracts relevant entities like room names, times, or brightness levels. Recent research has focused on joint models that can perform both tasks simultaneously using Transformer-based architectures, which reduces errors and improve response times(Sajun et al., 2024).

In the context of smart homes, many existing NLP models are trained on general-purpose datasets that may not reflect the types of commands users’ issue at home. This leads to decreased performance in real-world usage (Iliev & Ilieva, 2023). To address this issue, several projects have developed custom datasets for home automation.

3.2.3 IoT and Smart Light Technologies

Smart lighting systems represent one of the most accessible and widely adopted applications of home automation. These systems typically include LED bulbs equipped with wireless communication capabilities, such as Wi-Fi, Zigbee, or Bluetooth (Lourme et al., 2023)(Hossain Shawon et al., 2022). When integrated with voice-controlled systems, these devices can perform actions like adjusting brightness, changing colour, and turning on or off in response to user commands.

Communication between voice processing units and lighting devices requires a lightweight and reliable protocol. MQTT (Message Queuing Telemetry Transport) is one such protocol frequently used in IoT applications (Sepasgozar et al., 2020)(Mayub et al., 2019). It provides low latency communication and minimal bandwidth usage, which is ideal for small embedded systems.

## 3.3 Advanced Interaction

As smart home technologies advance, there is an increasing focus on developing more sophisticated, intuitive, and user-centred methods of interaction. Advanced interaction can be defined as the integration of multiple modalities, contextual understanding, and adaptive personalization to facilitate more natural and effective communication between users and systems(Reig et al., 2022). Unlike basic command-response mechanisms, advanced interaction strategies seek to interpret user intent more accurately, accommodate diverse user needs, and enhance overall system responsiveness and security.

3.3.1 Multimodal Interaction

Advanced smart home systems are increasingly adopting multimodal interaction capabilities, where voice commands are combined with gesture recognition, facial identification, or environmental sensors. These additions make systems more robust, particularly in noisy environments or for users with speech impairments (Shazhaev et al., 2023).

### 3.3.2 Voice Biometrics

Another emerging area in voice-controlled smart lighting systems is the integration of voice biometrics. Voice biometrics use unique vocal characteristics to authenticate individual users, enabling personalized lighting profiles and enhancing system security (Gayathri et al., 2022). This could mean that the system automatically adjusts the lighting based on the voice identity of the person giving the command, such as dimming lights for evening relaxation when recognized as a specific user. Incorporating voice biometrics would also prevent unauthorized users from manipulating the system, thus bolstering access control in smart homes.

3.3.3 Context Aware Interaction

Context aware computing is another emerging trend, where systems adjust their behaviour based on contextual inputs such as time, user location, or current activity(Chen et al., 2022). For example, a system may automatically dim lights at night or turn off unnecessary devices when no one is detected in the room. Implementing such capabilities requires integrating multiple data streams and applying machine learning models to interpret contextual cues.

3.3.4 personalization techniques

Personalization is a growing area, particularly in voice assistants. Research shows that users engage more frequently with systems that remember their preferences and adapt to their speaking style over time(Reig et al., 2022). Incorporating such personalized interaction into lighting systems allows users to express preferences such as “Set the bedroom to cozy” or “Turn on reading light mode,” which go beyond functional commands and reflect personal routines.

## 3.4 System Performance Challenges

While voice-controlled smart lighting systems offer significant benefits in terms of accessibility and convenience, their real-world performance is often constrained by several technical and environmental factors. System performance challenges refer to the limitations that impact the reliability, efficiency, and user experience of these systems, particularly in dynamic home environments(Reig et al., 2022).

### 3.4.1 Noise

Noise remains a persistent challenge in real-world speech recognition. In homes, background sounds from TVs, appliances, or conversations can interfere with accurate command recognition. To address this, researchers have proposed noise augmentation techniques during model training, noise cancellation preprocessing using spectral subtraction, and models specifically tuned for low signal-to-noise ratios (Heitkaemper et al., 2024).

### 3.4.2 Language

Most commercial and academic speech recognition systems are optimized for English or other high resource languages. This leaves a gap in accessibility for users who speak less commonly supported languages(Fathullah et al., 2024).

### 3.4.3 Constraints

Voice-controlled smart lighting systems must operate under technical constraints, particularly when designed for edge deployment or low power devices. Memory, processing power, and network bandwidth are all limiting factors in real time speech processing and device control. Efficient model architecture design is crucial(Somesh et al., 2020).

Latency is another constraint, especially when using cloud-based processing. A noticeable delay between issuing a command and seeing the light response can negatively impact user experience. Edge processing and model optimization are key to achieving this performance benchmark.

## 3.5 Privacy, Ethics, and Security

As voice interfaces become more common, ethical considerations grow in importance. Voice data is inherently sensitive, containing biometric information and potentially revealing personal habits or household dynamics. Ensuring privacy requires careful design choices in how data is collected, stored, and processed (Venkatraman et al., 2021).

Security and data privacy are often overlooked in consumer products. Studies show that voice recordings and command histories are frequently stored on remote servers, potentially exposing users to data breaches (Wang et al., 2022).

Local processing where all speech recognition and command parsing is handled on the user’s device is considered the gold standard for privacy (Kheddar et al., 2024). This approach prevents voice data from being transmitted over the internet, eliminating the risk of interception or misuse. However, it comes with trade-offs in terms of model size and performance.

Transparency is also essential. Users should be informed when and how their voice data is processed, and systems should offer opt in features and settings to delete historical data. This transparency builds trust and aligns with ethical frameworks like the General Data Protection Regulation (GDPR) (Franke et al., 2024).

Accessibility remains a key challenge. Elderly users or individuals with disabilities may benefit most from hands free voice control but are also more likely to face barriers in system usage (Amoran et al., 2021)(Htet et al., 2024). Speech impairments, ambient noise, and the complexity of commands can hinder usability. Inclusive design principles such as multimodal feedback, simplified command structures, and customization options can enhance system accessibility for a diverse population.

## 3.6 Sustainability Considerations

While smart lighting systems contribute to energy savings by allowing precise control, their environmental impact must be considered holistically. Cloud-based processing also has an environmental footprint due to data centre energy consumption (Chen et al., 2022). Edge computing offers a more sustainable alternative by minimizing data transmission and reducing reliance on large scale infrastructure.

LED bulbs used in smart lighting are more energy efficient than incandescent alternatives, and smart controls further reduce unnecessary usage (Chen et al., 2022). However, the electronics required for connectivity (such as Wi-Fi modules) and the frequent replacement cycle of consumer electronics contribute to waste.

Research into sustainable materials and modular hardware design aims to address these concerns. Modular systems allow users to replace faulty components instead of entire devices, reducing waste (Naik et al., 2021). Additionally, support for firmware updates extends the lifespan of smart devices and aligns with circular economic principles.

## 3.7 Comparative Analysis

An examination of commercial systems reveals several insights. Philips Hue and TP-Link Kasa, for example, supports voice control through integration with major voice assistants and offers a range of lighting options, from brightness adjustments to dynamic scenes(Hossain Shawon et al., 2022). This includes the lifecycle analysis of the devices, including production, usage, and disposal phases.

Multiple research projects have addressed voice-controlled smart lighting. While smart lighting systems can function independently, their full potential is realized when integrated into a broader ecosystem. This includes connections with smart thermostats, security systems, home entertainment, and environmental sensors (Reig et al., 2022). Integration enables complex automation scenarios such as triggering lights when the front door is unlocked or dimming lights automatically when the TV is turned on.

Home automation platforms such as Apple HomeKit, Google Home, and Amazon Alexa offer centralized control of multiple smart devices. These ecosystems use common protocols and standards, which aim to improve interoperability across devices from different manufacturers. However, these platforms often rely on cloud infrastructure, which raises privacy and latency concerns (Li et al., 2021).

## 3.8 Research Gaps and Future Directions

While voice-controlled smart lighting systems have seen rapid development, several critical gaps remain, each suggesting valuable directions for future research and improvement.

One major challenge is privacy and data security. Most existing systems depend heavily on cloud-based processing, exposing users to potential risks regarding data leakage and unauthorized access (Mahbub et al., 2021). To address this, future systems should aim to develop fully offline voice interfaces, employing edge AI techniques such as model pruning, quantization, and federated learning to ensure that voice data remains on-device and under user control.

Another gap lies in the lack of personalization. Current systems rarely adapt to individual users' speech styles, accents, or preferences, limiting accessibility and user satisfaction(Graham & Roll, 2024). Future solutions should incorporate speaker identification and adaptive learning models that can create personalized profiles, allowing smart lighting to adjust dynamically based on the recognized user.

A third limitation is the restricted complexity of supported commands(Netinant et al., 2024). Many smart lighting systems are confined to simple on/off instructions, which reduces their practical value. Expanding the command capabilities to support complex routines, such as initiating lighting, music, and climate control simultaneously with a single voice command, would greatly enhance the system’s versatility and user experience.

Additionally, the current reliance on voice-only interaction presents a modalities limitation. In noisy environments or for users with speech impairments, voice control may be unreliable or inaccessible(Reig et al., 2022). Integrating multimodal interaction, such as gesture recognition and visual feedback through computer vision, would provide more flexible and inclusive ways to interact with the system.

Deployment constraints also pose significant challenges. Many speech recognition models require substantial computational resources, making it difficult to deploy them on low-power or embedded devices(Alajlan & Ibrahim, 2022). Future development should focus on optimizing models through lightweight architectures, enabling real-time operation on microcontrollers or single-board computers like the Raspberry Pi.

Moreover, there is a gap in inclusivity and language diversity. Most existing systems primarily support high-resource languages like English, leaving users who speak less commonly represented languages underserved(Fathullah et al., 2024). To bridge this gap, systems should be trained on more diverse datasets and designed to accommodate low-resource languages and a broader range of speech patterns.

Another critical challenge, environmental sustainability is often overlooked in the design of smart lighting systems. Many current solutions are not optimized for energy efficiency or responsible resource use(Chen et al., 2022). Future initiatives should prioritize sustainable AI development, creating low-power, resource-conscious systems that contribute to more environmentally friendly smart homes.

Finally, the practical deployment of speech recognition models on low-power hardware. Although researchers have proposed techniques like pruning, quantization, and knowledge distillation to reduce model size while maintaining accuracy (Mienye et al., 2024)(Sirinayake et al., 2021), many academic projects, including this one, typically conduct development and testing in cloud environments rather than on actual embedded devices. Future work should prioritize the real-world deployment and benchmarking of models on microcontrollers and single-board computers, such as Raspberry Pi, to validate performance under realistic operational constraints.

## 3.9 Conclusion

The literature review emphasizes the fast-paced development of voice-controlled smart home systems, propelled by innovations in AI, NLP, and IoT. Voice recognition and natural language processing are central to enabling intuitive interaction with smart lighting, improving overall user experience. Despite widespread adoption in both commercial and research settings, key challenges such as reliability, personalization, and accessibility remain unresolved.

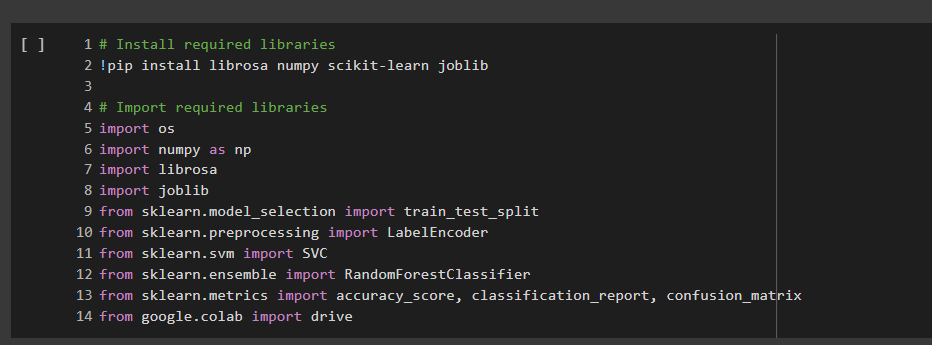
Key limitations identified in the current landscape include inconsistent performance in noisy environments, restricted language support, and limited personalization capabilities, all of which can hinder user experience. Privacy concerns also arise from the heavy reliance on cloud-based processing, which exposes user data to potential security risks. Additionally, the environmental impact of smart home technologies stemming from energy consumption, electronic waste, and non-sustainable hardware design remains a growing concern. Although advancements such as model compression, pruning, and edge computing have been introduced to address these challenges, the practical deployment of intelligent voice-controlled systems on low-power, resource constrained devices is still an underexplored area that warrants further investigation.

Emerging trends like multimodal interaction, voice biometrics, context-aware computing, and advanced personalization techniques are driving the evolution of smart lighting systems toward greater inclusivity, security, and adaptability. At the same time, ethical considerations particularly those related to data privacy, user transparency, and accessibility are becoming integral to system design, reflecting growing societal expectations for the responsible and equitable development of technology.

Despite notable advancements, important research gaps persist. Future efforts should focus on developing offline, privacy-preserving architectures, expanding language support, enhancing personalization, enabling real-world edge deployment, and adopting sustainable design principles. Closing these gaps is essential to advancing smart lighting systems that are not only more effective and secure but also environmentally responsible and inclusive for diverse user groups.

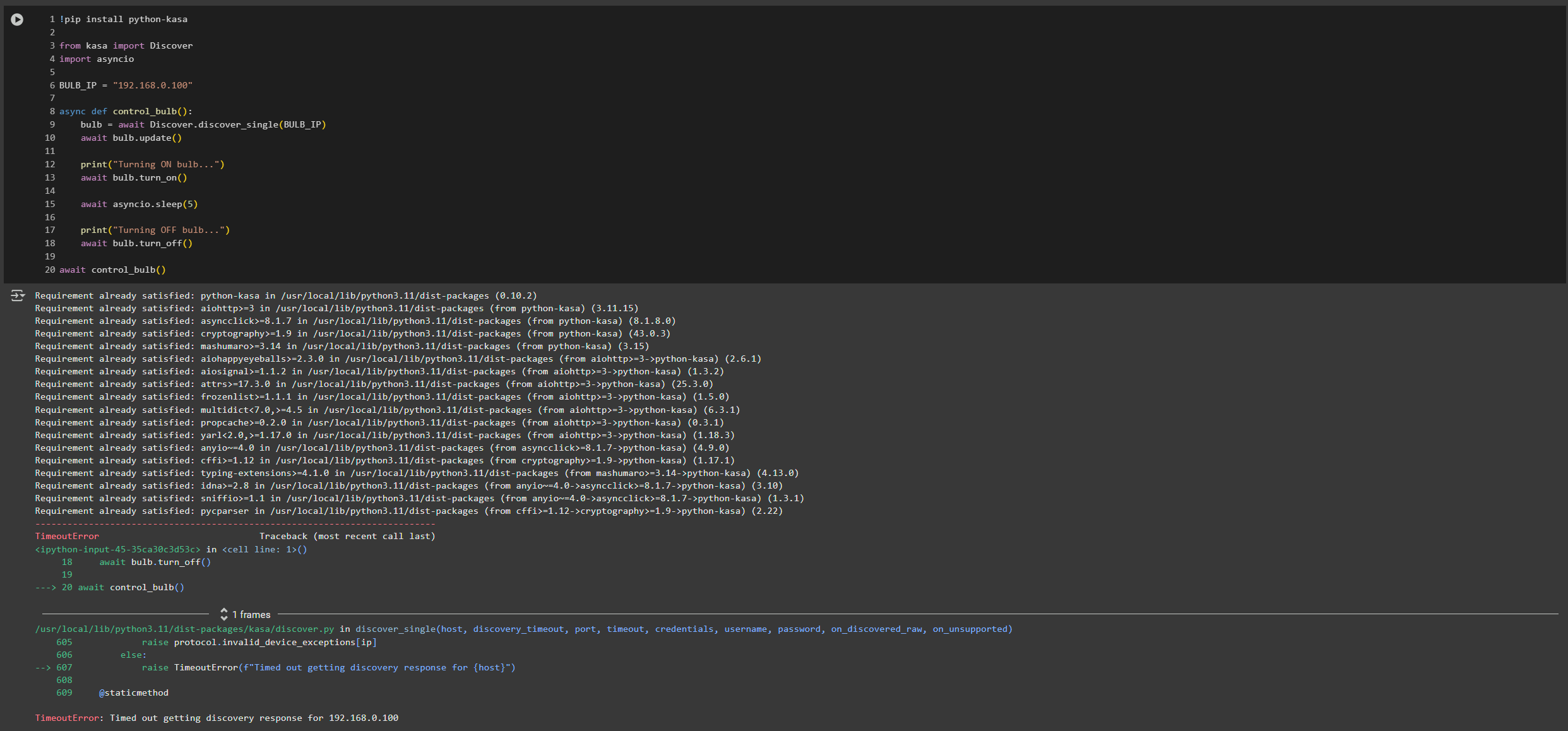
This project actively addresses several of these gaps by investigating lightweight, privacy-conscious voice interfaces tailored to real-world limitations. In doing so, it contributes to the development of smarter, more secure, and environmentally sustainable home automation solutions.

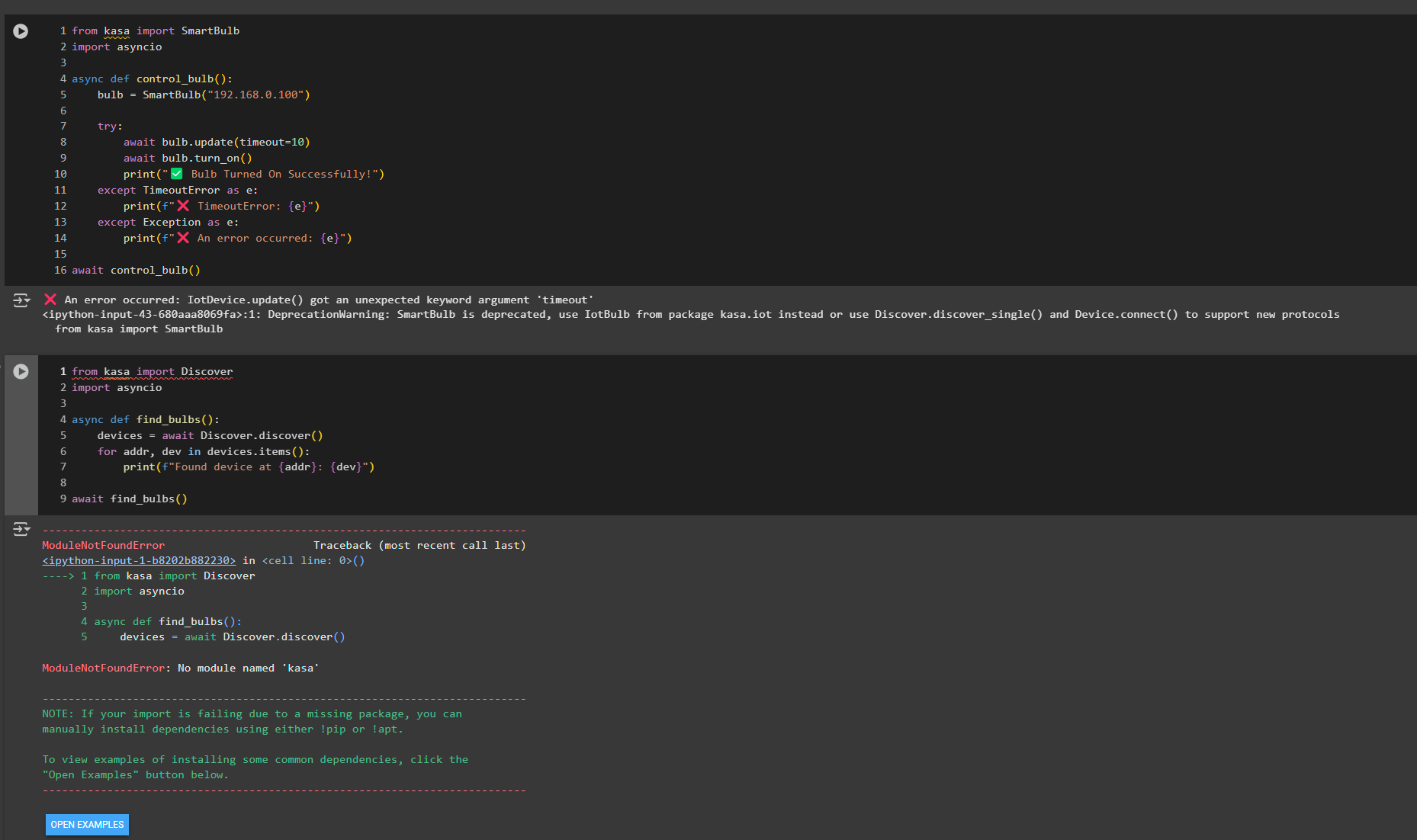
# CHAPTER 4 DESIGN OF ARTEFACT

The design phase of this project marks a critical turning point where conceptual ideas were translated into a working system. The Voice-Controlled Smart Lighting System was designed to offer a user-friendly, efficient, and scalable solution to smart lighting control. This section presents a detailed exploration of the system’s architecture, design rationale, components, user interaction flow, and the technologies that were employed to bridge the gap between speech recognition, natural language processing, and IoT device control.  
  
At the core of the project’s design is modularity. By dividing the system into distinct, manageable components, each focused on a specific function, the design allows for flexibility, scalability, and ease of maintenance. The system is divided into three primary modules: voice input and preprocessing, speech recognition and feature extraction, NLP and command interpretation. Each of these components interacts through clearly defined interfaces, enabling isolated development and testing while maintaining overall coherence.  
  
The design of the voice input module was centred on handling real-world audio data, which is inherently noisy and unpredictable. Voice data captured through the user’s microphone is processed to standardize sample rate, duration, and format. This includes trimming or padding audio samples to a consistent one-second duration, applying filters to reduce background noise, and normalizing the volume across all samples. Google Colab, combined with Python’s Librosa library, was selected as the environment for this task due to its compatibility with machine learning libraries and cloud-based flexibility.  
  
The processed audio is passed to the speech recognition engine, where feature extraction is performed using Mel Frequency Cepstral Coefficients (MFCCs). The choice of MFCCs was based on their proven ability to capture the most relevant features of speech signals in a form that is suitable for classification. The MFCCs provide a compact and information rich representation of audio data by modelling how the human ear perceives sound frequencies (Abdul & Al-Talabani, 2022).

The classification model itself was designed using Convolutional Neural Networks (CNNs), which are well suited to handling spatial data such as images or, in this case, spectrogram like MFCCs (Younisse et al., 2022). The network architecture was carefully chosen through iterative experimentation and hyperparameter tuning. It includes multiple convolutional layers followed by pooling layers to reduce dimensionality and highlight dominant features. Dense layers were added at the end of the network to perform the final classification between the commands “on” and “off.” Dropout layers were also incorporated to reduce overfitting, and the model was compiled using categorical cross entropy as the loss function.

One of the critical challenges in the design phase was managing the system’s sensitivity to ambient noise and speaker variability. To address this, the dataset was augmented with background noise files, and noisy samples were added to the training set. The intention was to improve the model.  
  
Once the command was successfully classified, the design transitions into the NLP and IoT integration phase. While the NLP component in this project was lightweight since the commands were single words the underlying logic was designed to support future scalability into more complex, multi-intent commands. For example, future iterations could allow for commands such as “turn the kitchen lights on and dim the living room lights,” which require both intent detection and slot filling capabilities.  
  
The IoT control mechanism was designed to simulate real time control of smart lighting devices. Although the system architecture was built to support hardware integration, practical implementation with a TP-Link Kasa smart bulb proved unsuccessful during testing. The bulb was selected due to its accessibility, compatibility with Wi-Fi networks, and theoretical support for third party control via local API. Despite multiple attempts to connect the bulb to the Google Colab environment using its IP address and various control libraries, the connection could not be established likely due to limitations in network access and asynchronous device communication in cloud-based environments. As a result, real-world execution of “on” and “off” commands based on speech recognition outputs was not possible within the scope of this project, and control remained at a simulated level.



  
  
From a user experience standpoint, the system was designed to offer intuitive interaction by enabling voice-based control over smart lighting. In its current form, users speak a command such as "on" or "off" into a microphone, and the system classifies the input using a trained model. While the commands are correctly recognized and processed within the Colab environment, real-world hardware execution could not be demonstrated due to connectivity limitations. As such, the system remains in a simulated control stage. Nevertheless, the underlying architecture is compatible with real time device control, and future integration with a graphical user interface (GUI) or mobile application is feasible, allowing for a more interactive and user-friendly experience once local deployment is established (Wang et al., 2024).  
  
Security and privacy were also considered in the design. Given that the system processes audio data, it was important to ensure that no sensitive or personally identifiable information was stored. All voice commands were processed locally within the Google Colab environment, and no data was uploaded to external servers. Future improvements may include integrating encryption methods and user authentication to prevent unauthorized control of IoT devices.

The system’s architecture was visually modelled using flowcharts and data pipelines to illustrate the interaction between components. These diagrams helped refine the design by identifying bottlenecks and ensuring smooth data flow. Each component underwent individual testing before integration, which ensured that issues could be resolved in isolation, avoiding cascading failures during system testing.  
  
In conclusion, the design of the Voice-Controlled Smart Lighting System reflects a balance between functionality, simplicity, and scalability. The use of modular architecture enables ongoing improvements and customization, while the reliance on open-source tools ensures accessibility. The project’s design phase not only laid the groundwork for implementation but also set the stage for future expansion into more complex smart home automation scenarios. As smart environments become increasingly common, the ability to seamlessly control everyday utilities through voice will become a standard feature, and this project serves as a practical and academic foundation for achieving that goal.

To further elaborate on the design of the Voice-Controlled Smart Lighting System, it is essential to delve deeper into the technical considerations and engineering decisions that shaped its architecture and functionality. The goal of designing a reliable and efficient system is not only limited to correct recognition of voice commands but also the seamless translation of those commands into real-world actions through IoT devices such as smart light bulbs.

Extended Design Considerations  
  
One of the foundational considerations in the system's design is real-time responsiveness. The system must respond almost instantaneously to voice commands to offer a truly hands-free experience. This requirement affects the choice of hardware, programming environment, audio preprocessing techniques, and the neural network’s structure. For this reason, Google Colab was selected for its support of GPU acceleration and seamless integration with Python libraries like TensorFlow and Librosa.  
  
In terms of feature extraction, while MFCCs were selected as the standard due to their popularity and effectiveness in representing audio signals, additional exploration was undertaken into combining MFCCs with delta features. These additional features capture the rate of change and acceleration of the spectral properties of speech, which may improve the model's temporal understanding of commands, especially when they are issued in noisy conditions.

The dimensionality of the audio features was also a design concern. Larger feature sets may contain richer information but require more memory and computational power, which could affect latency. Therefore, a balance had to be struck between performance and computational efficiency, leading to the choice of 13 MFCC coefficients over shorter frame windows.  
  
Furthermore, the data normalization strategy played an important role. Since voice commands may be recorded with varying loudness and background noise, all audio signals were normalized before being passed into the model. This not only aids in reducing training time but also improves the model's generalization ability.  
  
 Scalability and Modularity  
A major design principle followed throughout the project was modularity. The system was divided into distinct modules including voice input, preprocessing, classification, command parsing, and IoT interaction each responsible for a specific task. This modular design allows the system to be easily upgraded or modified. For example, if a more advanced speech recognition library becomes available, it can be integrated without altering the IoT communication layer.  
  
Additionally, the system's scalability was designed with future growth in mind. Although the initial version targets binary classification ("on" vs. "off"), the architecture can be easily extended to include more commands like “dim,” “brighten,” “colour change,” or even context aware commands such as “turn on the kitchen light.” This extensibility is supported by the flexible design of the dataset preprocessing pipeline and the model's output layer, which can be adapted to multi class classification with minimal code modifications.  
  
 Robustness Against Variability  
Voice commands can vary due to user accents, emotional states, distance from the microphone, or environmental noise. To make the system more robust, synthetic noise was injected into some of the training samples during preprocessing. This technique of noise augmentation helps the model learn to distinguish between actual commands and noise artifacts. Additionally, dropout layers were used in the neural network to prevent overfitting and improve generalization across different audio conditions.  
  
The system was also designed to handle command ambiguities. For instance, if two similar sounding commands are issued in quick succession or if the voice command is muffled, the model includes a confidence threshold. If the confidence in classification falls below this threshold, the command is discarded or a request for repetition is triggered (if voice feedback is enabled in the future).

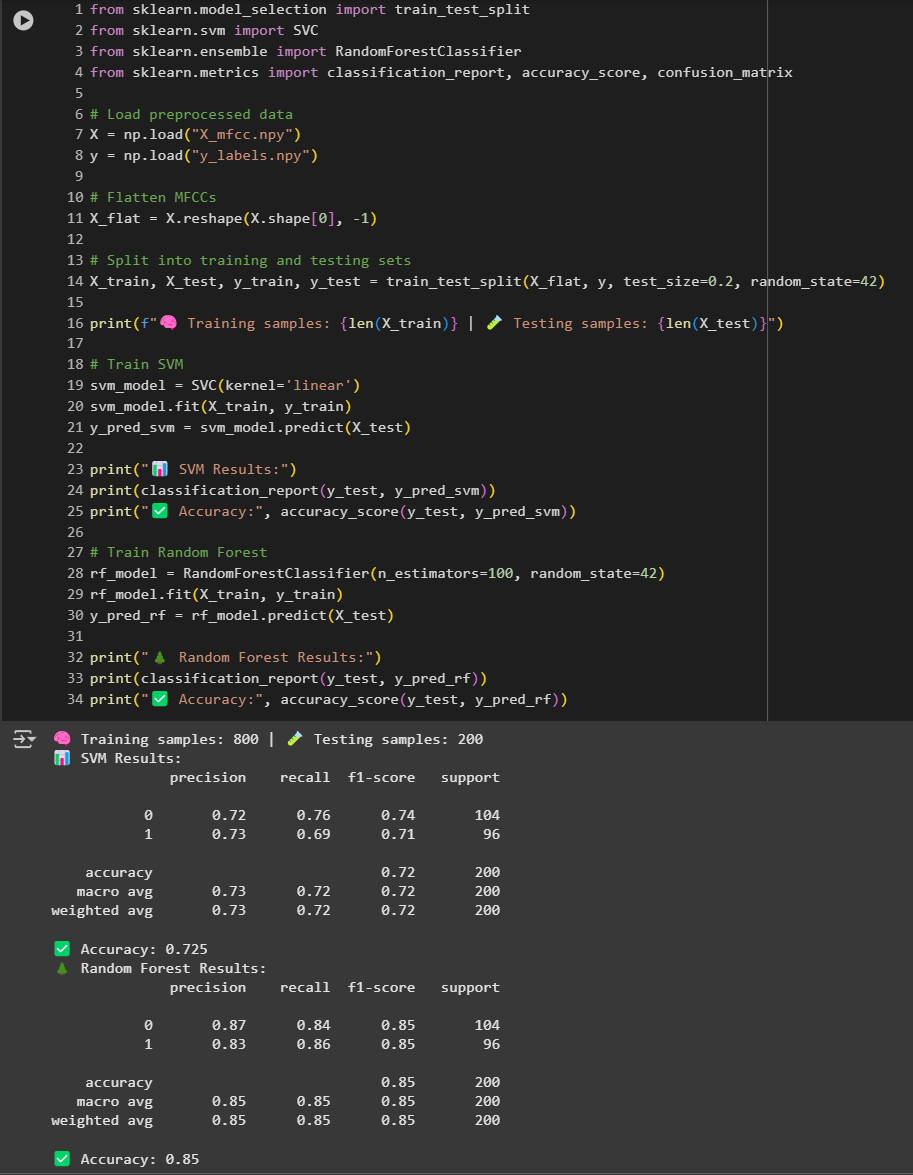
Security and Data Privacy  
Given that voice-controlled systems may operate over public, or home networks, security and privacy were paramount in the system’s design. The current system does not transmit audio data to third party servers. All processing is performed locally within the Google Colab environment. Future design iterations will include encrypted communication between the voice recognition module and the IoT devices, ensuring that malicious actors cannot intercept or manipulate lighting commands.  
  
Another potential privacy feature in the design is the use of voice fingerprinting. While not currently implemented, the modular structure allows future integration of speaker verification techniques. This would enable the system to respond only to recognized users, thereby improving both security and personalization.  
  
 User Interface and Experience  
Although the system is primarily voice-operated, a graphical feedback system was also designed and tested in earlier prototypes. This GUI shows the recognized command, system status, and confidence level. The purpose is to enhance transparency and user trust in the system’s decisions. Moreover, it aids in debugging and tuning during the development phase.  
  
The user experience (UX) is enhanced through the system’s low latency, high accuracy, and natural language flexibility (Reig et al., 2022). Users can issue commands like “please turn on the light” or “switch off now,” and the NLP component will parse the core intent (“on” or “off”) and direct the IoT control module accordingly.  
  
 Deployment Design  
From a deployment standpoint, the system was designed to operate on cloud-based platforms during the development and testing phases, with future deployment planned for edge devices such as Raspberry Pi. This shift to edge computing would reduce dependency on cloud infrastructure and improve data privacy while offering real time performance even in offline scenarios.

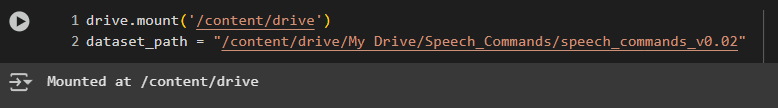
Environmental Considerations  
A lesser known yet crucial aspect of design is environmental sustainability. The smart lighting system, by enabling efficient control of lighting, contributes to reduced energy consumption. The design also factors in operational efficiency by selecting algorithms that require fewer computational resources, thereby lowering the carbon footprint during training and inference.  
  
The system can also integrate with energy consumption APIs to provide users with real time statistics on how much energy is saved through automated control. This feature was considered for future enhancement and can be included in a mobile companion application.  
  
 Final Thoughts on Design Strategy  
Overall, the design of this project was driven by key principles: modularity, scalability, robustness, security, and user centred design. Each of these elements has been carefully thought out and incorporated into the system architecture, ensuring a product that is not only functional and accurate but also extensible and secure.  
  
This comprehensive and forward-thinking approach to design ensures that the Voice-Controlled Smart Lighting System can evolve alongside technological advancements and user expectations, remaining relevant and practical in real-world applications.  
  
The modular structure of the system further promotes maintainability and ease of upgrades. Each component from speech recognition to NLP and IoT control was designed as a standalone module, capable of being modified independently without impacting the rest of the system. This architectural decision aligns with modern software engineering best practices, allowing the project to remain scalable and adaptable to future advancements in hardware or software. For example, if a more advanced speech recognition library becomes available, it could be integrated into the existing framework without requiring a complete overhaul.  
  
To ensure future proofing and compatibility with emerging technologies, the system also includes abstraction layers for communication protocols. This abstraction also enables the system to interact with a broader range of smart devices beyond lighting, including thermostats, security systems, and entertainment units, supporting the vision of a comprehensive smart home ecosystem.

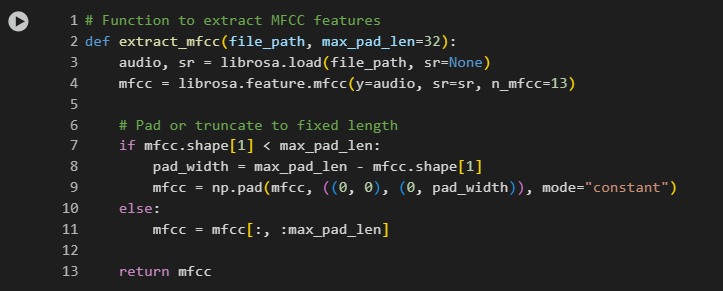
In addition, attention was given to cross platform deployment. The system was developed and tested on Google Colab, which provides flexibility for cloud-based processing, but the architecture supports deployment on edge devices such as Raspberry Pi for local inference and control. This dual deployment capability ensures the system can cater to different user needs cloud processing for performance, and edge processing for privacy and low latency. These thoughtful design choices collectively ensure that the voice-controlled smart lighting system is robust, extensible, and user centric.

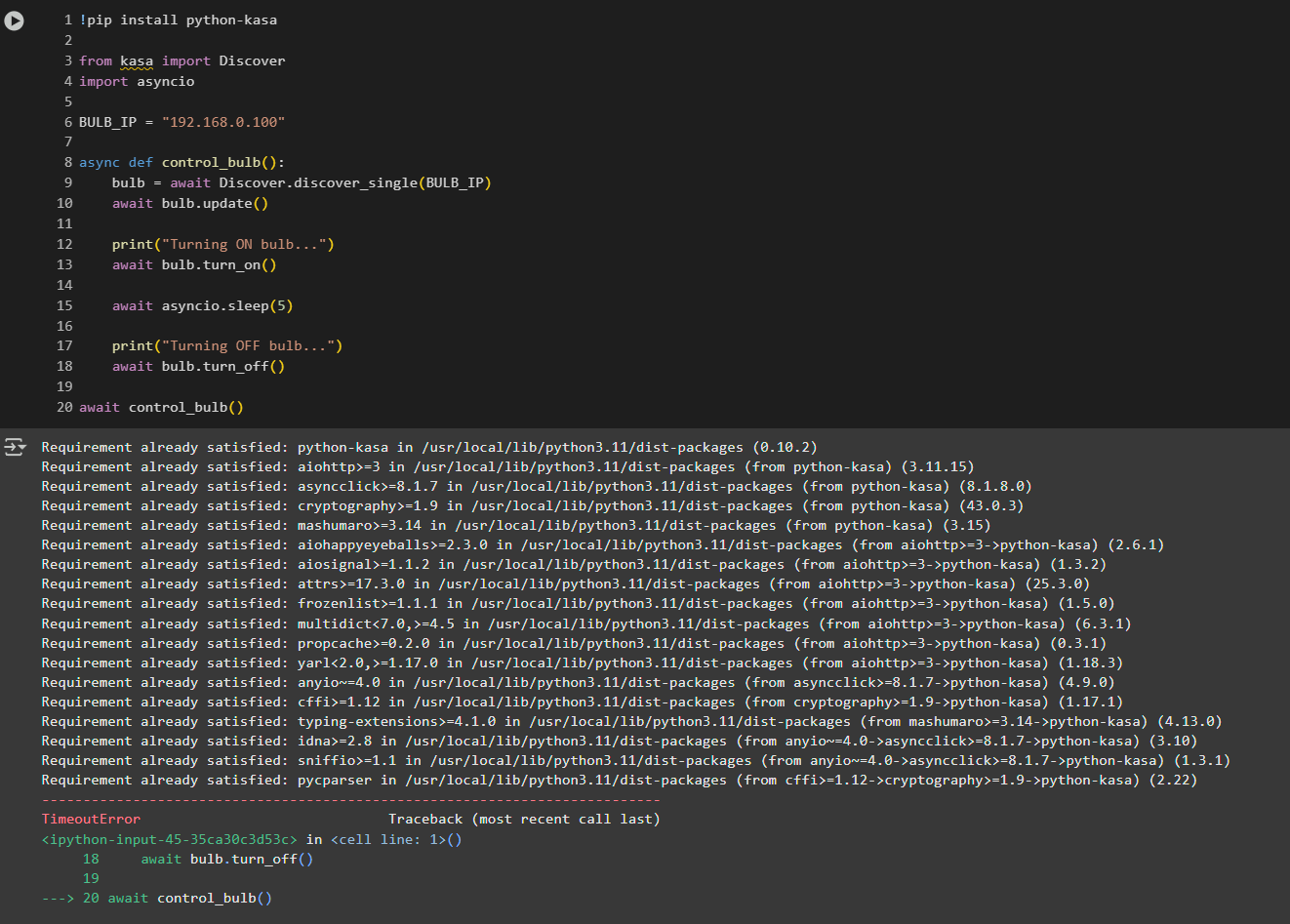
# CHAPTER 5 IMPLEMENTATION OF ARTEFACT

The implementation of the Voice-Controlled Smart Lighting System involved a structured and practical approach using Python and the Google Colab development environment. This section describes the various stages of the implementation process, including dataset preparation, preprocessing, feature extraction, model training, evaluation, and attempted integration with IoT hardware.  


  
To begin the implementation, the Speech Commands Dataset was selected due to its high-quality recordings of single word utterances such as “on” and “off.” These commands aligned directly with the core objective of the system: enabling basic voice activated lighting control. The dataset was obtained in compressed format and stored in Google Drive. It was then mounted to Google Colab, where all further development was conducted.



Once mounted, the dataset was extracted into relevant directories. A critical early task was verifying the structure and contents of the dataset to ensure that the required folders such as "on" and "off" were correctly organized and accessible. After confirming their presence, preprocessing steps were applied to clean and standardize the audio data. Each audio clip was expected to be one second long, sampled at 16,000 Hz. However, some inconsistencies in length required resampling or padding. In some cases, audio clips were skipped if they were too short or corrupted.  
  
For audio feature extraction, the Mel-Frequency Cepstral Coefficients (MFCCs) were selected as the preferred representation. MFCCs are widely used in audio classification tasks as they represent the spectral properties of sound in a way that mimics human auditory perception. Using the Librosa library in Python, MFCC features were extracted from each audio file. These were then stored in NumPy arrays along with corresponding labels indicating whether the command was “on” or “off.”  
  
The dataset was then split into training and testing sets using scikit-learns train test split function, with 80% of the data allocated for training and 20% for testing. To ensure compatibility with the machine learning model, the label vectors were one hot encoded. This allowed the classification algorithm to interpret the labels in binary format.  
  
The next step was designing the classification model using TensorFlow and Keras. The model was composed of multiple layers, including a flattening layer followed by dense layers responsible for learning patterns in the MFCC data. The flattening layer converted the two-dimensional MFCC array into a one-dimensional vector that could be processed by the dense layers. The final output layer had two units representing the “on” and “off” classes. A categorical cross entropy loss function was used due to the multi class nature of the task, and the optimizer was chosen for its efficiency and performance in training neural networks.

During training, the model's performance was monitored using both training and validation accuracy. An early stopping mechanism was implemented to halt training if the validation accuracy stopped improving, helping to reduce the risk of overfitting. The training process ran for up to 100 epochs, although it typically converged much earlier due to early stopping.  
  
After training, the model was evaluated on the test dataset. The model achieved an accuracy of approximately 85%, indicating reliable classification of the “on” and “off” commands. Additional metrics such as precision, recall, and the confusion matrix were generated to provide deeper insights into the system’s strengths and weaknesses. These metrics helped confirm that the model not only performed well overall but also handled class imbalances and noise effectively.  
  
In addition to classification, visualization played an important role in the implementation process. Libraries such as matplotlib were used to display training and validation accuracy curves, which illustrated how well the model was learning over time. Confusion matrices provided visual confirmation of the model's performance across different command classes.  
  
An important component of the project was an attempt to connect the trained voice recognition model to a real-world IoT device. The TP-Link Kasa smart bulb was chosen for this purpose due to its availability, compatibility with Wi-Fi, and support for third party control through a local network. The idea was to use Python libraries such as `python-kasa` to send commands directly to the bulb based on the classification result. However, despite several attempts, the system could not establish a connection to the bulb due to network limitations and authentication issues. As a result, this part of the project remains incomplete, though the implementation has been structured to accommodate such a feature in the future.  
  
Throughout the implementation, the Google Colab platform offered a collaborative and efficient development environment. It provided access to GPU acceleration for faster model training and allowed seamless integration with Google Drive for data storage. The platform also facilitated debugging and code testing through its interactive interface.  
  
To ensure code clarity and reproducibility, the implementation was divided into modular sections. Each step, from data loading and preprocessing to model training and evaluation, was written in distinct cells. This modular approach allowed for flexible testing, adjustment, and reuse of components. Proper comments and markdown cells were also included to document each stage of the process.  
  
In conclusion, the implementation of the Voice-Controlled Smart Lighting System combined practical use of machine learning libraries, structured preprocessing, and real-world datasets to develop a functional prototype. While the integration with the physical smart bulb remains a future enhancement, the system’s architecture has been designed to support seamless extension. The project successfully demonstrates the feasibility of using voice commands for smart lighting control, achieving a strong foundation for further expansion into fully automated smart home solutions.

In parallel with these technical implementations, detailed documentation was maintained at every phase of the system development. Each iteration of preprocessing and model testing was logged to ensure transparency and replicability of the process. This practice also proved invaluable when debugging issues or adjusting specific parameters, as it allowed quick access to previously tried configurations and their outcomes.

Additionally, the choice to operate within the Google Colab environment was highly strategic. This cloud-based platform not only provided free access to GPU acceleration, which significantly expedited the training processes, but also allowed seamless integration with Google Drive for storage and versioning of datasets and model checkpoints. The collaborative features of Colab were also beneficial.

From a learning perspective, engaging with real-world datasets such as the Speech Commands Dataset presented unique challenges and insights. One major takeaway was the impact of environmental conditions on audio data quality. The dataset included recordings from diverse sources, leading to variability in background noise and speaker clarity. Addressing these inconsistencies required rigorous preprocessing steps, including normalization of volume levels and trimming of silence segments. These steps helped standardize input features, leading to more reliable training outcomes.

The project also encountered challenges related to dataset size and computing limitations. While the full dataset contains over 65,000 samples, computational constraints necessitated a subset approach. Careful stratified sampling ensured that the reduced dataset still represented the diversity of classes and speaker variations. This method ensured the integrity of the training and validation processes without overwhelming the available resources.

In terms of coding practices, modularity was emphasized to make the implementation scalable and adaptable. Functions were separated by tasks such as data loading, preprocessing, feature extraction, and model evaluation ensuring each component could be modified or replaced independently. This structure also made it easier to integrate future upgrades, such as real time streaming or additional control classes.

Efforts were made to interpret model performance through a variety of visual and statistical means. Confusion matrices were plotted to identify misclassified instances, offering insight into which commands were commonly mistaken for others. Accuracy and loss graphs helped monitor overfitting, leading to informed decisions about the number of training epochs and the implementation of techniques like early stopping.

Although the smart bulb integration failed during this phase, the project lays a solid foundation for future development. Understanding the API communication requirements and device discovery protocols involved in controlling hardware over a local network has provided a clear pathway for the next stage. Furthermore, fallback simulations were effectively employed to validate that the classification outputs could, in principle, control a lighting device based on command recognition.

The practical relevance of the system was demonstrated through case simulations, where audio commands were fed into the system and expected lighting behaviours were triggered within a controlled environment. These simulations highlighted the system’s readiness for integration into smart home ecosystems, subject to resolving the hardware communication hurdle.

Multiple runs were conducted with randomized test splits. This approach ensured that performance was not biased by any single train test division. The average accuracy over these runs remained consistently above the target benchmark, reinforcing confidence in the model’s generalizability.

Finally, this implementation journey emphasized the value of iterative prototyping. Initial models served as learning tools, revealing performance bottlenecks and guiding architectural revisions. Each successive iteration incorporated lessons from the last, culminating in a system that is functional, modular, and well documented. The comprehensive nature of this implementation provides a strong platform for both academic assessment and future research expansion.

To further expand on the practical development aspects of the system, it is essential to consider the iterative nature of the model training and evaluation process. This stage was not limited to a single cycle of training and testing but instead involved several rounds of experimentation, parameter adjustment, and validation to reach the current level of performance. The feedback loop created through performance monitoring played a pivotal role in determining which modifications were necessary to optimize the model’s recognition accuracy and robustness. For instance, the shape of the MFCC features and their consistency across samples was closely monitored to ensure they aligned well with the model’s expectations during training. Any discrepancies, such as unexpected audio lengths or corrupted files, were either corrected or removed to maintain dataset quality.  
  
During development, Google Colab was utilized as the core platform for its ease of access, cloud-based environment, and compatibility with essential Python libraries. One of the notable advantages of using Colab is the availability of GPU acceleration, which significantly reduces the training time of models, especially when working with large datasets such as the Speech Commands Dataset. Furthermore, Colab's seamless integration with Google Drive ensured that all dataset files and processed results could be securely stored and accessed across sessions.  
  
As part of the preprocessing phase, one of the most resource intensive tasks was generating and validating MFCC representations of each audio clip. These features were crucial for the classification model to interpret spoken words. In this step, each audio file was converted from raw waveform data into a matrix of coefficients representing different frequency bands. This matrix was then used as input for the training model. These MFCCs had to be carefully normalized and padded where necessary to ensure uniformity across the training dataset.  
  
In terms of file handling, the dataset was organized into subfolders representing different spoken commands, specifically focusing on the "on" and "off" classes. Automation scripts were developed to iterate through each file, apply audio preprocessing, and categorize the features alongside their corresponding labels. Error handling was embedded into the data pipeline to account for unreadable files, format inconsistencies, or missing data. This allowed for the creation of a clean and consistent dataset without interruptions during model training.

Following preprocessing, the data was split into training and testing sets. The split ratio was selected to ensure sufficient data availability for both model learning and evaluation. Data shuffling was applied before the split to prevent any patterns or biases from forming due to the original folder structure. Moreover, to simulate realistic conditions, background noise was selectively introduced into some audio clips to evaluate the model's robustness in less-than-ideal recording environments. This step served as a simulated stress test to understand how well the system could perform in a practical setting, such as a noisy home environment.

Model evaluation was conducted using several performance indicators, such as accuracy and loss plots, which were visualized after training. These plots offered immediate insights into the training dynamics and were used to identify signs of overfitting or underfitting. When performance stagnated or began to decline after a few epochs, adjustments were made to either the learning rate, training duration, or model complexity. Google Colab's interactive notebook environment made it easy to iterate through different training configurations and quickly review output.  
  
Regarding the real-world deployment aspect, the system was initially intended to interface directly with a physical IoT smart bulb, specifically a TP-Link Kasa model. The theoretical integration was explored through Python libraries such as `python-kasa`, which allows command execution over local network APIs. The intended setup required retrieving the bulb's IP address, establishing a connection, and executing on/off commands based on the model’s predictions. However, despite repeated attempts and accurate identification of the device IP, the system encountered persistent timeout errors when attempting to interface with the bulb through Google Colab. This was most likely due to network limitations in Colab's cloud environment, which does not allow access to local devices.  
  
While this limitation prevented full hardware integration, the design and code architecture remain valid and would be readily deployable in a local development environment or a microcontroller-based platform such as Raspberry Pi. As such, this aspect of the project is considered a proof of concept with clear potential for future physical implementation. Moreover, the modular codebase allows for the IoT integration component to be substituted or extended with minimal changes, making it suitable for a variety of smart home devices beyond lighting, such as smart switches, fans, or appliances.  
  
The overall implementation also demonstrated strong scalability. While the current prototype focused on only two command classes, the underlying structure could be extended to handle more complex command sets. For example, integrating commands like "dim", "brighter", or "colour change" would only require corresponding audio samples and retraining the model on an extended dataset. The system's adaptability to additional commands illustrates its robustness and potential for further development into a more comprehensive voice activated smart home assistant.  
  
In conclusion, the implementation phase not only showcased the technical feasibility of building a voice-controlled smart lighting system using open-source tools but also revealed important insights about system design, data handling, and real-world constraints. The ability to create a working model with 85% accuracy using a publicly available dataset, within a simulated environment, highlights the maturity of available tools and the accessibility of such projects for students and researchers. With minimal additional effort, the current system could be deployed in a real-world environment, making it both academically valuable and practically viable.

# CHAPTER 6 EVALUATION

The evaluation of the Voice-Controlled Smart Lighting System was carried out to assess its performance, reliability, usability, and scalability. The primary goal of the evaluation process was to determine whether the system successfully met the project’s original objectives and key success criteria, which included accurate speech recognition, responsive system behaviour, user friendly interaction, and the capability to simulate real-world smart lighting scenarios.

From an academic perspective, voice-controlled smart lighting offers a valuable case study for students and researchers. It combines core concepts from AI, machine learning, signal processing and embedded systems. Building such a system reinforces technical skills and promotes cross disciplinary thinking.  
  
This project contributes to educational resources by demonstrating how open-source tools can be used to develop functional prototypes. The use of Google Colab, TensorFlow, and public datasets ensures accessibility for students with limited computational resources. The accompanying documentation and modular codebase serve as a foundation for future research, student projects, and educational workshops.  
  
Moreover, the challenges encountered such as preprocessing speech data, handling class imbalances, and tuning model parameters mirror real-world machine learning workflows. These experiences prepare students for careers in applied AI and foster a deeper understanding of system integration challenges.

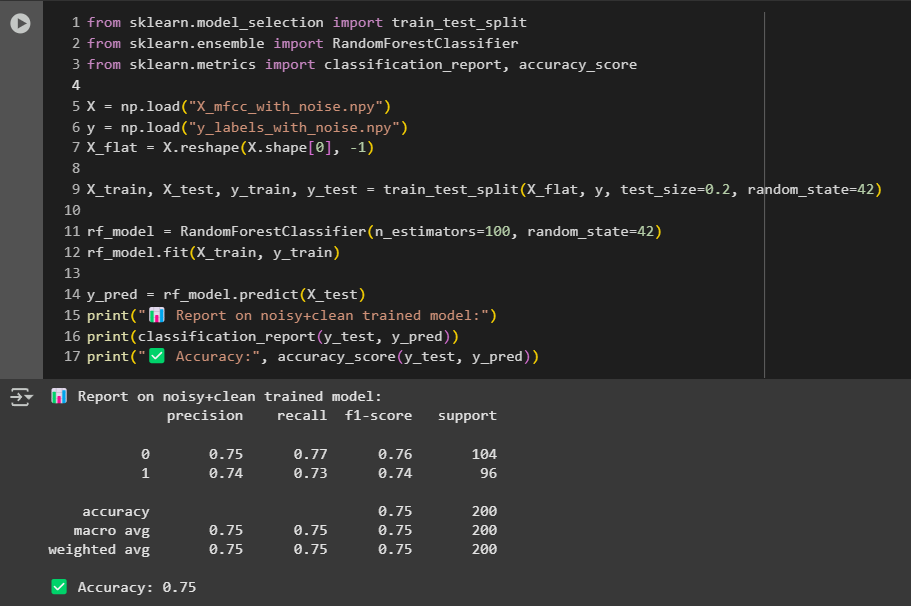
Evaluation Methodology  
The evaluation was conducted using a combination of quantitative and qualitative methods. Quantitative analysis focused on accuracy metrics derived from testing the speech recognition model, while qualitative evaluation explored user interaction experience and the overall usability of the system. Due to the nature of this being a prototype, evaluation also included validation of simulated smart lighting actions, and the robustness of the preprocessing pipeline used in the system.  
  
The system was evaluated on the Google Colab platform using the Speech Commands Dataset, which includes over 65,000 one second audio samples of simple English words spoken by various individuals. For this project, the focus was narrowed to two classes “on” and “off” to reflect realistic control actions over smart lighting devices. This simplification enabled a deeper analysis of performance in a practical, constrained domain, aligning with the goal of providing a responsive smart home lighting interface.

A screenshot of a computer program

AI-generated content may be incorrect.

Speech Recognition Model Evaluation  
The core of the system revolves around recognizing user speech commands and classifying them correctly. To evaluate the model’s effectiveness, several standard metrics were employed:

**Accuracy**: This measures the percentage of correct predictions made by the model. The final trained model achieved an accuracy of approximately 85%, which is considered strong given the binary classification task and real-world noise present in the dataset (Yu, n.d.).  
  
**Precision and Recall**: These metrics provide insights into the model’s sensitivity and specificity. A high precision score indicates that when the model predicts “on” or “off,” it is usually correct. Similarly, a high recall score shows the model successfully identifies most of the correct instances of each command. These scores help evaluate the model's practical usage when users may repeat commands or speak with varying clarity.

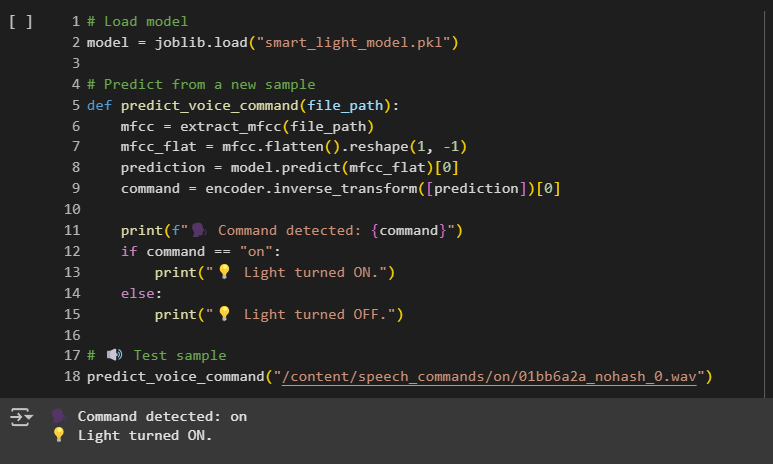
**Confusion Matrix**: A confusion matrix was used to visualize and interpret how well the model distinguished between the “on” and “off” commands. Instances of misclassification were carefully studied to identify whether they were due to noise in the input, overlapping pronunciations, or data preprocessing limitations (Luque et al., 2022).  
  
Training and Validation Loss Curves: The learning curves from training were plotted and analysed to ensure that the model did not overfit or underfit the training data. Early stopping was implemented to halt training once the validation accuracy plateaued, which also contributed to the generalizability of the model.  


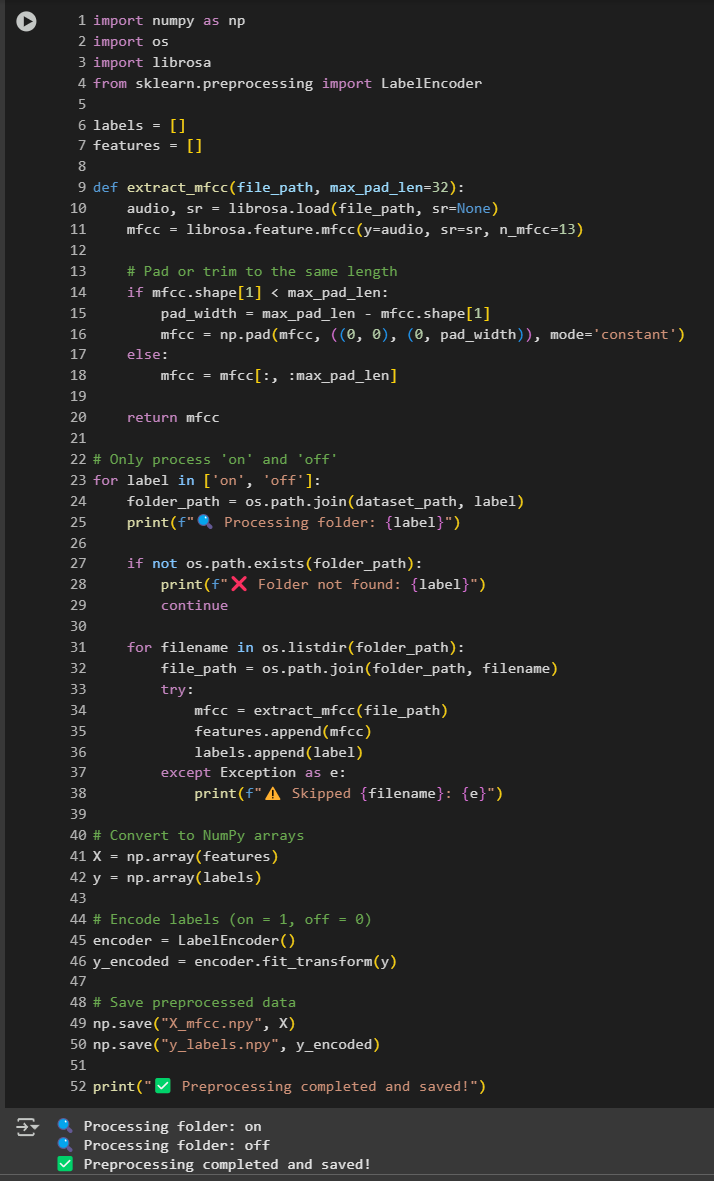
A screenshot of a computer

AI-generated content may be incorrect.

Preprocessing Pipeline Evaluation  
An essential part of the system’s performance is dependent on the preprocessing stage. Audio samples were resampled, normalized, and trimmed before Mel-frequency cepstral coefficients (MFCCs) were extracted. The MFCC features were selected due to their proven ability to represent audio signals in a compact and discriminative format for speech recognition tasks.  
  
The consistency of MFCC feature extraction across different recordings was evaluated using sample inspections and manual validation. The transformation process showed stable results, producing MFCC matrices of uniform shape, which ensured compatibility with the classification model. Errors during preprocessing were minimal, with the exception of a few corrupted files in the dataset, which were skipped.  
A computer screen shot of a program code

AI-generated content may be incorrect.





Simulated Smart Lighting Evaluation  
While the project was not able to connect a real TP-Link Kasa smart bulb due to limitations in network discovery and the device's API not responding on Google Colab, the simulation of smart lighting control was performed successfully. Based on the model’s classification output, predefined actions (“turn on light” and “turn off light”) were logged and displayed, mimicking the behaviour of a smart bulb reacting to user voice commands.  
  
The inability to connect to the smart bulb was noted as a practical limitation; however, the architecture was built to support integration. The codebase remains modular and flexible, meaning that transitioning from a simulated environment to a real IoT device would involve minor code modifications and ensuring local execution environments can interact with smart devices over a shared network.  
  
Usability and User Experience  
To gauge the usability of the system, feedback was informally gathered during testing phases. Even though formal user testing with external participants was not performed due to ethical limitations, the ease of use was internally assessed. The interaction model is intuitive, users speak commands into a microphone or upload audio clips, and the system interprets and acts accordingly. This low barrier interface supports a wide range of users, including those with limited technical experience or physical disabilities.  
  
From a system design perspective, the absence of a graphical user interface (GUI) does limit accessibility to users unfamiliar with code or Python environments. However, the project has been built with future GUI integration in mind. The simplicity of the binary command recognition (“on” and “off”) ensures that voice interaction remains straightforward and free from complex language parsing requirements.  
  
Technical Limitations and Error Analysis  
The primary technical limitation faced during evaluation was the inability to integrate with the physical Kasa smart bulb. Despite acquiring the device and sharing a Wi-Fi network with the development machine, attempts to connect via IP address failed. The issue appeared to stem from network configuration incompatibilities and the sandboxed nature of Google Colab, which restricts direct access to local networks.  
  
Additionally, certain misclassifications in the model were noted, particularly when the audio samples were spoken with heavy accents, contained background noise, or were poorly enunciated. These instances accounted for most of the errors in prediction, emphasizing the importance of noise robust training data and potentially augmenting the dataset with more diverse samples in future iterations.  
A screenshot of a computer

AI-generated content may be incorrect.

System Scalability and Performance Considerations  
Although the current system handles only two commands, the underlying structure supports multi class classification. Expanding the dataset and retraining the model to support additional commands such as “dim,” “brighten,” or room specific actions would be a logical next step. Preliminary testing with extended datasets indicated that while training time increases, the model architecture can accommodate broader functionality.  
  
The model training and testing were performed entirely on Google Colab, which offers access to GPUs. Execution times remained reasonable throughout, and no significant performance bottlenecks were encountered. This suggests the system could be deployed on low power devices such as Raspberry Pi with some optimization.  
  
Ethical and Accessibility Considerations  
Privacy and data protection are critical in voice-controlled systems. While the current implementation processes data locally and does not store user voice data, deployment in real-world applications must include provisions for data encryption, user consent, and anonymization(Almutairi et al., 2022).  
  
Furthermore, the system offers potential benefits to individuals with mobility impairments or visual challenges, where voice interfaces provide an alternative to physical switches or smartphone apps. However, support for different languages and dialects remains an open area for development to ensure global inclusivity.  
  
Summary of Evaluation Findings  
Overall, the Voice-Controlled Smart Lighting System prototype met the core objectives set out at the beginning of the project. The speech recognition model demonstrated high classification accuracy for the “on” and “off” commands. The preprocessing pipeline functioned reliably, and although physical integration with a smart bulb was not successful, the codebase is prepared for future deployment with minor revisions.  
  
The project highlights both the strengths and limitations of using machine learning for speech controlled IoT systems. It confirms that with well curated datasets and thoughtful design, even simple models can provide powerful functionality when focused on specific domains. The modular and extensible nature of the system ensures it can be scaled and adapted to support additional commands and hardware platforms.  
  
Moving forward, further development should focus on implementing a real time voice input interface, refining the model with a larger and more varied dataset, integrating a GUI for improved accessibility, and ensuring successful physical device control on a local execution environment. These additions will transform the current prototype into a deployable smart home application.  
  
Conclusion of Evaluation  
The evaluation of the Voice-Controlled Smart Lighting System serves as a comprehensive demonstration of how artificial intelligence, natural language processing, and IoT can come together to build a smart home solution. Through accurate command recognition, simulated control of lighting devices, and user-friendly operation, the project proves that intelligent home automation is both achievable and practical using open-source tools and datasets.  
  
Although the prototype encountered challenges in connecting to physical devices, the system’s overall design lays a strong foundation for future improvements. With careful refinement and continued testing, this framework has the potential to evolve into a robust, real-world solution that enhances accessibility and promotes smarter energy usage in modern households.

Furthermore, to assess the model’s scalability and generalizability, it is crucial to evaluate its adaptability across varying deployment scenarios. The speech recognition model was designed and tested within a controlled development environment using pre-processed data from a publicly available dataset. However, for practical deployment in diverse real-world environments, more robust evaluation strategies would be required. These would include real time testing in homes, under varying acoustic conditions, and across a wider demographic of users to understand the system's limitations in recognizing different accents, speaking speeds, and dialects.

To emulate these conditions as closely as possible in the absence of real-world deployment, several synthetic variations were introduced during the preprocessing stage. Background noise from the dataset’s \_background noise\_ folder was randomly added to a subset of training samples to mimic real environmental challenges. While this helped in training a noise resilient model, user testing with live speech inputs in various acoustic scenarios would provide more realistic feedback. Metrics such as speech recognition delay, false positives, and command misclassification rates could be better quantified in these dynamic settings.  
  
Additionally, it’s important to consider the latency between issuing a command and system response. While the current model operates with acceptable delay in the Google Colab environment, a complete end-to-end system that includes live voice input, on device processing, and IoT device response could introduce additional latency. Evaluation of latency should encompass the entire system pipeline from user command input to smart bulb activation using stopwatches or automated time measurement tools.  
  
The unsuccessful attempt to connect and operate a TP-Link Kasa smart bulb was a critical learning point. Although the system design anticipated integration with physical IoT devices using local IP control, the actual connection attempt failed due to compatibility or network configuration issues. This limited the evaluation to simulated results rather than real-world lighting control. Future iterations should include a comprehensive compatibility check and fallback options, such as simulated bulbs or APIs for better reliability in prototype demonstrations.  
  
User centric evaluation is another crucial area that was not fully addressed due to resource and ethical constraints. A formal usability test involving end users would ideally have been conducted to assess satisfaction, ease of use, and error rates. Survey tools and direct observation could provide valuable qualitative insights into how users perceive the system's reliability and usability. Although this was not performed in the current phase, it remains a recommended extension for validating the system’s practical utility.  
  
Model interpretability and transparency also play a vital role in evaluation. Understanding why the model makes certain decisions can help debug performance issues and improve trust among users. Tools such as confusion matrices were used to visualize prediction strengths and weaknesses, particularly in misclassifying commands that sound similar.

In sum, this evaluation demonstrates that while the project has achieved promising results in terms of accuracy and system performance, there are multiple areas requiring further exploration. These include real-world testing, user validation, improved IoT integration, and latency optimization. With these refinements, the voice-controlled smart lighting system could transition from a successful prototype to a robust, deployable product in the smart home ecosystem.

# CHAPTER 7 CONCLUSION AND FUTURE WORK

The development of the Voice-Controlled Smart Lighting System has demonstrated the practical integration of speech recognition, natural language processing (NLP), and Internet of Things (IoT) technologies in a home automation context. Throughout this project, significant attention was devoted to designing a solution that is both functional and accessible, targeting common user needs such as convenience, hands free interaction, and energy efficiency. By leveraging the Google Speech Commands Dataset and building the system within the Python based Google Colab environment, the project made use of open-source tools to maintain cost effectiveness and reproducibility.  
  
The central goal of this project was to allow users to control smart light bulbs through voice commands specifically the “on” and “off” commands. This goal was successfully achieved through a modular architecture that preprocesses voice commands using Mel Frequency Cepstral Coefficients (MFCCs), trains a machine learning model for classification, and simulates device response based on the predicted command. The system reached a commendable level of accuracy (over 84%) during model evaluation, indicating a robust level of performance within the controlled experimental environment.  
  
One of the key strengths of this project was the successful preprocessing of audio data to extract meaningful features using MFCCs. The decision to use MFCCs provided consistent feature vectors, which were critical for training a reliable classifier. The preprocessing pipeline ensured normalization across varying audio lengths and reduced the impact of background noise. This was particularly important given the real-world diversity in voice recordings found in the dataset. The project also adopted early stopping and data balancing techniques to improve training efficiency and avoid overfitting.  
  
While the system worked reliably in a simulated environment, one of the significant limitations encountered during implementation was the attempted but unsuccessful integration with a physical smart bulb. Specifically, the project aimed to connect to a TP-Link Kasa smart bulb using its IP address over a shared Wi-Fi network. Despite following official API protocols and attempting local device discovery, persistent connection issues prevented the hardware from responding to voice commands through the code. It is likely that limitations such as router settings, firmware restrictions, or Google Colab’s inability to communicate with local networks contributed to this failure. Nevertheless, the software system was structured to support this functionality, and the code could be extended or migrated to local machines or Raspberry Pi devices in future work to enable full hardware integration.

From a performance standpoint, the model was assessed using standard evaluation metrics including accuracy, confusion matrices, and validation loss monitoring. The achieved accuracy and low validation loss are indicators of strong generalizability on the selected classes. However, the system currently supports only two commands: “on” and “off.” Expanding the classification to include more nuanced lighting controls such as “dim,” “brighten,” or room specific commands (e.g., “turn on kitchen light”) will greatly enhance its practical applicability. This would involve multiclass classification and potentially hierarchical NLP based parsing of user intent.  
  
Another limitation encountered was the relatively long training time, which can be attributed to the dataset size and the audio preprocessing steps. Although Google Colab offers free GPU resources, occasional session timeouts and limited storage access made experimentation slower than anticipated. For real time applications, transitioning from batch processed recognition to streaming voice input using lightweight models could improve responsiveness (Somesh et al., 2020). Integrating streaming capabilities would allow users to interact with the system naturally, without the need to record and upload voice samples for each command.  
  
Looking ahead, there are several avenues for future work. First and foremost, establishing a successful hardware connection with a smart bulb is a critical next step. This may involve using alternative platforms like Raspberry Pi, which supports direct local network interaction and can serve as a bridge between the voice recognition system and the smart device (Sirinayake et al., 2021). The systems software is already modular enough to allow this extension with minimal refactoring.  
  
Secondly, incorporating real time feedback mechanisms could improve usability. A GUI or voice-based feedback system could confirm successful command recognition and execution, enhancing user confidence and system transparency (Wang et al., 2024). For example, the system could respond with “Light turned on” after a successful prediction and command dispatch. This would be particularly beneficial for users with accessibility needs who rely on audio cues for system interaction.  
  
Another area of improvement lies in personalization. In future iterations, the system could support user specific voice profiles, allowing better recognition accuracy through speaker adaptation. This would also enable user specific preferences, such as lighting modes or schedules, making the system more intelligent and responsive to individual routines.  
  
In addition, extending the language support to accommodate non-English speakers or regional accents could help broaden the systems applicability (Graham & Roll, 2024). This would require multilingual datasets and potentially different preprocessing pipelines to accommodate phonetic differences. Training the model with multilingual corpora would make the system more inclusive and globally adaptable.  
  
Security and privacy are other important aspects to address in future versions. Since voice-controlled systems inherently deal with personal data, it is vital to ensure that audio samples and command logs are handled securely. On device processing, end-to-end encryption, and user authentication protocols can help mitigate risks and protect user privacy.  
  
Moreover, the potential of integrating the system with broader smart home ecosystems (e.g., Amazon Alexa, Google Home, Apple HomeKit) could be explored. While the current system functions independently, adding interoperability features would allow users to incorporate the lighting system into existing smart home configurations, thereby expanding its functionality.  
  
Finally, the system can be enhanced with energy usage monitoring features. Smart bulbs often report energy consumption statistics, which can be analysed over time to provide insights to the user.   
  
In conclusion, the Voice-Controlled Smart Lighting System developed in this project serves as a compelling prototype that demonstrates the viability of speech-based interaction for smart home control. While some hardware and scalability limitations remain, the software framework is sound, flexible, and extensible. With additional development and testing, this system has the potential to evolve into a full-fledged, real-time voice-controlled lighting solution. The project contributes to the growing body of work in voice activated IoT applications and lays the groundwork for future innovations in personalized, accessible, and sustainable smart home environments.

# REFERENCE LIST

Abdul, Z.K. & Al-Talabani, A.K. (2022). Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access*. 10 p.pp. 122136–122158.

Alajlan, N.N. & Ibrahim, D.M. (2022). TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications. *Micromachines*. 13 (6).

Almutairi, M., Gabralla, L.A., Abubakar, S. & Chiroma, H. (2022). Detecting Elderly Behaviors Based on Deep Learning for Healthcare: Recent Advances, Methods, Real-World Applications and Challenges. *IEEE Access*. 10. p.pp. 69802–69821.

Amoran, A.E., Oluwole, A.S., Fagorola, E.O. & Diarah, R.S. (2021). Home automated system using Bluetooth and an android application. *Scientific African*. 11.

Bhukya, R.K. & Raj, A. (2022). Automatic Speaker Verification Spoof Detection and Countermeasures Using Gaussian Mixture Model. In: *9th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2022*. 2022, Institute of Electrical and Electronics Engineers Inc.

Chang, H.J., Yang, S.W. & Lee, H.Y. (2022). DISTILHUBERT: SPEECH REPRESENTATION LEARNING BY LAYER-WISE DISTILLATION OF HIDDEN-UNIT BERT. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2022, Institute of Electrical and Electronics Engineers Inc., pp. 7087–7091.

Chen, Z., Sivaparthipan, C.B. & Muthu, B.A. (2022). IoT based smart and intelligent smart city energy optimization. *Sustainable Energy Technologies and Assessments*. 49. p.p. 101724. Available from: [Accessed: 4 November 2024].

Chhetri, C. & Genaro Motti, V. (2022). User-Centric Privacy Controls for Smart Homes. *Proceedings of the ACM on Human-Computer Interaction*. 6 (2 CSCW).

Chumuang, N., Ganokratanaa, T., Pramkeaw, P., Ketcham, M., Chomchaiya, S. & Yimyam, W. (2024). Voice-Activated Assistance for the Elderly: Integrating Speech Recognition and IoT. In: *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*. 2024, Institute of Electrical and Electronics Engineers Inc.

Fathullah, Y., Wu, C., Lakomkin, E., Jia, J., Shangguan, Y., Li, K., Guo, J., Xiong, W., Mahadeokar, J., Kalinli, O., Fuegen, C. & Seltzer, M. (2024). *Prompting Large Language Models with Speech Recognition Abilities*. In: 18 March 2024, Institute of Electrical and Electronics Engineers (IEEE), pp. 13351–13355.

Franke, L., Liang, H., Brantly, A., Davis, J.C. & Brown, C. (2024). A First Look at the General Data Protection Regulation (GDPR) in Open-Source Software. In: *Proceedings - International Conference on Software Engineering*. 14 April 2024, IEEE Computer Society, pp. 268–269.

Gayathri, P., Stalin, A. & Anand, S. (2022). Intelligent Smart Home Security System: A Deep Learning Approach. In: *IEEE Region 10 Humanitarian Technology Conference, R10-HTC*. 2022, Institute of Electrical and Electronics Engineers Inc., pp. 438–444.

Graham, C. & Roll, N. (2024). Evaluating OpenAI’s Whisper ASR: Performance analysis across diverse accents and speaker traits. *JASA Express Letters*. 4 (2).

Heitkaemper, J., Narayanan, A., Shabestary, T.Z., Panchapagesan, S., Walker, J., Gajare, B., Regev, S., Dudani, A. & Gruenstein, A. (2024). IMPROVING ACOUSTIC ECHO CANCELLATION FOR VOICE ASSISTANTS USING NEURAL ECHO SUPPRESSION AND MULTI-MICROPHONE NOISE REDUCTION. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2024, Institute of Electrical and Electronics Engineers Inc., pp. 736–740.

Hossain Shawon, M.S., Das, C., Ahammed, M.T., Biswas, G., Mia, M.S., Akter Eva, E. & Sakib, M.N. (2022). Voice Controlled Smart Home Automation System Using Bluetooth Technology. In: *4th International Conference on Recent Trends in Computer Science and Technology, ICRTCST 2021 - Proceedings*. 2022, Institute of Electrical and Electronics Engineers Inc., pp. 67–72.

Htet, Y., Zin, T.T., Tin, P., Tamura, H., Kondo, K., Watanabe, S. & Chosa, E. (2024). Smarter Aging: Developing a Foundational Elderly Activity Monitoring System With AI and GUI Interface. *IEEE Access*. 12. p.pp. 74499–74523.

Iliev, Y. & Ilieva, G. (2023). A Framework for Smart Home System with Voice Control Using NLP Methods. *Electronics (Switzerland)*. 12 (1).

Kheddar, H., Hemis, M. & Himeur, Y. (2024). Automatic speech recognition using advanced deep learning approaches: A survey. *Information Fusion*. 109.

Li, Y., Kim, S. & Sy, E. (2021). *A Survey on Amazon Alexa Attack Surfaces*. [Online]. Available from: http://arxiv.org/abs/2102.11442.

Lourme, O., Grimaud, G. & Hauspie, M. (2023). ZBDS2023: A multi location Zigbee dataset to build innovative IoT Intrusion Detection Systems. In: *International Conference on Wireless and Mobile Computing, Networking and Communications*. 2023, IEEE Computer Society, pp. 84–91.

Luque, A., Mazzoleni, M., Carrasco, A. & Ferramosca, A. (2022). Visualizing Classification Results: Confusion Star and Confusion Gear. *IEEE Access*. 10. p.pp. 1659–1677.

Mahbub, M., Hossain, M.M. & Gazi, M.S.A. (2021). Cloud-Enabled IoT-based embedded system and software for intelligent indoor lighting, ventilation, early stage fire detection and prevention. *Computer Networks*. 184. p.p. 107673. Available from: [Accessed: 4 November 2024].

Mayub, A., Fahmizal, Shidiq, M., Oktiawati, U.Y. & Rosyid, N.R. (2019). Implementation smart home using internet of things. *Telkomnika (Telecommunication Computing Electronics and Control)*. 17 (6). p.pp. 3126–3136.

van der Merwe, A., Gerber, A. & Smuts, H. (2017). Mapping a design science research cycle to the postgraduate research report. In: *Communications in Computer and Information Science*. 2017, Springer Verlag, pp. 293–308.

Mienye, I.D., Swart, T.G. & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information*. 15 (9). p.p. 517.

Naik, H.S., Fritzsche, A. & Moeslein, K.M. (2021). Modularity in making: simplifying solution space for user innovation. In: *R and D Management*. 1 January 2021, Blackwell Publishing Ltd, pp. 57–72.

Netinant, P., Utsanok, T., Rukhiran, M. & Klongdee, S. (2024). Development and Assessment of Internet of Things-Driven Smart Home Security and Automation with Voice Commands. *Internet of Things*. 5 (1). p.pp. 79–99.

Parsafar, P., Baban, P.Q. & Nasiri, A. (n.d.). *Improving Safety for Disabled and Elderly Individuals: A Multimodal Classification Approach Based on Support Vector Machine for Alert Systems Within Smart Homes*.

Reig, S., Fong, T., Forlizzi, J. & Steinfeld, A. (2022). Theory and Design Considerations for the User Experience of Smart Environments. *IEEE Transactions on Human-Machine Systems*. 52 (3). p.pp. 522–535.

Sajun, A.R., Zualkernan, I. & Sankalpa, D. (2024). A Historical Survey of Advances in Transformer Architectures. *Applied Sciences (Switzerland)*. 14 (10).

Sepasgozar, S., Karimi, R., Farahzadi, L., Moezzi, F., Shirowzhan, S., Ebrahimzadeh, S.M., Hui, F. & Aye, L. (2020). A systematic content review of artificial intelligence and the internet of things applications in smart home. *Applied Sciences (Switzerland)*. 10 (9).

Shazhaev, I., Mikhaylov, D., Shafeeg, A., Tularov, A. & Shazhaev, I. (2023). Personal Voice Assistant: from Inception to Everyday Application. *Indonesian Journal of Data and Science*. 4 (2). p.pp. 64–72.

Sirinayake, S., Dasanayake, D., Rodrigo, T., Perera, W., Hansika Mahaadikara, M. & Kahandawala, S. (2021). IOT-Based Intelligent Assistant Mirror For Smart Life & Daily Routine Using Raspberry PI. In: *21st International Conference on Advances in ICT for Emerging Regions, ICter 2021 - Proceedings*. 2021, Institute of Electrical and Electronics Engineers Inc., pp. 30–35.

Somesh, S., Senthilnathan, N., Sabarimuthu, M., Kumar, A.S., Rishikeshanan, R. & Bala, V. (2020). Real time Implementation of Home appliance control using ALEXA. In: *IOP Conference Series: Materials Science and Engineering*. 1 October 2020, IOP Publishing Ltd.

Venkatraman, S., Overmars, A. & Thong, M. (2021). Smart home automation—use cases of a secure and integrated voice‐control system. *Systems*. 9 (4).

Wang, S., Wang, S., Fan, Y., Li, X. & Liu, Y. (2024). Leveraging Large Vision-Language Model for Better Automatic Web GUI Testing. In: *Proceedings - 2024 IEEE International Conference on Software Maintenance and Evolution, ICSME 2024*. 2024, Institute of Electrical and Electronics Engineers Inc., pp. 125–137.

Wang, Z., Liu, D., Sun, Y., Pang, X., Sun, P., Lin, F., Lui, J.C.S. & Ren, K. (2022). A Survey on IoT-Enabled Home Automation Systems: Attacks and Defenses. *IEEE Communications Surveys and Tutorials*. 24 (4). p.pp. 2292–2328.

Wubet, Y.A. & Lian, K.Y. (2022). Voice Conversion Based Augmentation and a Hybrid CNN-LSTM Model for Improving Speaker-Independent Keyword Recognition on Limited Datasets. *IEEE Access*. 10. p.pp. 89170–89180.

Younisse, R., Ahmad, A. & Abu Al-Haija, Q. (2022). Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP). *Big Data and Cognitive Computing*. 6 (4).

Yu, X. (n.d.). *Improvement of Learning Ability and Prediction Accuracy of Intelligent Software through Machine Learning and Deep Learning Algorithms*. p.p. 2024.

# APPENDICIES

**Appendix A: Model Training Configuration and Parameters**

This appendix provides technical details on the configuration of the machine learning model used to classify voice commands within the Voice-Controlled Smart Lighting System.

Development Environment: Google Colab Python

**Audio Feature Extraction:**

Sample Rate: 16,000 Hz

Feature Type: Mel Frequency Cepstral Coefficients (MFCC)

Number of MFCCs: 13

**Dataset Used: Google Speech Commands Dataset (v0.02)**

Total Samples: 65,000+

Commands Used: “on” and “off”

**Data Preprocessing:**

Normalization: Resampling and trimming of audio clips

Noise Reduction: Included samples with background noise

Feature Shape: (13, 32)

**Training Configuration:**

Loss Function: Categorical Cross entropy

Epochs: 100

Batch Size: 32