Embedded AI Project Report

Aditya Somasundaram, Likhith Ayinala, Waasi Jagirdar

1 Problem Statement

A smart mirror to help with everyday fashion choices: to suggest matching outfits, outfits based on weather prediction, styling everyday attires, and more! We plan to attach a camera and a Raspberry Pi to mirrors, along with a microphone and a speaker, to enable interactivity. We plan to use Large Language Models (LLMs) to suggest fashion choices. If provided by the user, the closet inventory will be added to improve the feasibility of suggestions. We aim to keep the interaction as local as possible to protect the user's privacy. On a lighter note, we would like our users to follow the foot steps of evil queen [1].

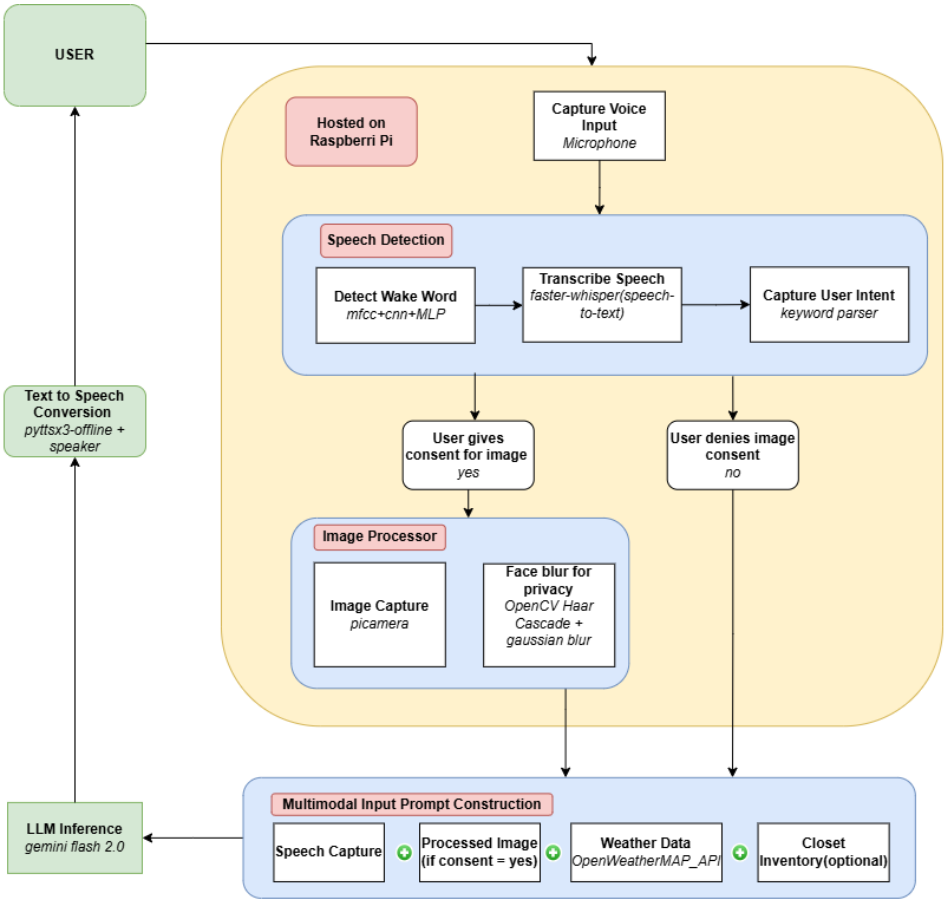

Figure 1: Architecture of the System

2 Prototype: Fashion Frame 1.0 (Code)

2.1 Initialization and Data Collection

During setup, the user is prompted with an initialization stage. Here, the user is asked to repeat the word "yes" and "no" a few times to fine tune the keyword detection model [2]. The user is also requested for the location (city) and an inventory of the wardrobe.

2.2 Voice Activation and Speech-to-Text

The system is activated using a lightweight, offline keyword spotting mechanism powered by the faster whisper model [3, 4]. We record 2 second audio segments using the system microphone and transcribe them using Whisper's quantized variant (int8) optimized for edge devices [5]. If the transcription contains predefined keywords, the system switches to active listening mode. A longer, 8 second recording is then captured to extract user intent in detail.

2.3 Image Capture and Face Blur

Once activated, the system requests the user's permission to capture an image. Upon consent, the Raspberry Pi camera captures the user's image. During our tests, we use our desktop webcam for initial testing. To preserve privacy, the OpenCV Haar cascade classifier [6] is used to detect facial regions, which are then blurred using Gaussian blur. The resulting anonymized image is saved locally for further processing.

2.4 LLM Based Outfit Recommendation (Gemini Flash 2.0)

To increase the relevance of output suggestions, the system uses the user's mentioned location (if specified 2.1). Subsequently, it queries the OpenWeatherMap API [7] using this location to fetch real time weather conditions. We then have a predefined prompt framework that contains four primary characteristics: user speech, real time weather data, blurred user image, and wardrobe inventory (see section 2.1). This prompt is sent to the Gemini Flash 2.0 model [8] using the 'google.generativeai' client [9]. The model, capable of multimodal reasoning, interprets the image and context to generate a tailored outfit recommendation. This interaction is handled securely using API keys.

2.5 Text-to-Speech Feedback

The generated response is converted into speech using 'pyttsx3' [10], an offline text-to-speech engine ensuring responsiveness. This feedback is delivered to the user in a natural voice, completing the conversational loop.

2.6 Edge Aware Design

All of audio recording, keyword spotting, image capture, face detection, and text-to-speech are planned to be executed locally on the Raspberry Pi to minimize latency and protect privacy. Only the blurred image and text prompt are sent to Gemini Flash for outfit generation, making the system suitable for smart home deployment in resource constrained environments.

3 Hardware Configuration and Deployment

Figure 1 presents a detailed illustration of the system architecture. Figures 2 and 3 display the physical device as showcased in the demonstration. The overall extension of components is minimal, with the hardware elements discreetly positioned behind the mirror surface. The Raspberry Pi is connected to the internet via a LAN cable and powered through a standard electrical socket. A Pi camera and microphone are interfaced with the Raspberry Pi and extended outward to face the user. The components are secured in place using adhesive tape.

4 Results (Test Example)

The initialization and video demos are available here. The pipeline successfully demonstrated the following:

- **Offline speech-to-text conversion:** The Whisper model transcribed the user's spoken input accurately, and the system correctly detected activation keywords.
- **Input information:** The image was captured, and OpenCV successfully applied Gaussian blur to all detected facial regions. The system parsed the location taking during initialization (2.1) accurately and retrieved real time weather information from the OpenWeatherMap API.
- **Multimodal LLM inference:** The blurred image, combined with the user's voice command and weather context, was sent to the Gemini Flash 2.0 API. The model returned a relevant and descriptive outfit recommendation.

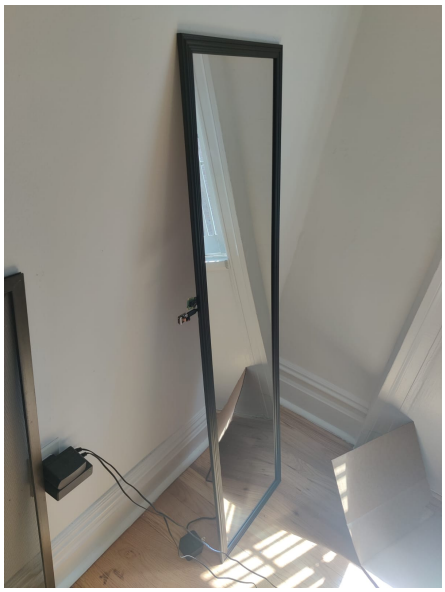

Figure 2: Fashion Frame!

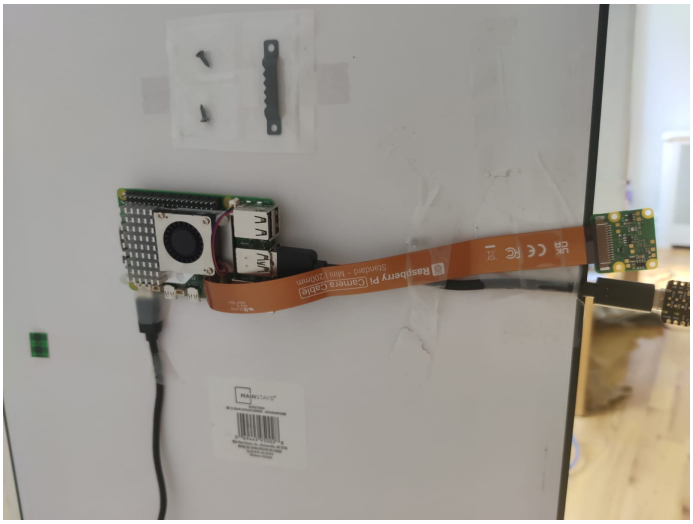

Figure 3: The attachment of the Pi camera and speaker is shown here. The Raspberry Pi is taped to the back of the mirror.

5 Future Works: Building Fashion Frame 2.0

We plan to upgrade to a more human sounding text-to-speech system, like Alexa or Siri. This would give the user a better feel for during the use cases. We would also enable a conversationally interactive system. Here, the user would be able to ask the Mirror a variety of questions, not restricting themselves to the fashion related questions. We would also like to create a feature wherein the mirror accesses the wardrobe inventory dynamically. This would let the model understand what clothing options are presently available to the user, giving it more context.

It would be amazing to connect Fashion Frame to an existing smart home system. This would allow for the mirror to be connected to a centralized environment, enabling access to smart devices, mobile phones, computers, watches, and more.

References

[1] *Snow White and the Seven Dwarfs*. 1937.

[2] Waasi Jagirdar Aditya Somasundaram Likhith Ayinala. *embedded-ai-project*. https://github.com/likhithayinala/embedded-ai-project. Accessed: 2025-04-21.

[3] Guillaume Klein. *Faster-Whisper: Fast and accurate automatic speech recognition*. https://github.com/guillaumekln/faster-whisper. Accessed: YYYY-MM-DD. 2023.

[4] Alec Radford et al. *Whisper: Robust Speech Recognition via Large-Scale Weak Supervision*. https://github.com/openai/whisper. Accessed: YYYY-MM-DD. 2022.

[5] Muhammad Yasir Shabir, Gianluca Torta, and Ferruccio Damiani. "Edge ai on constrained iot devices: Quantization strategies for model optimization". In: *Intelligent Systems Conference*. Springer. 2024, pp. 556–574.

[6] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. IEEE. 2001, pp. I–I.

[7] OpenWeather. *OpenWeatherMap API*. Accessed: 2025-04-21. 2024. URL: https://openweathermap.org/api.

[8] Rohan Anil et al. *Gemini: A Family of Highly Capable Multimodal Models*. Accessed: 2025-04-21. 2023. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[9] Google LLC. *google-generativeai: Google Generative AI Python SDK*. Version 0.8.5. 2025. URL: https://pypi.org/project/google-generativeai/.

[10] Parente Bate and contributors. *pyttsx3 on GitHub*. Accessed: 2025-04-21. 2023. URL: https://github.com/nateshmbhat/pyttsx3.