

# Design Documentation

---

After uploading the software to, e.g., a Raspberry Pi and placing its camera next to the desired TV area, it starts capturing real-time images from the panelists' living area according to specified intervals. The faces of the panelists are then extracted from each frame, as well as an estimate of their ages, genders, and emotions. The software can also recognize which panelists have been detected before by using a database that automatically stores their features. This way, the software can track the panelists' watching behavior and save all their information in log files, which can then be used to calculate useful statistics and gain insights for the television audience measurement.

The software assures users' privacy and only sends the required anonymized information to the server. Furthermore, the software comes with a UI that allows users to enter their names and update other information.

The detection and recognition of the faces and their features is done through pre-trained machine learning models, which can easily be replaced in the code by new or self-implemented ones. The performance of these models can be tested through the benchmark testing code.

The attentiveness of a panelist had to be estimated using a rule-based approach that depends on the status of the eyes: (semi-)closed or open. A better approach can be implemented (ML-based or eye tracking) but this was out of the scope of this project.

The code is located in the `Implementation` directory. The Automatic Panelist Detection Software can be started using the `main.py` file. In this file, both the graphical user interface and the API are started. Files that belong to the API are located in the `api` directory. The GUI is created

in the `gui/gui.py` file. Alongside displaying information about the detected people, this module handles the processing of the captured images. For this purpose, it uses the detection and recognition functions defined in the `detection/detection.py` file.

The GUI module also provides the feature to manually update the predicted data such as the user's age. It is responsible for storing both the manual and the predicted information of users in the user database which is the csv file `database/family.csv`.

The communication with the database happens using the `database.py` file. It also updates the database with every new prediction. Furthermore, it stores images of all users in the `database` directory.

With every new prediction, the GUI module calls the logging function located in the `detection/logging.py` file.

This creates a new entry in the `log.csv` file.

Benchmark reports can be created using the `benchmark.py` file. Similar to the main program it uses the detection and recognition functions defined in the `detection/detection.py` file.

Statistics can be calculated using the `statistics/calculate_statistics.py` file. For calculating the statistics it uses the `logs.csv` file that is created during the detection process.

Tests are located in the `tests.py` file that also uses the detection and recognition functions defined in the `detection/detection.py` file.