EFFICIENT DESIGN AND IMPLEMENTATION OF ELECTRONIC NOSE

Nir Katchinskiy, Greeshma Kayala, Katherine Liu, Daniela Villasmil, and Mrinal Mandal
Department of Electrical and Computer Engineering
University of Alberta, Edmonton, AB, CANADA
katchins@ualberta.ca, kayala@ualberta.ca, kliu3@ualberta.ca, dvillasm@ualberta.ca, mmandal@ualberta.ca

ABSTRACT
The Electronic Nose is a device used to distinguish between scents. The hardware portion of the electronic nose is comprised of an array of eight different gas sensors installed in an open chamber. Scents were characterized using different features extracted from sensor response curves, and then recognized by finding the minimum Euclidian distance in multi-space between the features. The final product is generic in its applications, compact, portable, adaptable to future development and user friendly. The resulting product has an operation of 2.5 minutes per sample and the total clearing time between samples is 1.75 minutes on average. The results show that the data recognition ability of the E-nose improved as the number of samples of a given scent increased. A recognition success rate of 95% was achieved.

KEY WORDS
Electronic Nose, Feature Extraction, Gas Sensors, Open System, Wireless, Portable, User Interface, Nearest Neighbour Classifier, Baseline Manipulation

1. Introduction
An Electronic Nose (E-nose) is an artificial device designed to identify and quantify changes in the environment caused by the presence of volatile organic molecules for which it has been trained. In the past few years, the system has been subject to extensive research and development to allow for its entry into different industries as the electronic nose has tremendous potential in the medical, entertainment, and public safety industries. Recently, it has been demonstrated that the electronic nose can be used for detailed breath analysis to diagnose cancer. The electronic nose has been shown to detect the presence of volatile organic compounds indicative of cancer in the breaths of lung cancer patients. Furthermore, it can be used by food inspection agencies to monitor product quality. It can also be used to improve public safety through the detection of potential hazards, drugs, or explosives in public places, such as airports.

Given the relevance of the topic to society, the objective of this work is to create a prototype of a generic electronic nose that is versatile and can be trained to recognize a wide range of odors.

Castro et al. [1] proposed an electronic nose based on Figaro sensors. Although the e-nose has a small implementation cost, it had some limitations in terms of operation speed, lack of compactness, and portability. The average operation time is between six to ten minutes for a single smell cycle [1]. Furthermore, the complete system consists of several linked sections arranged physically apart from each other such as sensor chamber, air pumps and valves, control board, sensor interface board and data acquisition board. All these elements on their own occupy a lot of space and limit the compactness of the system. Because of its size and the complexity of the connections involved, the overall system also lacks portability. Tang et al., 2010, implemented a portable, however not wireless, electronic nose using eight commercially available gas sensors from Figaro sensors, and were able to distinguish between three types of fruits. Their device operated at temperature between 24-28°C [2]. Song et al., 2011, implemented a wireless electronic nose capable of remotely detecting the combustible gases, methane and hydrogen [3].

In this paper, we propose an efficient E-nose with the following characteristics: fast operation time, generic in its applications, compact in size, easy to transport, adaptable to future applications, and user friendly interface. In the following sections: hardware implementation, software implementations, data recognition algorithm, and test results, are presented.

2. Proposed Electronic Nose
2.1 Hardware Implementation
The corresponding block diagram for the hardware portion of this project is presented in Figure 1 below. At the input end, there is an array of 8 gas sensors from Figaro Sensors, which will be stimulated by the presence of a smell sample. Two LM35 temperature sensors are also located at the input to provide temperature readings from inside the chamber during operation. These two arrays of inputs generate electrical signals that are read by an ATMEGA2560 microcontroller unit. The microcontroller unit acquires the information and sends it through serial communicate with the MATLAB user interface. Because the complete project involves a user interface, there is also a flow of information in the reverse direction. That is, the user...
is able to, through the user interface, control the MCU and determine when to start/stop the sensors, when to start/stop the heating module for the sensors and how to operate the suction fan circuitry.

![Block Diagram of E-nose Hardware Section](image)

In terms of hardware, the design approaches taken to achieve the project goals involved primarily the use of an open system as opposed to the closed system, used in [1]. The purpose of using an open system design was to achieve reduction in size of the overall product as well as minimize the distance travelled by the VOC molecules to reach the surface of the gas sensors. The reduction in size allows the implementation of a product that is compact and portable. Secondly, an array of 8 gas sensors was selected to detect a wide range of gas samples, and thus, satisfying generic applications. Furthermore, in the chamber design, the proximity of the sensor board to the sample along with the use of a suction fan allowed a reduction in operation speed per cycle. In terms of future adaptability, the E-nose can easily accommodate future developments. The sensor array and power boards can be replaced for more specific applications while keeping the same compact and portable chamber. Lastly, the E-nose offers a user-friendly interface through the combination of hardware and software design.

### 2.2 Gas Sensors

The sensor array represents a crucial section of the E-nose. At this stage, all the data regarding the gas sample is gathered to be processed and analyzed by the rest of the system. The types of sensors considered for the purpose of this project are Taguchi sensors from Figaro Engineering Inc. The motivation behind choosing this particular vendor stems from the fact that these sensors were previously used in references [1, 2]. This approach aimed to fulfill our goal of creating a generic product that is suitable for various applications. Another motivating point for choosing these sensors was their commercial availability at a reasonable price. The gas sensors are made of metal-oxide semiconductor material that offers good sensitivity and a long lifetime [4].

The principle of operation is highly dependent on the semiconductor properties of the sensors. As the metal oxide crystal, such as Silicon Oxide, is heated to high temperatures, the surface becomes negatively charged due to oxygen absorption on the crystal surface. Once this occurs, donor electrons within the crystal surface are transferred to the absorbed oxygen creating a layer of positive charges. The aforementioned phenomenon leads to the formation of a potential difference on the surface of the sensors that is later used for measurement purposes. The mechanism involved in the detection of different gases is based on the relationship between the resistance of the sensors and the concentration of a given gas as described by Equation 1 [5]. The output voltage of each sensor is recorded and plotted with respect to sampling time to obtain a sensor response curve.

\[
R_s = \frac{A(C)^\alpha}{\text{C}}
\]  

(1)

where \(R_s\) is the resistance of the sensor, \(C\) is the gas concentration, \(A\) is a constant, and \(\alpha\) is the slope of \(R_s\).

### 2.3 Temperature Sensors

Two temperature sensors also form part of the input arrays to the microcontroller unit. A 5V voltage regulator powers the IC sensor units. The purpose for including these devices was to monitor the internal temperatures of the E-nose during operation. The model chosen for the temperature sensors was LM35 and the reason for choosing this model was based on its adaptability to our circuit conditions and their reasonable price. These sensors can operate in a temperature range from \(-55^\circ\text{C}\) to \(+150^\circ\text{C}\); they are calibrated to measure temperature in Celsius and have an accuracy of 0.5°C which is sufficient for the purposes of monitoring.

### 2.4 Microcontroller Unit

The microcontroller unit selected for this project was ATMEGA 2560. The ATmega2560 is a low-power 10-bit CMOS microcontroller. The device achieves throughputs approaching 1 MIPS (Million instructions per second) allowing the optimization of power consumption versus processing speed [4]. The power requirement for the unit is 5V and it is driven by a 16MHz crystal clock signal. The microcontroller also has room for 16 10-bit analog to digital inputs which was an attractive feature since a total of 10 inputs (8 gas sensors and 2 temperature sensors) were required. If more sensors or more complexity is to be implemented in a future system, the same microcontroller can be used without need of replacement, as more pins are available on the microcontroller for use. Another attractive feature of this particular MCU is its adaptability for serial communication (USART), which allows the unit to communicate with a wireless device that will later send the information to the user interface.

### Table 1: Selected Gas Sensors for E-nose

<table>
<thead>
<tr>
<th>Sensor Model</th>
<th>Gas Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGS 825</td>
<td>Hydrogen Sulfide (H₂S)</td>
</tr>
<tr>
<td>TGS 826</td>
<td>Ammonia (NH₃)</td>
</tr>
<tr>
<td>TGS 880</td>
<td>Volatile vapors from food</td>
</tr>
<tr>
<td>TGS 2180</td>
<td>Water vapors from food</td>
</tr>
<tr>
<td>TGS 2600</td>
<td>Air Contaminants</td>
</tr>
<tr>
<td>TGS 2610</td>
<td>Hydrocarbon gases</td>
</tr>
<tr>
<td>TGS 2611</td>
<td>Methane</td>
</tr>
<tr>
<td>TGS 2620</td>
<td>Volatile Organic Vapors</td>
</tr>
</tbody>
</table>
2.5 X-Bee Pro Wireless Device

For the microcontroller unit to communicate wirelessly with the user interface, it is necessary to incorporate a wireless transceiver in the hardware. The device selected for this task was XBee Pro. This wireless module uses minimal power, 50mW, and provides reliable delivery of data between remote devices with a sensitivity of -102Dbm. It does, however, require a regulated supply voltage in the range from 3 to 3.3V [7]. The motivation behind choosing this particular model was based on previous experience with the device and its programming tools, its compatibility with the ATMEGA 2560 microcontroller unit in terms of USART serial communication, and the transmission range that it offers which ranges from 0m to around 90m. The device also offers a maximum RF Data rate of 250,000 bps, which is sufficient for the purpose of this project.

In the wireless transmission, each packet that is sent appears as “N10:1024E” approximately. Assuming this is sent, there are a total of 9 characters and each character has 16 bits. Combined with the fact that there are 10 sensors, including the temperature sensors, a total of 1440 bits per second are transmitted. Therefore, the wireless transceiver was set to operate at a baud rate of 9600 bps.

### Table 2: Electrical Characteristics of E-nose

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated Voltage</td>
<td>12V</td>
</tr>
<tr>
<td>Rated Current</td>
<td>2.5A</td>
</tr>
<tr>
<td>Operating Current (Smelling Mode)</td>
<td>1.12A</td>
</tr>
<tr>
<td>Operating Current (Clearing Mode)</td>
<td>0.29A</td>
</tr>
<tr>
<td>Max Power Consumption</td>
<td>13.8W</td>
</tr>
</tbody>
</table>

3. Software Implementation

The E-Nose operates in two modes, teaching and smelling. In teaching mode, the user introduces a new sample to the E-Nose and manually types the name of the sample. Once the smell cycle is complete, the data is added to the known samples database. In smelling mode, the user introduces an unknown sample to the E-Nose. Once the smell cycle ends, the data is processed and the sample name is determined through the data recognition algorithm.

3.1 User Interface

The user interface was developed using the MATLAB GUI tools. The user interface displays the sensor responses and temperature inside the E-nose as data is collected. Additionally, the UI allows the user to input sample name during a smell cycle, perform baseline manipulation, feature extraction, data recognition, data storage and retrieval. The UI can also be used to control the E-nose hardware, such as controlling the E-nose fan settings, sensor heating and power settings. Data acquisition was coded using the most basic MATLAB tools in order to allow users that do not have data acquisition tool box to use the E-Nose.

3.2 Data Storage and Retrieval

The sensor response collected from a scent sample will be used in either teaching or learning modes. If the E-Nose operates in teaching mode, the operations performed on the raw data include baseline manipulation and feature extraction. The new data will be saved in the database; data will be saved only when the E-nose is operating in the teaching mode. If the E-Nose operates in smelling mode, the operations performed on the raw data include baseline manipulation, feature extraction and data recognition. The raw data for each smell cycle is stored as a MATLAB storage file with the date, time and sample name in the filename. The raw data is stored in case the feature vectors are lost and needs to be regenerated. If future researchers decide to change the feature extraction or data recognition methods, there is no need to obtain new sensor response, as the raw data is already stored. In addition to the raw data, all feature vectors are stored in one MATLAB file that is automatically updated following each teaching cycle. The feature vectors are loaded to the computer memory as a variable matrix called MainFeature with a size of Nx137 once the UI is loaded. The first 136 columns contain the features for each sample, and the last column contains an ID number made from the date and time the sample was taken. The ID number is linked to the sample name through a mapping database called mappingDB. During data recognition, explained later, the ID number of the feature vector that corresponds to the unknown sample is found and used to print out the sample name.

3.3 Baseline Manipulation

Baseline manipulation of the sensor response curves is necessary to account for volatile organic compounds (VOCs) present in the air that might be altering the sensor response
to a given smell sample. Therefore, a sensor baseline response \( W_{e,s}(k) \) is taken without the scent sample under the E-nose for N number of samples \( (N_{BL}) \). The ideal \( N_{BL} \) was determined, through trial and error, to be sampled from 20-60 seconds of the sensor response. For each sensor \( W_{e,s}(k) \) is averaged over the \( N_{BL} \) samples \( \text{(shown in Equation 2)} \) to form the baseline sensor array average \( W_{BL,e,s} \) which will be used to perform the baseline manipulation. The baseline manipulation is performed using the difference method. This takes the difference between the sensor response \( W_{e,s}(k) \) of the scent sample and the baseline sensor array \( W_{BL,e,s} \) \( \text{(shown in Equation 3)} \) to acquire a pre-processed sensor response \( X_{e,s}(k) \). The sensor response to the sample corresponds to the sensor response curve after the E-Nose is exposed to the scent sample. The duration of the sensor response to the scent sample \( (W_{e,s}(k)) \) was determined to be 60-145 seconds from experimentation. Features can now be extracted from the pre-processed signal, \( X_{e,s}(k) \), and these features will be used to form a feature vector used to characterize a scent.

As discussed above, many methods exist for feature extraction; however, the feature extraction methods that are used in this project are the Steady-State Transient Derivative (STD) method and Ascending Transient Derivative (ATD). These methods were chosen to extract features due to their high success rate during the operation of the proposed Electronic Nose by R. Castro et al. [1]. STD and ATD are performed by first calculating the derivative between all consecutive samples in the ascending and steady-state regions of the sample response, depicted in Equation 7. Then, an average of the derivative over \( N_e \) which indicates number features per sensor, equal sized intervals is taken, where each interval is of equal \( k_i \) samples. Through experimentation it was determined that dividing the ascending and steady-state region into 17 equal intervals of 5 seconds each will result in optimal data recognition. This leads to each scent being represented by 136 features. A compromise between resolution and long computation time results from averaging the derivative every 5 seconds. The sensors used are very sensitive. On average, the ascending region which contains most of the sample information lasts 15 seconds. This allows for the extraction of 3 features per sensors or 24 features per scent. The features of the ascending region provide enough resolution to distinguish between scent samples. Each feature vector for a given smell is stored as a row in the “MainFeature” vector, while the last column of the “MainFeature” vector contains an ID number. This ID number is unique to the particular sample, date, and time it was sampled at.

### 3.5 Data Recognition

In the smelling mode operation of E-Nose, a data recognition method is necessary to compare the unknown feature vector to the database of known feature vectors to determine the identity of the unknown scent. The data recognition method used is called the nearest-neighbor classifier. The nearest-neighbor method is used because it was shown to have a high success rate during the operation of the proposed Electronic Nose by R. Castro et al. [1]. The nearest-neighbor classifier works by calculating the Euclidian distance in a multi-dimensional space between two vectors. Then, by finding the minimum Euclidian distance, the identity of the unknown smell sample is determined. The Euclidian distance is calculated using Equation 4 shown below. Once the minimum distance is found, the corresponding ID number of the sample is retrieved from the “MainFeature” vector and mapped to the sample name using the “mappingDB”, and printed to the user interface.

\[
d_{u,e} = \sqrt{\sum_{m=1}^{N_e} (Z(u,m) - Z(e,m))^2} \quad (4)
\]

where \( N_e \) is the number of samples, \( Z(u,m) \) is the unknown feature vector, \( Z(e,m) \) is the know feature vector, and \( d \) the Euclidean distance between the unknown feature vector to the known feature vector stored in the database. \( u,e \) is

### 3.6 Microcontroller Operations

The MCU operates in a cyclic mode where on each cycle a set of procedures are performed. In each cycle, the MCU first reads the XBee for instructions. Subsequently, the

---

**Figure 4**: The User interface with the sensor response curve and other controls
MCU compares the instruction string to a set of strings within the program in order to determine what operation to perform. The MCU first checks if the instruction is to “Turn ON FAN LOW”, and if the result is “yes”, the MCU sets the appropriate pins on high (5V) or low (0V). However, if the result is “no”, the MCU moves on to compare the next string in its code. In the same manner, the MCU compares the string received from the XBee to its programmed strings. When the instruction “Turn ON sensors” is sent to the MCU, the MCU sets the appropriate pins on high and also sets the variable ReadSensor to 1. When the instruction “Turn OFF sensors” is sent to the MCU, the MCU sets the appropriate pin on low and sets the variable ReadSensor to 0. ReadSensor variable is used in every MCU cycle. The MCU checks if ReadSensor =1, and if this is true the MCU takes a reading from all 10 sensors (8 gas and 2 temperature) and sends the data to XBee. The XBee sends the data to the XBee that is connected to the PC, and the sensor response data is then stored in the PC for analysis. At the end of each cycle, the MCU has a delay of one second before starting a new cycle; this is done to achieve a sample rate of one sample per second.

4. Performance Evaluation

4.1 Smell Cycle

In this project, a smell cycle represents the physical and computational procedures involved in the acquisition and storage of a scent sample. During a complete cycle, the user first clears the E-Nose by positioning it horizontally and turning on the fan at a high speed for 45 seconds. Once the E-Nose has been cleared, it is positioned upright, the fan is turned off, and the E-Nose sensors and heating elements are turned on. The user, then, monitors the internal temperature of the chamber, and once the temperature reaches 57°C the data acquisition process starts. For the first 60 seconds, only the environment is sampled. Once 60 seconds have been reached, the user positions the sample inside the E-Nose, and data is in acquisition for a period of 85 seconds until 145 seconds. Once 145 seconds have been reached, the user removes the sample from the E-Nose chamber and clears it for a new cycle.

4.2 Testing and Results

The E-nose was taught a total of 20 samples corresponding to different scent classes as listed in Table 3. The 20 scent samples are comprised of 6 different pairs of scents with each pair being similar, and 8 unique scents. During the teaching and recognition process, three main factors were held constant in the sampling process to ensure consistency of the results. Firstly, the physical amount of the smell sample, which translates to the concentration of the scent within the volume of the E-nose chamber, was kept constant. For example, when the nose was taught the smell of Tide detergent, 1 Tbsp of Tide was used for each scent sample response. Secondly, the operating temperature at which each smell cycle was started remained fixed at 57°C. The selected sensors are highly sensitive to temperature changes; therefore, keeping the starting temperature constant reduced the possibilities of response discrepancies from sample to sample. Thirdly, the clearing period used to remove the VOCs from the chamber was kept constant at 45 seconds.

Table 3: A List of 20 Scent Samples used in the Experiments

<table>
<thead>
<tr>
<th>SunType Grapefruit Juice</th>
<th>Very Ripe Banana</th>
<th>Chloe Perfume</th>
<th>Almond Extract</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cinnamon Extract</td>
<td>Green Banana</td>
<td>Listerine Original</td>
<td>Bounce Sheet</td>
</tr>
<tr>
<td>Tide Detergent</td>
<td>Feta Cheese</td>
<td>Nail Polish</td>
<td>Cilantro</td>
</tr>
<tr>
<td>Sunlight Detergent</td>
<td>Parmesan Cheese</td>
<td>Peanut Butter</td>
<td>Dried Red Chilies</td>
</tr>
<tr>
<td>Zesty Italian Dressing</td>
<td>Budlight Beer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

On average, each smell was taught 4-5 times to improve the recognition capabilities. Following the initial teaching round, a total of 89 samples were stored in the database. Three different experiments were carried out to evaluate the performance of this product:

A qualitative experiment was performed in order to test its ability to distinguish between different scent samples. The experiment tested a list of 20 randomly generated samples using Microsoft Excel.

A quantitative experiment was performed to determine its ability to detect differences in the concentration of the sample used. The only sample tested for this experiment was nail polish. The E-nose was able to successfully learn and recognize the difference between one and three brush swipes of nail polish.

A maturity experiment was performed to test its ability to differentiate sample maturity. The tested samples were green (unripe) banana, and dark (very ripe) banana. The E-nose was able to successfully distinguish between the two types of bananas.

The results obtained for the three experiments described above are presented as percentage success rate and are listed in Table 4.

Table 4: Table of Results from the Three Types of Experiments

<table>
<thead>
<tr>
<th>Type of Experiment</th>
<th>Success Rate</th>
<th># of Samples in Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualitative Experiment #1</td>
<td>75%</td>
<td>125</td>
</tr>
<tr>
<td>Qualitative Experiment #2</td>
<td>65%</td>
<td>145</td>
</tr>
<tr>
<td>Qualitative Experiment #3</td>
<td>95%</td>
<td>165</td>
</tr>
<tr>
<td>Quantitative Experiment</td>
<td>100%</td>
<td>24</td>
</tr>
<tr>
<td>Maturity Experiment</td>
<td>100%</td>
<td>10</td>
</tr>
</tbody>
</table>

5. Conclusion

In conclusion, the proposed Electronic Nose successfully meets our design goal, for it has the following
characteristics: generic in its applications, compact, portable, adaptable for future improvements, user friendly, and has an operation time faster than 6 minutes. The resulting product has an operation time of 2.5 minutes per sample and the total clearing time between samples is 1.75 minutes on average. After completing qualitative, quantitative and maturity experiments, it can be concluded that the data recognition ability of the E-nose improved as the number of samples of a given scent increased after each testing cycle. Furthermore, for the nail polish sample tested, the amount of the scent sample could be successfully distinguished. This result indicates that the recognition method can be used to quantify the scent sample. In addition, the E-nose succeeded in recognizing item maturity with samples of unripe and very ripe banana. This result could be helpful in detecting how old an unknown smell sample is.

Acknowledgement

The authors would like to thank Loren Wyard-Scott, and Dr. Mojgan Daneshmand, University of Alberta, for their help in the implementation of the E-nose hardware.

References