

# Configuration Manual

MSc Research Project  
Artificial Intelligence for Business

Stephanie Mae Armitstead  
Student ID: 23346230

School of Computing  
National College of Ireland

Supervisor:     Anderson Simiscuka

**National College of Ireland**  
**MSc Project Submission Sheet**  
**School of Computing**



**Student Name:** Stephanie Mae Armitstead  
.....  
23346230  
**Student ID:** .....  
MSc in AI for Business ..... 2025  
**Programme:** ..... **Year:** .....  
Practicum 2  
**Module:** .....  
Anderson Simiscuka  
**Lecturer:** .....  
**Submission Due Date:** 15/09/2025  
.....  
Designing a Multimodal AI Framework for Real-Time Behavioral and  
**Project Title:** Verbal Analysis of Virtual Client Interactions.....  
**Word Count:** 669..... **Page Count:** 3.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** ..... *Stephanie Armitstead* .....  
15/09/2025  
**Date:** .....

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

|                                                                                                                                                                                           |                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Attach a completed copy of this sheet to each project (including multiple copies)                                                                                                         | <input type="checkbox"/> |
| <b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).                                                                  | <input type="checkbox"/> |
| <b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | <input type="checkbox"/> |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| <b>Office Use Only</b>           |  |
|----------------------------------|--|
| Signature:                       |  |
| Date:                            |  |
| Penalty Applied (if applicable): |  |

# Configuration Manual

Stephanie Mae Armitstead  
Student ID: 23346230

## 1 System Overview

### 1.1 Purpose of the Artefact

This artefact is a multimodal AI system designed to integrate with video conferencing workflows, capturing both visual and audio cues in real time to deliver actionable sentiment and behavioral insights. It performs:

- Real-time body language and gesture tracking using MediaPipe Pose/Holistic.
- Speech sentiment & emotion detection using fine-tuned transformer-based NLP models.
- Live visualization of sentiment results in the user interface.
- Session data export for post-analysis.

The system is intended for sales and relationship management contexts, where understanding client engagement in virtual meetings can improve outcomes.

### 1.2 Methodology Justification

The design of this artefact draws on a review of state-of-the-art multimodal analysis frameworks and the practical requirements of real-time business communication. MediaPipe Pose and Holistic were selected over alternatives such as OpenPose due to their lightweight performance, CPU compatibility, and built-in landmark normalization, which reduce preprocessing overhead. HuggingFace Transformers were chosen for speech sentiment and emotion analysis because of their pre-trained, domain-adaptable models and straightforward integration via the pipeline API.

Streamlit with streamlit-webrtc was preferred over traditional Flask/Django deployments for its ability to provide a browser-based, interactive dashboard with minimal latency and without complex server setup. This stack balances accuracy, interpretability, and deployment speed, directly aligning with the project's research objectives and constraints.

### 1.3 High-Level Architecture

The system consists of the following components:

1. Video Processing Module – Captures webcam stream, applies pose/gesture detection via MediaPipe.
2. Audio Capture & Sentiment Module – Uses sounddevice for real-time audio stream capture and HuggingFace Transformers pipeline for sentiment/emotion analysis.
3. UI & Streaming Layer – Streamlit front-end with streamlit-webrtc for video/audio streaming, and Plotly for visualizing live sentiment.
4. Data Export Module – Outputs structured CSV logs containing timestamps, detected sentiment, emotion confidence scores, and behavioral metrics.

Data Flow:

Webcam/Audio → Pre-processing → AI Inference (pose, sentiment) → Real-time visualization → Optional CSV export.

## 1.4 System Requirements

Hardware

- CPU: Intel i5/i7 or AMD Ryzen 5/7 (Quad-Core or higher)
- RAM: Minimum 8 GB (16 GB recommended)
- Storage: 2 GB free space
- Webcam and microphone (USB or integrated)
- GPU: Optional, for faster inference (NVIDIA CUDA-supported)

Software

- OS: Windows 10/11 (tested), macOS, or Linux
- Python: 3.9+
- Package Manager: Conda or pip
- Web Browser: Chrome/Edge/Firefox (latest versions)
- IDE/Text Editor: Visual Studio Code or equivalent

Python Dependencies

Key libraries: opencv-python, mediapipe, streamlit, streamlit-webrtc, plotly, sounddevice, transformers, numpy, pandas, av

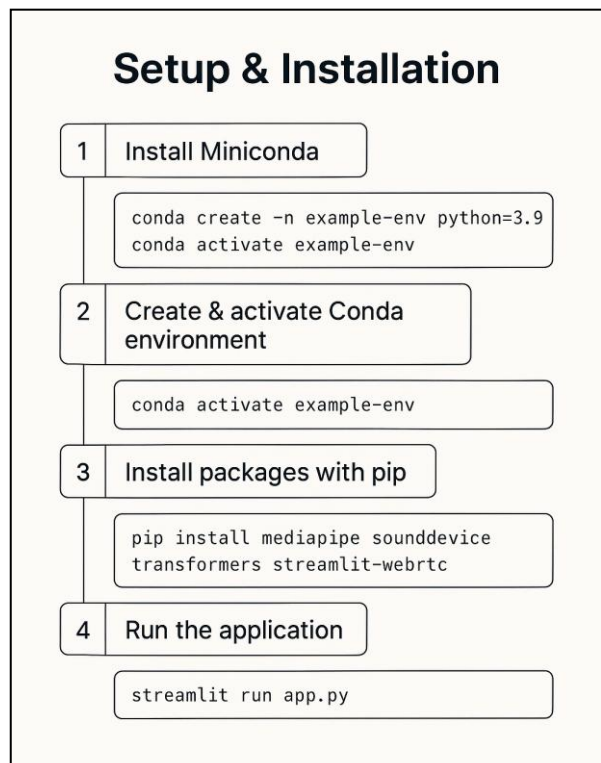
## 2 Installation & Configuration

### 2.1 Prerequisites

- Install Miniconda or Anaconda from <https://docs.conda.io/en/latest/miniconda.html>.
- Install Visual Studio Code from <https://code.visualstudio.com/>.

### 2.2 Setup Steps

1. Create Conda Environment:
  - a. `conda create -n multimodal_ai python=3.9`
  - b. `conda activate multimodal_ai`
2. Install Required Packages
  - a. `pip install opencv-python mediapipe streamlit streamlit-webrtc plotly sounddevice transformers numpy pandas av`
3. Clone or Download Repository
  - a. `git clone https://github.com/[your-repo-name].git`
  - b. `cd [your-repo-name]`
4. Environment Variables
  - a. Create an .env file in the project root if API keys or credentials are needed.
  - b. Example:  
`API_KEY=your_api_key_here`



## 3 Running the System

### 3.1 Launching the Application

From the terminal inside your Conda environment: `streamlit run app.py`

- Default browser will open at `http://localhost:8501`.
- Grant permission for camera and microphone when prompted.

### 3.2 Using the Interface

- Mode Selector: Choose between *Pose Detection* or *Holistic Detection*.
- Start Video/Audio: Click Start to begin analysis.
- Live Sentiment Label: Sentiment/emotion label will display in real time above the video feed.
- Stop Session: Click Stop to end capture and save logs.

### 3.3 Output

Real-time Display:

- Video feed with overlay of detected pose landmarks.
- Live sentiment classification label with confidence.

Exported Data:

- CSV file saved in `/exports` folder with:
  - Timestamp
  - Sentiment label
  - Emotion confidence score
  - Pose/gesture metadata

### 3.4 Testing

Run system using a short, pre-recorded test clip (optional).

Verify:

- Video feed is displayed with pose landmarks.
- Sentiment label updates in real time.
- CSV output is generated and contains valid timestamps and predictions.

### 3.5 Troubleshooting

| Issue                 | Possible Cause      | Solution                                                 |
|-----------------------|---------------------|----------------------------------------------------------|
| Webcam not found      | Permission blocked  | Enable camera/mic in browser                             |
| Audio not captured    | Wrong audio device  | Check system sound settings                              |
| High CPU usage        | Running without GPU | Reduce resolution in app.py settings                     |
| Streamlit not loading | Port conflict       | Run <code>streamlit run app.py --server.port 8502</code> |

## References

- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Yong, M., Lee, J., & Chang, C. (2019). MediaPipe: A framework for building perception pipelines. *Google Research*. Retrieved from <https://github.com/google/mediapipe>
- McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51–56).
- NumPy Developers. (2023). NumPy: Fundamental package for array computing with Python. *NumPy Documentation*. Retrieved from <https://numpy.org>
- Plotly Technologies Inc. (2015). Collaborative data science. *Plotly Documentation*. Montréal, QC: Plotly Technologies Inc. Retrieved from <https://plotly.com/python/>
- Python Software Foundation. (2023). Python Language Reference, version 3.9. Retrieved from <https://www.python.org>
- Streamlit Inc. (2023). Streamlit: The fastest way to build data apps. *Streamlit Documentation*. Retrieved from <https://streamlit.io>
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38–45). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-demos.6>