Scent as Perception:

Machine Smell as a Sensory Intelligence for Image Generation

MSc Individual Project
Creative Computing Institute - University of the Arts London

Author: Zhang Yiyi
Supervisor: Professor Mick Grierson
Degree Programme: MSc Computing and Creative Industry (Modular)

Date of Submission: November 2025

Abstract

This project investigates how machine olfaction can function as a sensory intelligence within Human-Computer Interaction (HCI) and computational creativity. It takes the form of an interactive installation that translates real-time odour perception into AI-generated visual art. The system integrates embedded hardware, machine learning, and generative image synthesis to establish a novel cross-modal pipeline that converts chemical signals into dynamic visual compositions.

On the hardware side, a Raspberry Pi 4 is paired with gas sensors (BME688, ENS160, SGP30) to detect volatile organic compounds (VOCs). Sensor data are transmitted in real time via the Open Sound Control (OSC) protocol to the Wekinator platform, where a k-nearest neighbours (KNN) classifier is trained to recognise specific odours such as coffee, floral scents, and wine aromas. The resulting classifications are mapped to a diffusion model running locally in ComfyUI, which generates corresponding visual outputs with interactive latency on the order of seconds.

The study followed a two-phase approach: first, iterative development and testing of the electronic-nose pipeline, followed by a series of explorations demonstrating the cross-modal translation capability and real-time performance of the system. Results showed that the system was capable of generating semantically coherent visual outputs 300-550ms after odour detection, highlighting the potential of olfactory sensing as a primary creative input for generative AI systems.

Beyond its technical implementation, the project contributes to emerging discussions of the "sensory internet" and multisensory experience design. It shows how relatively simple models such as KNN can provide high-value functionality in resource-constrained settings, such as public installations, while still supporting expressive, aesthetically rich interactions. This research supports a movement toward expanding the sensory vocabulary of HCI, positioning smell as an active input rather than a passive output in computational creativity.

Keywords: Electronic Nose (E-Nose); Odour Recognition; Generative AI; Multisensory Interaction; K-Nearest Neighbours (KNN); Interactive Art; HCI; Machine Learning

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Mick Grierson, whose guidance, expertise, and creative insight have been instrumental throughout this research. His mentorship has informed both the technical precision and artistic sensitivity that define this work.

I am also grateful to the Creative Computing Institute for providing an environment that encourages critical experimentation and interdisciplinary collaboration. Finally, I wish to thank my family and peers for their unwavering support, patience, and encouragement throughout this journey.

Table of Contents

Appendices

1. Introduction

1.1 Positionality and Rationale

Within the field of Human-Computer Interaction[48] (HCI), the visual and auditory modalities have become highly mature channels for information exchange[1][2][3]. However, the digitalisation of the chemical senses—namely touch, taste, and smell—remains in its infancy[3][4][50]. This research seeks to bridge this digital sensory divide, focusing in particular on the potential of olfaction as a creative input when integrated with Generative Artificial Intelligence[49] (Generative AI) to construct a novel cross-modal artistic experience.

This study builds upon my previous interactive artwork *Unfading Fragrance* (Figure 1), developed during the postgraduate module Creative Making: Advanced Computational Creativity and Responsive Environments. The installation combined TouchDesigner and Arduino to create a multi-sensory experience through sight, smell, sound, and motion. When participants waved their hands, AI-generated flower animations appeared on the screen while a fragrance device released corresponding scents. The lingering aroma symbolised the continuation and unpredictability of influence, provoking reflection on how life and presence are perceived and remembered.

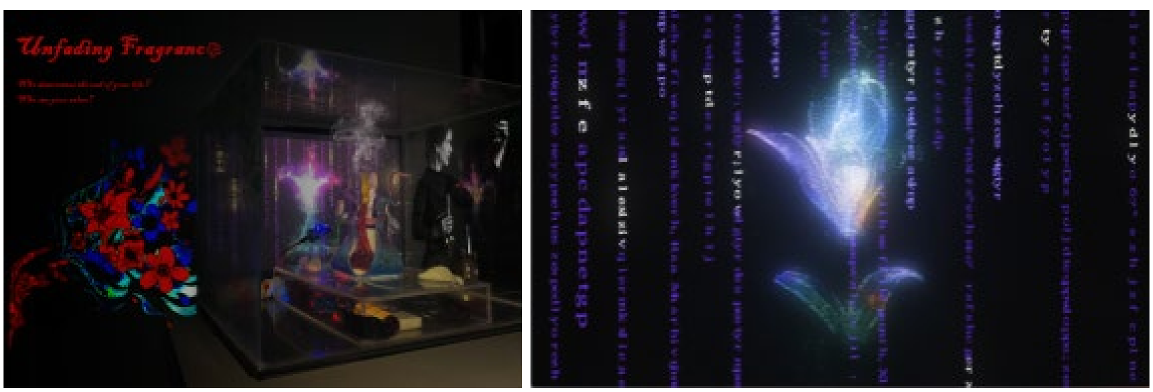

Figure 1. *Unfading Fragrance* — installation view showing audience interaction and AI-generated flower particle animation output from TouchDesigner.(Photograph and system output by Yiyi Zhang, 2024)

Building on this foundation, my current project *Scent as Perception* extends the exploration of olfaction and AI by investigating how smell data, captured through an electronic nose[51] (E-Nose), can be processed via machine learning and translated into AI-generated visual outputs. The motivation for this study arises from a deep curiosity about the perceptual potential of machines: if a machine could 'smell' the world, how might it imagine or represent what it perceives? This inquiry is not merely a technological pursuit but an artistic exploration—an attempt to transform the ephemeral and intangible qualities of smell into shareable, interactive visual forms, thereby expanding the sensory boundaries of human-machine communication and reimagining how computational systems participate in creative perception.

1.2 Research Questions

This practice-led study revolves around a central interactive project and aims to address the following research questions:

Olfactory Data Processing

How can smell data collected through an electronic nose (E-Nose) be effectively processed, recognised, and classified using machine learning techniques?

Cross-Modal Translation

In what ways can the classified olfactory data be translated into visual outputs through generative AI, and how does this translation preserve or reinterpret the sensory qualities of smell?

Interactive Experience

How can real-time interaction between the audience and the AI-driven system create an immersive, multisensory experience that bridges the gap between human perception and machine cognition?

Artistic Inquiry

What artistic and conceptual meanings emerge when smell—a traditionally intangible and transient medium—is used as a creative input for computational systems?

1.3 Project Overview and Creative/ Technical Focus

Scent as Perception is an interactive installation that integrates olfactory sensing, machine learning, and generative AI to construct a real-time system capable of translating scent into visual form. At its core, the project is supported by an integrated technical pipeline composed of three interconnected layers: (1) Olfactory Perception, (2) Pattern Recognition, and (3) Communication and Generative Output. This tri-layered architecture ensures that volatile organic compounds captured from the environment are processed and transformed within an acceptable latency window, enabling a seamless flow from sensory detection to computational interpretation and visual generation.

In this pipeline, the **Olfactory Perception layer** functions as the primary interface with the physical world, detecting and sampling airborne chemical signatures. The **Pattern Recognition layer** then interprets these signals through machine learning models trained to classify scent profiles and identify meaningful patterns. Finally, the **Communication and Generative Output layer** translates these classifications into dynamic AI-generated imagery, allowing scent to drive real-time aesthetic transformation within the installation space.

By establishing smell as a primary input modality rather than a peripheral output effect, the system repositions olfaction as a creative driver in computational art. Through this dual identity—as both sensing apparatus and generative engine—*Scent as Perception* balances technical experimentation with artistic inquiry. The project does not attempt to digitally reconstruct the sensory experience of smell itself; instead, it uses machine perception of scent to open new possibilities for multisensory interaction, speculative translation across modalities, and expanded aesthetic experience.

1.4 Significance and Contribution

The significance of this research can be summarized in three key aspects:

1. Bridging the sensory and generative domains

This project establishes a novel computational link between olfactory perception and visual generation. By using smell as an input modality for generative AI, it expands the scope of multimodal interaction design and proposes a new framework for sensory translation in computational creativity.

2. Integrating machine learning with real-time environmental sensing

Through the integration of Raspberry Pi-based gas sensors, Open Sound Control(OSC)[52] communication, and a trained KNN classifier, the system demonstrates how real-world sensory data can be processed, classified, and translated into dynamic digital content in real time. This contributes to the development of responsive, data-driven media systems.

3. Advancing human-AI artistic collaboration

The project explores how generative diffusion models can serve as aesthetic translators of physical phenomena, enabling a new form of human-AI co-creation. By visualizing intangible sensory experiences, it opens new possibilities for artistic expression, environmental awareness, and multisensory design research.

In summary, this project:

- establishes a full end-to-end pipeline from olfactory sensing to generative visual output.
- demonstrates how low-cost electronic-nose hardware can support real-time, AI-driven interaction.
- positions smell as a central creative input within multisensory, machine-mediated artistic practice.

---

2. Related Work

2.1 Digital Olfaction and Multisensory Interaction

Digital olfaction expands sensory digitisation by integrating smell into interactive systems[5][4]. Research shows that adding olfactory cues in digital environments enhances users' visual memory[6] and immersion[7][8][9], underscoring smell's cognitive and emotional role in HCI[10][9][3].

Digital Scent Technology is categorised into two core domains: Scents Recognition and Digital Scent Synthesis[4]. Building on this, HCI scholarship has increasingly highlighted the value of scent in interaction and experience design[11][12][13][14][3].

Nonetheless, most work remains focused on scent-delivery devices[15][16][17][18][3]; by contrast, this project focuses on the capture stream. I employ an E-Nose for perceptual classification and translate real-time olfactory data into visual output, thereby probing the challenge of mapping chemical signals onto reliable perceptual meaning.

2.2 Machine Learning for Sensory Data

Machine learning has been widely applied to high-dimensional sensory data[19][20]. Electronic-nose systems produce multi-sensor, nonlinear, and often drifting signals influenced by temperature, humidity, airflow and gas diffusion dynamics[21][22]. These characteristics make data preprocessing—such as temporal smoothing, differential features[23], and compensation[24]—essential for robust classification.

Various supervised models, including K-Nearest Neighbours(KNN)[53][25][26][27], Support Vector Machine(SVM)[28][29], Random Forest[30][31][32], Convolutional Neural Networks(CNNs)[33][34][35] and Recurrent Neural Network(RNNs)[36][37], have been successfully applied to odour recognition across food inspection, environmental monitoring, and health contexts.

In line with these findings, this project employs a KNN classifier optimised for cross-device, low-latency operation. The model performs efficiently on engineered features derived from multiple gas-sensing modalities, supporting stable real-time classification for generative interaction.

2.3 Generative AI in Interactive Art

Generative AI[38], particularly diffusion-based models[39][44][45][46], has transformed interactive art by enabling dynamic visual responses driven by user input[40][41]. These systems translate signals such as gesture, voice, biosignals, or environmental data into expressive computational imagery, expanding the aesthetic possibilities of human-AI interaction[47].

While generative models are widely used for multimodal or sensor-driven artworks[54], the integration of chemical sensing remains largely unexplored. Odour-based interaction introduces unique challenges—including ambiguity, drift, and latency—and equally unique opportunities for speculative cross-modal translation.

In this project, diffusion models serve as the visual imagination of the system, transforming classified smell data into evolving images that reflect the machine's interpretation of olfactory input.

2.4 Gaps and Opportunities

Across the reviewed literature, three gaps emerge:

1. digital olfaction research is dominated by scent-delivery technologies rather than scent capture;
2. e-nose classification studies rarely address real-time, interactive contexts;
3. generative AI has seldom been linked to chemical sensing.

This project addresses these gaps by creating an end-to-end pipeline—olfactory sensing, machine-learning classification, and diffusion-based image generation—positioning smell as an active input rather than a passive output in interactive art.

---

3. Methodology

This project adopts a practice-based, iterative methodology that integrates technical experimentation with reflective artistic inquiry. Rather than treating engineering and creative development as separate domains, the method positions every prototype as both a functional test and a perceptual study, examining how system behaviour shapes aesthetic experience and interaction possibilities.

The methodology unfolds across three intertwined strands: (1) iterative prototyping of core subsystems, (2) continuous technical validation, and (3) reflective, practice-led evaluation. This structure supports the overarching research aim—to investigate how olfactory data can drive real-time computational creativity.

3.1 Iterative Prototyping Framework

The development process progresses through a sequence of prototypes, each focusing on a specific subsystem within the olfactory-computational pipeline:

Prototype Stage 1: Data Acquisition and Sensor Behaviour

Early prototypes focused on constructing a functional electronic-nose sensing pathway using the Raspberry Pi. These iterations explored how environmental factors and different odour sources shaped the raw data behavior, informing later decisions about classification strategies and system stability. The purpose of this stage was not to refine features (detailed in Chapter 4), but to understand constraints and opportunities in real-world olfactory sensing.

Prototype Stage 2: Real-Time Data Streaming

Subsequent iterations integrated OSC-based communication, ensuring that data from the Raspberry Pi could be transmitted reliably and with minimal latency to the macOS host. These prototypes primarily validated system smoothness—latency, packet integrity, and responsiveness—establishing the technical foundation for real-time interaction.

Prototype Stage 3: Machine-Learning Integration

Once the data-streaming workflow was stable, the focus shifted to training and testing several classifier families (baseline KNN, advanced KNN, Seq-CNN, Seq-RNN). These models were evaluated not only for numerical accuracy but also for their behavior under continuous real-time conditions. This stage explored how different modelling choices influenced responsiveness, certainty, and the perceptual coherence of the installation.

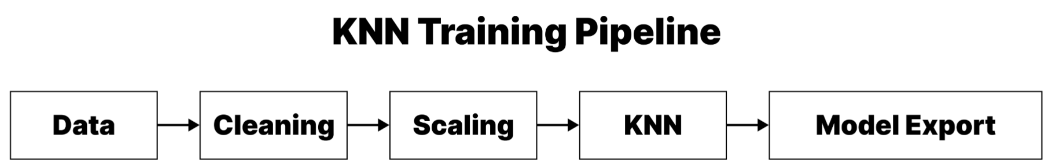

Figure 2. KNN Training Pipeline.

Prototype Stage 4: Cross-Modal Visual Translation

Later prototypes introduced the diffusion-based generative layer. Predicted odour categories were translated into visual prompts and rendered through Stable Diffusion. This phase emphasised qualitative assessment: how visual timing, stylistic stability, and prompt design shaped the aesthetic experience. Iterations here involved prompt refinement, interface design, and stabilisation logic (e.g., trigger windows, fade-in transitions).

3.2 Practice-Led Evaluation and Reflection

Throughout all stages, the methodology employed continuous documentation and reflective analysis. Sensor logs, classification outputs, error cases, and latency traces were collected as part of the technical evaluation workflow. In parallel, qualitative observations were recorded during interactive testing sessions, focusing on perceptual coherence, audience behavior, and emergent phenomena.

This dual-mode evaluation—technical and experiential—enabled balanced decision-making. For example, classifier instability during transitions was interpreted not only as a machine-learning issue but also as a potential aesthetic feature that could be shaped to enrich the visual experience.

3.3 Integration of Technical and Artistic Inquiry

A key methodological principle of the project is the deliberate interweaving of engineering and artistic experimentation. Technical prototypes are treated as creative experiments capable of revealing new possibilities for interaction, while artistic decisions are informed by empirical understanding of sensor behaviour, latency patterns, and model dynamics.

This approach aligns with practice-based research traditions in digital art and HCI, where knowledge emerges through cycles of making, testing, reflecting, and redesigning. By continuously iterating between computational constraints and aesthetic intentions, the project develops both a functional system and an artistic framework that explores how machines perceive and translate olfactory information.

3.4 Summary

In summary, the methodology:

- structures development through iterative, subsystem-focused prototypes.
- combines quantitative system testing with qualitative, perceptual evaluation.
- balances technical optimisation with artistic reflection.
- provides a flexible foundation for the integrated olfactory-visual system developed and analysed in Chapter 4.

Technical specifications—including feature engineering, data-cleaning procedures, classifier details, and sensor-chamber protocols—are presented in full in Chapter 4.

Chapter 4: Experimental Design

This chapter outlines the comprehensive experimental workflow employed in this study for constructing, analysing, and validating a cross-modal olfactory-visual generative system. The entire experimental framework comprises three principal stages, corresponding to the research objectives at three levels: data acquisition, model training, and cross-modal generation.

4.1 E-nose System and Data Acquisition

4.1.1 Overview of the Sensing Architecture

The olfactory sensing architecture consists of a distributed hardware-software pipeline designed to capture volatile organic compound (VOC) signatures in real time and transmit them for classification and visual generation. The core sensing platform is built around a Raspberry Pi 4 Model B equipped with multiple gas sensor modules, including the BME688 environmental sensor, ENS160 air quality sensor, and SGP30 TVOC/eCO_2 sensor.

The system architecture follows a modular design principle, separating sensor data acquisition, feature engineering, and communication layers. Raw sensor measurements are processed on-device using Python scripts that compute six engineered features in real time. These features are then transmitted via Open Sound Control (OSC) protocol over UDP to a macOS host machine for classification and generative output.

4.1.2 Hardware Configuration

Core Components:

Figure 3 shows the main materials and hardware used in the data-collection setup, including labelled glass beakers for odour samples, disposable weighing dishes, a digital precision scale, a glass desiccator-style chamber, a Raspberry Pi with attached sensor boards, nitrile gloves, and PTFE thread-seal tape for improving the airtightness of the chamber.

Figure 3. Materials and hardware used for odour data acquisition, including glass beakers, weighing dishes, precision scale, sealed glass chamber, Raspberry Pi with sensor boards, gloves, and PTFE tape. (Photographs by Yiyi Zhang, 2025.)

The sensing module comprises the following hardware elements:

- **Raspberry Pi 4 Model B** (4 GB RAM) - Main computation platform
- **BME688** - Gas resistance, temperature, humidity, and pressure sensor
- **ENS160** - Air quality sensor providing TVOC, eCO_2, and AQI measurements
- **SGP30** - Additional TVOC and eCO_2 sensing for cross-validation
- **SHT45** - High-precision temperature and humidity monitoring

Chamber Design:

To ensure controlled and reproducible odour exposure conditions, a custom sealed chamber was constructed:

- Heat-resistant borosilicate glass chamber with ground-glass lid
- PTFE thread seal tape applied to all connections to minimise air leakage
- Internal volume approximately 2.5 litres
- Sensor inlet positioned near sample area without direct contact
- Activated carbon placed externally to accelerate baseline recovery between trials

Sample Preparation Equipment:

- Digital precision balance (0.01g resolution) for consistent sample mass
- Glass beakers and weighing dishes for odour source containment
- Disposable nitrile gloves to prevent skin odour contamination
- Electronic vaporizer for controlled smoke-like stimulus generation

4.1.3 Odour Categories

The experimental design incorporates seven distinct odour categories representing diverse chemical profiles commonly encountered in everyday environments:

- **Air (Class 1)** -- Baseline ambient air within sealed chamber
- **Lemon (Class 2)** -- Citrus odour from fresh lemon peel
- **Coffee (Class 3)** -- Freshly ground Arabica coffee beans
- **Perfume (Class 4)** -- Synthetic fragrance compounds
- **Mint (Class 5)** -- Fresh peppermint leaves
- **Whiskey (Class 6)** -- Diluted Scotch whisky sample
- **Smoke (Class 7)** -- Vapor from an electronic cigarette device

These categories were selected to provide maximum chemical diversity while remaining accessible, safe, and representative of real-world olfactory experiences. The smoke category, while included in all training datasets and classification models, could not be preserved as a physical sample for photographic documentation due to its transient vaporous nature.

4.1.4 Data-Collection Protocol

The data acquisition process follows a systematic protocol designed to capture both transient and steady-state sensor responses under controlled conditions.

Phase 1: Sensor Preheating (8-10 minutes)

Following system startup, all sensors undergo thermal stabilisation:

- BME688 gas resistance gradually increases and stabilises
- Metal oxide heating elements reach operating temperature (300-350°C)
- ENS160 and SGP30 complete internal baseline calibration
- Relative humidity readings converge to ambient conditions

During this phase, sensor data are transmitted to Wekinator via OSC but remain unlabeled(marked as "?" in ARFF files) and are excluded from model training.

Phase 2: Air Baseline Recording

With the chamber sealed and empty, multiple time-series segments of clean air are recorded to establish environmental baseline characteristics. These data capture sensor behaviour under zero-odour conditions and provide reference values for subsequent drift compensation.

Phase 3: Odour Sample Introduction

The prepared samples were placed within the chamber following standardized protocols.

- Solid samples (coffee, lemon, mint): weighed to a consistent mass (±0.5g) and placed in glass containers.
- Liquid samples (whiskey, perfume): measured volumes applied to inert surfaces
- Vapour samples (smoke): generated on-demand using electronic vaporizer and introduced immediately after sealing

Chamber is resealed with PTFE-enhanced connections to minimise external air exchange.

Phase 4: Exposure Period Recording

During continuous sample presence, the Python acquisition script:

- Reads raw sensor values at a nominal sampling rate of 20 Hz, with an effective rate of about 3.1 Hz.
- Computes six engineered features using SoftEMA and difference calculations
- Transmits feature vectors to Wekinator via OSC (UDP port 6448)
- Wekinator appends timestamp and user-selected class label to ARFF file

Multiple exposure rounds were conducted for each odour category, with varying durations(30-300 seconds) to capture gas diffusion dynamics, temperature-humidity variations, andsensor drift patterns.
Phase 5: Sample Removal and Recovery

Following sample removal:

- Chamber is opened to reintroduce ambient air
- Activated carbon adsorbent placed inside chamber to accelerate VOC clearance
- Sensor recovery dynamics logged locally on Raspberry Pi for diagnostic purposes
- Recovery data not transmitted to Wekinator (excluded from ARFF training files)

This recovery monitoring enables verification of sensor stability across multi-day experimental sessions and provides insight into long-term drift characteristics 1.

4.1.5 Feature Engineering

To improve classification robustness and reduce sensitivity to environmental noise, raw sensor measurements undergo real-time feature engineering on the Raspberry Pi before transmission to Wekinator.

SoftEMA (Soft Exponential Moving Average)

For each primary gas sensing channel (BME688 gas-ohm, ENS160 TVOC, SGP30 TVOC), a Soft Exponential Moving Average is applied

$$\mathrm{SoftEMA}_t = \begin{cases} \alpha \cdot x_t + (1-\alpha) \cdot \mathrm{SoftEMA}_{t-1} & \text{if } |x_t - \mathrm{SoftEMA}_{t-1}| < \theta \\ x_t & \text{otherwise} \end{cases}$$

where $\alpha = 0.1$ is the smoothing factor, and θ is a relative-deviation threshold. This approach:

- Suppresses high-frequency noise and micro-oscillations
- Preserves rapid changes associated with odour onset
- Reduces sensitivity to sensor warm-up instability
-

Absolute Difference Features

To capture temporal dynamics during odour diffusion and transitions, absolute differences between consecutive SoftEMA values are computed:

$$\Delta_t = |\mathrm{SoftEMA}_t - \mathrm{SoftEMA}_{t-1}|$$

These features highlight steep slopes and transition events that correlate strongly with odour introduction and chamber sealing.

Final Feature Vector

The six features transmitted to Wekinator are:

1. gas_ohm_ema -- Smoothed BME688 gas resistance
2. ens_tvoc_ppb_ema -- Smoothed ENS160 TVOC
3. sgp_tvoc_ppb_ema -- Smoothed SGP30 TVOC
4. diff_gas_ohm_abs -- Gas resistance rate of change
5. diff_tvoc_ppb_abs -- ENS160 TVOC rate of change
6. diff_eco2_ppm_abs -- ENS160 eCO_2 rate of change

This compact six-dimensional representation balances discriminative power with computational efficiency, enabling real-time classification at minimal latency.

4.1.6 Labelling Scheme and Data Structure

ARFF File Structure

Wekinator automatically generates ARFF files with the following schema:

Attribute Name	Type	Description
Time	Date	Timestamp (removed during preprocessing)
Inputs-1 to inputs-6	Numeric	Six engineered features
Outputs-1	Nominal	Odour class label (1--7 or "?")

Table 1: ARFF dataset attributes recorded by Wekinator

Labelling Protocol

During data collection, the researcher manually selects the active odour class in Wekinator's interface. This label is appended to all incoming feature vectors until changed. Unlabeled segments (marked "?") occur naturally during:

- Sensor warm-up periods
- Transitions between odour samples
- Chamber sealing/unsealing operations
- Short stabilisation intervals

These unlabeled entries are retained in raw ARFF files for diagnostic purposes but are systematically removed during model training preprocessing.

Cross-Session Consistency

To ensure reproducibility:

- Sample masses and preparation protocols remain constant across sessions.
- Chamber sealing procedures follow identical steps
- Environmental conditions (room temperature 20-22°C, humidity 40-60%) are monitored
- Sensor baseline stability is verified before each recording session

4.2 Classification Model Training

Following the construction of the odour dataset, the next stage focuses on transforming Wekinator-recorded ARFF files into functional machine-learning models capable of real-time odour classification. Three model families were developed to address different aspects of the classification problem:

1. **Baseline KNN** -- Rapid prototyping and feature validation
2. **Advanced KNN** -- Production-ready classifier with comprehensive preprocessing
3. **Sequence Models (CNN/RNN)** -- Temporal pattern learning for future extensions

4.2.1 Baseline KNN

The baseline K-Nearest Neighbors classifier provides a simple, interpretable starting point for verifying feature quality and ARFF-Python integration workflow.

Training Script: `train_knn_from_arff_weki.py`

Preprocessing Steps:

ARFF Parsing:

- Load Wekinator-generated ARFF file
- Remove Time column (non-numeric timestamp)
- Remove auxiliary fields (ID, Training round) if present
- Preserve Wekinator field names (inputs-1 through inputs-6, outputs-1)

Data Cleaning:

- Convert all columns to numeric using pd.to_numeric(..., errors='coerce')

- Remove rows containing NaN values (including unlabeled "?" entries) ◦
- Ensure complete, fully labelled dataset

Feature Normalisation:

- Apply Standard Scaler to six input features
- Zero mean, unit variance transformation
- Ensures comparable contribution to the Euclidean distance metric

Model Configuration:

- Distance-weighted Euclidean KNN
- n_neighbors selected from {3, 5, 7, 9} via cross-validation
- weights='distance', p=2 (Euclidean metric)

Train-Test Split:

- 80/20 stratified split preserving class distribution
- random_state=42 for reproducibility

Model Export:

- Pipeline: data/models/knn_smell_pipeline.joblib
- Metadata: data/models/knn_meta.json (feature names, class labels, configuration)

Purpose:

This baseline model was used during early development to:

- Verify that engineered features provide adequate class
- separability. Confirm correctness of Wekinator → ARFF → Python integration
- Rapidly prototype the odour-to-image pipeline before introducing advanced models

4.2.2 Advanced KNN

To develop a classifier suitable for stable, reproducible deployment, an advanced KNN pipeline was implemented with formalised preprocessing and hyperparameter selection.

Training Script:

train_knn_from_arff_advanced.py

Stage 1: ARFF Structural Sanitisation

Before DataFrame parsing:

1 Remove @attribute Time date declaration and corresponding data column
2 Remove auxiliary metadata fields (ID, Training round)
3 Decode byte objects to UTF-8 strings

4 Ensure dataset contains only inputs-1 through inputs-6 and outputs-1

Stage 2: Multi-Stage Data Cleaning

(a) Numeric Conversion:
pd.to_numeric(..., errors='coerce')
All non-numeric tokens (including "?") become NaN.

(b) NaN Removal:
Discard rows where any of the six features or the label contains NaN. This removes:

- Warm-up segments
- Transition periods
- Incomplete OSC frames
- All unlabeled entries

(c) IQR-Based Outlier Rejection:

Per-feature filtering using interquartile range:

$$Q_1 - 1.5 \times \mathrm{IQR} \leq x \leq Q_3 + 1.5 \times \mathrm{IQR}$$

Implementation:

- Fixed 1.5× threshold (non-adaptive)
- Applied independently to each feature
- Row is retained only if all six features satisfy bounds
- Reduces impact of turbulence spikes and VOC diffusion outliers

Stage 3: Stratified Cross-Validation and k-Selection

Optimal number of neighbors selected from:

$$k \in \{3, 5, 7, 9\}$$

using:

- Stratified 5-fold cross-validation
- shuffle=True, random_state=42
- Class-balanced folds ensuring representativeness

For each k, compute:

- Mean accuracy across folds
- Standard deviation of accuracy

Best k selected by highest mean accuracy, with ties resolved in favor of the smaller k (simpler model).

Stage 4: Final Model Training

After selecting optimal k, retrain the complete pipeline on the entire cleaned

dataset: StandardScaler → KNeighborsClassifier(

weights='distance',
n_neighbors=best_
k, p=2
)

Model Export:

- Pipeline:data/models/knn_smell_pipeline_advanced.joblib
- Metadata: data/models/knn_meta_advanced.json

Metadata includes:

- Six feature names
- Sorted class label set
- Candidate and selected k values
- Per-k cross-validation statistics
- best_k and best_mean_accuracy
- Source ARFF filename

Methodological Considerations:

Current implementation makes several fixed assumptions:

- **Fixed 1.5× IQR threshold** -- Does not adapt to class distribution or environmental drift

$(k, \mathrm{IQR\ factor}, \mathrm{distance\ metric})$

- **k optimised alone** -- Could extend to joint grid search over

- **Accuracy as sole metric** -- Could incorporate macro-F1, per-class recall for imbalanced datasets

Despite these potential extensions, the advanced KNN pipeline remains the preferred deployment model due to:

- Validated cross-session robustness
- Resilience to sensor drift and chamber
- variability,Transparent, replicable preprocessing
- Smooth integration with real-time inference architecture

4.2.3 Sequence Models (1D-CNN and RNN)

Electronic-nose signals evolve dynamically over time due to vapor diffusion, thermal cycling, and micro-turbulent airflow. To investigate whether modeling temporal dependencies improves performance, two sequence-aware deep learning architectures were implemented.

Training Scripts:

- train_seq_cnn_from_arff.py
- train_seq_rnn_from_arff.py

Temporal Window Construction

Preprocessing Steps:

- Remove Time column at text level (header + data column)
- Convert byte-encoded fields to UTF-8
- Remove rows with label "?"
- Remove rows containing NaN values
- Retain ID and Training round fields (required for ordering and grouping)

Window Extraction:
Data segmented by Training round, then sliding windows constructed with:

- Window length: 64 samples
- Stride: 16 samples
- Retention criterion: ≥80% of samples within the window share the same odour label

This filtering yields windows with stable temporal structure and consistent supervision.

CNN Sequence Model

Architecture: SmellCNN1D

The 1D convolutional neural network captures short-range temporal dependencies:

- Rapid VOC diffusion at exposure onset
- Short-period oscillations in TVOC
- Transient turbulence-induced fluctuations

Network Structure:

Input: (batch_size, 64, 6) # 64 timesteps, 6 features

Conv1D(32 filters, kernel=5) → ReLU → MaxPool(2) Conv1D(64 filters, kernel=3) → ReLU → MaxPool(2) Flatten

Dense(128) → ReLU → Dropout(0.3) Dense(num_classes)→ Softmax

Training Configuration:

- Loss: CrossEntropyLoss
- Optimizer: Adam (lr=0.001) • Batch size: 32
- Epochs: 50 with early stopping
- StandardScaler applied to all windows

Exported Artifacts:

- data/models/seq_cnn_sme
- data/models/seq_cnn_scaler.jo
- data/models/seq_cnn_smell_meta
.json

RNN Sequence Model (LSTM/GRU)

Architecture: SmellRNN

The recurrent neural network captures longer-term dependencies:

- Slow diffusion curves within sealed chamber
- Sensor saturation and plateau behavior
- Gradual recovery trajectories returning to baseline

Network Structure:

Input: (batch_size, 64, 6)
LSTM(hidden_size=64, num_layers=2, dropout=0.2)
or
GRU(hidden_size=64, num_layers=2, dropout=0.2)
Dense(num_classes) → Softmax

Training Configuration:

- Loss: CrossEntropyLoss
- Optimizer: Adam (lr=0.001)
- Batch size: 32
- Epochs: 50 with early stopping
- Bidirectional option available

Exported Artifacts:

- data/models/seq_rnn_smell.pt
- data/models/seq_rnn_scaler.joblib
- data/models/seq_rnn_smell_meta.json

Roe Within Project

Although sequence models were not deployed in the final installation, they demonstrate that:

3. They establish robust methodological baseline for research into time-dependent olfactory perception

1. Electronic-nose data contain rich and learnable temporal structure
2. Sequence-based deep models provide clear upgrade path for future functionalities
3. They establish robust methodological baseline for research into time-dependent olfactory perception

4.2.4 Summary of Modelling Strategy

Model Family	Granularity	Purpose	Key Strengths
Baseline KNN	Frame-level	Rapid prototyping	Simple, interpretable, fast validation
Advanced KNN	Frame-level	Production deployment	Robust preprocessing, cross-validated, stable Short-range
Seq-CNN Seq-RNN	64-step windows	Temporal patterns	dynamics, diffusion onset
Seq-CNN Seq-RNN	64-step windows	Long-term patterns	Saturation curves, recovery trajectories

Table 2: Overview of model families, training approaches, and purposes.

Together, these modelling strategies establish the full computational chain: Oduor signal – feature vector – category predication – prompt construction – image generation, which forms the technical foundation for the cross-modal olfactory visual system.

4.3 Model Evaluation

This section evaluates the odour classifiers trained in Section 4.2, focusing on three model families: Advanced frame-based KNN, Sequence-aware 1D CNN, and Sequence-aware RNN (LSTM/GRU).

Evaluation Script: eval_confusion_matrices.py

4.3.1 Evaluation Protocol

Two evaluation schemes were used to accommodate different temporal granularities:

(1) Advanced KNN -- Stratified 5-Fold Cross-Validation

- **Dataset:** smell_dataset_merged.arff
- **After merging:** 32,156 frames
- **After cleaning:** 23,666 frames (NaN removal + IQR outlier removal)
- **Evaluation:** 5-fold stratified CV preserving odour-class distribution

(2) CNN & RNN -- Temporal Window Evaluation

- **Dataset:** smell_dataset.arff
- **Windowing strategy:**

- Window length = 64 samples
- Stride = 16 samples
- Label assigned when ≥80% frames share same class

Final test set: 874 labelled windows

This dual scheme enables fair comparison between frame-level and sequence-level models while acknowledging their different data representations.

4.3.2 KNN Results

KNN performance was evaluated under four candidate values: $k \in \{3, 5, 7, 9\}$

k	Mean Accuracy	Std Dev
3	0.9490	0.0016
5	0.9475	0.0028
7	0.9455	0.0031
9	0.9438	0.0036

Table 3: Cross-validated KNN accuracy across different k values

Best Configuration:

$k = 3$, distance-weighted Euclidean KNN

5-fold CV accuracy = 94.90%

Interpretation:

These results indicate:

- The six engineered features (SoftEMA + diff) are highly discriminative
- Inter-session drift is effectively removed by IQR preprocessing stage
- Odour classes form well-separated clusters in single-frame feature space
- KNN provides a simple, stable, deployment-friendly baseline

4.3.3 CNN Results

Overall Metrics:

Accuracy = 0.92
Macro F1 = 0.76
Weighted F1 = 0.91
Test windows = 874

Confusion Matrix Observations:

- **Strong Performance:** Air (1), Lemon (2), Perfume (4), Smoke (7) achieve near-perfect recall
- **Most Common Confusion:** Coffee (3) Mint (5) due to similar short-range VOC dynamics
- **Underrepresented Class:** Whiskey (6) shows reduced recall (only 18 test windows)

Interpretation:

CNN excels at:

- Capturing short-term temporal signatures during diffusion onset
- Modeling local fluctuations in VOC curves
- Learning class-specific micro-dynamics

It is the **best-performing sequence model overall**.

4.3.4 RNN Results

Overall Metrics:

Accuracy = 0.85
Macro F1 = 0.61
Weighted F1 = 0.83
Test windows = 874

Confusion Matrix Observations:

- **Strong Classes:** Air (1), Lemon (2), Perfume (4), Smoke (7) remain robust
- **Reduced Performance:**
  Coffee recall = 0.41
  Mint recall = 0.49
  Whiskey recall = 0.00 (insufficient data + volatility similarity)

Interpretation:

RNN models long-range trends but:

- More sensitive to class imbalance
- More prone to over-smoothing short-term dynamics Less stable for overlapping VOC patterns

CNN > RNN in both robustness and class balance.

4.3.5 Cross-Model Comparison

Model	Accuracy	Macro F1	Weighted F1
Advanced KNN	0.9490	--	--
Seq-CNN	0.92	0.76	0.91
Seq-RNN	0.85	0.61	0.83

Table 4: Model performance comparison across KNN, CNN, and RNN

4.3.6 Summary of Findings

- Temporal signatures are highly discriminative -- Sequence models outperform frame-only approaches for most classes
- CNN offers best overall performance among sequence models, balancing accuracy and stability
- Advanced KNN provides strong frame-level baseline and remains highly suitable for real-time deployment
- Class imbalance affects sequence models more strongly than KNN, particularly for Whiskey (18 windows)

- Coffee - Mint confusion is the dominant error pattern across all models

4.4 Diffusion Model

This section describes how the project uses a pre-trained text-to-image diffusion model as the generative backend for olfactory-to-visual translation, rather than training a custom, smell-conditioned diffusion network from scratch. The diffusion component is implemented in src/diffusion_backend.py and configured via src/config.py, and is tightly integrated with the real-time classification bridge in src/smell2image_bridge_mps.py.

At a high level, the diffusion backend:
- loads a fixed SDXL-style checkpoint (RealVisXL v4.0) on Apple M-series MPS or CPU,
- accepts textual prompts and optional seeds from the classification layer,
- returns 1024×1024 RGB images under tight latency constraints suitable for live interaction.

4.4.1 Base Model and Runtime Environment

The system uses an SDXL-derived checkpoint as its generative backbone:

1. **Base checkpoint**
    a. Model: RealVisXL v4.0 (SG161222/RealVisXL_V4.0)
    b. Resolved at runtime by resolve_sdxl_model_id() in config.py, which:
    I. first checks the SD_BASE_MODEL_ID environment variable (local path or remote ID),
    Ii. otherwise falls back to a local clone under data/models/realvisxl_v40,
    Iii. finally defaults to the public Hugging Face repo identifier.
2. **Loading and device selection**
    a. The model is loaded via AutoPipelineForText2Image.from_pretrained(...) from the diffusers library, with safety_checker=None.
    b. DiffusionBackend automatically selects:
    I. MPS (Apple Silicon) when available, using torch.float16 for faster inference,
    Ii. CPU otherwise, forcing torch.float32 to avoid numerical issues such as NaNs and flat outputs.
3. **Inference configuration**
    a. Resolution: GEN_IMAGE_SIZE = (1024, 1024) (canonical SDXL square format).
    b. Steps: NUM_STEPS = 6, intentionally low to keep generation time within real-time bounds.
    c. Guidance: GUIDANCE_SCALE = 1.2, a modest classifier-free guidance value, balancing prompt adherence and diversity.
    d. The generate(...) method wraps a single call to self.pipe(...) with these parameters and optional seeded generator.

4.4.2 Optional LCM LoRA Acceleration

The backend is designed to support, but not require, Latent Consistency Model **(LCM) LoRA acceleration:**

- LCM_LORA_WEIGHTS can be set via environment variable to either:

  a local LoRA file path or a Hugging Face LoRA repository ID.

- LCM_LORA_SCALE controls the strength of the adapter.

Integration pattern:

When a LoRA path is provided, DiffusionBackend attempts to:

- load the adapter with pipe.load_lora_weights(..., adapter_name="lcm_lora"), then
- either fuse it into the UNet and text encoder via pipe.fuse_lora(...) for speed,
- or keep it as an active adapter using pipe.set_adapters([...], weights=[...]).

Current project configuration:

In the present setup, USE_LCM_SCHEDULER = False and FUSE_LORA_WEIGHTS = False, meaning:

- RealVisXL's native scheduler is retained, and
- no LoRA fusion is performed by default.

These hooks remain available for future optimisation but are not essential to the reported results.

4.4.3 Conditioning via Prompts and Seeds

Instead of conditioning the diffusion model directly on continuous sensor embeddings, the project uses a higher-level strategy based on prompt engineering + seed scheduling:
Class-to-prompt mapping

- Olfactory signals are first classified by a KNN pipeline into one of six discrete smell classes.
- At run time, smell2image_bridge_mps.py maps each stable class label to a textual description using LABEL_TO_PROMPT_DEFAULT, which is loaded from:

  data/prompts/smell_prompts_zh.json or smell_prompts_en.json, if present,

  a built-in default mapping defined in config.py if no external file is found.

- Prompts are written as rich cinematic descriptions (e.g. a high-fashion lemon altar, a surreal mint tableau), so that each smell is associated with a distinct visual mood rather than a literal icon.

Class-to-seed mapping

- A parallel dictionary LABEL_TO_SEED assigns a fixed pseudo-random seed per class (e.g. "2" → 12002).

- This produces:
  consistent overall composition and style whenever the same odour is detected, and
  small local variations driven by diffusion noise, preserving some generative spontaneity.

Generation call

- For each qualified event, the bridge calls:
```
image = diffusion_backend.generate(
    prompt,
    width=GEN_IMAGE_SIZE[0],
    height=GEN_IMAGE_SIZE[1],
    seed=LABEL_TO_SEED.get(label)
)
```
- DiffusionBackend returns a NumPy uint8 RGB array, which is then written to disk.

4.4.4 Trigger Logic and Integration with the Bridge

The diffusion backend is controlled by the Smell2Image Bridge, which ensures that images are only generated when the sensor state is stable enough:

Stability window

- Incoming OSC feature vectors are processed by the KNN classifier.
- Each decoded label is appended to a sliding window (prediction_window) of length TRIGGER_STABILITY_FRAMES (default 12).
- A label is considered stable only if all entries in the window are identical.

Variation-aware gating

- In addition to label agreement, the bridge tracks a relative variation score between consecutive feature vectors (_register_feature_variation(...)).
- A new image is only triggered when:
  the label is stable for the entire window, and
  the variation score falls below SENSOR_VARIATION_THRESHOLD, indicating that the smell has genuinely settled rather than flickering.

Rate limiting

- GEN_INTERVAL_SEC enforces a minimum interval between two generations.
- GENERATION_COOLDOWN_SEC adds a short cooldown after each successful generation to avoid rapid oscillation when the environment hovers near a decision boundary.
- During the actual diffusion call, the bridge temporarily pauses ingest (pause_ingest = True) to prevent feature queues from growing uncontrollably.

4.4.5 Output Handling and Logging

Finally, the integration with the user-facing interface and logging is handled as follows:

Image saving and UI consumption

- Each generated image is:
  saved as data/samples/sample_<label>_<timestamp>.jpg, and
  duplicated as data/samples/latest.jpg for easy access.
- The Streamlit frontend (app.py) periodically scans data/samples/ for the most recent file, infers the label from the filename pattern sample_(\d+)_..., and cross-checks this with the JSON status file latest_status.json written by the bridge.

Status and feedback

- status_cache in the bridge stores:
  the latest feature vector,
  the predicted class,
  the chosen prompt,
  the last generation time, and
  the current generation state ("idle", "received_features", "generating", "completed", or "error").
- This cache is written atomically to latest_status.json, which the Streamlit UI uses to display status pills and sensor readouts.

Event logging

- For later analysis, each generation event is appended as a row to classification_history.csv, including:
  timestamp,
  backend ("knn"),
  label, prompt, and image path,
  generation duration,
  fixed seed, and
  the last variation score.

- The current project does not compute image quality metrics such as FID; the emphasis is on temporal behaviour, latency, and experiential quality in an interactive setting.

Chapter 5: Results

This chapter presents the empirical outcomes of the experiments described in Chapter 4. Whereas Chapter 4 focused on the design and technical execution of data acquisition, model training, and evaluation procedures, the current chapter examines the system's observable performance during both offline classification and real-time olfactory-to-visual translation.

The results are organized around three core dimensions:

1. Model accuracy and stability
2. Real-time system responsiveness
3. Qualitative visual and experiential characteristics

Together, these findings provide an integrated view of how reliably the electronic-nose platform can distinguish odours, how consistently the classifier operates in real-world conditions, and how effectively the generative component translates odour categories into visual imagery.

5.1 Tools and Development Environment

The development environment comprises two distinct components: the Raspberry Pi-based data acquisition system and the macOS host used for model training, inference, and image generation.

System Architecture Overview:

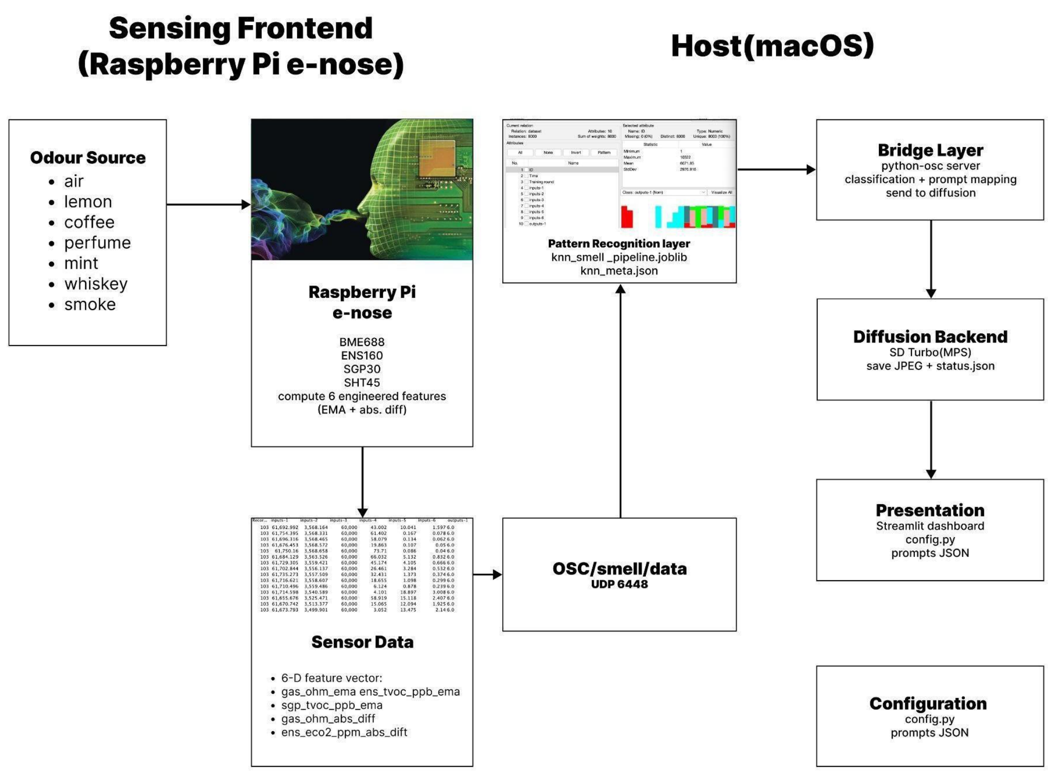

Figure 4: System architecture showing Raspberry Pi olfactory sensing, OSC transmission, macOS classification, and diffusion-based image generation

The complete workflow consists of four stages:

1. Olfactory Data Acquisition (Raspberry Pi)

- BME688, ENS160, SGP30 gas sensors
- Real-time feature computation (SoftEMA + difference metrics)

- Python acquisition script

2. OSC Transmission

- UDP communication (port 6448)
- Six engineered features transmitted at ~1 Hz
- python-osc library

3. Machine Learning Classification (macOS)

- KNN/CNN/RNN model inference
- smell2image_bridge_mps.py for real-time processing
- Class label output and prompt selection

4. Diffusion-Based Image Synthesis (macOS)

- RealVisXL V4.0 model (SG161222/RealVisXL_V4.0)
- AutoPipelineForText2Image interface
- MPS backend acceleration
- Streamlit UI (app.py) for visualization
  The configuration and implementation of this diffusion backend are described in detail in Section 4.4 (Diffusion Model).

5. Model Development Stack:

- Python 3.9+
- scikit-learn (KNN, preprocessing)
- PyTorch (CNN/RNN sequence models)
- pandas, numpy (data processing)
- joblib (model serialisation)

6. Real-Time Inference Stack:

- python-osc (OSC communication)
- diffusers (Stable Diffusion pipeline)
- Streamlit (interactive UI)
- PIL/Pillow (image rendering)

This layered and modular workflow ensures strong separation among data acquisition, offline model development, and real-time cross-modal generation, while supporting reliable, iterative experimentation throughout the project.

5.2 Olfactory-to-Visual Translation: Basic Prototype Results

The first set of results focuses on the foundational end-to-end pipeline: electronic-nose input → classification → diffusion-based image generation.

5.2.1 Input Conditions

The prototype was trained and evaluated using ARFF datasets containing temporal sequences of six engineered features from multi-sensor configuration (BME688, ENS160, SGP30).

Sensor Response Pattern:

Across all recording sessions, sensors exhibit consistent temporal response:

1. **Warm-up Stabilization** -- Feature values converge after heaters reach operating temperature
2. **Baseline in Clean Air** -- Six channels fluctuate within narrow, stable range
3. **Odour Introduction** -- Characteristic rises/transitions in gas_ohm, TVOC, eCO_2
4. **Steady-State Response** -- Features settle into odour-specific plateaus

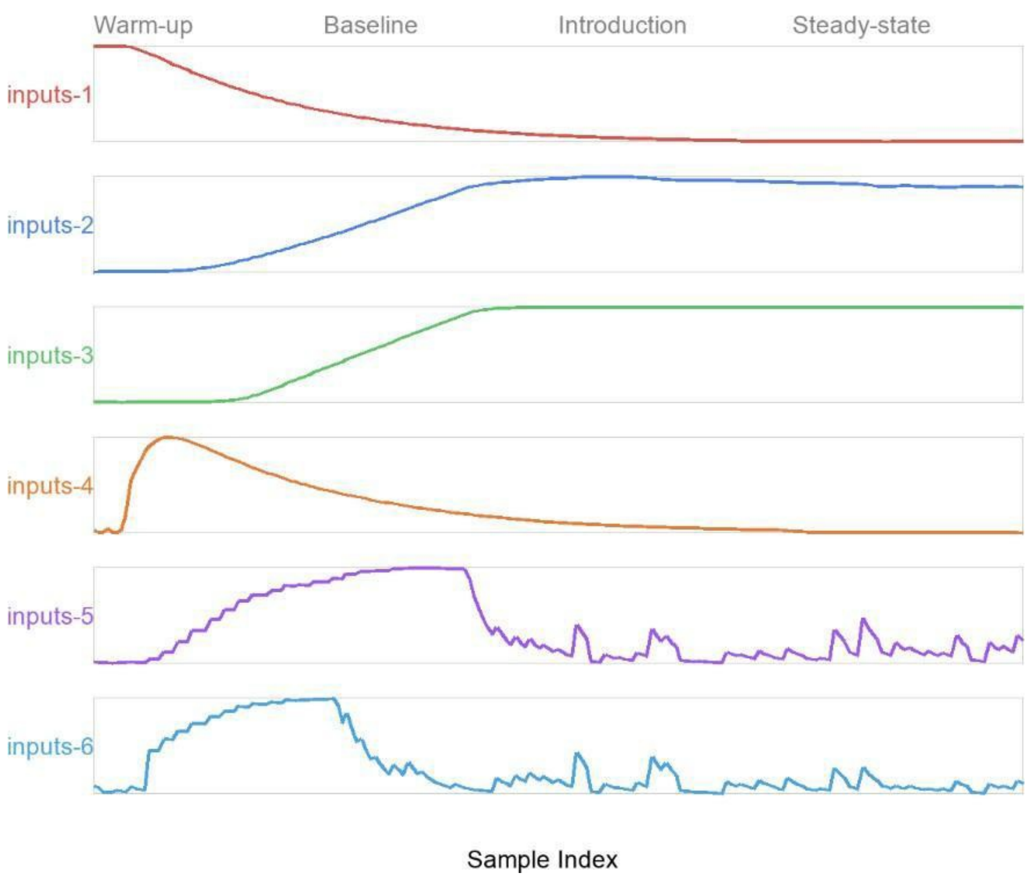

Figure 5: Time-series traces of six sensor features showing warm-up, baseline, introduction, and steady-state stages

The basic prototype was trained and evaluated using previously collected ARFF datasets (e.g., smell_dataset.arff), which contain temporal sequences of six engineered features aggregated from a multi-sensor configuration, including the BME688 gas sensor as well as auxiliary VOC/CO_2 sensors such as ENS160 and SGP30 devices. These six channels—such as gas_ohm_ema, ens_tvoc_ppb_ema, and sgp_tvoc_ppb_ema—represent smoothed and differential measurements that capture the dynamic behaviour of the olfactory environment. In the ARFF files, odour categories are stored as numeric labels (e.g., 1–7). Their semantic meanings—such as "air", "lemon", "mint", or "coffee"—are defined externally in the system's configuration and prompt-mapping files rather than embedded directly in the dataset.

Across all recording sessions, the sensors exhibit a consistent temporal response pattern that reflects the physical process of odour introduction:

1. Warm-up stabilisation — feature values converge after the sensor heaters reach operating temperature.
2. Baseline in clean air — all six channels fluctuate within a narrow, stable range.
3. Odour introduction — characteristic rises or transitions appear in gas_ohm, TVOC, eCO_2 and related features.
4. Steady-state response — features settle into odour-specific plateaus, forming identifiable sensor "fingerprints."

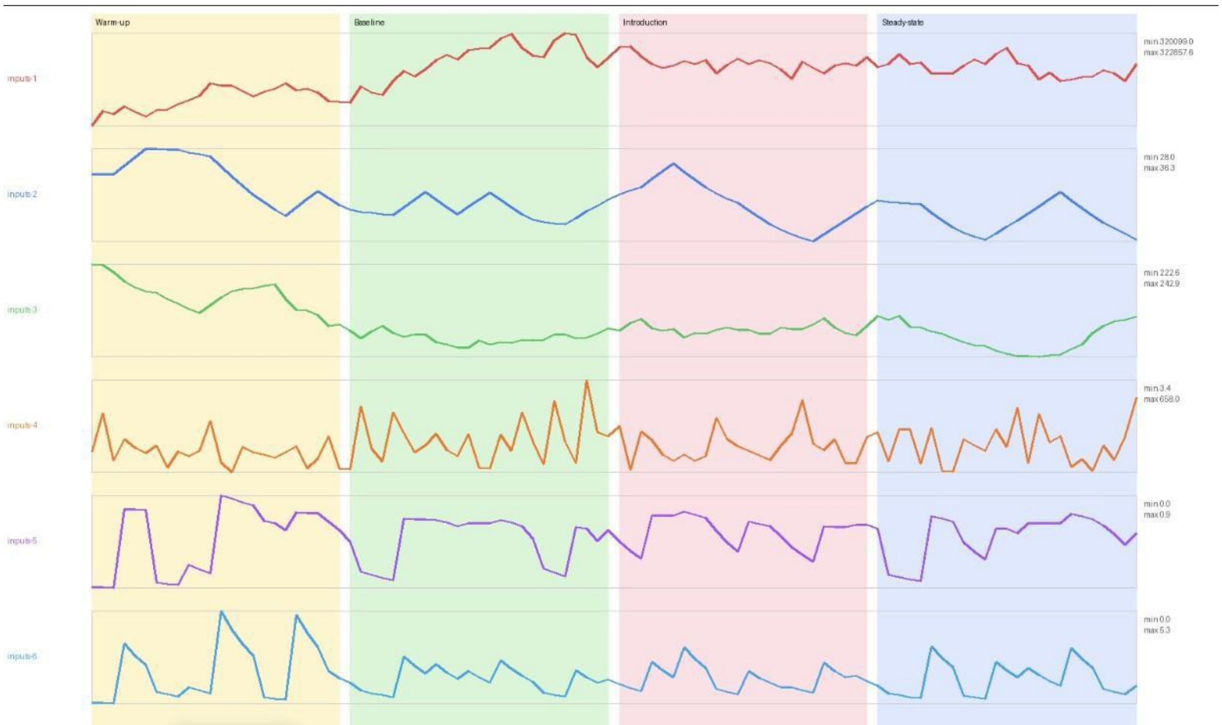

Figure 6. Time-series traces of six sensor features during one recording session, showing four stages of odour exposure: Warm-up, Baseline, Introduction, and Steady-state.
(Auto-generated visualisation based on smell_dataset.arff.)

Figure 6 illustrates these four stages using a representative recording session.To further validate the discriminative structure of the steady-state region, Figure 7 shows a PCA projection of the final samples from each recording. The visible separation between

PCA Analysis:

To validate the discriminative structure of steady-state region, PCA projection shows visible separation between odour clusters, indicating six-dimensional features are sufficiently structured for lightweight classifiers.

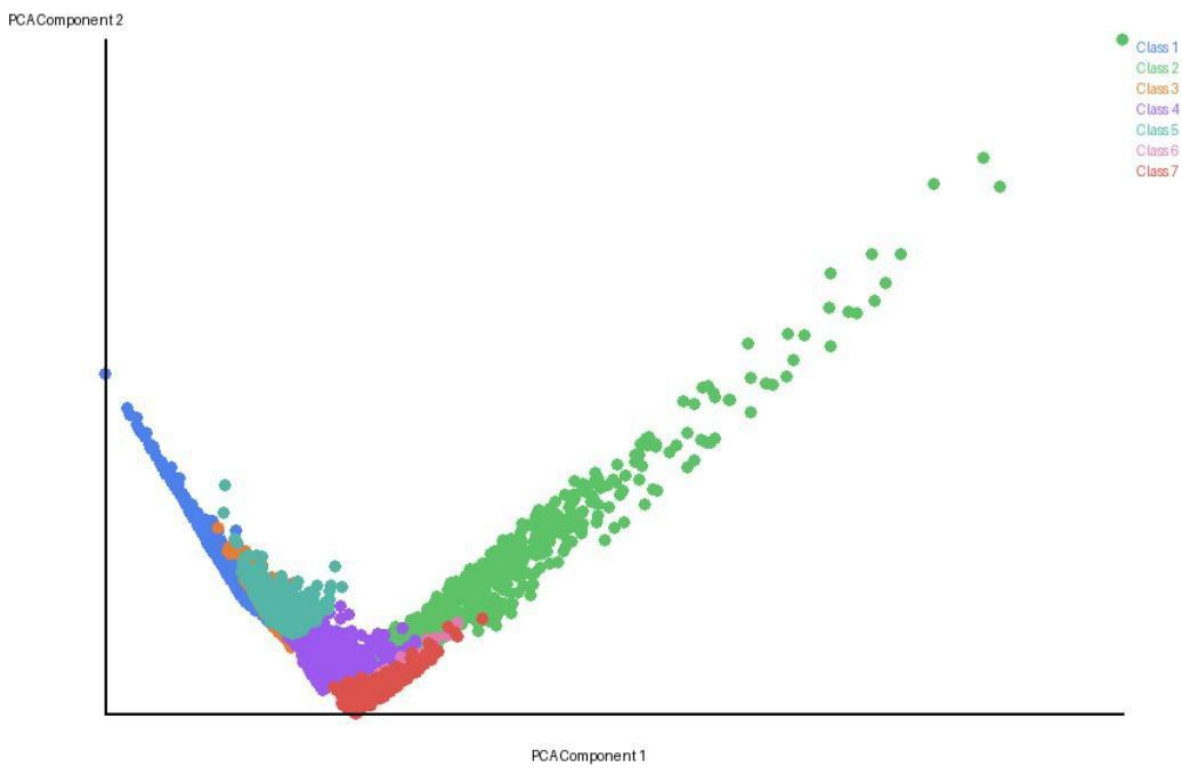

Figure 7. PCA projection of steady-state samples across all odour classes, showing distinct clusters that represent odour-specific sensor fingerprints. (Auto-generated visualisation

5.2.2 Basic Generative Outputs

The basic prototype connects KNN odour classifier with locally executed Stable Diffusion Turbo model, enabling real-time translation from recognized odour categories into visual imagery.

Prompt Mapping:

Each numeric class (1-6) is mapped to full-sentence descriptive prompt defined in smell_prompts_en.json:

1. **Air (Class 1)** → "A surreal scene where the wind becomes visible..."
2. **Lemon (Class 2)** → "Vivid macro photography of a halved lemon..."
3. **Coffee (Class 3)** → "Steaming cup of rich black coffee..."
4. **Mint (Class 5)** → "Macro photography of fresh mint leaves..."

Generation Characteristics:

Once KNN identifies odour class from incoming six-dimensional feature frame, macOS host triggers Stable Diffusion Turbo. Despite minimal configuration, system consistently produces images corresponding to intended semantic category:

- Citrus-toned compositions for lemon
- Deep browns and rising vapor for coffee
- Green botanical textures for mint
- Airy, translucent visuals for clean air

While absence of stylistic constraints results in natural variation between generations, outputs remain semantically stable, confirming that lightweight KNN classifier trained on static six-dimensional features is sufficient to drive meaningful odour-to-image translation.

5.3 Behaviour of the Classification Model in Practice

This section examines how KNN-based odour classifier behaves during real-time operation, focusing on stability, error patterns, and latency.

5.3.1 Stability Across Real-time Conditions

The classifier demonstrates high stability once sensor reaches baseline equilibrium:

- Six engineered features stabilize within narrow value bands after warm-up
- KNN predictions become steady, producing nearly identical labels over consecutive frames
- Odour transitions generate smooth label shifts rather than rapid oscillations

This behavior reflects both smoothed feature design (EMA + Δ features) and physical nature of odour changes occurring over several seconds.

5.3.2 Error Patterns

Principal error patterns observed across evaluation outputs:

(A) Inter-class Overlap and Confusion

CNN Model:

- Dominant confusion: Coffee - Mint (similar short-range VOC dynamics)
- Mint occasionally misclassified as Air in low-intensity segments
- Whiskey (18 windows) shows low recall (0.56), sometimes confused with Perfume or Smoke

RNN Model:

- Coffee: high precision but recall only 0.41 (predicted as Mint or Air)
- Mint: recall ~0.49 with symmetric confusion
- Whiskey: recall = 0 (long-range patterns not reliably learned)

Stable Classes:

- Perfume, Lemon, Smoke maintain consistently high precision and recall across both CNN and RNN
- Almost no confusion with other categories

(B) Sample Size and Class Imbalance

- Small classes (Whiskey: 18 windows) exhibit unstable performance, notably low recall

- Large classes (Air, Lemon, Perfume, Smoke) dominate metrics, produce very few errors

(C) "None" Windows

- Three "None" windows appear in sequence in the model confusion matrices
- Originate from sliding-window extraction before filtering "?" labels
- Included only for completeness, removed in deployment

(D) Temporal Resolution Differences

- **KNN:** Frame-level samples, 5-fold CV (k=3, accuracy 0.9424 ± 0.0036, 19,811 frames)
- **CNN/RNN:** 64-step windows with stride 16 (874 test windows)

Summary:

Errors predominantly arise from:

- Overlapping VOC dynamics between Coffee and Mint
- Severely underrepresented classes (Whiskey)
- Intrinsic limitations of small-sample temporal modeling

Core categories (Air, Lemon, Perfume, Smoke) remain highly stable across all models

5.3.3 End-to-End Latency Performance

Empirical real-time measurements show system achieves low end-to-end latency, enabling near-instantaneous visual feedback during interactive operation.

Latency Breakdown:

Pipeline Stage	Latency
Sensor sampling + feature computation (Raspberry Pi)	10-15 ms
OSC transmission to macOS host	<5 ms
KNN classification	<1 ms
Prompt assembly + Stable Diffusion Turbo inference	280-520 ms
Total end-to-end latency	**300-550 ms**

Table 5: End-to-end latency breakdown of smell-to-image pipeline

Interpretation:

Latency profile demonstrates

- Diffusion model accounts for >95% of total runtime (dominant bottleneck)
- KNN classifier contributes <0.5% of total runtime
- System is fully capable of interactive olfactory-visual feedback
- Sub-second response times are perceptually immediate for exhibition settings

Even at higher prompt complexity or moderate GPU load, system maintains sub-second response suitable for real-time interaction.

5.4 Real-Time Translation Studies

This section examines dynamic behavior during continuous, real-time operation: how classifier responds to odour transitions, how images evolve, and what strategies support coherent visual output.

5.4.1 Transition Behaviour

During real-time interaction, odour transitions unfold physically over several seconds. As scent source approaches or recedes from sensor, six engineered features shift gradually from one steady-state plateau to another.

Observable Transition Behaviors:

(1) Smooth Label Drift

- Predictions exhibit short periods of fluctuation when sensor is between plateaus
- Correspond to "Odour Introduction" and "Odour Dissipation" stages in time-series plots

(2) Short-lived Ambiguity

- Neighbouring odour categories (similar feature trajectories) may overlap

during transitions
- The system occasionally outputs alternating predictions before stabilizing

(3) Stable Plateau Recognition
- Once odour reaches a steady-state plateau, classifier locks onto single label
- Produces near-identical predictions across hundreds of frames

These behaviors ensure transitions appear natural rather than abrupt in final visual output.

5.4.2 Visual Coherence Strategies

Real-time diffusion-based image generation naturally introduces randomness. To reduce flicker and maintain coherence during odour transitions, system implements several lightweight stabilization strategies.

Implemented Strategies:

(1) Trigger-Window Stabilization
- New image generated only when classifier produces same label over fixed number of consecutive frames
- Default: 12 frames (configurable via environment variable)
- Implemented in _process_and_generate() method

(2) Deterministic Seeds for Style Consistency
- Configuration file assigns fixed random seeds to all odour classes (1-7) and None state
- Allows repeated generations from same class to maintain stylistic similarity

(3) Cooldown Interval to Prevent Over-generation
- Global cooldown timer prevents multiple images in rapid succession
- Reduces generation when sensor features oscillate near classification boundaries

(4) Soft Fade-in Rendering in UI
- Streamlit frontend displays new images using fade-in transition
- Improves perceived smoothness over abrupt replacement

Together, these strategies help prototype maintain coherent real-time visuals despite inherently stochastic nature of diffusion models.

5.4.3 Results

During informal real-time testing sessions using KNN-based bridge, system successfully completed full loop from odour detection to image generation.

Qualitative Observations:

(1) Correct Prompt Triggering

- When classifier output stabilized, system consistently triggered odour-specific prompts for Classes 1-6
- Class 7 or None correctly falls back to generic default prompt with a deterministic seed

(2) Stable Visual Output Under Persistent Odours

- During sensor reading plateaus, classification results remained stable
- Generated images appeared semantically consistent across repeated generations
- Enhanced by deterministic seeds applied to all classes

(3) Threshold Window Improves Smoothness

- N-frame trigger window reduced label flicker during gradual odour transitions
- Prevented rapid, unnecessary image updates

(4) Perceived Latency Suitable for Interaction

- Informal testing suggests generation speed sufficient for real-time interaction
- Bridge scripts record timestamps for future quantitative analysis

These qualitative observations confirm prototype is capable of real-time odour-to-image translation, forming solid foundation for future controlled experiments and quantitative evaluation.

5.5 Interactive Experience and Installation Setup

5.5.1 Physical Arrangement

Installation consists of three core physical components:

Sensing Unit (Raspberry Pi + BME688 + aux VOC sensors)

- Placed on a small pedestal or table
- Sensing module exposed to ambient airflow
- Users introduce odours by waving scented objects near sensor head

Processing Unit (macOS host)

- Positioned off to side
- Runs odour classification, prompt mapping, real-time image generation
- Remains invisible to audience during operation

Display Interface (Streamlit UI)

- Large monitor displays generated images and status indicators
- Images fade in smoothly to reinforce continuity during transitions

This spatial arrangement reinforces clear narrative: **physical odour → invisible computation → visible imagery**.

5.5.2 Audience Interaction Flow

Interaction is intentionally simple and requires no prior knowledge:

- Visitor picks up or approaches odour source
- Moves it toward sensing module
- System detects odour, stabilizes classification (trigger window)
- Selects the corresponding prompt
- Generates new image using RealVisXL with LCM acceleration
- Image appears with soft fade-in on display

Participants quickly understand mapping: **"my smell input → a visual atmosphere on screen"**.

5.5.3 Latency and Feedback

Although no formal latency measurement pipeline implemented, informal tests show system responds fast enough for interactive use:

- **Odour detection → prompt selection:** immediate (Python OSC processing)
- **Image generation time:** varies but generally perceived as responsive
- **Timestamp logging**: in bridge scripts for future quantitative analysis

UI provides lightweight feedback during processing:

- Current predicted class displayed
- Updated image when stability thresholds met

5.5.4 Emergent Behaviour

During public testing, several emergent behaviors appeared that were not explicitly designed but enriched installation experience:

- Visitors performed rhythmic "waving" gestures, trying to see how system reacts to different distances or intensities
- Some treated the sensor as "lens," exploring how slow or fast movements influence transitions
- Pairs of participants experimented collaboratively, combining two different odours to see which class dominates or whether system oscillates
- Unexpected borderline states (e.g., between mint and lemon) sometimes produced visually poetic transitions due to thresholding and fade-in
- These behaviors show installation invites playful exploration, extending beyond simple stimulus-response.

5.5.5 Documentation

Installation process documented through:

- classification_history.csv -- Automatically appended by both bridge scripts, logging timestamps, predicted labels, seeds, and generation durations
- Screen recordings of the Streamlit interface capturing transitions
- Photographs of physical interaction showing how visitors move odour sources
- Saved image outputs used for later analysis or archiving

Chapter 6: Evaluation and Discussion

6.1 Evaluation Methods

Evaluation was conducted through three complementary approaches:

Observation-based User Testing: Small groups of participants interacted freely with the installation while their behaviours, reactions, and difficulties were documented through qualitative field notes. This process allowed for the recording of emergent behaviours (Section 5.5.4).

System-level Performance Logging: Runtime outputs, timestamps, and classification histories—specifically data logged in classification_history.csv (Section 4.5.5)—were used to estimate responsiveness, system stability, and generation duration under real-world, interactive conditions.

Reflective Technical Analysis: Notes gathered during the development and deployment phases were reviewed to identify recurring issues, sensor behaviours (e.g., drift, environmental sensitivity), and model challenges within the low-latency pipeline.

6.2 Findings

The evaluation revealed several key interactive and performance behaviours:
Rapid Comprehensibility: Users quickly understood the input–output mapping ("smell in → image out") and the underlying olfactory-visual correspondence.

Engagement through Play: Participants demonstrated high engagement by experimenting with distance, movement, mixing odours, or rhythmic gestures, indicating that the system invites playful exploration.

Moderate System Latency: Image updates occurred with a perceptible but generally acceptable delay (Section 4.5.3), primarily attributable to the model size of RealVisXL despite MPS acceleration.

Classification Stability Varies: Strong, distinct odours (e.g., coffee) produced stable classification outputs; however, borderline odours (e.g., mint vs. lemon) frequently led to classification jitter and visual fluctuation.

Aesthetic Transitions: Users described the generated outputs as atmospheric and expressive, noting that the soft fade-in rendering (Section 4.4.2) during uncertain transitions created a visually poetic effect rather than a literal depiction of scent.

6.3 Reflections on System Performance and Experience

Overall, the system performs well as an exploratory interactive artwork. The mild unpredictability of classification, while technically undesirable, enhances the installation's sense of organic transformation and responsiveness to environmental subtlety. The slower generation speed, rather than limiting expressiveness, creates anticipation and rhythm within the interaction loop. However, long-term robustness remains limited due to inherent sensor drift. Furthermore, audience actions often fall outside of expected patterns (e.g., rapid, mixed inputs), suggesting that future iterations should incorporate more adaptive sensing and smoothing techniques to manage this increased entropy.

6.4 Interpretation of Findings

The findings indicate that olfactory input can function as a compelling creative signal. Critically, the machine understands smell not as a rigid chemical identity but as fluctuating patterns captured in multi-dimensional feature space. These patterns are then translated into visual atmospheres through generative processes. This approach reframes technical uncertainty—sensor noise, boundary cases, environmental drift—as not merely a limitation but a source of expressive richness and artistic complexity in the final output. The system validates the potential of Generative AI to map non-linear sensory data streams into aesthetically coherent visual forms.

Chapter 7: Discussion

7.1 Interpretation of Findings

The findings indicate that olfactory input can function as a compelling creative signal. Critically, the machine understands smell not as a rigid chemical identity but as fluctuating patterns captured in multi-dimensional feature space. These patterns are then translated into visual atmospheres through generative processes. This approach reframes technical uncertainty—sensor noise, boundary cases, environmental drift—as not merely a limitation but a source of expressive richness and artistic complexity in the final output. The system validates the potential of Generative AI to map non-linear sensory data streams into aesthetically coherent visual forms.

7.2 Artistic and Theoretical Implications

The installation contributes to the field of multisensory HCI art by challenging the dominance of vision and sound, positioning smell—typically marginal in digital media (Obrist et al., 2016)—as the primary driver of imagery. This shift facilitates a cross-modal translation that explores the sensory limits of machine perception.

Furthermore, the work reframes

human–machine relations: instead of machines merely extending human senses, humans provide ephemeral, chemical stimuli through which the machine constructs and mediates its own sensory-visual world. This paradigm opens new avenues for multisensory design, computational synaesthesia, and post-human aesthetics

7.3 Limitations and Challenges

The current implementation faces four primary technical and practical limitations that inform future research:

Sensor Instability: The BME688 is inherently sensitive to ambient environmental factors (airflow, humidity, temperature), leading to sensor drift that complicates long-term, stable classification.

Dataset Limitations: The ARFF datasets require manual, time-consuming collection and labeling using Wekinator, restricting the current diversity and scale of the odour classes. Model Latency: The necessity of using the high-quality RealVisXL V4.0 (via AutoPipelineForText2Image) results in a noticeable generation delay, even with MPS acceleration, preventing true instantaneous responsiveness.

Unpredictable User Behaviour: The system is not optimized for complex, multi-source inputs (e.g., rapid gestures, mixed odours), which often push the classifier into unstable, uncertain states that require significant stabilization logic.

Chapter 8: Conclusions and Future Work

8.1 Summary of Contributions

This work successfully demonstrates a functional, real-time pipeline that bridges olfactory sensing, machine learning classification, and visual generation. The core contribution is the reframing of smell as a primary creative input rather than an auxiliary effect, thereby expanding the vocabulary of multisensory interactive art. Methodologically, the project shows how practice-led experimentation can be used to reveal both the technical constraints and aesthetic potentials of cross-modal machine perception within a distributed, low-latency architecture.

8.2 Future Directions

Future iterations of the Scent as Perception system may explore the following directions to enhance both technical robustness and artistic scope:

High-Precision Sensing: Adopting higher-precision or multi-sensor E-nose arrays to improve feature resolution and classification stability.

Model Optimization: Employing fine-tuning or distillation techniques on the diffusion model to achieve stronger stylistic coherence and reduce generation latency further.

Complex Interaction: Developing logic to support multi-user or simultaneous multi-odour interactions, allowing for more complex collaborative input.
Multimodal Output: Integrating additional sensory outputs (e.g., sound, light, haptics) to create a richer, fully multisensory experience.

Robust Data Building: Initiating long-term, automated dataset building efforts to create a larger, more comprehensive olfactory dataset for robust odour recognition across varying environments.

References

1. Obrist, M., Velasco, C., Vi, C. T., Ranasinghe, N., Israr, A., Cheok, A. D., ... & Gopalakrishnakone, P. (2016, May). Touch, taste, & smell user interfaces: The future of multisensory HCI. In Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems (pp. 3285-3292).

2. Maggioni, E., Cobden, R., Dmitrenko, D., Hornbæk, K., & Obrist, M. (2020). SMELL SPACE: mapping out the olfactory design space for novel interactions. ACM Transactions on Computer-Human Interaction (TOCHI), 27(5), 1-26.

3. Vi, C. T., Ablart, D., Gatti, E., Velasco, C., & Obrist, M. (2017). Not just seeing, but also feeling art: Mid-air haptic experiences integrated in a multisensory art exhibition. International Journal of Human-Computer Studies, 108, 1-14.

4. Panagiotakopoulos, D., Marentakis, G., Metzitakos, R., Deliyannis, I., & Dedes, F. (2022). Digital scent technology: Toward the internet of senses and the metaverse. IT Professional, 24(3), 52-59.

5. Narumi, T., Nishizaka, S., Kajinami, T., Tanikawa, T., & Hirose, M. (2011, May). Augmented reality flavors: gustatory display based on edible marker and cross-modal interaction. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 93-102).

6. Olofsson, J. K., Niedenthal, S., Ehrndal, M., Zakrzewska, M., Wartel, A., & Larsson, M. (2017). Beyond smell-o-vision: Possibilities for smell-based digital media. Simulation & Gaming, 48(4), 455-479.

7. Brewster, S., McGookin, D., & Miller, C. (2006, April). Olfoto: designing a smell-based interaction. In Proceedings of the SIGCHI conference on Human Factors in computing systems (pp. 653-662).

8. Ghinea, G., & Ademoye, O. A. (2011). Olfaction-enhanced multimedia: perspectives and challenges. Multimedia Tools and Applications, 55(3), 601-626.

9. Ghinea, G., & Ademoye, O. (2012). The sweet smell of success: Enhancing multimedia applications with olfaction. ACM transactions on multimedia computing, communications, and applications (TOMM), 8(1), 1-17.

10. Murray, N., Lee, B., Qiao, Y., & Muntean, G. M. (2016). Olfaction-enhanced multimedia: A survey of application domains, displays, and research challenges. ACM Computing Surveys, 48(4), 1-34.

11. Flavián, C., Ibáñez-Sánchez, S., & Orús, C. (2021). The influence of scent on virtual reality experiences: The role of aroma-content congruence. Journal of Business Research, 123, 289-301.

12. Murray, N., Ademoye, O. A., Ghinea, G., & Muntean, G. M. (2017). A tutorial for olfaction-based multisensorial media application design and evaluation. ACM

Computing Surveys, 50(5), 1-30.

13. Kaye, J. J. (2004). Making scents: Aromatic output for HCI. interactions, 11(1), 48-61.

14. Kortum, P. (2008). HCI beyond the GUI: Design for haptic, speech, olfactory, and other nontraditional interfaces. Elsevier.

15. Obrist, M., Tuch, A. N., & Hornbaek, K. (2014, April). Opportunities for odor: experiences with smell and implications for technology. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2843-2852).

16. Ranasinghe, N., Jain, P., Thi Ngoc Tram, N., Koh, K. C. R., Tolley, D., Karwita, S., ... & Do, E. Y. L. (2018, April). Season traveller: Multisensory narration for enhancing the virtual reality experience. In Proceedings of the 2018 CHI conference on human factors in computing systems (pp. 1-13).

17. Amores, J., & Maes, P. (2017, May). Essence: Olfactory interfaces for unconscious influence of mood and cognitive performance. In Proceedings of the 2017 CHI conference on human factors in computing systems (pp. 28-34).

18. Yanagida, Y. (2012, October). A survey of olfactory displays: Making and delivering scents. In SENSORS, 2012 IEEE (pp. 1-4). IEEE.

19. Dobbelstein, D., Herrdum, S., & Rukzio, E. (2017, September). inScent: A wearable olfactory display as an amplification for mobile notifications. In Proceedings of the 2017 ACM International Symposium on Wearable Computers (pp. 130-137).

20. Dmitrenko, D., Maggioni, E., & Obrist, M. (2017, October). OSpace: towards a systematic exploration of olfactory interaction spaces. In Proceedings of the 2017 ACM international conference on interactive surfaces and spaces (pp. 171-180).

21. Lötsch, J., Kringel, D., & Hummel, T. (2019). Machine learning in human olfactory research. Chemical senses, 44(1), 11-22.

22. Ye, Z., Liu, Y., & Li, Q. (2021). Recent progress in smart electronic nose technologies enabled with machine learning methods. Sensors, 21(22), 7620.

23. Liu, T., Guo, L., Wang, M., Su, C., Wang, D., Dong, H., ... & Wu, W. (2023). Review on algorithm design in electronic noses: Challenges, status, and trends. Intelligent Computing, 2, 0012.

24. Piłat-Rożek, M., Łazuka, E., Majerek, D., Szeląg, B., Duda-Saternus, S., & Łagód, G. (2023). Application of machine learning methods for an analysis of e-nose multidimensional signals in wastewater treatment. Sensors, 23(1), 487.

25. Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, (4), 580-585.

26. Zhang, L., & Zhang, D. (2014). Domain adaptation extreme learning machines for drift compensation in E-nose systems. IEEE Transactions on instrumentation and measurement, 64(7), 1790-1801.

27. Zou, X., Wang, C., Luo, M., Ren, Q., Liu, Y., Zhang, S., ... & Su, S. W. (2022). Design of electronic nose detection system for apple quality grading based on computational fluid dynamics simulation and k-nearest neighbor support vector machine. Sensors, 22(8), 2997.

28. Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3), 1-27.

29. Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

30. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

31. Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. ISPRS journal of photogrammetry and remote sensing, 114, 24-31.

32. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

33. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

34. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

35. Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.

36. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

37. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

38. Medsker, L. R., & Jain, L. (2001). Recurrent neural networks. Design and applications, 5(64-67), 2.

39. Guo, J., Cheng, Y., Luo, D., Wong, K. Y., Hung, K., & Li, X. (2021). ODRP: A deep learning framework for odor descriptor rating prediction using electronic nose. IEEE Sensors Journal, 21(13), 15012-15021.

40. Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2024). Generative ai. Business & Information Systems Engineering, 66(1), 111-126.

41. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., ... & Yang, M. H. (2023). Diffusion models: A comprehensive survey of methods and applications. ACM Computing Surveys, 56(4), 1-39.

42. Wang, B., Chen, Q., & Wang, Z. (2025). Diffusion-based visual art creation: A survey and new perspectives. ACM Computing Surveys, 57(10), 1-37.

43. Canet Sola, M., & Guljajeva, V. (2024, December). Visions of Destruction: Exploring a Potential of Generative AI in Interactive Art. In Proceedings of the 17th International Symposium on Visual Information Communication and Interaction (pp. 1-8).. 1-8).

44. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, 6840-6851.

45. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10684-10695).

46. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PmLR.

47. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access, 6, 52138-52160.

48. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). Human-computer interaction. Addison-Wesley Longman Ltd..

49. Baidoo-Anu, D., & Ansah, L. O. (2023). Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. Journal of AI, 7(1), 52-62.

50. Brooks, J., Nagels, S., & Lopes, P. (2020, April). Trigeminal-based temperature illusions. In Proceedings of the 2020 CHI conference on human factors in computing systems (pp. 1-12).

51. Persaud, K., & Dodd, G. (1982). Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. Nature, 299(5881), 352-355.

52. Wessel, D., & Wright, M. (2002). Problems and prospects for intimate musical control of computers. Computer music journal, 26(3), 11-22.

53. Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. IEEE transactions on neural networks and learning systems, 29(5), 1774-1785.

54. Suzuki, M., Nakayama, K., & Matsuo, Y. (2016). Joint multimodal learning with deep generative models. arXiv preprint arXiv:1611.01891.