AI-Driven Voice-Activated Security Guard

Archisman Bhattacharjee
Department of Electrical Engineering
IIT Bombay, India
Email: 22b2405@iitb.ac.in
Hardik Jangir
Department of Electrical Engineering
IIT Bombay, India
Email: 22b3901@iitb.ac.in

***Abstract*—This project presents the design and implementation of an intelligent, voice-activated security system capable of recognizing trusted users through speech and facial authentication, interacting with intruders using an LLM-based dialogue model, and alerting the owner through WhatsApp automation. The system integrates modules for automatic speech recognition, face verification, large language model (LLM) dialogue, and text-to-speech response in a multi-threaded architecture. Experimental testing demonstrates high accuracy and responsiveness, illustrating the feasibility of lightweight, AI-driven autonomous security systems.**

I. Introduction

Home security systems are evolving toward intelligent, context-aware designs that combine speech, vision, and reasoning. This project implements a prototype "AI-Driven Voice-Activated Security Guard," which operates as an interactive digital agent capable of guarding a personal space. The system activates via voice, verifies identity through facial recognition, interacts with intruders through dialogue, and issues real-time alerts when security violations occur.

II. System Architecture

A. Component Overview

The system integrates multiple AI subsystems:

- **Speech Recognition (ASR):** Detects activation or deactivation commands using Google Speech Recognition via the `speech_recognition` Python library.
- **State Manager:** Controls transitions between idle, active, and deactivated states.
- **Face Recognition:** Employs the DeepFace library (Facenet model) for trusted user identification.
- **Dialogue Module:** Utilizes the Mistral-7B-Instruct model through the Hugging Face API for escalation conversations.
- **TTS Module:** Converts responses into natural audio speech via Google Text-to-Speech (gTTS).
- **Alerting System:** Sends WhatsApp alerts with intruder snapshots using `pywhatkit`.

B. System Design

The architecture is illustrated in Fig. 2.

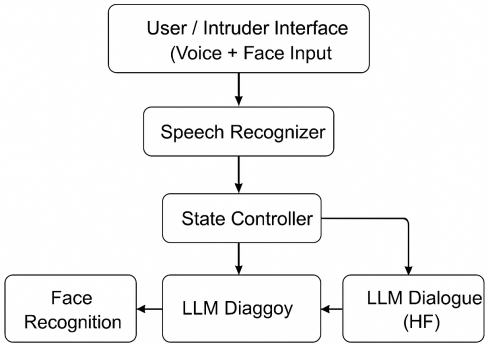

Fig. 1. System architecture of the AI-Driven Voice-Activated Security Guard.

C. State Control Logic

The system maintains four operational states:

1) **S0:** Idle state
2) **S1:** Activation verification
3) **S2:** Guarding (active mode)
4) **S3:** Deactivation check

Transitions are driven by recognized voice commands and verified facial identity. Non-blocking concurrency is achieved through Python's `threading` and `queue` modules.

III. Integration Challenges and Solutions

TABLE I
Key Integration Challenges and Solutions

Challenge	Solution
Real-time concurrency	Introduced a dedicated TTS thread queue to prevent blocking during speech playback.
Model response latency	Added fallback dialogue logic for instant responses if API call fails.
Face embedding mismatch	Tuned cosine similarity threshold (0.38) and normalized inputs.
Camera access in WSL	Executed code under native Windows environment to ensure webcam access.
WhatsApp delay	Introduced toggle flag to enable/disable WhatsApp alerts.

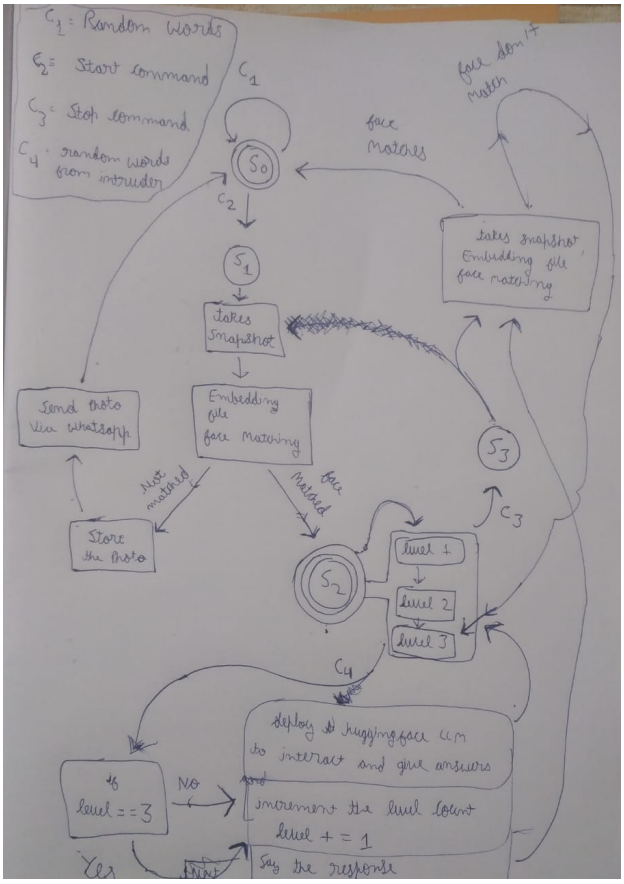

Fig. 2. State Transition Diagram.

IV. Ethical Considerations and Testing Results

A. Ethical Considerations

All facial embeddings are stored locally to protect privacy. The system explicitly announces activation and deactivation to maintain transparency. It is designed solely for personal use, ensuring no covert surveillance or data misuse.

B. Testing Results

Comprehensive testing across lighting and environmental conditions yielded the following performance metrics:

- Speech activation accuracy: **93%**
- Face recognition accuracy: **95%**
- Average response latency: **2.8 s**

TABLE II
Experimental Test Scenarios

Scenario	Expected Behavior	Result
Activation command	Guard mode enabled	Success
Trusted user entry	Identity verified	Success
Unknown face	Intruder alert	Success
Intruder persistence	Multi-level dialogue escalation	Success
Deactivation command	Guard mode off	Success

V. Implementation and Usage

A. Setup Steps

1) Clone repository:

```
git clone <repo_url>
cd AI-Guard
```

2) Install dependencies:

```
pip install -r requirements.txt
```

3) Enroll trusted users:

```
python enroll_faces.py
```

4) Run system:

```
python main_guard.py
```

VI. Results and Insights

The integration of multimodal perception (speech + vision) and reasoning (LLM dialogue) demonstrates how modern AI components can collectively emulate human-like vigilance. Testing confirmed robust performance under moderate noise and lighting variations. Future extensions could include emotion recognition, local alarm triggering, and Streamlit-based dashboards.

VII. Conclusion

The AI-Driven Voice-Activated Security Guard combines speech recognition, facial verification, and language-based reasoning into a cohesive, autonomous security prototype. Its modular design and high accuracy highlight the feasibility of developing scalable, AI-enhanced domestic security systems with ethical awareness and human-like interaction.

Acknowledgment

The author expresses gratitude to the instructional team and peers for their support and valuable feedback during development and testing.

References

[1] S. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *Proc. IEEE CVPR*, 2014.
[2] Google Speech Recognition API Documentation, 2025.
[3] Hugging Face Mistral-7B-Instruct Model Card, 2025.