UNIVERSITÀ DI PISA

Foodback

Sensing how consumers experience Food

Tommaso **Califano** · Giovanni **Ligato** · Nicola **Ramacciotti** · Gabriele **Suma**

Master's degree in Computer Engineering

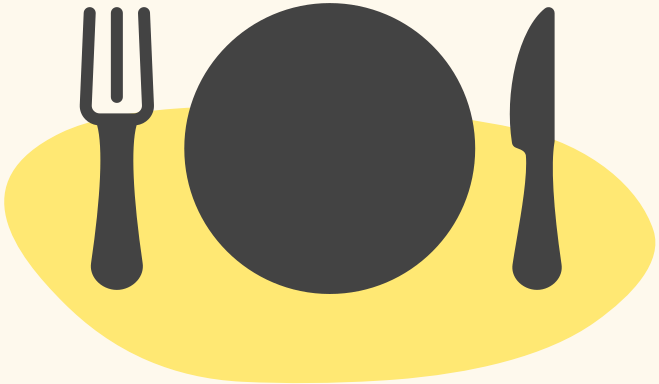
Mobile and Social Sensing Systems

Marco **Avvenuti** · Alessio **Vecchio**

Why Foodback?

Most dining experiences go unrecorded—people simply don't have the time, interest, or motivation to write **reviews**.

Setting the Table

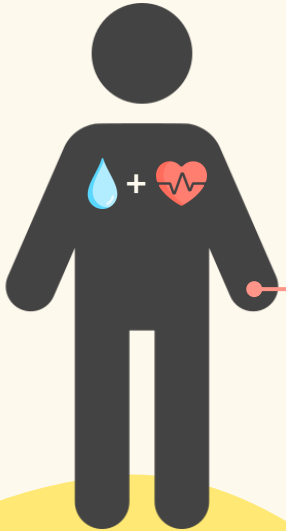


Yet every bite tells a story

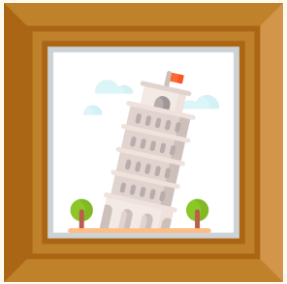
Our goal is to capture that story—**automatically**.

What we Propose

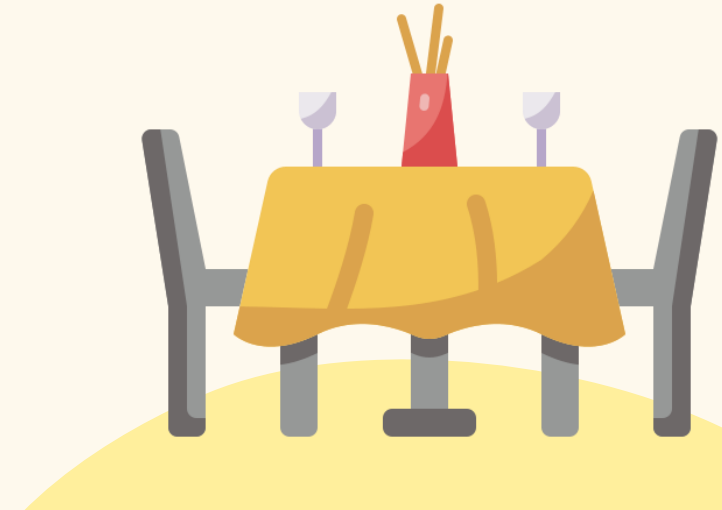
A mobile-based prototype that evaluates food experiences by analyzing *brain* and *body signals*. Bringing objectivity and automation to digital gastronomy.

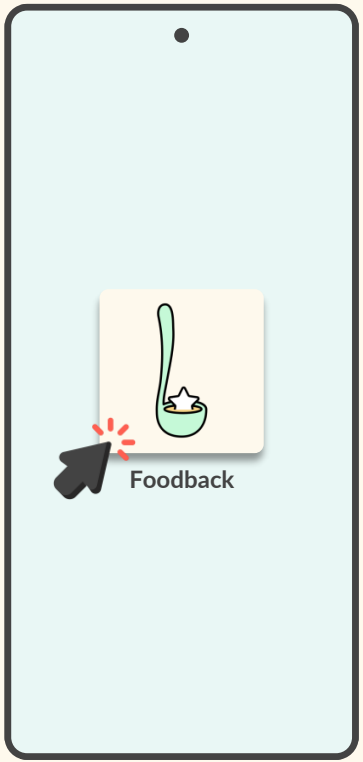


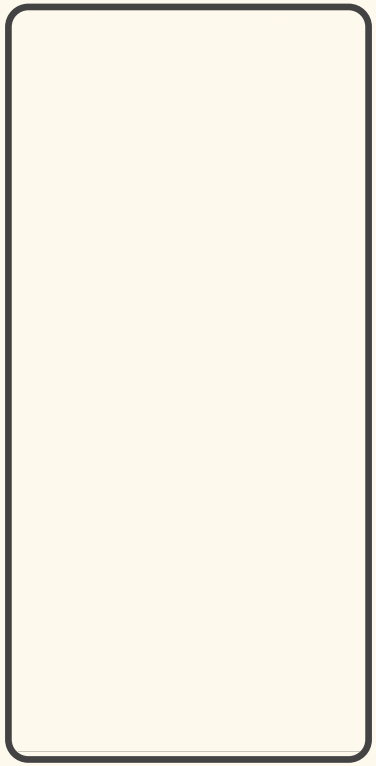
No user input required

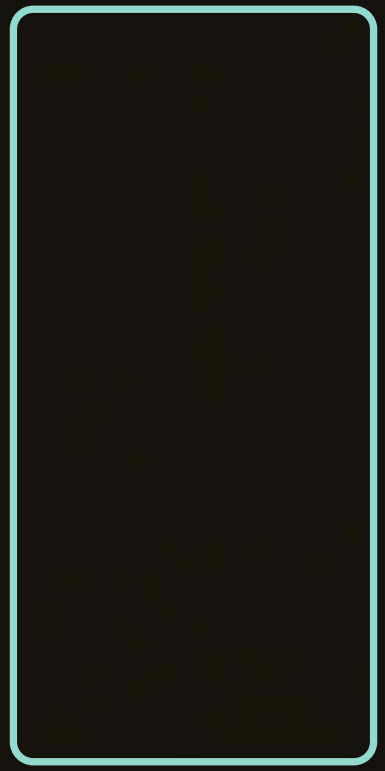


Inside the Foodback **Experience**





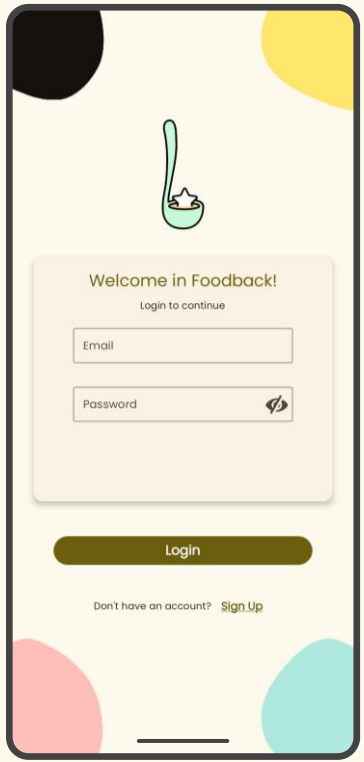




Splash Screen

Login Made Simple

Users can access Foodback's features quickly and securely using just their email and password—nothing more.

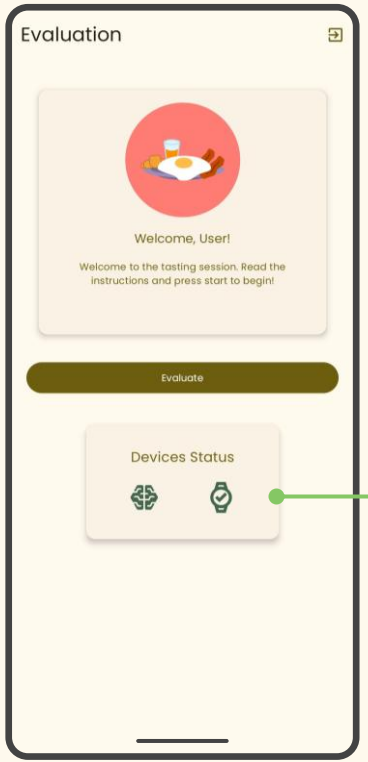


Login Screen

No account? No problem.

You can easily register for Foodback anytime through the sign-up page.

Sign-Up Screen

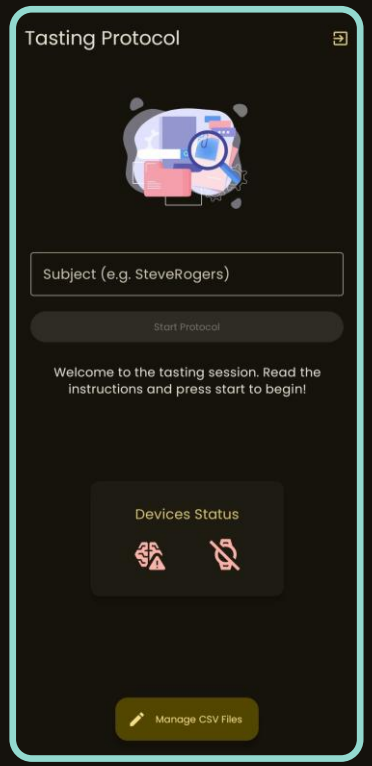


Evaluation Screen

From Login to Evaluation

After logging in, users are directed to their main dashboard, where the pre-trained machine learning model is ready to evaluate tasting experiences automatically.

Both the EEG cap and the smartwatch must be **connected**—indicated by green icons—before the app allows inference to proceed.



Data Collection Screen

The Secret Ingredient

If the logged-in user is an **Admin**, an alternate route is revealed. Instead of the standard dashboard, the Admin accesses a dedicated **data collection** interface, used to gather EEG and smartwatch signals for training the machine learning model.

This experimental page enables controlled sessions across multiple participants and will be detailed further in the protocol discussion.

SYSTEM ARCHITECTURE

EEG
MindRove ARC

Smartwatch
Google Pixel Watch 2

Heart Rate

Electrodermal
Activity

Mobile Phone

Backend
Flask

Database
MySQL

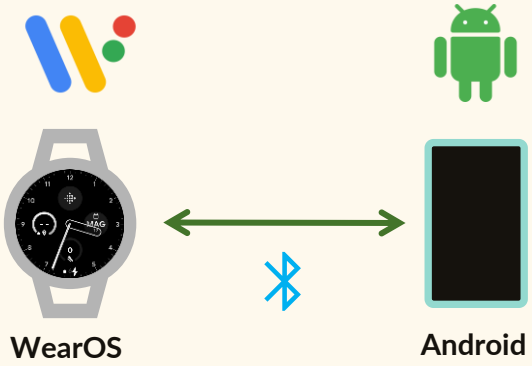
Companion **Application**

Powered by **WearOS**

Splash Screen

Main Screen

Android **Wearable API**



- Part of Google Play Services

- Provides a communication channel between wearable devices and connected handheld devices

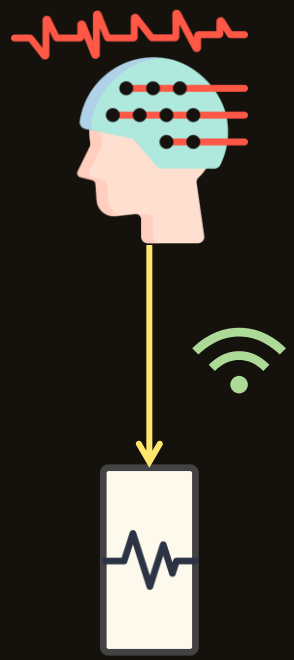
- Message Client API is a part of the Android Wearable API that enables sending short, one-way, real-time messages between connected devices

MINDROVE SDK

- **Mindrove SDK** 2.0 for Android

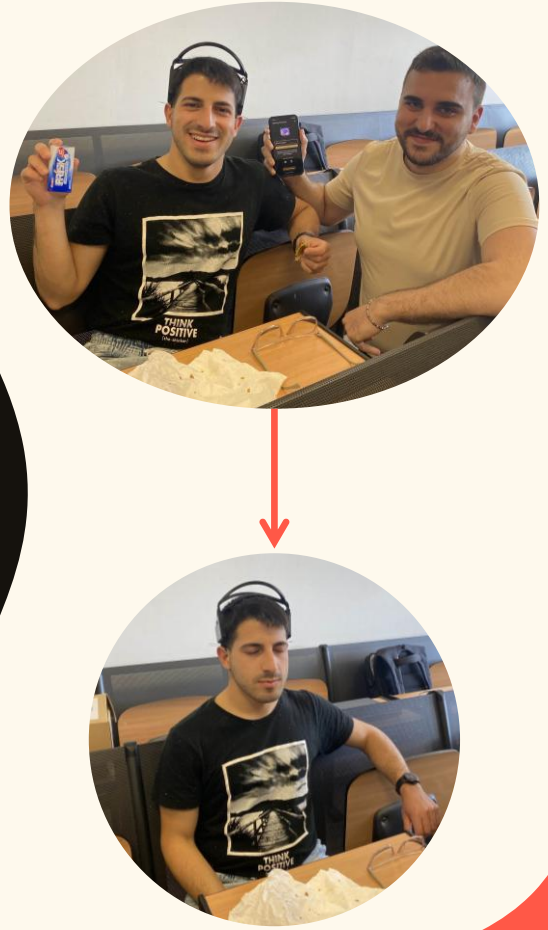
- The **ServerManager** class is responsible for managing a server thread and its interactions.

- **SensorData**:
  - Channel[1-6]
  - NumberOfMeasurements
  - Voltage

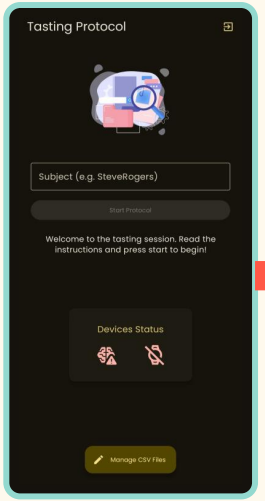


TASTING PROTOCOL

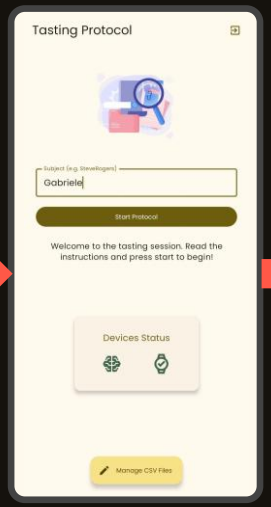
1. **Device Initialization:**
   - The protocol begins once EEG and wearable watch (HR & EDA) are connected and ready to transmit data
2. **Preparation Phase (5 seconds):** GET READY!
   - A first beep signals the subject to bring the food sample into the mouth
   - This is a 5-second window to prepare for tasting
3. **Recording Phase (10 seconds):** REC
   - A second beep marks the start of data recording
   - For the next 10 seconds, the system captures EEG, HR, and EDA signals, representing the subject's sensorial response to the food
4. **Rating Phase:**
   - A final beep ends the data capture
   - A rating interface appears for the administrator to input the subject's sensorial experience rating



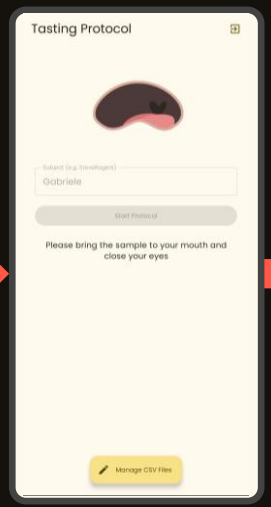
TASTING PROTOCOL (2)



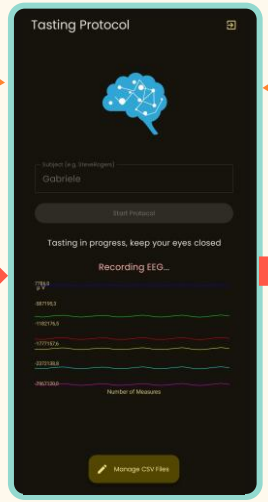
Health check on devices status



Starting the protocol

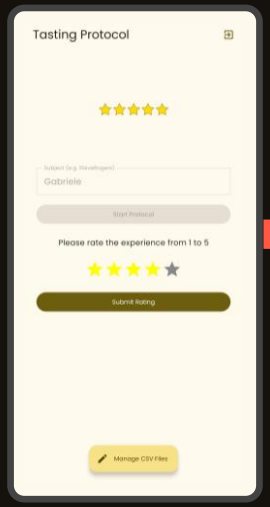


Preparation phase of 5 secs

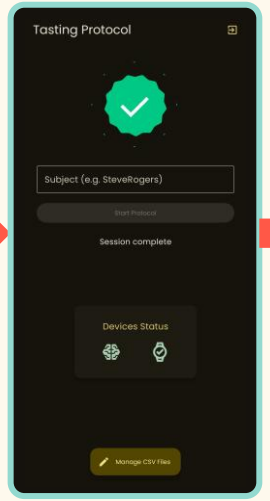


EEG and Wearable data recording phase of 10 secs

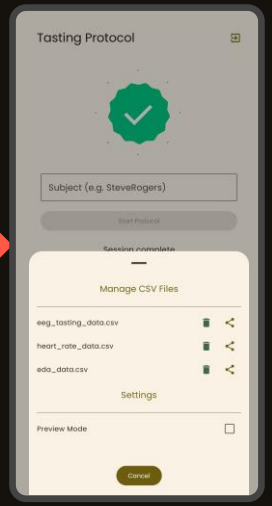
TASTING **PROTOCOL**
(3)



Rating the sensorial experience



Data collected and session completed



Manage your data in one-tap

TASTING **PROTOCOL**

PREVIEW MODE

A **complex** protocol requires **clear** instructions

How do we help subjects understand each step to **optimize** data quality?

Train participants with full protocol exposure

Preview Mode

All steps are performed exactly as in the real session, **but no data is saved**

Identify and fix **signal issues**

Our Homemade Dataset

Small but Gathered with Love

45

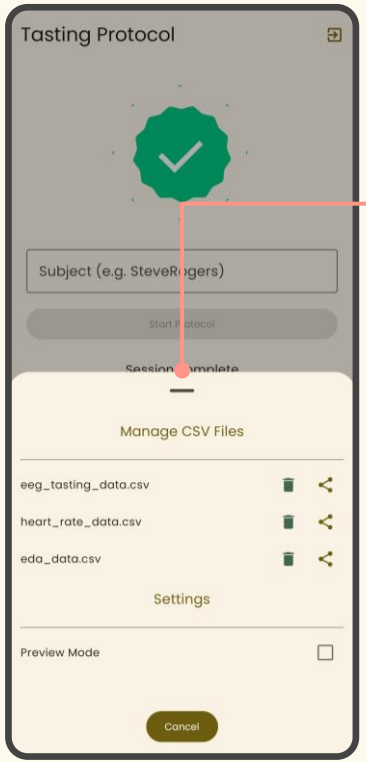
TOTAL TASTING SESSIONS

10

PARTICIPANTS
(ages 16–56)

Data collected via the App in Admin mode:

- Manages cues and rating input;
- Streams EEG from MindRove Arc;
- Triggers smartwatch HR and EDA recording;
- Saves one CSV per sensor, preserving native sampling rates.



CSV Management

A bottom sheet lets you delete or share CSV files individually, making it easy to export data for training after experiments.

Pre-Processing

Sampling
Frequency

CSV
EEG
500 Hz

CSV
HR
~1 Hz
Upsampling
125 Hz

CSV
EDA
~10 Hz
Upsampling
125 Hz

HOLD-OUT
SPLITTER

80/20
Split by
Experiment

TRAINING SET CSV
TEST SET CSV
TRAINING SET CSV
TEST SET CSV
TRAINING SET CSV
TEST SET CSV

What is an **Epoch?**

$$\frac{\textbf{10 s}\ \text{TASTING SESSION}}{\textbf{5}} = \textbf{2 s}\ \text{SEGMENT}$$

This is an **Epoch**

1 Experiment → 5 Epochs

INFO

The epochs form the **input units** for the machine learning model.

The **Kitchen-Ready** Classifier Adopted

EEGNet

Lawhern et al. (2018)

Compact **CNN** specifically designed for **EEG**.

Tiny two-layer footprint, ideal for **small** EEG datasets

Depthwise convolutions for learning spatial patterns.

1861
TOTAL PARAMETERS

1781
TRAINABLE

80
NON-TRAINABLE

Very few details

TENSORFLOW GRAPH

EMBEDDED **EEG** PRE-PROCESSING

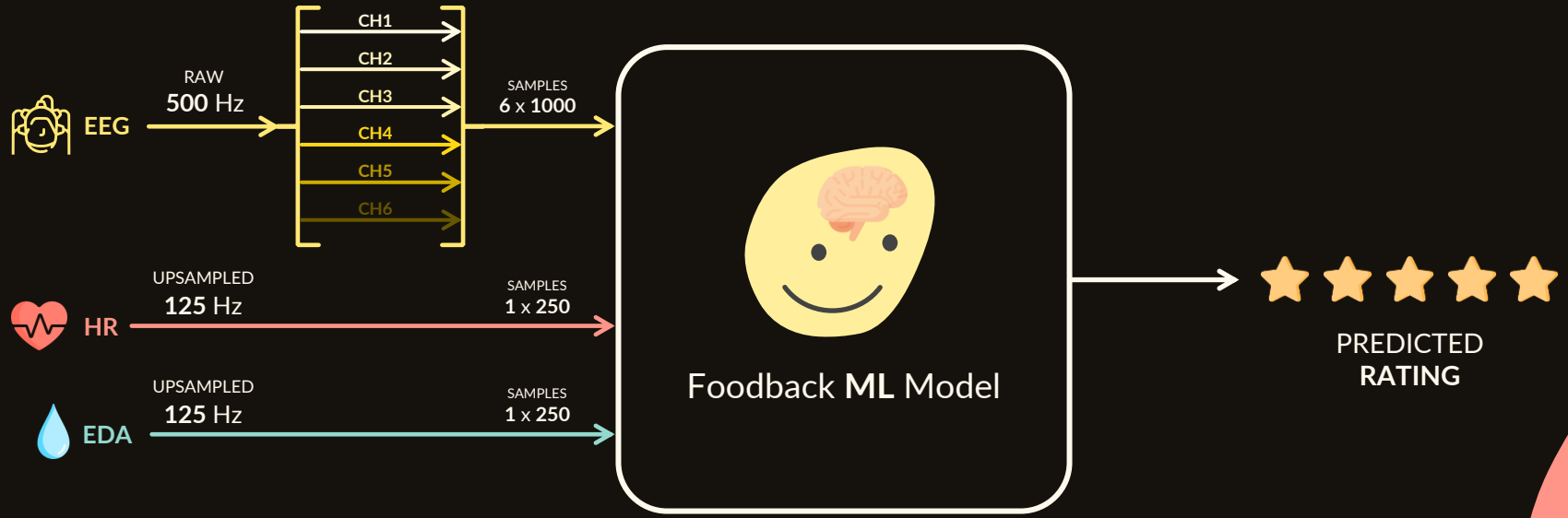
All EEG pre-processing steps are fused directly into the TensorFlow model, so no external DSP or signal handling is needed on-device.

Simplifies the deployment on Kotlin

A bit more detail

TENSORFLOW GRAPH

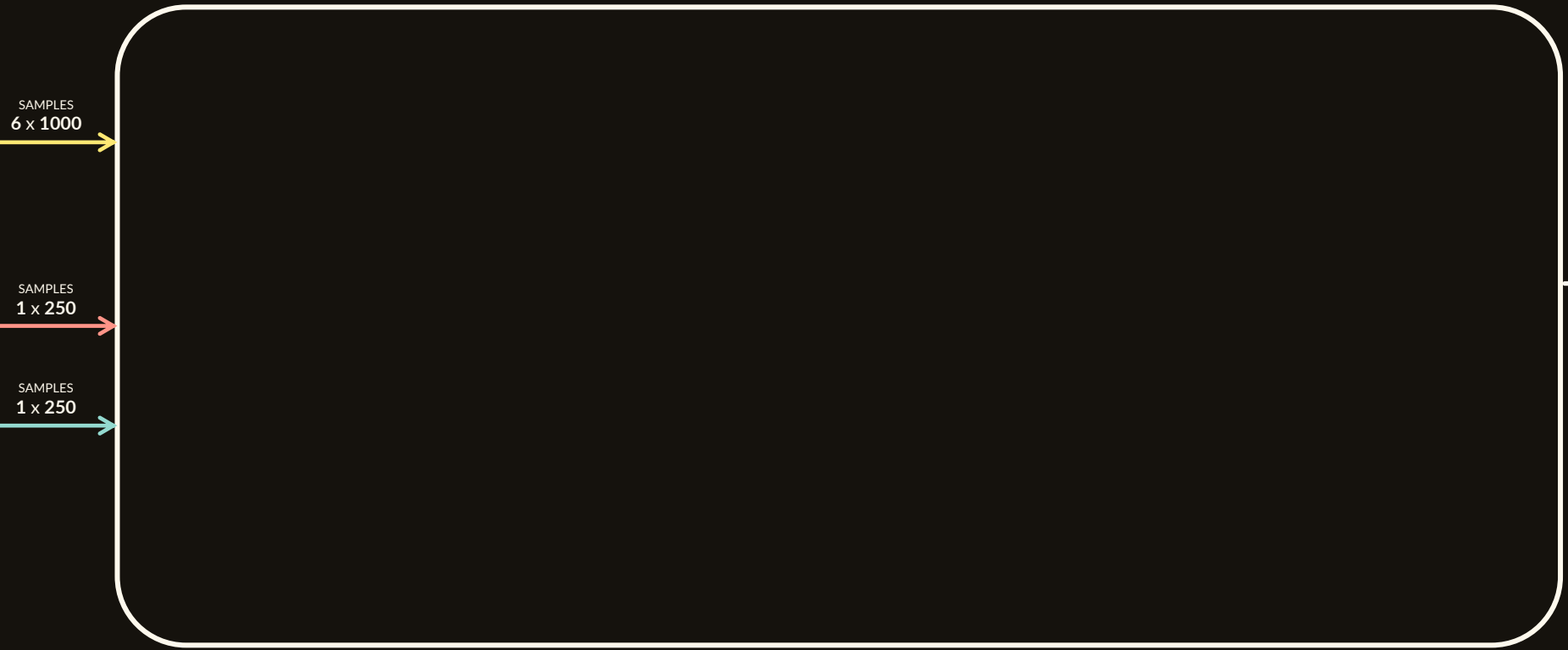
EMBEDDED **EEG** PRE-PROCESSING



A bit more of more detail

TENSORFLOW GRAPH

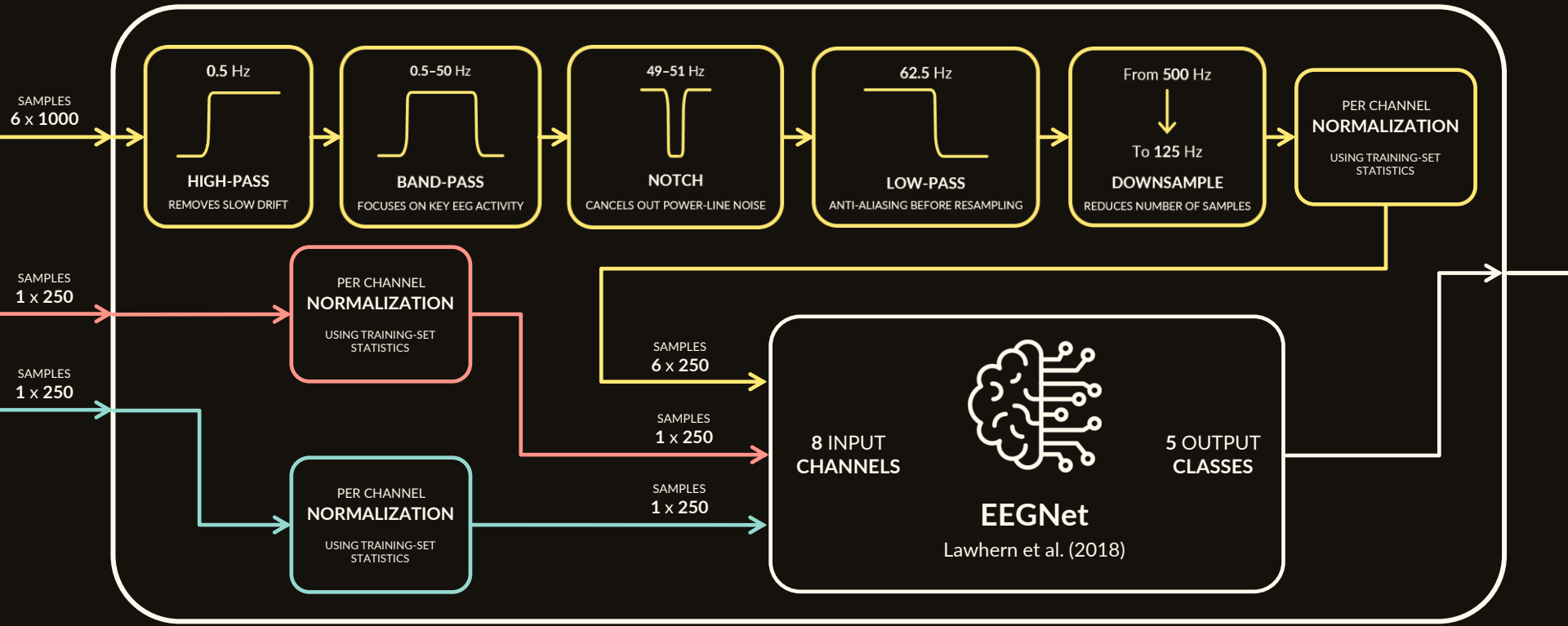
EMBEDDED **EEG** PRE-PROCESSING



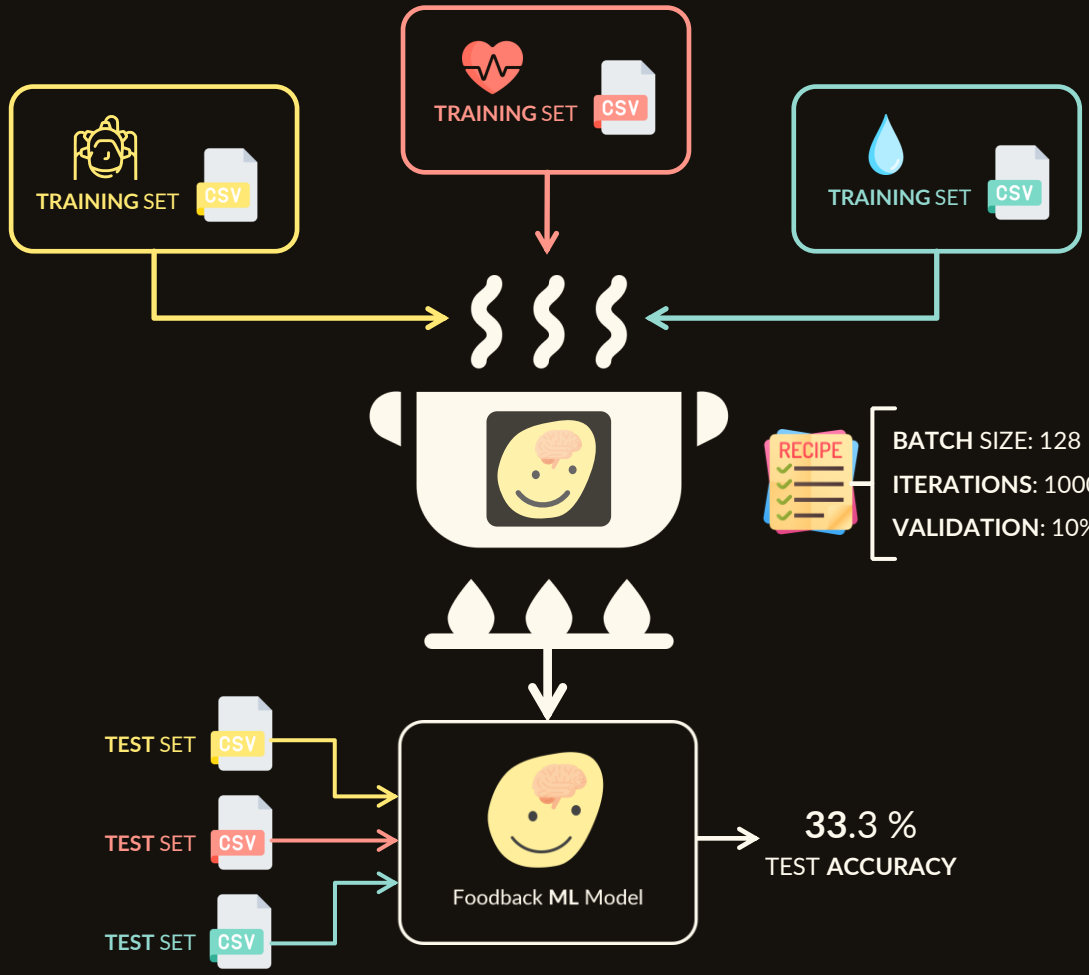
A bit more of more detail

TENSORFLOW GRAPH

EMBEDDED EEG PRE-PROCESSING



MODEL TRAINING & EVALUATION



DEPLOYMENT

Final model exported as a .tflite file, ready for on-device inference via the Foodback app.

A TEMPTING ATTEMPT

But **too sweet**
to trust

Epoch-Level Split

80/20 split applied randomly across **2-second epochs**

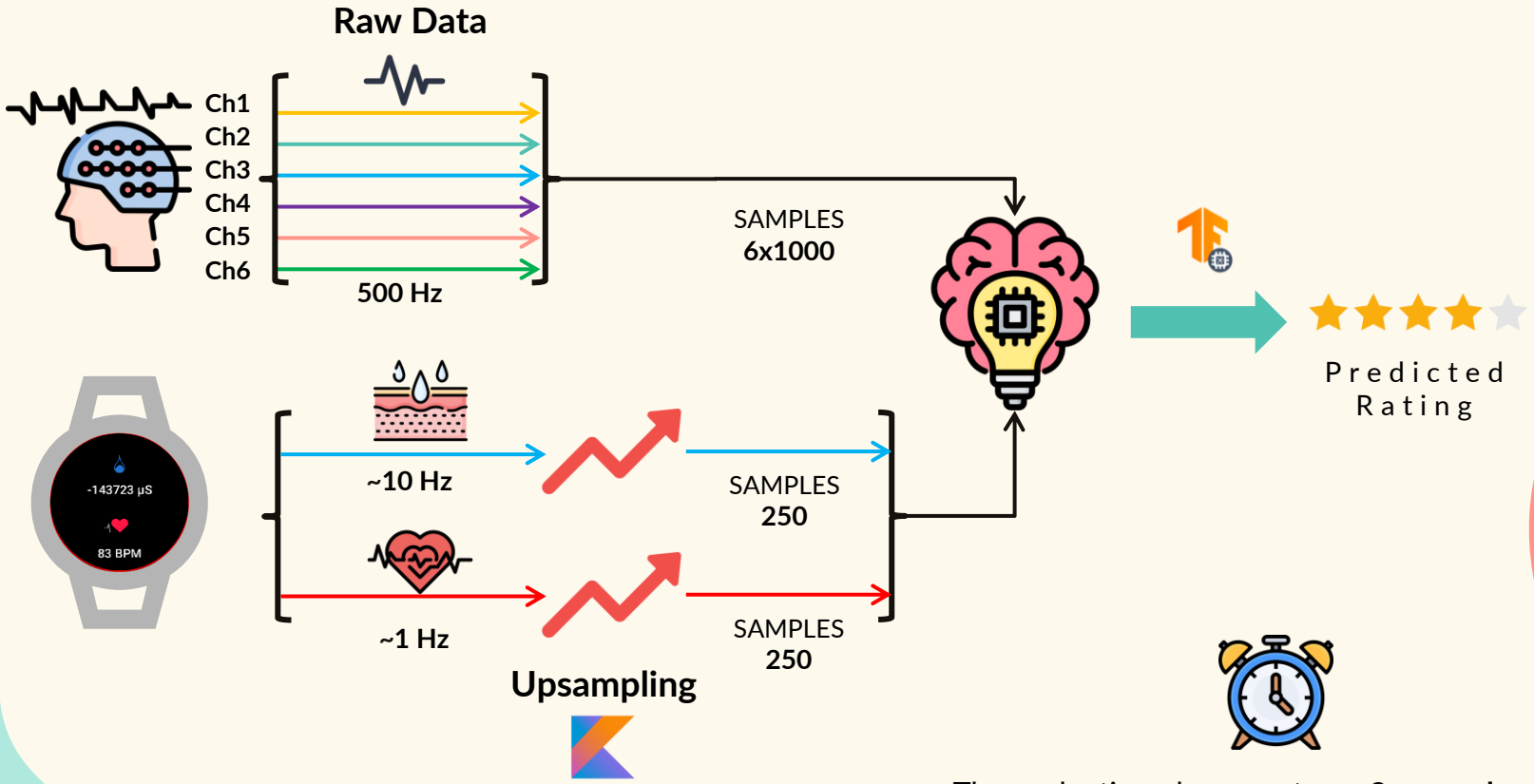
73.3 %

TEST **ACCURACY**

Why it's misleading:

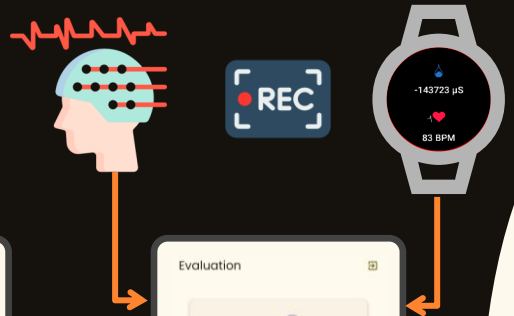
- Information leakage from overlapping sessions
- Model may memorize session-specific noise or context
- Overestimates real-world performance

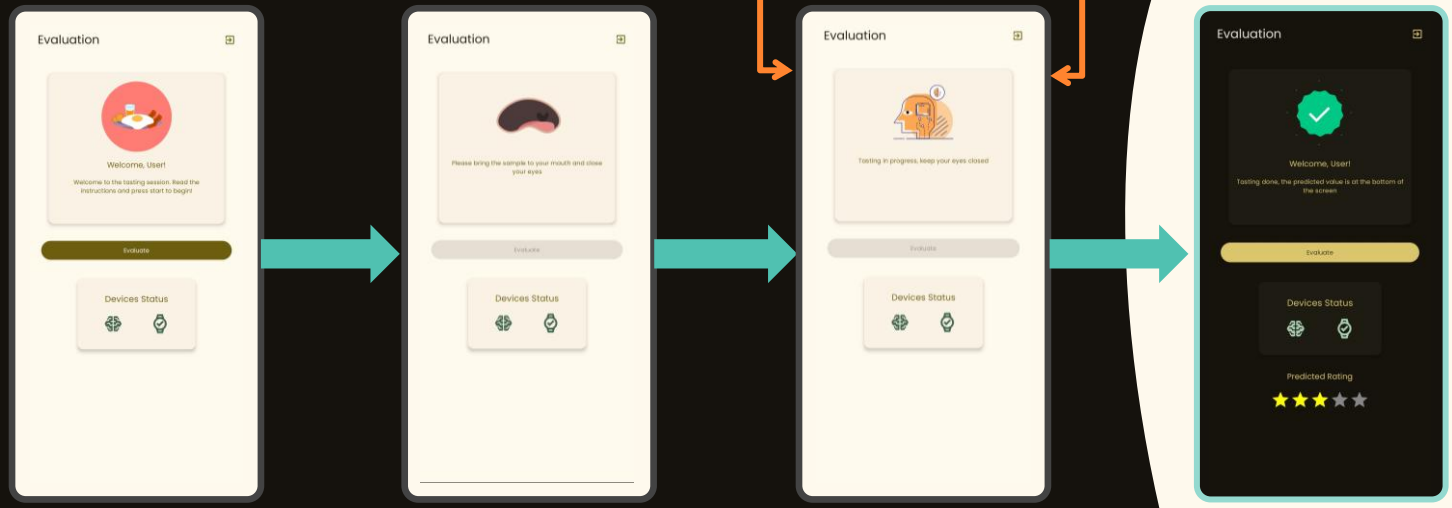
EVALUATION **PROTOCOL**



The evaluation phase captures **2-second** samples

EVALUATION PROTOCOL (2)





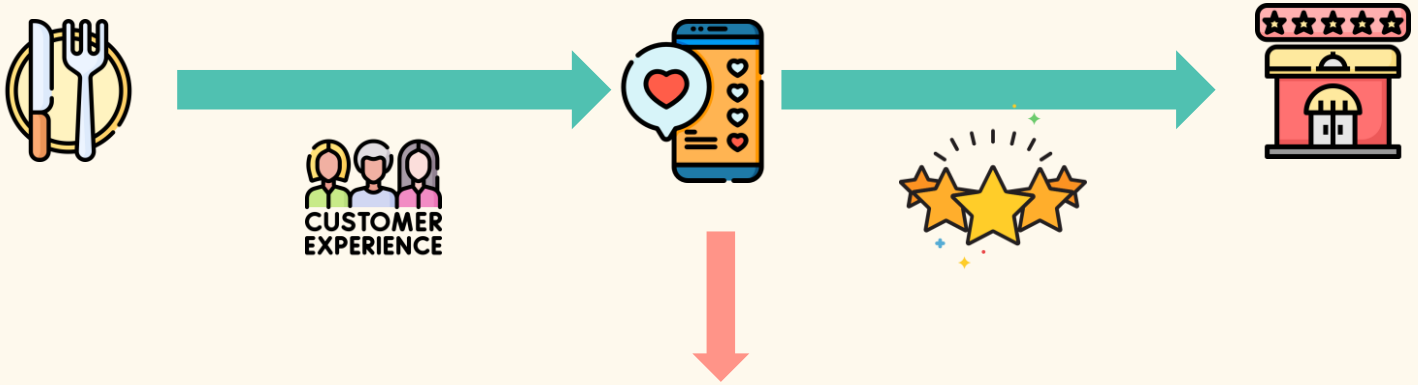
Health check on devices status

Preparation phase of 5 secs

Data collection and model prediction for 2 secs

Score prediction

FUTURE **WORK**



A key future goal is to **enhance the application** to automate the full review process and exploring richer output formats (e.g. free-text review generation)

To enhance model **accuracy** and **generalizability**, future work should include data collection from a **larger** and more **diverse** population

CONCLUSIONS

Just a **Prototype**

Observed classification performance reflects the **true complexity** of decoding human taste perception

Technical viability of real-time, on-device inference

What if a single bite could speak for itself now?

What if a single byte could speak for itself now?

Thanks for your attention