EE782: Assignment 2 – AI Guard Agent

Shounak Das (21D070068)

Varunav Singh
(21D070086)

1. Introduction

The objective of this assignment was to design and implement a real-time AI agent that could act as a virtual room guard monitoring and communicating via visual and audio mediums. Upon activation via voice command the AI agent will enter monitoring mode, recognize trusted individuals, and initiate an escalating conversation with unrecognized persons.

2. System Architecture

We have given a figure below which describes the system architecture as a flowchart. The architecture is fixed, and consists of the following flow-

1. **Listening Mode**: Listening to detect “Guard my room” in spoken form to activate guard mode
2. **Guard Mode**: Watch via webcam whether a person is seen who is not in the room. At the startup of this mode, the faces that have been given to the model are saved and kept to match to new faces.
    a. If trusted face detected: Guard Mode terminates and system stops.
    b. If trusted face is not detected:
        i. **1st Escalation**: System asks the person for password to identify themselves
        ii. **2nd Escalation**: System asks the person more firmly to give the password, and warns them that the area is under constant surveillance and they are as well.
        iii. **3rd Escalation**: System tells the person that a security alert has been sent, this is the last chance to answer with the password.
    c. If the password is said after any escalation, the system stops after letting the person know the password has been recognized. The system also terminates after Escalation 3 (ideally a mail or something should be sent but we were not asked to do so).
    d. For each of the 3 escalations, we have given 3 prompts. The prompts have been designed to accept a string variable which gives the LLM an example to model its response after, along with extra details to make sure it does not add anything extra such as non english characters or other effects. We also utilized meta-prompting using Gemini Flash 2.5 to ensure that the prompt is as crisp and delivers consistent results, as any non-english character or line changes may lead to a weird audio response from the system. We also ensured that the strings are good enough on their own that in case the LLM fails to respond, we can use the strings as the default response.

All information is conveyed verbally by the system as well as visually as part of terminal printout. The escalation prompts and strings have been given in Milestone 3 section for Escalation.

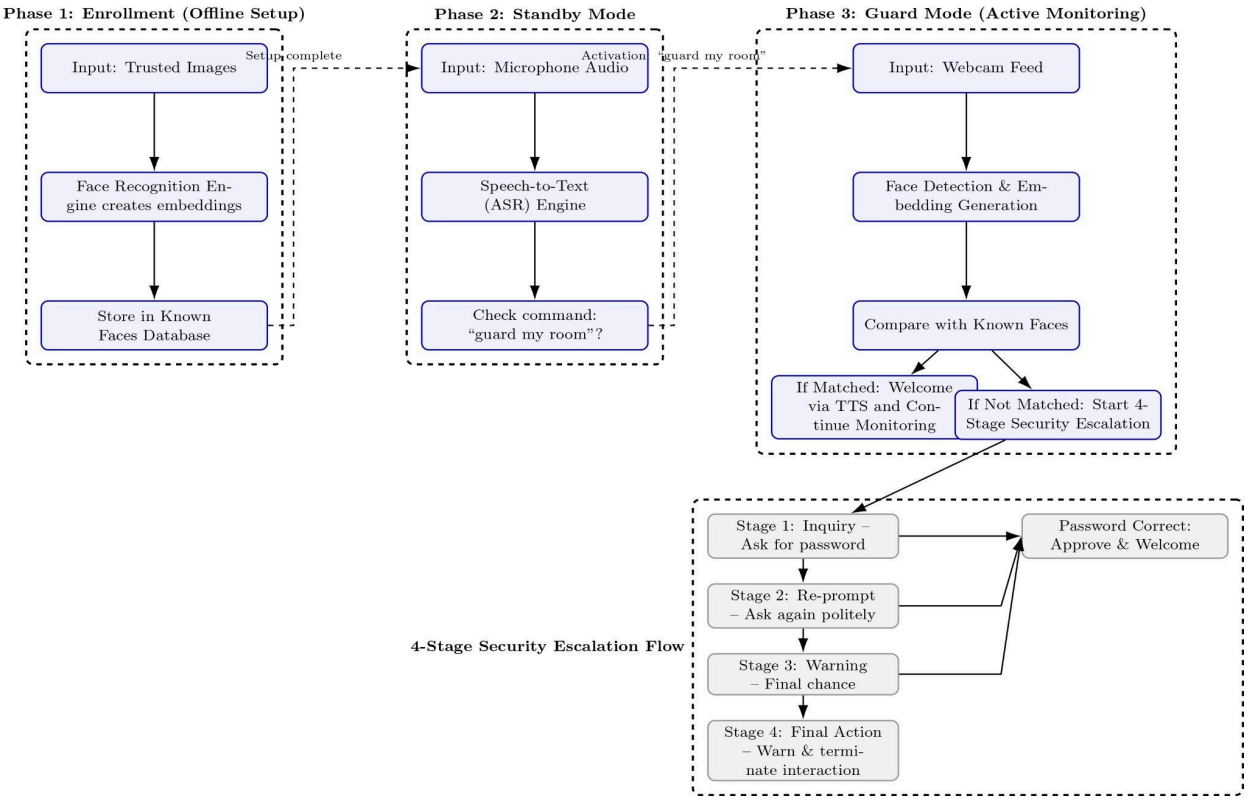

Figure 1: Flow chart of AI Guard Agent

3. Milestone 1 : Activation and Basic Input

The seamless integration is visible in the demo, where we have shown how smoothly all the transitions work without unnecessary lag, and if any time is being taken, we print out the reason for the time being taken, and showing what is happening during that process.

4. Milestone 2 : Face Recognition and Trusted User Enrollment

We successfully implemented and validated the face recognition and user enrollment module. To achieve this, a trusted individual ("Varunav") was enrolled using two distinct photographs to test intra-subject consistency, while an untrusted individual ("Shounak") was used to test inter-subject differentiation. The system's efficacy was verified by generating unique 128-dimensional face embeddings for each image. A visual analysis of these plotted embeddings revealed a high correlation between the two trusted images and a distinct pattern for the untrusted one. This finding was further substantiated by a quantitative analysis of reconstructed "ghost images," which yielded a high Structural Similarity Index Measure (SSIM) of 0.9765 and a low Mean Squared Error (MSE) of 0.0013 for the trusted pair. In contrast, comparisons against the untrusted individual resulted in significantly lower SSIM scores (~0.86) and higher MSE values (~0.011). This clear numerical and visual distinction confirms the module's accuracy in reliably differentiating between enrolled and unknown individuals.

5. Milestone 3 : Escalation Dialogue and Full Integration

We implemented a 4-level escalating dialogue to manage unrecognized individuals, centered around a password-based verification system. The interaction begins with a polite request for the password. If the response is incorrect or absent, the system escalates to a firmer tone, informing the individual that they are under surveillance. The final escalation level issues a stern warning, stating that a security alert has been triggered and providing one last opportunity to comply. This entire dialogue is powered by the Google Gemini Flash 1.5 model, which is dynamically prompted at each stage. We engineered a robust prompt that instructs the AI to paraphrase a predefined "base text" for each level, ensuring the core message is delivered politely but firmly in a clean, single line of text suitable for our Text-to-Speech (TTS) engine. This method not only provides a creative and non-repetitive interaction but also includes a fallback mechanism where the base text can be used directly if the LLM fails. All system responses are conveyed to the user through both spoken audio and text printouts in the terminal for clarity.

The **Prompt** is:

You are a Security Guard guarding a room. Paraphrase and give a response akin to '{base_text}' politely yet firmly without changing the meaning or omitting any information. The response must be a single line of plain text only, containing no introductory phrases, formatting characters (like * or **), or parenthetical actions.

The base text is one of these 3, depending on the escalation:

1. Please Identify yourself via password.
2. Password not recognized. Please leave or tell me the password. This room is being monitored.
3. Security alert triggered! This is your last attempt to tell the password. If you don't know the password, leave immediately

6. Milestone 4 : Polish and Stretch Goals (Bonus)

We **stretched** goals by adding a password based escalation logic as part of the bonus of stretching the goals. The password, while we have used a single word (“pineapple”) and can be called keyword detection, it can be multiple words and can be phrase detection as well.

To **polish**, we have mentioned in the System Architecture and Milestone 3 section the various measures (Meta prompting, prompt engineering: using appropriate adjectives and language, assigning proper roles (such as You

are a Guard)) we have taken to ensure the responses from Gemini are appropriate, concise. We also ensured the audio response from the system (including the LLM response) and the general system responses are always the same as the printed responses acting as subtitles, ensuring no response is ever missed. This makes the system messages impossible to ignore for any person in the room.

7. Results

This section presents the experimental results of the face recognition module. The objective is to demonstrate the system's ability to distinguish between a trusted individual ("Varunav") and an untrusted individual ("Shounak") by analyzing their face embeddings both visually and quantitatively.

7.1. Input Images

Three images were used for this analysis. Two distinct images of the trusted individual, Varunav, were captured under slightly different conditions to test intra-subject similarity. One image of an untrusted individual, Shounak, was captured to test inter-subject dissimilarity.

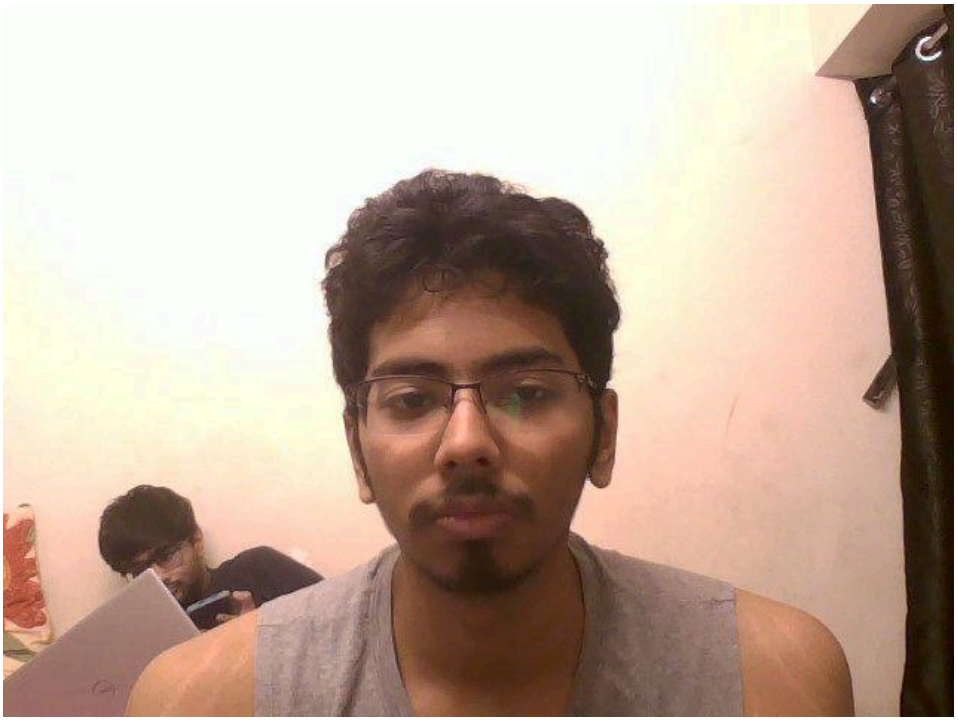

Figure 7.1: Trusted Person - Varunav (Image 1)

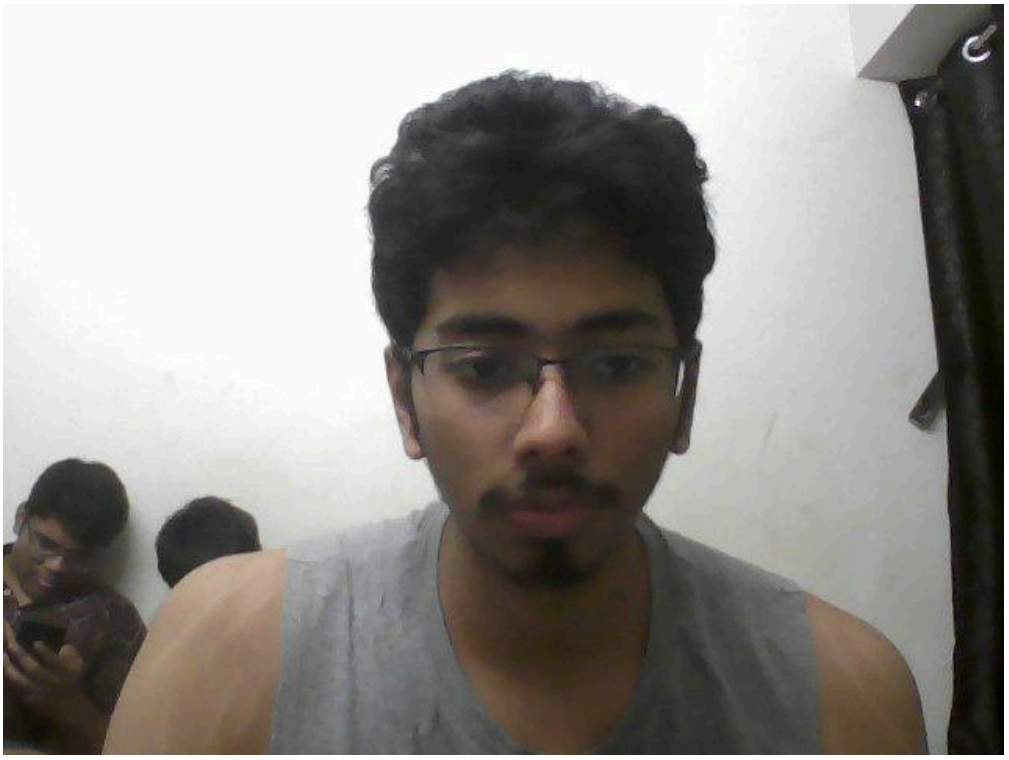

Figure 7.2: Trusted Person - Varunav (Image 2)

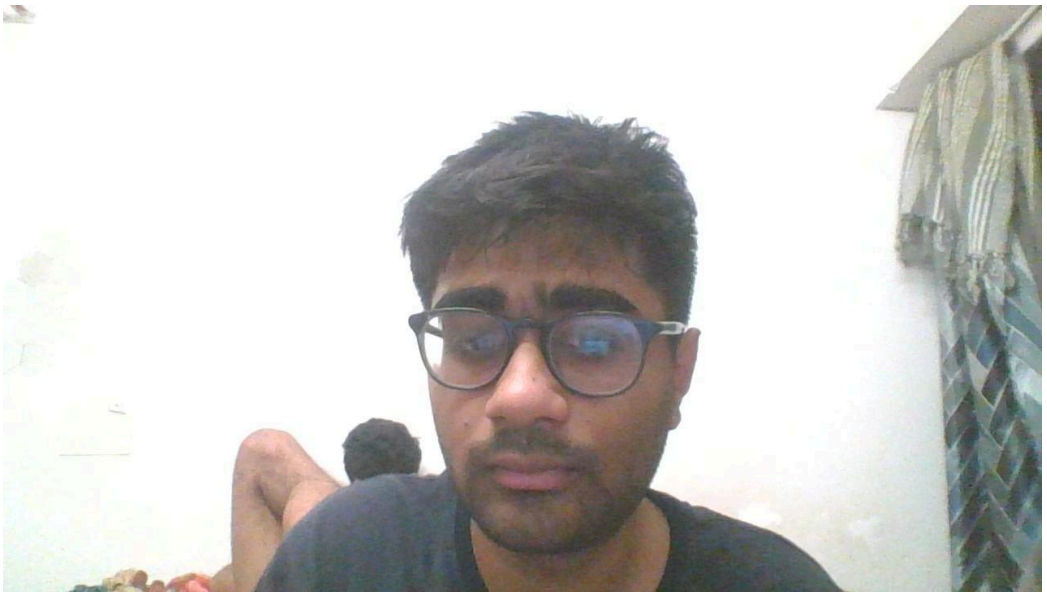

Figure 7.3: Untrusted Person - Shounak

7.2. Face Embedding Visualization

For each input image, the face recognition model generated a 128-dimensional vector, or "embedding." This vector serves as a unique numerical signature for the face. The embeddings for all three images are plotted below for visual comparison.

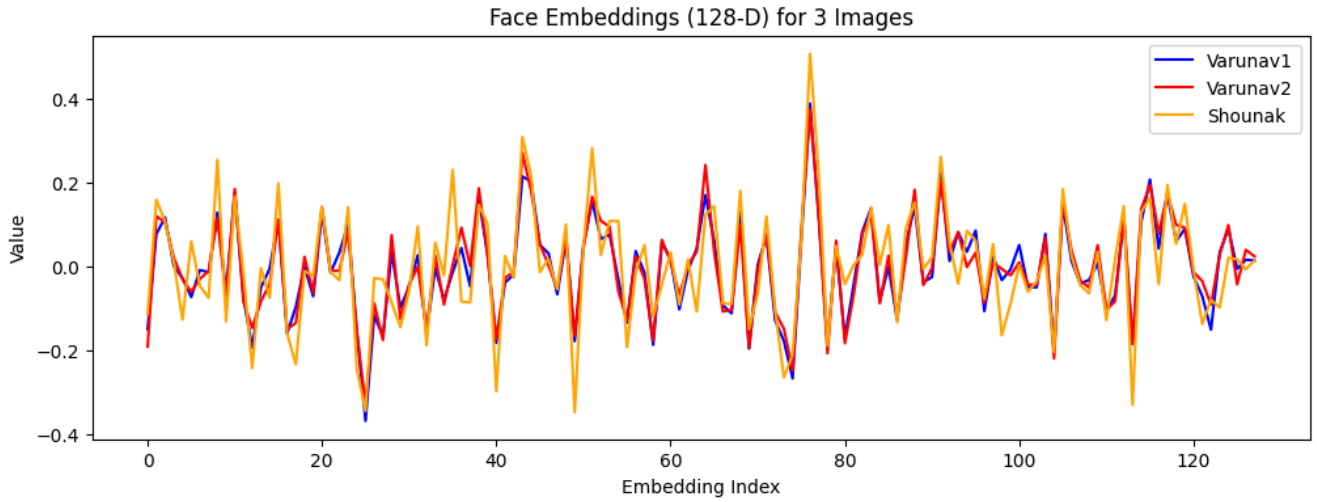

Figure 7.4: Comparison of 128-d Face Embeddings

As illustrated in Figure 7.4, the embeddings for Varunav 1 (blue) and Varunav 2 (Red) exhibit a very high degree of correlation. Their plots follow an almost identical trajectory, with minor variations attributable to differences in lighting and angle in the source images.

In contrast, the embedding for Shounak (Orange) shows a slightly different pattern. While some regions may coincidentally overlap, its key peaks and valleys are structurally different from those of Varunav, confirming that the model has generated a unique signature for this individual.

7.3. Quantitative Analysis: SSIM and MSE

To provide a numerical validation, we reconstructed a visual representation ("ghost image") from each of the three embeddings. We then performed a pairwise comparison of these reconstructed images using two standard metrics:

- Structural Similarity Index Measure (SSIM): A value from 0 to 1 indicating how similar two images are. A value of 1.0 means they are identical.
- Mean Squared Error (MSE): Measures the average squared difference between the pixels of two images. A value closer to 0 indicates higher similarity.

The results are summarized in the table below.

Comparison Pair	SSIM (Higher is Better)	MSE (Lower is Better)	Interpretation
Varunav 1 vs. Varunav 2	0.9765	0.0013	Very High Similarity
Varunav 1 vs. Shounak	0.8773	0.0116	Low Similarity
Varunav 2 vs. Shounak	0.8593	0.0110	Low Similarity

The quantitative results strongly support the visual findings. The SSIM score between the two images of Varunav is extremely high (0.9765), and the MSE is very low (0.0013), confirming that the system correctly identifies them as the same person.

Conversely, when comparing either of Varunav's images to Shounak's, the SSIM scores are significantly lower, and the MSE values are an order of magnitude higher. This large numerical distance corroborates that Shounak is a different

individual. These metrics provide a clear threshold for the system to differentiate between "trusted" and "untrusted" persons.