MULTI MODAL SMART HOME CONTROL WITH RIDGE CLASSIFIER INTEGRATION

A PROJECT REPORT

Submitted by

AARTHI J	**(310821243002)**
DHIVYA G	**(310821243014)**
RESHMA M	**(310821243043)**
SABITHA D	**(310821243045)**

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

JEPPIAAR ENGINEERING COLLEGE

CHENNAI – 600 119

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **“MULTI MODAL SMART HOME CONTROL WITH RIDGE CLASSIFIER INTEGRATION’’** is the bonafide work of **AARTHI J (310821243002), DHIVYA G (310821243014), RESHMA M (310821243043), SABITHA D (310821243045)** who carried out project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.T.R.CHENTHIL,M.E.,Ph.D.,	**Dr.T.R.CHENTHIL,M.E.,Ph.D.,**
HEAD OF THE DEPARTMENT,	**SUPERVISOR,**
Assistant professor,	Assistant Professor,
Department of Artificial Intelligence and Data Science,	Department of Artificial Intelligence and Data Science,
Jeppiaar Engineering College,	Jeppiaar Engineering College,
Chennai – 600 119.	Chennai – 600 119.

CERTIFICATE OF PROJECT APPROVAL

This is to certify that the Project report titled **"MULTI MODAL SMART HOME CONTROL WITH RIDGE CLASSIFIER INTEGRATION"** is the approved record of work done by **AARTHI J (310821243002) ,DHIVYA G (310821243014) ,RESHMA M (310821243043), SABITHA D (310821243045)** in partial fulfillment for the award of the Degree of Bachelor of Technology in **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** during the academic year 2024- 2025.

SUPERVISOR **HEAD OF THE DEPARTMENT**

(Signature with seal)

Date:

Submitted for the Project Viva Voce Examination held on………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We affirm that the project titled **"MULTI MODAL SMART HOME CONTROL WITH RIDGE CLASSIFIER INTEGRATION"** being submitted in partial fulfillment for the award of B.Tech., Artificial Intelligence and Data Science is the original work carried out by us. It has not formed the part of any other thesis submitted forwarded of any degree or diploma, either in this or any other University.

Signature of the Candidates

AARTHI J	(310821243002)
DHIVYA G	(310821243014)
RESHMA M	(310821243043)
SABITHA D	(310821243045)

I certify that the declaration made above by the candidate is true.

Signature of the Guide

Dr.T.R.CHENTHIL, M.E.,Ph.D.,

Department of Artificial Intelligence

And Data Science

ACKNOWLEDGEMENT

Behind every achievement lies and unfathomable sea of gratitude to those who actuated it. Without them, it would never have into existence. To them we lay the word of gratitude imprinted within us.

We delighted to thank (Late) Hon'ble Colonel **Dr. Jeppiaar, M.A., B.L., Ph.D.,** the Chairman and Managing Director, **Dr. M. Regeena Jeppiaar, B. Tech., M.B.A., Ph.D.,** Jeppiaar Engineering College.

We also take this opportunity to delicate our gratefulness to our honorable Principal **Dr. K. Senthil Kumar, M. E., Ph.D., FIE,** and whose undying spirit is our complimentary.

We also express our sincere thanks and performed gratitude to our Head of Department **Dr.T.R.Chenthil,M.E.,Ph.D.,** Artificial Intelligence and Data Science, for his excellent guidance, sustained inspiration, constant help, timely encouragement and excellent cooperation throughout the tenure of the dissertation work.

We wish to express our sincere thanks to our project Guide **Dr.T.R.Chenthil,M.E.,Ph.D.,** Assistant Professor, Artificial Intelligence and Data Science, for his blessings and help provided during the period of our project work.

We would like to extend our sincere thanks to all lab technicians for helping us in this venture, Unflinching support and encouragement from the members of my family, friends and staff members in **Jeppiaar Engineering College** helping me a long way to complete our project work. We thank them all from the bottom of our heart.

ABSTRACT

This project introduces an advanced smart home automation system integrating gesture and voice control to enhance user interaction. Utilizing a Ridge Classifier, it accurately recognizes sign language gestures through computer vision, while also detecting voice commands via keyword recognition. This dual-modality approach ensures reliable, real-time control of appliances, even in challenging conditions like low light or hands-free scenarios. Designed with inclusivity in mind, the system supports users with disabilities and promotes ease of use. Its cost-effective, modular, and scalable architecture makes it ideal for various smart home settings. By combining machine learning with accessible technology, the project marks a forward step in creating intelligent, personalized living environments. The system also promotes energy efficiency and home safety by enabling remote or automated control of appliances, helping users monitor and reduce unnecessary power usage. Furthermore, the design is future-ready, supporting integration with IoT platforms and virtual assistants like Alexa or Google Home, ensuring seamless scalability and interoperability with other smart systems.

Keywords: Smart Home Automation, Machine Learning, Sign Language Recognition, Voice Command Detection, Ridge Classifier, Gesture Control, Voice Control, Arduino Integration, Human-Computer Interaction, Multimodal Interface, Home IoT System.

TABLE OF CONTENT

LIST OF FIGURE

LIST OF ABBRIVATIONS

ASL	-	**A**merican **S**ign **L**anguage
CNN	-	**C**onvolutional **N**eural **N**etwork
FMCW	-	**F**requency **M**odulated **C**ontinuous **W**ave
HCI	-	**H**uman **C**omputer **I**nteraction
HGR	-	**H**and **G**esture **R**ecognition
IR	-	**I**nformation **R**etrieval
MW	-	**M**illimeter **W**ave
NUS	-	**N**ational **U**niversity of **S**ingapore
PEP	-	**P**ython **E**nhancement **P**roposal
PWM	-	**P**ulse **W**idth **M**odulation
SES	-	**S**peech **E**nabled **S**ervices
SHA	-	**S**mart **H**ome **A**utomation
SMPS	-	**S**witched **M**odel **P**ower **S**upply
SVM	-	**S**upport **V**ector **M**achine
ULP	-	**U**ltra **L**ow **P**ower
UML	-	**U**nified **M**odeling **L**anguage
UWB	-	**U**ltra **W**ide **B**andwidth
VAC	-	**V**olts **A**lternating **C**urrent
WSDL	-	**W**eb **S**ervice **D**escription **L**anguage

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In today's rapidly advancing technological landscape, smart home automation systems are becoming increasingly prevalent, offering users greater comfort, security, and efficiency in managing their living spaces. However, most existing systems primarily rely on traditional input methods such as mobile applications, remote controls, or touch-based interfaces. While effective for the general population, these methods can pose significant challenges for individuals with physical disabilities, speech impairments, or age-related limitations. This project was initiated to bridge that gap by creating an inclusive and intelligent home automation system that enhances accessibility and independence for all users. The system uniquely integrates two powerful and intuitive modes of interaction sign language gesture recognition and voice keyword detection. By using a live camera feed and applying a Ridge Classifier machine learning algorithm, the system can accurately identify various hand gestures associated with common commands, such as opening or closing a door or switching lights on and off. Ridge Classifier is particularly well-suited for this task due to its robustness in handling multi-class classification problems with high accuracy and minimal computational overhead. In addition to gesture recognition, the system employs voice recognition capabilities to detect predefined spoken keywords. This enables users to control smart home devices without the need for physical contact, promoting a hands-free experience that is especially beneficial for those with mobility issues. By combining visual and auditory inputs, the system ensures redundancy and flexibility, allowing users to choose the mode of interaction that best suits their needs and circumstances.

1.2 SCOPE OF THE PROJECT

The practical applications of this system are wide-ranging. It is ideally suited for smart homes, where convenience and automation are key priorities. It can also be implemented in elder care facilities to support independent living by allowing residents to control home appliances without physical effort. Moreover, in high-risk or sterile environments such as hospitals, laboratories, or during pandemic situations, the contactless nature of this system contributes to better hygiene and safety standards by reducing the need to touch physical switches or handles.

1.3 OBJECTIVE

The Ridge Classifier algorithm for real-time sign detection through a live camera and incorporating voice keyword recognition, the system aims to offer a seamless interaction experience for all users, especially those with physical or speech disabilities. The goal is to enhance convenience, safety, and independence within residential or care-based environments by eliminating the need for physical switches. Furthermore, the system is designed to communicate directly with hardware components, such as microcontrollers, to execute physical actions, making it a practical and scalable solution for smart home integration.

1.4 MOTIVATION OF THE PROJECT

The motivation behind this project stems from the desire to create a more inclusive and accessible living environment by harnessing the power of machine learning in smart home automation. Many individuals, especially those with speech, hearing, or mobility impairments, often face challenges interacting with conventional home control systems that rely heavily on physical interfaces or complex apps. As smart home technologies become increasingly popular, there is a growing need to ensure these systems cater not just to convenience but also to accessibility and inclusivity. The use of a Ridge Classifier algorithm for gesture recognition ensures reliable, real-time performance, while voice detection enables seamless control for users who may not be able to perform gestures.

1.5 PROBLEM OF THE STATEMENT

The primary problem addressed by this project is the lack of accessibility and inclusiveness in existing smart home automation systems, which often depend on touch-based interfaces, mobile applications, or remote controls. These traditional input methods can be difficult or even impossible to use for individuals with physical disabilities, speech impairments, or elderly users with limited mobility. As a result, a significant portion of the population is unable to fully benefit from the convenience and functionality offered by modern home automation technologies. Moreover, many current systems are not designed to accommodate alternative modes of communication such as sign language or voice commands tailored to diverse user needs. This creates a technological gap that limits independence and usability for people who rely on non-traditional means of interaction.

CHAPTER 2

REVIEW OF LITERATURE

2.1 PAPER 1

TITLE: Two-Dimensional Parallel Spatio-Temporal Pyramid Pooling for Hand Gesture Recognition

AUTHOR: FARZANEH JAFARI ANDANUPBASU

YEAR: 2023

ABSTRACT:

Hand Gesture Recognition (HGR) plays a crucial role in user-friendly interactions between humans and computers. In recent years, using the Convolutional Neural Network (CNN) has improved the accuracy of image processing problems. Recognizing different hand signs by Two-Dimensional Parallel Spatio Temporal Pyramid Pooling (2DPSTPP) features with deep learning methods reduces the size of the map, minimizes training complexity, and by paying attention to more details, improves detection performance.

2.2 PAPER 2

TITLE: Accuracy Enhancement of Hand Gesture Recognition Using CNN

AUTHOR: GYUTAE PARK, VASANTHA KUMAR CHANDRASEGAR AND JINHWAN KOH

YEAR: 2023

ABSTRACT:

Human gestures are immensely significant in human-machine interactions. Complex hand gesture input and noise caused by the external environment must be addressed in order to improve the accuracy of hand gesture recognition algorithms. To overcome this challenge, we employ a combination of 2D-FFT and convolutional neural networks (CNN) in this research. The accuracy of human-machine interactions is improved by using Ultra Wide Bandwidth (UWB) radar to acquire image data, then transforming it with 2D FFT and bringing it into CNN for classification. The classification results of the proposed method revealed that it required less time to learn than prominent models and had similar accuracy.

2.3 PAPER 3

TITLE: Improved Static Hand Gesture Classification on Deep Convolutional Neural Networks Using Novel Sterile Training Technique

AUTHOR: JOSIAH W. SMITH , SHIVA THIAGARAJAN, RICHARD WILLIS, YIORGOS MAKRIS AND MURAT TORLAK

YEAR: 2021

ABSTRACT:

In this paper, we investigate novel data collection and training techniques towards improving classification accuracy ofnon-oving(static) hand gestures using a convolutional neural network(CNN)and frequency-modulated-continuous-wave (FMCW) millimeter-wave (mmWave) radars. Recently, non-contact hand pose and static gesture recognition have received considerable attention in many applications ranging from human-computer interaction(HCI), augmented/virtual reality (AR/VR), and even therapeutic range of motion for medical applications.While most current solutions rely on optical or depth cameras, these methods require ideal lighting and temperature conditions.

2.4 PAPER 4

TITLE: STeSH: Intelligent Speech Technology Enabled Smart Home Automation Using IoT

AUTHOR: Santosh Kumar Sahoo1, Sushant Kumar Pattnaik2, Soumya Ranjan Samal34 , Chinmay Kumar Nayak5, Jitendra Kuma Das2 and Vladimir Poulkov3

YEAR: 2022

ABSTRACT:

Due to the advancement in recent technologies and emergence of Internet of Things (IoT), Smart Home Automation (SHA) plays a vital role in today's lifestyle. The scheme implements a system named as "Speech Technology enabled Smart Home (STeSH)", which supports voice assisted home automation with extreme security and range-free localization.

2.5 PAPER 5

TITLE: A COMPARISON OF OPEN-SOURCE HOME AUTOMATION SYSTEMS

AUTHOR: BRIAN SETZ DESISLAVA IVANOVA 2,ALEXANDER TIESSEN 2, AND MARCOAIELLO

YEAR: 2021

ABSTRACT:

Homes are becoming an ecosystem of digital devices and appliances, which can be inter connected and controlled. This interconnection can be facilitated by a central smart hub on which home automation software is deployed. Commercially available hubs, while easy to install and use, often support a limited set of devices and protocols, and have a high total cost of ownership.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the current landscape of home automation, most systems rely on physical interaction through switches, mobile applications, or remote controls to manage appliances like doors and lights. While some advanced setups incorporate voice assistants such as Alexa, Google Assistant, or Siri, these systems still have limitations, particularly for users with speech or hearing impairments. Gesture-based control is not widely implemented, and where it does exist, it is often limited to predefined hand movements without support for full sign language recognition. Additionally, many of these systems lack integration with machine learning algorithms for adaptive and intelligent control. Most existing solutions also do not offer real-time hardware interaction based on both voice and gesture inputs, making them less accessible and less responsive for users requiring hands-free or assistive control options.

DISADVANTAGES:

- Most systems rely on physical touch or speech, making them unsuitable for individuals with physical disabilities or speech impairments.
- Existing systems do not incorporate sign language recognition, limiting gesture-based control options.
- Many systems do not support real-time gesture or voice command recognition, resulting in delays or low responsiveness.

3.2 PROPOSED SYSTEM

The proposed system introduces an intelligent, contactless home automation solution that utilizes both sign language recognition and voice keyword detection to control door and light operations. Using a live camera feed and a Ridge Classifier algorithm, the system accurately detects hand gestures in real-time, while simultaneously recognizing voice commands to provide an alternative mode of input. This dual-input approach ensures enhanced accessibility for users with physical or speech impairments, making home automation more inclusive. Once a gesture or command is recognized, the corresponding instruction is sent to a connected hardware module, such as an Arduino or Raspberry Pi, to perform the desired action. The system is designed to be scalable, cost-effective, and adaptable, with potential for integration into broader IoT frameworks.

ADVANTAGES

- Enables door and light operations without physical touch, enhancing hygiene and user convenience.
- Supports gesture-based input, making it accessible for users with speech or hearing impairments.
- Allows hands-free operation using voice keywords, offering an alternative control method.

3.3 PROPOSED SYSTEM ALGORITHM

3.3.1 RIDGE CLASSIFIER

The Ridge Classifier is a linear model for classification tasks that adapts Ridge Regression to classify discrete class labels. Unlike logistic regression, which models probabilities using a log-loss function, the Ridge Classifier applies a least-squares loss with L2 regularization and then uses the sign of the prediction to assign class labels.

3.3.2 BASIC PRINCIPLE

The Ridge Classifier treats classification as a regression problem by minimizing the squared error loss with an L2 regularization term. It solves the following optimization problem:

$$L(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \alpha \|w\|_2^2$$

Where:

the feature vector for the i-th sample

the target label (typically encoded as -1 and +1)

the weight vector

the regularization parameter

the squared L2 norm of the weight vector

After training, the model predicts the class label as:

$$\hat{y} = \text{sign}(w^T x + b)$$

This method helps reduce model variance and improves generalization, especially in the presence of multicollinearity or noise in the data.

3.3.3 APPLICATIONS

The Ridge Classifier is effectively used in:

- Text categorization: For instance, classifying emails as spam or not spam based on word frequencies.
- Image classification (with feature-extracted data): As a fast baseline model for labeled image datasets.

3.3.4 ADVANTAGES

- Handles multicollinearity well: L2 regularization mitigates the impact of correlated features.
- Fast and efficient: Especially suited for large-scale problems where a simple linear model is acceptable.
- Stable predictions: Regularization reduces the risk of overfitting, enhancing model generalization.

3.3.5 LIMITATIONS

- Linear decision boundaries: Cannot model complex, non-linear class relationships.
- Lacks probabilistic output: Unlike logistic regression, Ridge Classifier does not naturally provide class probabilities.
- Not suitable for highly imbalanced datasets: Without additional techniques like class weighting, performance may degrade.

3.3.6 CONCLUSION

The Ridge Classifier is a robust, regularized linear model well-suited for high-dimensional classification problems. While it lacks the flexibility of non-linear models like SVMs or neural networks, it offers a strong baseline with fast training, solid performance, and excellent generalization in structured data scenarios. It's especially favored when speed, interpretability, and resistance to overfitting are required.

CHAPTER 4

SYSTEM DESIGN

4.1 BLOCK DIAGRAM

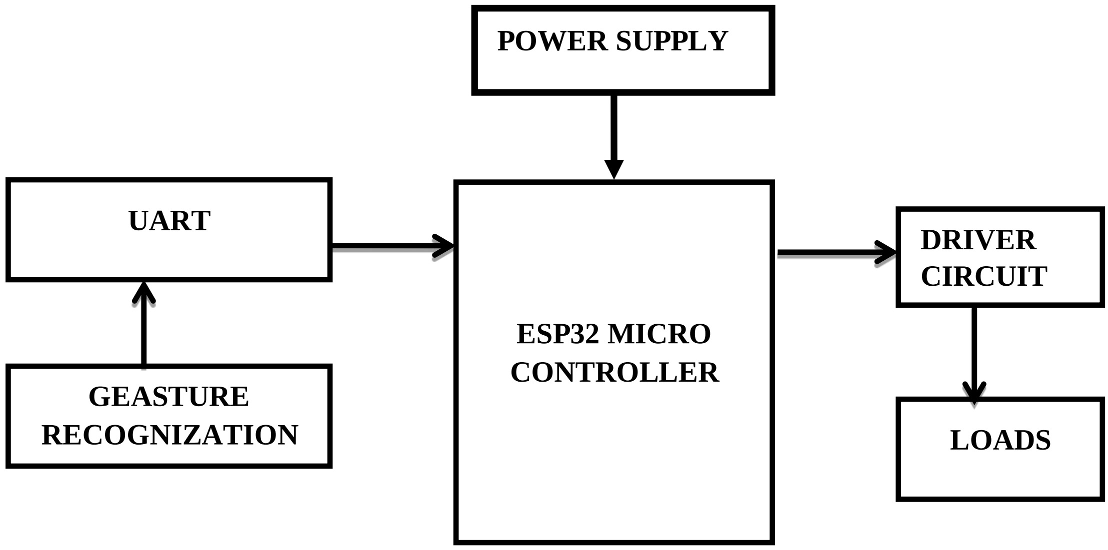


Figure 4.1 Block Diagram Of a Gesture-Controlled System Using an ESP32 Microcontroller.

4.2 ARCHITECTURE DIAGRAM

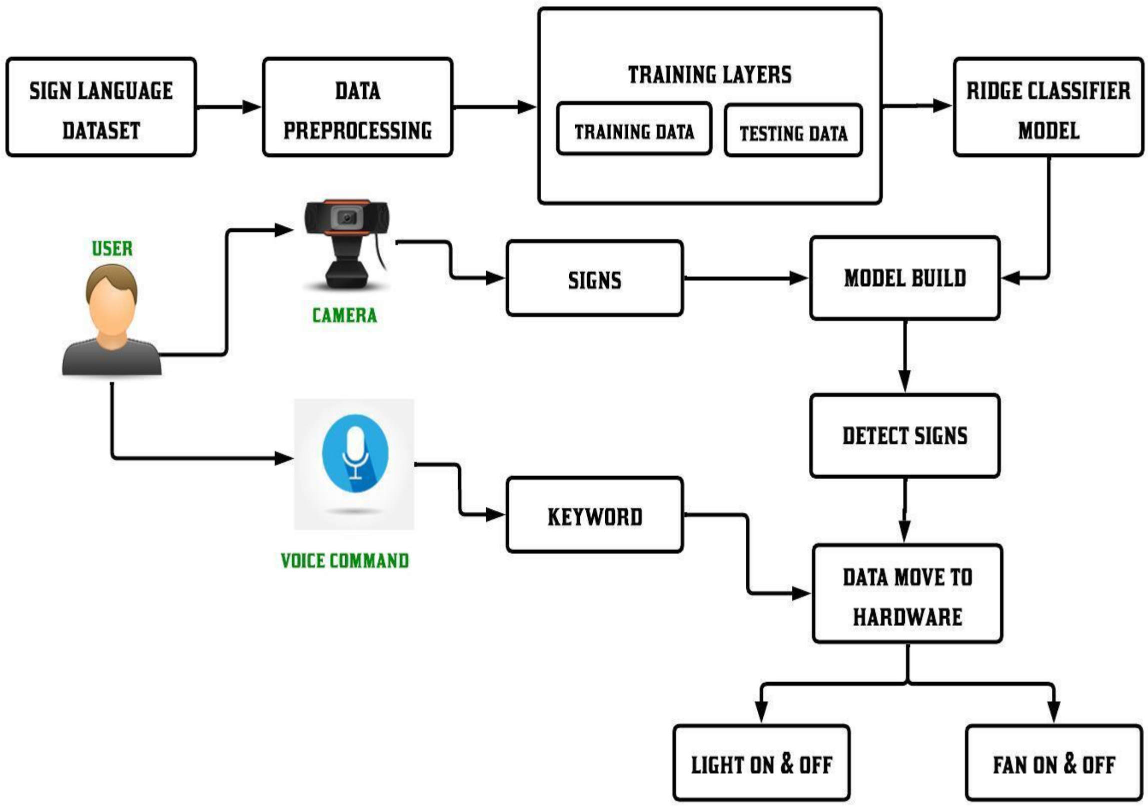


Figure 4.2 Architecture Diagram of a Smart Gesture-Controlled Home System.

4.3 UML DIAGRAMS

USE CASE DIAGRAM

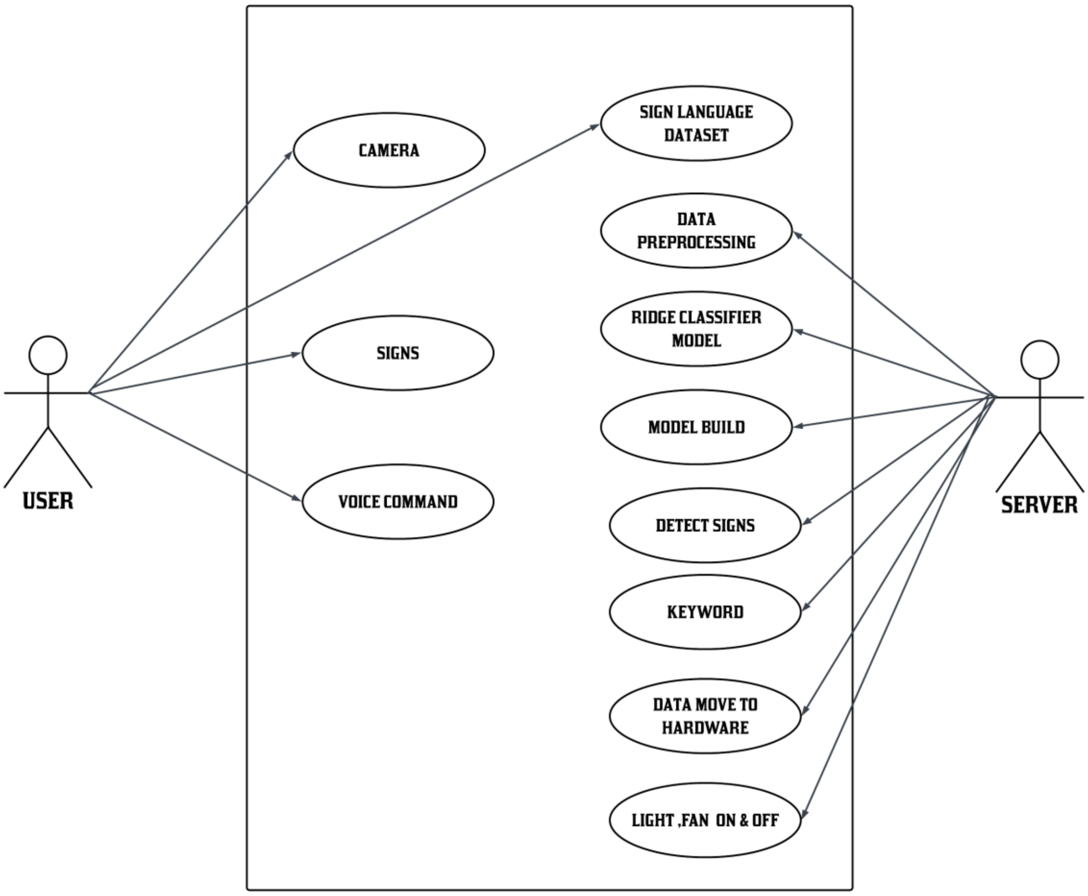


Figure 4.3 Use Case Diagram for a Smart Gesture-Controlled Home System

4.4 ACTIVITY DIAGRAM

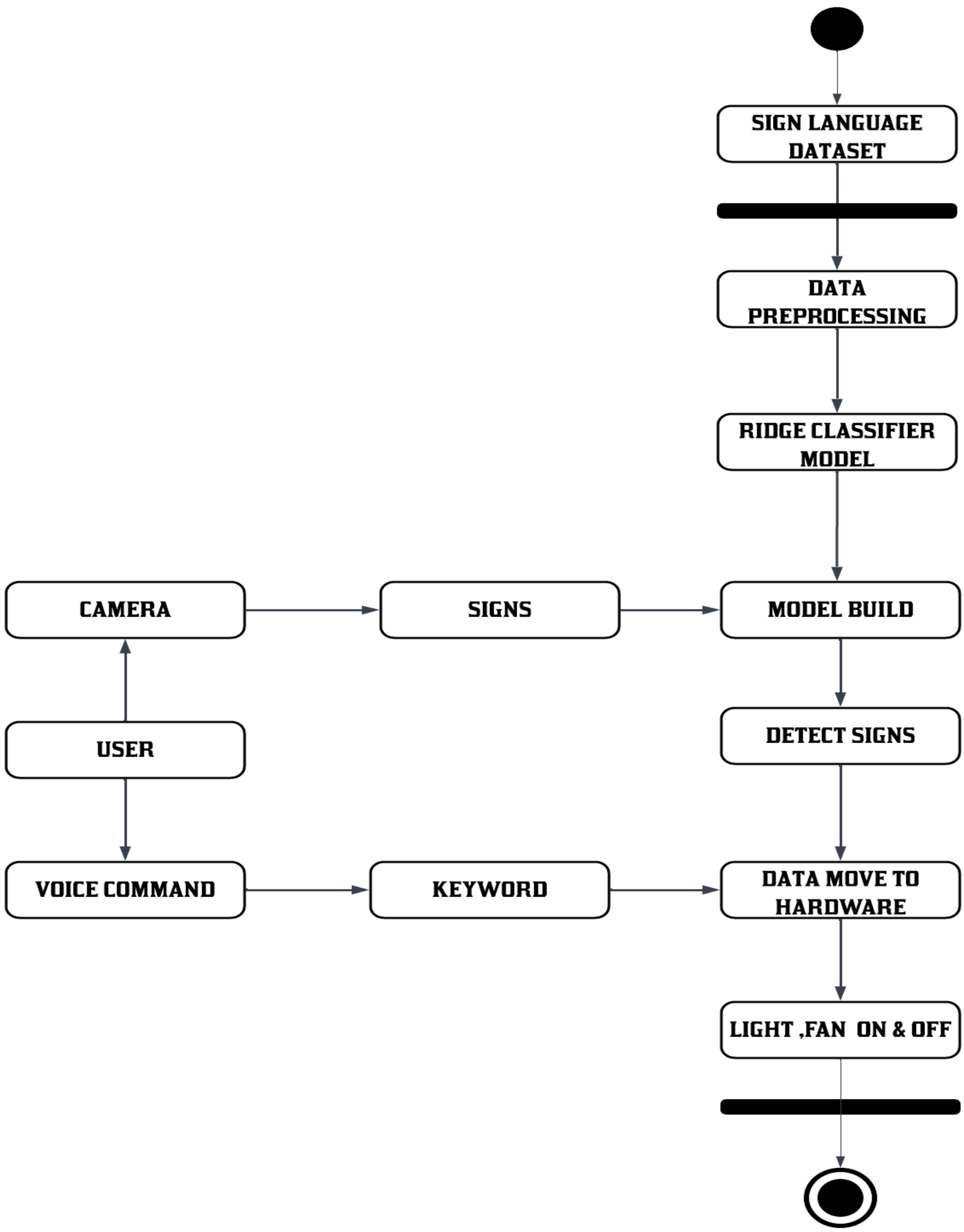
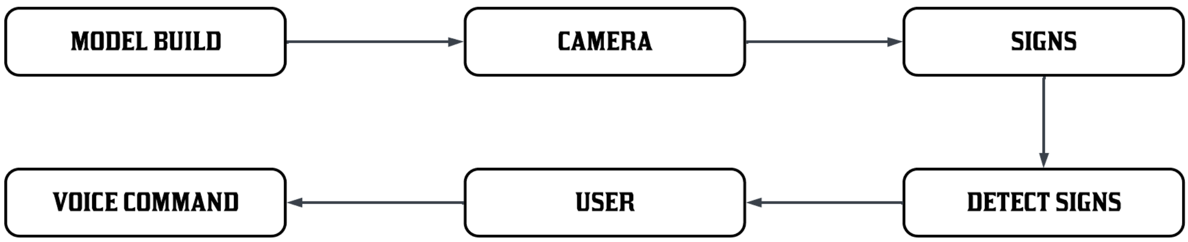


Figure 4.4 Activity Diagram for a Smart Gesture-Controlled Home System**.**

4.5 DATA FLOW DIAGRAM

LEVEL 0

LEVEL 1



LEVEL 2

Figure 4.5 Data Flow Diagram (DFD) for a Smart Gesture-Controlled Home System**.**

4.6 OVERALL DIAGRAM

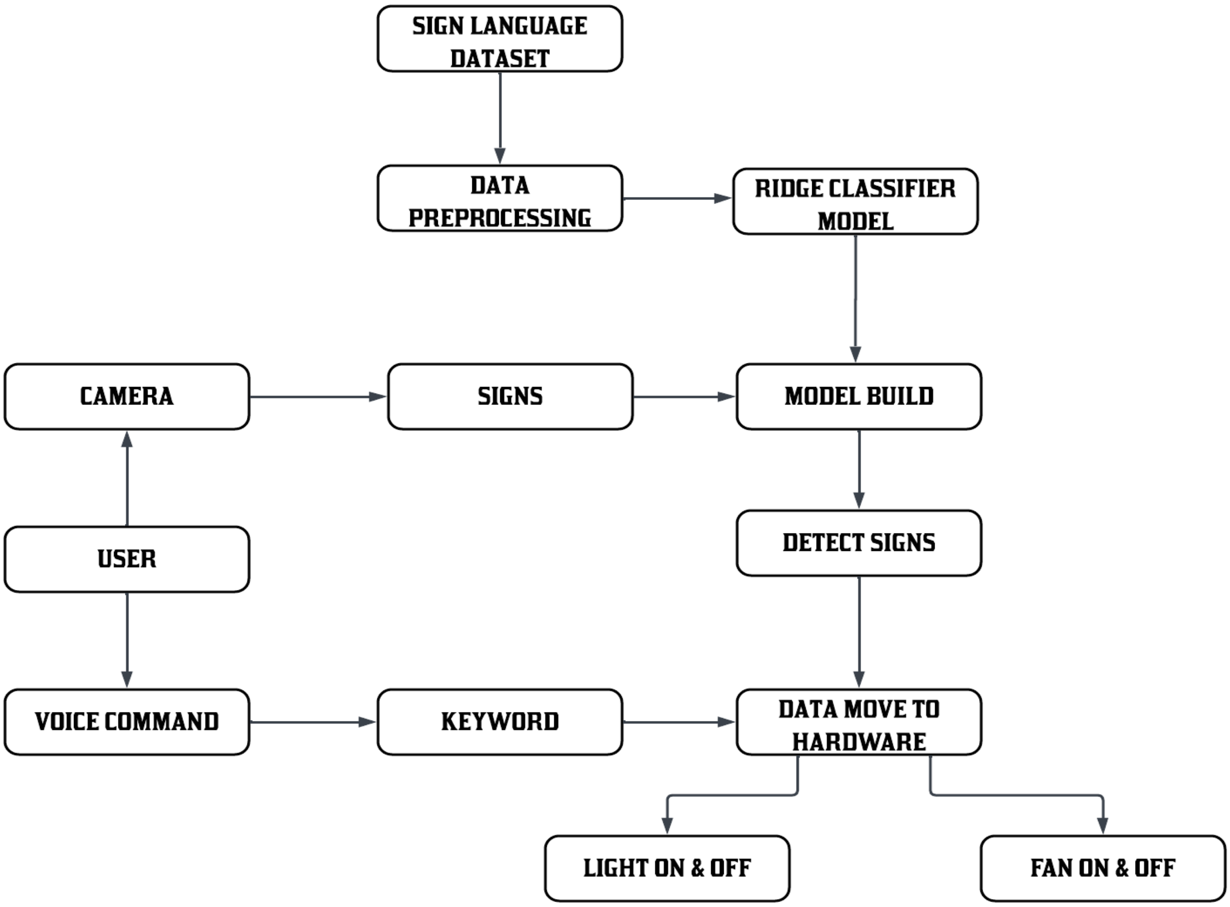


Figure 4.6 Overall Diagram

CHAPTER 5

SYSTEM SPECIFICATION

5.1 SYSTEM CONFIGURATION

5.1.1 H/W SYSTEM CONFIGURATION

- Processor – I3, i5,i7
- RAM - 8 Gb
- Hard Disk - 500 GB

5.1.2 S/W SYSTEM CONFIGURATION

- Operating System - Windows 8/10
- Scripts – Python (3.10.8)
- Tool – IDLE (Python)

5.2 HARDWARE REQUIREMENTS

- ESP 32 MICRO CONTROLLER
- POWER SUPPLY
- UART
- DRIVER CIRCUIT
- LOAD

5.3 SOFTWARE REQUIREMENTS

- EMBEDDED C
- ARDUINO IDE
- PYTHON IDE

CHAPTER 6

MODULE DESCRIPTIONS

6.1 GESTURE DETECTION MODULE

The system utilizes MediaPipe, a powerful and lightweight real-time hand tracking framework developed by Google, to detect and extract hand landmarks from the video stream. MediaPipe provides 21 key landmark points per hand, which serve as precise feature coordinates representing hand positions and finger movements.

6.2 COMMAND INTERPRETATION AND INTEGRATION MODULE

This intermediate module acts as the decision-making unit. It receives classified input from both the gesture and voice modules and maps the recognized gestures or voice keywords to specific hardware control actions. This module resolves conflicts and ensures the most recent or valid command is executed.

6.3 HARDWARE CONTROL MODULE

This module interfaces with physical hardware components like ESP32. Based on commands received from the interpretation module, it sends digital signals to actuators such as door locks, relays, or lights. For instance, if the command is “turn on light,” the ESP32 activates the corresponding pin to power a light bulb.

6.4 POWER SUPPLY ADAPTER

6.4.1 GENERAL DESCRIPTION

An adapter is a device that converts attributes of one electrical device or system to those of an otherwise incompatible device or system. Some modify power or signal attributes.

6.4.2 PRODUCT DESCRIPTION

An electric power adapter may enable connection of a power plug, sometimes called, used in one region to a AC power socket used in another, by offering connections for the disparate contact arrangements, while not changing the voltage. An AC adapter, also called a "recharger", is a small power supply that changes household electric current from distribution voltage) to low voltage DC suitable for consumer electronics. Some modify power or signal attributes, while others merely adapt the physical form of one electrical connector to another. For computers and related items, one kind of serial port adapter enables connections between 25-contact and nine-contact connectors, but does not affect electrical power- and signalling-related attributes.

Figure 6.4.3 Image of Power Supply Adapter

6.4.4 FEATURES

- Output current:1A
- Supply voltage: 220-230VAC
- Output voltage: 12VDC
- Reduced costs

- Increased value across front-office and back-office functions
- Access to current, accurate, and consistent data
- It generates adapter metadata as WSDL files with J2CA extension.

6.4.5 APPLICATIONS

- Back-end systems which need to send purchase order data to oracle applications send it to the integration service via integration server client.
- SMPS applications.

6.5 MICROCONTROLLER ESP32

6.5.1 WHAT IS ESP32?

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC.It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica's 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

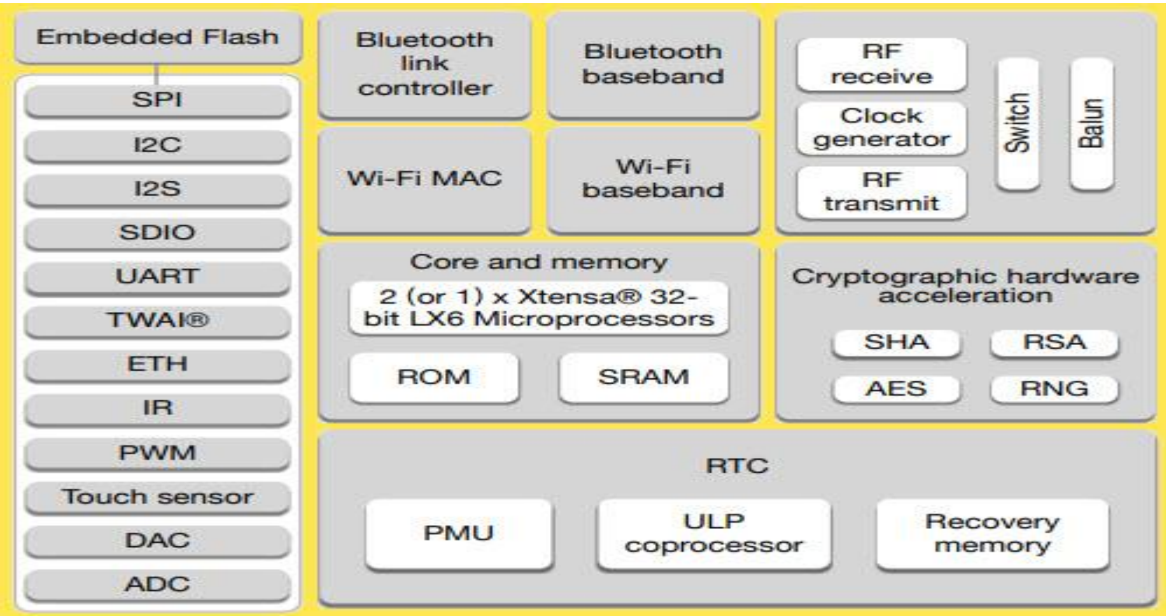


Figure 6.5.2 Block Diagram Of an ESP32 Microcontroller**.**

6.5.3 GENERAL DESCRIPTION

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process.

Figure 6.5.4 Pinout Diagram Of the ESP32-WROOM-32 Module.

CHAPTER 7

SOFTWARE DESCRIPTION

7.1 INTRODUCTION TO PYTHON

Python is a high-level object-oriented programming language that was created by Guido van Rossum. It is also called general-purpose programming language as it is used in almost every domain we can think of as mentioned below:

- Web Development
- Software Development
- Game Development
- AI & ML
- Data Analytics

7.1.1 WHY PYTHON PROGRAMMING?

Every Programming language serves some purpose or use-case according to a domain. for eg, Javascript is the most popular language amongst web developers as it gives the developer the power to handle applications via different frameworks like react, vue, angular which are used to build beautiful User Interfaces. Similarly, they have pros and cons at the same time. They can easily focus on business logic and Its demanding skills in the digital era where information is available in large data sets.

7.1.2 HOW DO WE STARTED?

Following are references where we can start our journey:

Official Website: https://www.python.org/

YouTube: https://www.youtube.com/watch?v=_uQrJ0TkZlc

CodeAcademy: https://www.codecademy.com/catalog/language/python

1) Visual Studio: https://visualstudio.microsoft.com/

2) PyCharm: https://www.jetbrains.com/pycharm/

3) Spyder: https://www.spyder-ide.org/

4) Google Colab: https://research.google.com/colaboratory/

7.1.3 REAL-WORLD EXAMPLES

1) NASA (National Aeronautics and Space Agency):

One of Nasa's Shuttle Support Contractors, United Space Alliance developed a Workflow Automation System (WAS) which is fast.

2) Netflix: There are various projects in Netflix which use python as follow:

- Central Alert Gateway
- Chaos Gorilla
- Security Monkey
- Chronos

7.1.4 APPLICATIONS OF PYTHON PROGRAMMING

1. Web Development: Python offers different frameworks for web development like Django, Pyramid, Flask. This framework is known for security, flexibility, scalability.
2. Artificial Intelligence and Machine Learning: There is a large number of open-source libraries which can be used while developing AI/ML applications.

CHAPTER 8

SYSTEM TESTING

8.1 SOFTWARE TESTING

8.1.2 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

8.1.3 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.1.4 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

CHAPTER 9

SOURCE CODING

9.1 INSTALLING PACKAGES

This code snippet sets up a basic framework for a machine learning project that uses computer vision and classification. It begins by importing essential libraries: mediapipe for detecting and tracking human landmarks (such as hands or pose), cv2 (OpenCV) for handling video and image processing, and pandas for data manipulation. It also includes key tools from scikit-learn, such as train_test_split for dividing data into training and testing sets, make_pipeline and StandardScaler for preprocessing data and building a scalable machine learning pipeline, and RidgeClassifier, a linear model used for classification tasks. This setup is commonly used in gesture recognition or pose-based activity classification projects.

This Python script performs a machine learning workflow to train and save a classification model using the Ridge Classifier algorithm. It begins by importing necessary libraries such as mediapipe, cv2, pandas, and scikit-learn modules. The dataset is loaded from a CSV file, with features (X) separated from the target class labels (y). The data is split into training and testing sets, and a pipeline is created that standardizes the feature values using StandardScaler and trains a RidgeClassifier. The model is trained on the training set and evaluated on the test set. Predictions are generated, and although the accuracy is computed in the loop, it's not printed or stored. Finally, the trained Ridge Classifier model is saved to a file named model.pkl using Python's pickle module for future use. This script provides a foundational example of building and exporting a machine learning model pipeline.

If you are looking to safely install global command line tools, see Installing stand alone command line tools.

Import Libraries

Import necessary libraries for:

MediaPipe (for landmark detection)

OpenCV (for video capture and processing)

Pandas (for data handling)

Scikit-learn (for ML tasks)

```
import mediapipe as mp
import cv2
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import RidgeClassifier
```

2. Capture Video Input

Initialize webcam capture using OpenCV.

```
cap = cv2.VideoCapture(0)
```

3. Initialize MediaPipe

Set up MediaPipe hands/pose/face detection.

mp_hands = mp.solutions.hands

hands = mp_hands.Hands()

mp_draw = mp.solutions.drawing_utils

4. Frame Processing Loop

Loop over each frame from the webcam:

Read the frame

Convert it to RGB

5. Extract Landmark Coordinates

If landmarks are found:

Extract (x, y, z) coordinates

Store in a structured format (e.g., list or DataFrame)

6. Label Data

Manually or programmatically label the landmarks with the corresponding gesture/pose.

7. Prepare Dataset

Convert the list of landmarks and labels to a Pandas DataFrame.

```
df = pd.DataFrame(data)  # data = list of landmarks + labels
X = df.drop('label', axis=1)
y = df['label']
```

8. Split Data

Split the dataset into training and testing sets.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

10. Train the Model

Train the model using training data.

```
model.fit(X_train, y_train)
```

11. Evaluate the Model

Test the model and check accuracy.

```
accuracy = model.score(X_test, y_test)
```

FRONTEND

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>Sign Language Recognition</title>

  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&displa
y=swap" rel="stylesheet">

  <style>

    * {

      margin: 0;

      padding: 0;

      box-sizing: border-box;

    }

    body {

      font-family: 'Roboto', sans-serif;

      background: url('static/2.jpg') no-repeat center center/cover;

      height: 100vh;

      display: flex;

      align-items: center;

      justify-content: center;

      backdrop-filter: blur(2px);
```

```
  color: white;
 }
 .container {
   background: rgba(255, 255, 255, 0.1);
   border-radius: 20px;
   padding: 50px 40px;
   text-align: center;
   box-shadow: 0 8px 32px rgba(0, 0, 0, 0.3);
   backdrop-filter: blur(12px);
   -webkit-backdrop-filter: blur(12px);
   border: 1px solid rgba(255, 255, 255, 0.18);
   animation: fadeIn 1s ease-in-out;
   max-width: 600px;
   width: 90%;
 }
.button {
  display: inline-flex;
  align-items: center;
  gap: 10px;
  padding: 14px 28px;
  font-size: 16px;
```

```
  padding: 14px 28px;
  font-size: 16px;
  font-weight: 600;
  color: #fff;
  background: linear-gradient(135deg, #00c6ff, #0072ff);
  border: none;
  border-radius: 10px;
  cursor: pointer;
  transition: 0.3s ease;
  text-decoration: none;
  box-shadow: 0 4px 14px rgba(0, 0, 0, 0.2);
 }
.button:hover {
  background: linear-gradient(135deg, #ff416c, #ff4b2b);
  transform: translateY(-3px) scale(1.05);
 }

.button {
  width: 100%;
  justify-content: center;
 }
```

```
</style>
</head>
<body>
  <div class="container">
    <h1>Multi Modal Smart Home Control</h1>
<a href="{{ url_for('generate_frames') }}" class="button">
      <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path
d="M12 5v14m7-7H5"/></svg>
      Open Camera
    </a>
    <a href="/speak" class="button" onclick="startSpeech()">
      <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path
d="M12 1a4 4 0 0 1 4 4v6a4 4 0 1 1-8 0V5a4 4 0 0 1 4-4z"/><path d="M19
10v2a7 7 0 0 1-14 0v-2"/><path d="M12 19v4m-4 0h8"/></svg>
      Speech
    </a>

    <div id="loadingMessage"> Listening... Please speak clearly.</div>
  </div>
  <script>
```

```
    function startSpeech() {
      const button = document.querySelector('a[href="/speak"]');
      const loadingMessage = document.getElementById('loadingMessage');
      button.style.pointerEvents = 'none';
      button.style.opacity = '0.5';
      loadingMessage.style.display = 'block';
setTimeout(() => {
        loadingMessage.style.display = 'none';
        button.style.pointerEvents = 'auto';
        button.style.opacity = '1';
      }, 5000);
    }
  </script>
</body>
</html>
```

CHAPTER 10

RESULT ANALYSIS

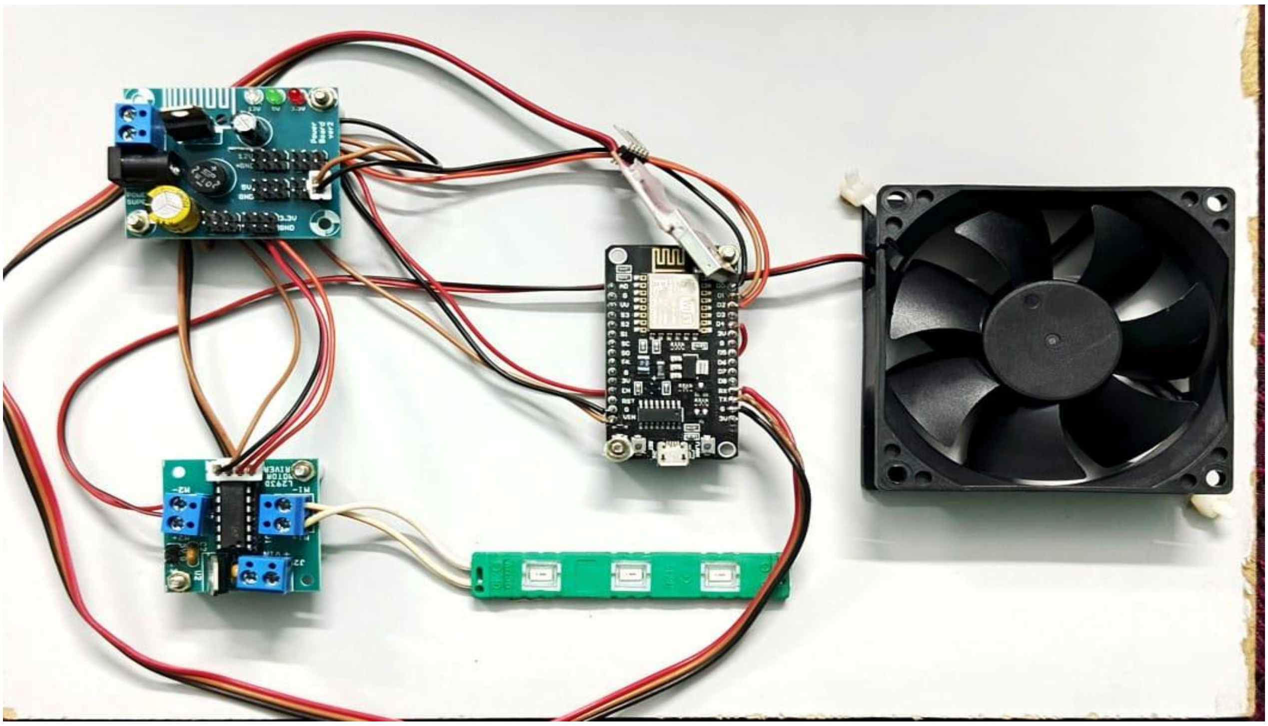


Figure 10.1 IoT-Based Smart Cooling and Lighting Control System

This image showcases an IoT-based hardware setup involving several commonly used electronic components. At the center of the configuration is an ESP8266 NodeMCU microcontroller, which acts as the brain of the system, providing Wi-Fi connectivity and control logic. To its left is a L298N motor driver module, which is used to control high-power devices such as the DC fan visible on the right side of the image. The L298N receives control signals from the NodeMCU and powers the fan accordingly. Above the L298N is a voltage regulator module (likely LM2596), which ensures a stable voltage supply to the circuit, converting higher voltages (like from a battery pack) down to 5V or 3.3V as needed. At the bottom center is a strip of SMD RGB LEDs, which are likely being used for status indication or aesthetic feedback. The modules are connected via jumper wires, with power and signal lines routed carefully to ensure proper functioning. This setup is likely used in a smart home or automation project, possibly for environmental control (like automated fan control based on temperature or humidity sensed remotely).

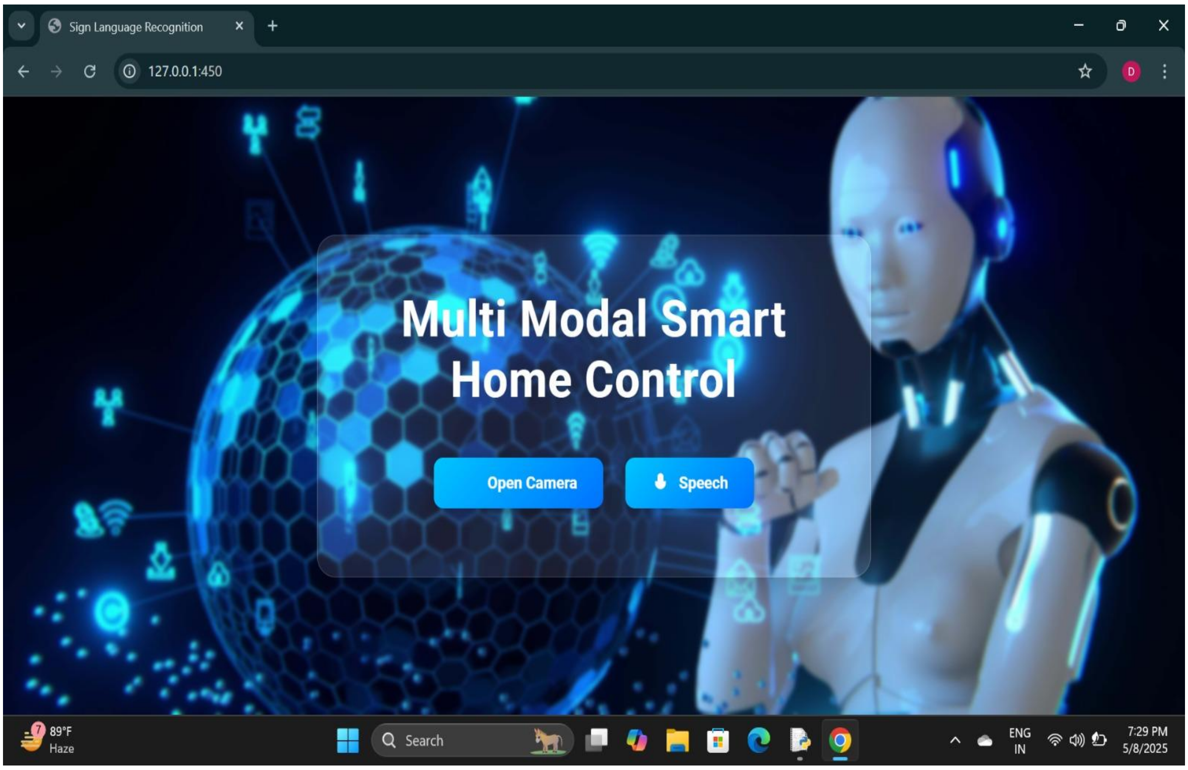


Figure 10.2 Multimodal Interface for AI-Based Smart Home Control

This image displays the user interface of a Multi Modal Smart Home Control system running on a local web server (as indicated by the address 127.0.0.1:450). The interface features a futuristic design, showcasing a humanoid robot and a digitally rendered globe surrounded by IoT icons, representing various smart home devices and technologies. At the center, the title “Multi Modal Smart Home Control” highlights the system’s capability to operate using multiple input modes. Below the title, there are two interactive buttons labeled

"Open Camera" and "Speech", suggesting that the system can be controlled via real-time camera input (likely for gesture or sign language recognition) and speech commands, enabling intuitive and accessible home automation. The webpage is titled "Sign Language Recognition," which further implies that the system integrates computer vision to recognize hand gestures or signs for controlling home appliances. This project merges AI, computer vision, and IoT, aiming to provide a smart and inclusive environment, particularly useful for users with speech or mobility impairments.

CHAPTER 11

CONCLUSION

In conclusion, the proposed sign language and voice command-based home automation system offers an innovative and accessible solution for controlling essential household appliances, such as doors and lights. By integrating machine learning for real-time gesture recognition and voice keyword detection, the system provides a seamless and intuitive user experience, particularly for individuals with disabilities or those in need of contactless control. The ability to interface directly with hardware platforms, like Arduino or Raspberry Pi, ensures the system's practicality and scalability.

FUTURE ENHANCEMENT

In the future, this smart home automation system can be enhanced with several advanced features to further improve usability, scalability, and accessibility. One key enhancement would be the integration of deep learning models such as Transformer-based architectures to improve the accuracy and robustness of both gesture and voice recognition, especially under varying lighting and noise conditions. The system can also be expanded to recognize a broader set of sign language gestures, enabling control over a wider range of smart appliances such as fans, air conditioners, or security systems. Cloud connectivity and IoT integration can allow users to monitor and control their homes remotely via a mobile app, while also enabling data logging for personalized automation patterns.

CHAPTER 12

REFERENCE

[1] F. Jafari and A. Basu, "Two-Dimensional Parallel Spatio-Temporal Pyramid Pooling for Hand Gesture Recognition," in IEEE Access, vol. 11, pp. 133755-133766, 2023, doi: 10.1109/ACCESS.2023.3336591.

[2] G. Park, V. K. Chandrasegar and J. Koh, "Accuracy Enhancement of Hand Gesture Recognition Using CNN," in IEEE Access, vol. 11, pp. 26496-26501, 2023, doi: 10.1109/ACCESS.2023.3254537.

[3] J. W. Smith, S. Thiagarajan, R. Willis, Y. Makris and M. Torlak, "Improved Static Hand Gesture Classification on Deep Convolutional Neural Networks Using Novel Sterile Training Technique," in IEEE Access, vol. 9, pp. 10893-10902, 2021, doi: 10.1109/ACCESS.2021.3051454.

[4] S. K. Sahoo, S. K. Pattnaik, S. R. Samal, C. K. Nayak, J. K. Das and V. Poulkov, "STeSH: Intelligent Speech Technology Enabled Smart Home Automation Using IoT," in Journal of Mobile Multimedia, vol. 18, no. 6, pp. 1471-1496, November 2022, doi: 10.13052/jmm1550-4646.1861.

[5] B. Setz, S. Graef, D. Ivanova, A. Tiessen and M. Aiello, "A Comparison of Open-Source Home Automation Systems," in IEEE Access, vol. 9, pp. 167332-167352, 2021, doi: 10.1109/ACCESS.2021.3136025.

[6] L. Yuan, ‘‘Remote sensing image classification methods based on CNN: Challenge and trends,’’ in Proc. CONF-SPML, Stanford, CA, USA, Nov. 2021, pp. 213–218.

[7] K.Sha que,B.A.Khawaja,F.Sabir, S. Qazi, and M. Mustaqim, Internet of

Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios, IEEE Access, vol. 8, pp. 2302223040, 2020.

[8] S. Skaria, A. Al-Hourani, and R. J. Evans, ‘‘Deep-learning methods for hand-gesturerecognitionusingultra-widebandradar,’’IEEEAccess,vol.8, pp. 203580–203590, 2020, doi: 10.1109/ACCESS.2020.3037062.

[9] M. E. Yanik, D. Wang, and M. Torlak, Development and demonstra tion of MIMO-SAR mmWave imaging testbeds, IEEE Access, vol. 8, pp. 126019126038, 2020.

[10] J. W. Smith, M. E. Yanik, and M. Torlak, Near- eld MIMO-ISAR millimeter-wave imaging, in Proc. IEEE Radar Conf. (RadarConf), Sep. 2020, pp. 16.