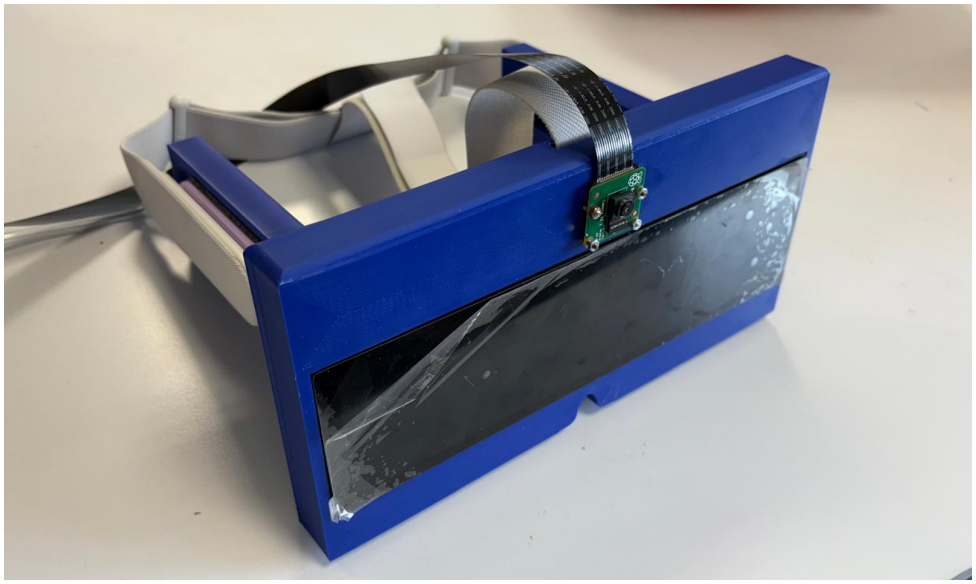
SORBONNE UNIVERSITÉ

Smart Glasses

Authors :
Sarah El Zeghendy
Nacim Talaoubrid
Evanthia Virginia Anastasopoulou
Mohamed El Oualid Boudemagh
Romane Couedel

Master 2 : MSR 2025-2026



SORBONNE UNIVERSITÉ

Table des matières

1 Introduction

1.1 Project Context

With the rise of wearable technology, smart glasses are undergoing significant development. Companies such as Meta, with their Ray-Ban Meta glasses, are now offering devices capable of capturing images and video, interacting via voice commands, and accessing AI-based assistants.

At the same time, within the field of assistive technology, several research projects and commercial products aim to support visually impaired or blind individuals specifically through obstacle detection, scene description, and the reading of visual information. Furthermore, recent breakthroughs in artificial intelligence, particularly in natural language processing (NLP) and computer vision, are opening up new possibilities to enhance visual assistance tools.

1.2 Objectives

In this context, our project follows a similar path, but with a specific focus on social interaction and assistance. The objective is to design smart glasses capable of detecting both the emotions of the person facing the user and any obstacles within the environment.

However, to ensure reliable operation and straightforward interaction, these two features do not run simultaneously. The system is organized into different operating modes, accessible through an interface of physical buttons integrated into the glasses. This allows the user to switch between "obstacle detection" and "emotion detection" modes, depending on their immediate needs.

This approach simplifies data processing, improves the clarity of haptic feedback, and provides the user with greater control over the device's behavior.

2 Materials Used

Component	**Number**	**Reference / Model**	**Description / Role**
Camera	1	Camera Raspberry module 2	Sony IMX219 8-megapixel sensor
Raspberry Pi	1	Raspberry Pi 3B+	Used to process image data and send it to the computer
Arduino	1	Arduino Uno	Used to control the haptic feedback system (ie pneumatic actuators...)
Differential Pressure sensor	2	DFRobot SEN0343	Used to detect pressure into balloons
Motor driver	3	L298N	Used to control 2 motors per driver
Valves	2	DFR0866	Used to switch between inflate and deflate

Component	Number	Reference / Model	Description / Role
Pumps	4	DFRobot 370 Mini Vacuum Pump	2 used to inflate and 2 to deflate balloons
Balloons	2	Standard party balloons	Used to do haptic feedback
Push buttons	2	Standard push buttons	Used to switch between modes
Computer	1	PC linux	Used to run the AI models for emotion and obstacle detection
Battery 12V	1	lead acid battery 12V	Used to power the pumps
Battery 5V and <2.5A	2	Standard Power bank 5V	Used to power the Raspberry Pi and Arduino

TABLE 1: Main hardware components used in the project

3 Implementation

3.1 Overall Architecture

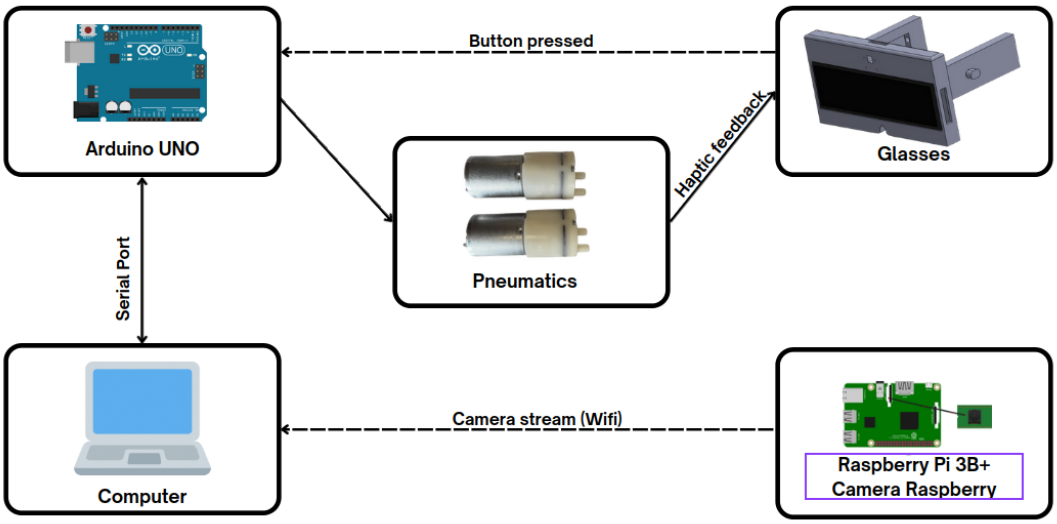


FIGURE 1 – Overall system architecture of the smart glasses. The system comprises four main components : (1) the user wearing the glasses with integrated balloons controlled by a pneumatics system (2) the Arduino microcontroller managing the pneumatic feedback system, (3) the Raspberry Pi capturing and streaming video frames via Wi-Fi, and (4) the host computer running AI models for emotion and obstacle detection. Communication flows bidirectionally : the camera stream is transmitted from the Raspberry Pi to the computer, while control signals and pressure commands are exchanged between the computer and Arduino via serial communication. User mode selection is handled through physical buttons connected to the Arduino.

3.2 Camera and Raspberry pi

Initially, we attempted to implement a stereo camera system using an Arducam module to obtain depth information directly from two synchronized camera feeds. However, the image quality was insufficient for reliable image processing, which led to poor performance in depth estimation. Moreover, the Raspberry Pi 4B frequently crashed when using the Arducam module, likely due to insufficient processing power and high resource demands of the stereo setup. This instability made the configuration unreliable for real-time operation.

As a result, we switched to using a single camera, combined with AI processing on the computer block to infer depth information from the captured images. This approach allowed us to maintain accurate obstacle and emotion detection while simplifying the hardware setup.

For transmitting the camera stream, we initially experimented with USB gadget mode on a Raspberry Pi 4B. However, this setup proved unstable and presented power supply challenges, as the USB port could not simultaneously provide enough power and transmit data reliably.

To overcome these issues, we opted for a Raspberry Pi 3B+ equipped with Wi-Fi for streaming the camera feed to the computer. Although this solution introduces a small amount of latency, it provides a stable and functional system for real-time image analysis.

All the configuration steps and installations required to use the Raspberry Pi 3B+ with Wi-Fi for the system are detailed in the file `procedure_camera.md`.

3.3 Pneumatic/haptic system

The haptic feedback mechanism employed in this project draws inspiration from the OpenPneu platform developed by Tian et al. [1], which presents a compact architecture for multi-channel pneumatic actuation. This approach was selected for its ability to provide precise, controllable haptic sensations through pressure modulation in soft actuators.

System Architecture

The pneumatic system comprises four principal components per haptic channel : (i) an inflatable balloon serving as the soft actuator, (ii) a differential pressure sensor (DFRobot SEN0343) for real-time pressure monitoring, (iii) a 3/2-way solenoid valve (DFR0866) for airflow direction control, and (iv) a dual-pump configuration consisting of one inflation pump and one deflation pump (DFRobot 370 Mini Vacuum Pump). The complete system architecture is illustrated in Figure 2.

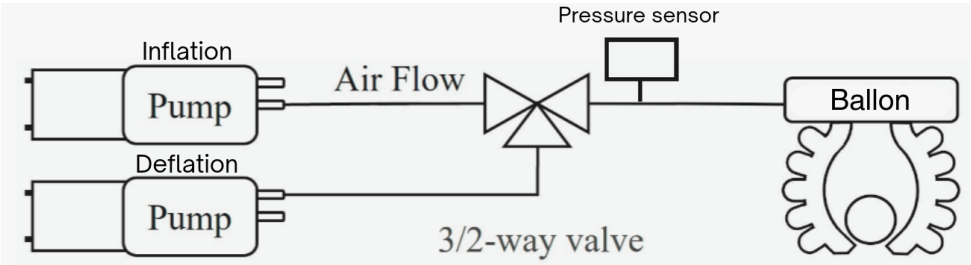


FIGURE 2 – Schematic diagram of the pneumatic actuation system for a single haptic channel, adapted from OpenPneu platform [1]. The system integrates a 3/2-way valve, differential pressure sensor, inflation and deflation pumps, and soft balloon actuator.

Valve Configuration and Operation

The 3/2-way solenoid valve features three ports and two stable positions, enabling bidirectional airflow control. In its default state, one port remains permanently open while the valve alternately connects either the inflation or deflation pathway. When the valve switches position, the previously open port closes while the opposite port opens, thereby reversing the airflow direction. This configuration allows for independent control of inflation and deflation processes without requiring additional valves.

Pressure Control Algorithm

The system implements a closed-loop pressure control scheme based on proportional feedback. The pressure setpoint P_{target} is continuously compared with the measured pressure P_{measured} from the differential sensor to compute the control error :

$$e(t) = P_{\text{target}} - P_{\text{measured}} \tag{1}$$

The control logic operates as follows :

— If $e(t) > 0$ (positive error), the valve switches to the inflation configuration, activating the inflation pump until the measured pressure reaches the target value.

— If $e(t) < 0$ (negative error), the valve switches to the deflation configuration, activating the deflation pump to reduce pressure until equilibrium is achieved.

— When $|e(t)|$ falls below a predefined threshold, both pumps are deactivated to prevent oscillations and reduce power consumption.

This feedback control approach ensures rapid and precise pressure regulation, enabling the generation of distinct haptic patterns for different feedback modalities (obstacle detection and emotion recognition). The dual-pump architecture permits simultaneous inflation and deflation capabilities, significantly improving response time compared to passive deflation systems.

Dual-Channel Implementation

To provide bilateral haptic feedback corresponding to the left and right sides of the glasses frame, the entire pneumatic system described above is duplicated, resulting in two independent actuation channels. Each channel operates autonomously with its own balloon actuator, pressure sensor, valve, and pump pair. The Arduino microcontroller manages both channels simultaneously, receiving target pressure commands from the host computer via serial communication.

I2C Address Conflict and Sensor Simulation

During implementation, a significant hardware limitation was encountered related to the pressure sensors. Both DFRobot SEN0343 differential pressure sensors communicate via the I2C protocol and are factory-configured with identical, non-configurable I2C addresses. Since the Arduino Uno possesses only a single I2C bus, it is unable to distinguish between the two sensors, preventing simultaneous operation of both channels with hardware pressure feedback.

To circumvent this constraint, a software-based sensor simulation approach was developed. The functional pressure sensor was first used to empirically characterize the system's

pressure dynamics under three operational conditions : (i) active inflation, (ii) active deflation, and (iii) idle state (no pump activity). Through systematic experimentation, the pressure variation rates were quantified for each condition.

Based on these empirical measurements, a virtual pressure variable `pressure_simulated` was implemented in the Arduino firmware. This variable is incremented or decremented at each control cycle according to the current valve state and pump activity, effectively modeling the expected pressure evolution. The control algorithm then operates using this simulated pressure value rather than direct sensor readings.

While this approach successfully enabled dual-channel operation, it presents inherent limitations. The simulated sensor cannot detect external disturbances such as physical contact with the balloon, which would cause unexpected pressure variations. A genuine pressure sensor would immediately detect such anomalies and adjust the control accordingly, whereas the simulation continues to assume nominal behavior. Despite this limitation, the system performance proved satisfactory for the intended application, as the glasses are designed to be worn in a relatively stable configuration where external interference with the actuators is minimal.

3.4 AI Models and Processing

The algorithms described below are not executed manually from a terminal, but are launched through the `launcher.py` script, which manages communication with the Arduino and handles mode switching. To check the required installations needed to run the scripts, refer to the `requirements.txt` file.

Emotion Detection

The emotion detection module uses the **FER** (Facial Expression Recognition) library, which is based on deep neural networks. FER detects seven facial emotions : happy, surprise, sad, angry, fear, disgust, and neutral.
To simplify the haptic feedback system and reduce computational load, the seven emotions are grouped into three main categories :

— **Happy** : happy, surprise
— **Sad** : sad, fear, disgust
— **Neutral** : angry, neutral

A confidence threshold of 0.35 is applied to ignore low-confidence detections. Each reduced emotion is associated with a confidence score computed from the FER probabilities.

To allow real-time processing, several optimizations are applied :

— **Frame skipping** : only one frame out of three is processed (FRAME_SKIP = 3)
— **Resizing** : images are scaled to 60% of their original size
— **Centered detection** : only the face closest to the center of the image is processed

Each reduced emotion triggers a specific haptic pattern using the pneumatic balloons :

— **Happy** : right balloon inflates and deflates, left balloon inactive
— **Sad** : left balloon inflates and deflates, right balloon inactive
— **Neutral** : both balloons remain inactive

The Arduino receives the commands through a serial connection and controls the pneumatic system.

The program receives JPEG-compressed video frames through a UDP socket from the Raspberry Pi. Each frame is decoded and processed by FER at regular intervals. The corresponding haptic commands are then sent to the Arduino. A live display shows the detected face and the recognized emotion.

The emotion is detected only once in order to avoid overstimulating the user.

Obstacle Detection

The obstacle detection module is based on the **YOLO** (You Only Look Once) object detection model combined with the **Depth Anything V2** network for depth estimation. YOLO is used to detect relevant objects in the scene, while Depth Anything provides a depth map that allows estimating the relative distance of each detected object.

Only a subset of COCO classes is considered, corresponding to potential obstacles for the user (persons, chairs, beds, couches, benches, handbags, suitcases).

The processing pipeline is structured as follows :

— **Frame acquisition** : The program receives JPEG-compressed video frames via a UDP socket from the Raspberry Pi.

— **Object detection** : YOLO detects objects and provides bounding boxes for localization. To improve spatial accuracy, we employ the YOLOv11-seg variant, which extends the base YOLO architecture with instance segmentation capabilities. Unlike standard object detection that only provides rectangular bounding boxes, YOLO-seg generates pixel-precise segmentation masks for each detected object (see Figure 3). These masks identify all pixels belonging to the object, providing a significantly more accurate representation of the object's spatial extent compared to the entire bounding box region.

— **Depth estimation** : Depth Anything computes a dense depth map from the RGB image (Figure 4).

— **Depth aggregation** : For each detected object, the average depth is computed by combining information from the depth map and the segmentation mask. The depth map provides a depth value for every pixel in the image, while the segmentation mask identifies which pixels belong to the detected object. By indexing the depth map with the mask coordinates, we extract the depth values corresponding exclusively to pixels within the object. These depth values are then averaged to obtain a single representative depth estimate for the object.

— **Region of Interest (ROI)** : Only objects located inside a trapezoidal region in front of the user are considered (illustrated in Figure 5). This trapezoidal shape is specifically designed to respect the natural perspective of human vision : wide at the bottom (near field) and narrow at the top (far field). The geometry of this ROI closely approximates the user's walking path and immediate surroundings, focusing detection efforts on the zone where obstacles pose an actual risk during navigation. Objects detected outside this region, even if visible in the camera frame, are filtered out as they are sufficiently distant or laterally displaced to be irrelevant for immediate obstacle avoidance. This spatial filtering reduces computational load and prevents unnecessary haptic alerts for obstacles that the user is unlikely to encounter.

— **Temporal filtering** : An object must be detected continuously for at least one second to be considered stable.

— **Intensity and side selection** : Only one object per side (left and right) is kept, corresponding to the closest obstacle on each respective side. When obstacles are

detected on both sides, the haptic feedback intensity is computed using a proportional relationship based on relative distances. The closest object, regardless of which side it appears on, is always assigned a maximum haptic intensity of 100%. The intensity for the more distant object is then scaled proportionally according to the depth difference between the two obstacles.

FIGURE 3 – Example of YOLOv11-seg object detection. The image shows detected obstacles with bounding boxes (rectangles) and pixel-precise segmentation masks (colored overlays).

FIGURE 4 – Dense depth map generated by Depth Anything V2 from an RGB input frame. Warmer regions correspond to closer objects, while colder regions indicate more distant areas.

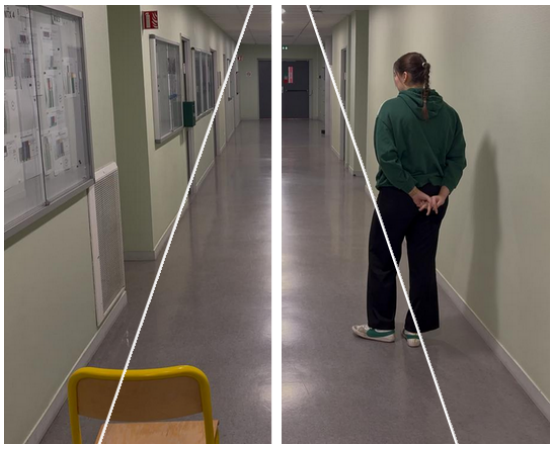


FIGURE 5 – Trapezoidal Region of Interest (ROI) overlaid on the camera view. The trapezoid is wide at the bottom (near field) and narrow at the top (far field), respecting natural perspective. Only obstacles detected within this region trigger haptic feedback, focusing attention on the user's immediate walking path.

The resulting haptic commands are sent to the Arduino, which controls the pneumatic system to provide feedback to the user. The left and right balloons correspond to obstacles detected on the respective sides of the user.

3.5 Communication Arduino - Computer

Communication between the Arduino and the computer is carried out via a serial port (USB). The Arduino sends mode commands to the PC using the format `MODE:X`, where X represents the mode selected by the user through the push buttons.

Computer

A launcher program (`launcher.py`) running on the computer continuously listens to the serial port. Upon receiving a mode command, it manages the start and stop of the corresponding Python processes :

- **Mode 1** : Launch of the emotion detection module (`emotion_detection2_pneu.py`)
- **Mode 2** : Launch of the obstacle detection module using YOLO (`objet.py`)
- **Mode 3** : Stop of all active scripts (neutral mode)

This orchestration system ensures that only one AI module runs at a time, optimizing the use of computational resources and simplifying the delivery of haptic feedback to the user. In the event of an interruption (mode change), the launcher cleanly stops the current process before starting a new one.

Arduino

The Arduino Uno acts as the embedded real-time controller responsible for button handling, serial communication, and pneumatic actuation. During operation, it continuously reads the two push buttons to detect mode transitions and sends the corresponding `MODE:X` command to the computer via serial. Depending on the active mode, the Arduino routes incoming serial commands to the appropriate control loop : in emotion mode, it executes open-loop inflation and deflation sequences of a fixed duration, while

in obstacle detection mode, it runs the closed-loop pressure control algorithm described in Section 3.3 on both left and right channels independently.

3.6 User Interface and CAD Design

The mechanical design of the smart glasses was developed with a strong focus on ergonomics, stability, and integration of electronic and pneumatic components. The entire structure was modeled using **SolidWorks**, allowing precise placement of each element and iterative improvements before manufacturing.

Mechanical Design and 3D Printing

The glasses frame was fully designed in SolidWorks and subsequently manufactured using **3D printing**. The chosen material was **PLA**, due to its ease of printing, sufficient mechanical rigidity for prototyping, and lightweight properties, which are essential for wearable devices.
The front part of the glasses integrates both the display and the camera module. The display is positioned in front of the user's field of view and fixed to the structure using **M2.5 screws (20 mm)** to ensure mechanical stability. The camera module is mounted securely using **M2 screws (16 mm)**, allowing accurate alignment with the user's visual axis.

Figure 6 shows the front view of the glasses, highlighting the positioning of the display and the camera.

FIGURE 6 – Front view of the smart glasses showing the display and camera mounting

Integration of the Pneumatic Haptic System

Inside each branch of the glasses, a rectangular cavity was designed to accommodate the inflatable balloon used for haptic feedback. This cavity allows the balloon to expand and deflate while remaining protected within the structure.
An additional hole was integrated inside this rectangular cavity to allow the pneumatic tubes to exit toward the exterior side of the glasses. This design ensures clean cable and

tube routing, reduces mechanical stress on the tubing, and maintains user comfort.

Figure 7 illustrates the internal structure of the glasses branch, including the balloon housing and tube routing.

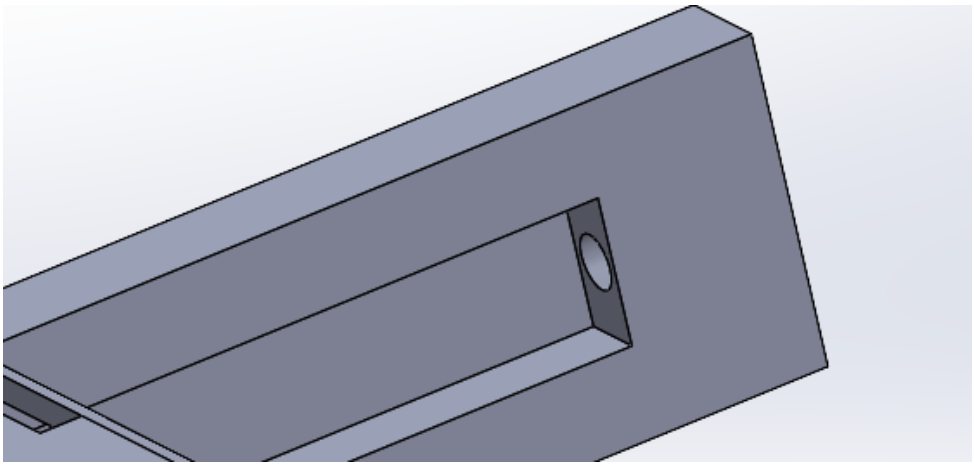


FIGURE 7 – Internal view of the glasses branch showing the balloon cavity and tube exit

Stability and Head Strap Adapter

To improve stability during use and ensure proper positioning of the glasses on the user's head, a head strap adapter was designed and integrated into the frame. This system is inspired by commercial smart glasses such as the Meta smart glasses.
The head strap significantly enhances comfort and stability, especially during movement, by distributing the weight of the device and preventing unwanted slipping. This design choice is particularly important given the additional weight introduced by the camera, display, and pneumatic system.

Figure 8 presents the head strap adapter mounted on the glasses.

FIGURE 8 – Head strap adapter designed to improve stability and comfort

User Comfort Considerations

Special attention was given to user comfort during prolonged use. A **nose cut-out** was designed in the front part of the glasses This opening reduces pressure on the nose bridge and improves overall wearability.
By combining lightweight materials, ergonomic design, and stabilization accessories, the final mechanical design achieves a balance between functionality, comfort, and robustness, making the smart glasses suitable for real-world usage scenarios.

4 Fonctionnalities and User Interaction

This section describes the functionalities offered by the smart glasses from the user's perspective, as well as the interaction principles used to control the system. The device is designed to be simple to use, relying on a limited number of physical inputs and clear haptic feedback to avoid cognitive overload.

4.1 Operating Modes

The smart glasses operate according to three distinct modes :
— **Emotion Detection Mode**
— **Obstacle Detection Mode**
— **Neutral Mode**

To ensure clarity and reliability, only one functional mode can be active at a time. This design choice avoids ambiguity in the haptic feedback and reduces computational load.

4.2 Mode Selection Interface

The user interacts with the system through two physical push buttons integrated into the glasses frame. Each button is associated with a specific operating mode :

— **Button 1** : Emotion Detection Mode
— **Button 2** : Obstacle Detection Mode

When no button is pressed, the system automatically enters the **neutral mode**, in which no detection algorithm is running and no haptic feedback is generated. This mode acts as a safe idle state.

Pressing a button immediately switches the system to the corresponding mode and launches the associated processing pipeline. Releasing all buttons stops any active functionality and returns the system to the neutral mode.

4.3 Emotion Detection Functionality

When the user presses **Button 1**, the glasses activate the emotion detection mode. In this mode, the system analyzes the facial expressions of the person located in front of the user using a computer vision and deep learning pipeline.

Once an emotion is detected with sufficient confidence, the information is conveyed to the user through haptic feedback generated by the pneumatic balloons integrated into the glasses :

— Inflation of the **right balloon** indicates a positive emotion (e.g., happiness)
— Inflation of the **left balloon** indicates a negative emotion (e.g., sadness)
— No balloon activation corresponds to a neutral emotional state

To prevent overstimulation and ensure user comfort, the emotion is detected only once per interaction. The system remains inactive until the user explicitly switches mode or returns to neutral mode.

4.4 Obstacle Detection Functionality

When the user presses **Button 2**, the glasses enter obstacle detection mode. In this configuration, the system continuously analyzes the scene in front of the user to identify potential obstacles along the walking path.

Detected obstacles are translated into haptic feedback according to their relative position and distance :

— Obstacles detected on the **left side** activate the left balloon
— Obstacles detected on the **right side** activate the right balloon
— The **intensity** of the balloon inflation is proportional to the proximity of the obstacle

This intuitive mapping allows the user to quickly infer both the location and urgency of obstacles without relying on auditory cues, preserving environmental awareness.

4.5 Neutral Mode and Safety

The neutral mode is the default and fallback state of the system. It is activated automatically when no button is pressed or when the user intentionally releases all buttons. In this mode :

— All AI processing is stopped
— No haptic feedback is generated

— Power consumption is minimized

This behavior ensures safe operation, avoids unintended feedback, and gives full control to the user over when assistance is needed.

5 Initial Objectives and Achieved Outcomes

5.1 Initial Project Objectives

At the beginning of the project, several technical and functional objectives were defined in order to explore multiple sensing, interaction, and feedback modalities for assistive smart glasses. The initial objectives were the following :

— **Emotion detection** : Detect the emotional state of a person facing the user using computer vision and artificial intelligence techniques, and convey this information through intuitive feedback.

— **Obstacle detection** : Identify obstacles in the user's environment to assist navigation and improve safety during movement.

— **Depth perception using a stereo camera** : Implement a stereo vision system to estimate obstacle distances directly from synchronized dual-camera images.

— **Gaze tracking via animated eyes on a display** : Represent gaze direction using animated eyes displayed on an integrated screen to enhance social interaction cues.

— **Full on-board processing on a Raspberry Pi** : Run all perception, decision-making, and control algorithms directly on an embedded platform (Raspberry Pi), without relying on an external computer.

— **Integration of a pneumatic haptic feedback system** : Design and integrate a multi-channel pneumatic system capable of providing localized and intensity-modulated haptic feedback through inflatable actuators embedded in the glasses.

These objectives aimed to create a fully autonomous, wearable assistive device combining social interaction assistance, navigation support, and innovative haptic feedback mechanisms.

5.2 Achieved Objectives

Despite several hardware, computational, and integration constraints, a substantial portion of the initial objectives was successfully achieved :

— **Emotion detection** : A real-time emotion recognition system was implemented using deep learning models. Detected emotions are translated into clear and non-intrusive pneumatic haptic feedback patterns.

— **Obstacle detection with depth estimation** : A robust obstacle detection pipeline was developed by combining object detection with monocular depth estimation. This approach provides reliable relative distance information and effectively replaces the initially planned stereo camera system.

— **Full integration of a pneumatic haptic system** : A complete pneumatic feedback system was successfully designed, implemented, and integrated into the smart glasses. The system includes inflatable actuators, valves, pumps, pressure sensing, and control logic, enabling precise and bidirectional haptic feedback for both emotion and obstacle detection modes.

All achieved functionalities were integrated into a coherent end-to-end system, covering real-time image acquisition, AI-based perception, embedded control, and wearable haptic

feedback.

5.3 Unachieved Objectives and System Limitations

Some initial objectives could not be fully realized due to hardware limitations and computational constraints :

- **Gaze tracking display** : The gaze-following feature based on animated eyes displayed on an integrated screen could not be implemented, as the display hardware did not function correctly during the integration phase. Consequently, this visual interaction modality was abandoned.
- **Stereo camera system** : The stereo vision approach was not retained due to insufficient image quality and computational instability on the embedded platform, leading to unreliable real-time performance.
- **Fully embedded AI processing** : Running all AI algorithms (emotion recognition, object detection, and depth estimation) directly on the available Raspberry Pi board was not feasible due to limited computational resources. The processing load exceeded the capabilities of the embedded platform, resulting in unstable execution and unacceptable latency. To ensure reliable real-time operation, the system architecture was adapted by offloading intensive AI computations to an external computer while retaining the Raspberry Pi for image acquisition and data transmission.

Although not all planned objectives were achieved, the final system demonstrates a realistic and well-justified architectural trade-off. The successful integration of emotion recognition, obstacle-aware navigation, and a complete pneumatic haptic feedback system highlights the technical feasibility and relevance of the proposed smart glasses for assistive applications.

6 Conclusion

This project successfully demonstrates the feasibility of integrating emotion detection, obstacle awareness, and pneumatic haptic feedback into smart glasses. Despite the inherent constraints of a prototype development-limited resources, hardware reliability challenges, and computational trade-offs-the system achieves its core objectives through pragmatic engineering decisions.

However, as a proof-of-concept prototype, the system relies on external components-an external computer for AI processing and a lead-acid battery for power-which currently prevent full wearability. These dependencies reflect the reality of prototype development and highlight areas for future miniaturization and optimization. A production version would require embedded processors powerful enough to handle AI computations on-board and lightweight, integrated power solutions.

The integration of multiple technologies (computer vision, deep learning, embedded systems, and pneumatic actuation) demonstrates the technical feasibility of the proposed approach. While not all initially planned features were realized, the completed system provides a solid foundation for future iterations.

Our git : `github.com/romanecouedel/projetmsr_smart_glasses`

SORBONNE UNIVERSITÉ

Bibliographie

[1] Y. Tian, R. Su, X. Wang, N. B. Altin, G. Fang, and C. C. L. Wang, "OpenPneu : Compact Platform for Pneumatic Actuation with Multi-Channels," [Conference/Journal details], 2024.