IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 11, No. 1, January, 2024, pp. 263-272

Journal homepage: http://iieta.org/journals/mmep

Ululate: A Non-Intrusive, Wearable Tongue Gesture Detection System for Human-Computer Interaction

Dhuha F. Jasim*, Waleed F. Shareef

Department of Control and Systems Engineering, University of Technology-Iraq, Baghdad 10069, Iraq

Corresponding Author Email: cse.21.13@grad.uotechnology.edu.iq

https://doi.org/10.18280/mmep.110129

Received: 20 May 2023
Revised: 15 July 2023
Accepted: 24 August 2023
Available online: 30 January 2024

Keywords:
human-computer interaction, microgestures, tongue gestures, wearable computing, hands-free computer interaction, non-intrusive human-computer interaction, wearable human-computer interaction

ABSTRACT

Human-computer interaction (HCI) focuses on improving the user's interaction with the computer. HCI enhances user experience in a wide range of applications, such as medical, security, autonomous vehicles, and wearable smart devices. While several systems have already developed tongue-based HCI that aim to be used as an input device, the majority require tongue piercing, dental retainers, and multiple electrodes on the chin, in the mouth or ears. These approaches are generally unhygienic, intrusive, and unsuitable to use in public areas esthetically. In this study, we designed Ululate, a hygienic, unobtrusive, and non-intrusive tongue gesture detection system that detects tongue movement by measuring vibration on the neck. The proposed system uses a sensing unit (accelerometer) that can be positioned below the lower jaw on the Genioglossus muscle. Hence, it does not require any in-mouth installation. Classification is conducted using four types of supervised machine learning algorithms, namely K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Tree, and Random Forest, then the accuracy of each algorithm using five different accuracy matrices is compared. The initial result of tongue gestures demonstrates that random forest shows the highest accuracy (97%). The overall designed system is lightweight, low profile, and low cost, which makes it efficient for everyday use.

1. INTRODUCTION

Human-computer interaction is the study of the interface between humans and computers. The HCI system integrates various fields of study, such as computer science, human factors, psychology, and ergonomics. Furthermore, HCI deals with questions of human perception, intelligence, decision-making, and interactive techniques of visualization, so it centers mainly on supervised methods.

With the improvement and exploitation of the sensor technology, it is important to improve micro-interaction with wearable devices because its applications may affect the user experience, whether it is enabling computers to be accessed by individuals with quadriplegia or just being used on a regular basis.

Due to modern technology, a growing interest in human-computer interaction has been observed in the last few years. HCI is attracting significant interest due to its importance in improving the user experience in a wide range of domains. HCI applications can vary from medical applications to wearable devices, security, and automatic vehicles. In medical applications, using an advanced HCI system as an input device is vital for a wide range of users, including quadriplegic patients and those with mutism making it easier for them to engage with smart devices. Furthermore, stress detection, where HCI systems show an improvement in detecting human stress levels using ECG signals, may appear to be a challenging task. Still, with the development of wearable devices and the improvement in sensor technology, it is now possible to detect stress levels using ECG signals where the sensors are located on the wrist that can be detected, identified, and diagnosed. In addition to measuring stress, those systems can also detect fatigue of construction workers, drivers and measure glucose levels in the blood of patients with Diabetes [1-12]. Understanding human-computer interaction is critical for creating positive user experiences. Mental health specialists from all disciplines are increasingly creating and implementing Internet-based therapies for persons suffering from a variety of mental illnesses. Technology-enabled therapies for a variety of mental health conditions have several therapeutic and economic benefits. Despite that, the influence of HCI and associated design aspects on patient safety, efficacy, and treatment adherence for computer users who engage in e-mental health therapies still need to be discovered [13].

While most electronic courseware enables users to progress at their own rate, courseware designers frequently make two assumptions. Initially, all users are presumed to be able to synthesize graphical material with their existing experience and knowledge to facilitate learning without considering diverse cognitive types. Recognizing student characteristics becomes crucial in broadening access to electronic information. Secondly, learning is taken for granted rather than validated. Therefore, HCI systems employ data analysis techniques to differentiate between what individuals comprehend and what they do not. E-Learning websites could utilize the HCI system to improve the learning experience by collecting and analyzing user data when they use learning websites [14-17]. The security research community acknowledges user behavior as a significant factor in various security vulnerabilities. Characterizing people as the "weakest link in the security chain" has gained popularity. Designers must identify and tackle the root causes of undesired user behavior to develop effective security systems. Researchers can draw upon knowledge and techniques from human-computer interaction to proactively address and resolve these issues [18-22].

Hands are required in today's interface, which causes limitations when running or walking. Voice- based HCI might solve these problems, but it creates other issues. For example, using voice- based HCI in crowded areas is inappropriate for privacy reasons. Furthermore, voice-based HCI is inefficient in noisy places due to cross-talk and background noise. Autonomous systems powered by artificial intelligence (AI) are progressively infiltrating people's daily lives and work. Autonomous cars are a prominent category. In the last few years, researchers have shown much interest in researching smart cars. Integrating visual human- computer interactions (VHCIs) into intelligent vehicle systems is critical for closing the gap between autonomous and man-machine control systems. Concerned about autonomous systems' safety, researchers have emphasized that HCI systems for automatic cars must prioritize human factors in design for this new technology class. They suggest that society is undergoing considerable transformation due to the emergence of AI-based autonomous systems that differ from traditional automation. There needs to be more clarity between automation and AI-based autonomy, which may lead to unrealistic expectations and the misuse of technology [21-23].

The daily maintenance and repair of a specific type of equipment are often hampered by long fault diagnosis times and untimely repairs. By using HCI systems, analog electronic circuits and equipment can be diagnosed and identified for faults, preventing accidents from occurring [24, 25].

The main principle of creating an HCI system is to create a functioning system that considers various HCI system design concerns and solves them in a way that does not impair system efficiency while improving user experience. We may divide HCI system design limitations into usability and user experience, with usability constraints being any component that influences system function, such as efficiency, safety, and utility. On the other hand, user experience limitations relate to how users interact with the HCI system, so increasing system complexity might negatively impact the user experience.

The interest in HCI is focused on solving issues in various design aspects. The design of an HCI system must consider multiple factors that improve the user experience, such as privacy, aesthetics, hygiene, and system complexity.

Bulky gadgets on the hand or head are unsightly and obstruct daily activities. In addition, obtrusive HCI devices can be unattractive, especially in public places. As a result, while building an HCI system, we must consider numerous aesthetic factors like device size, weight, positioning, and employing origin to control the device.

Voice assistants are digital helpers that can understand and respond to human speech using computer-generated voices. The most popular voice assistants are Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant, which can be found on smartphones or smart speakers. Users can speak to their assistants to ask questions, control home automation devices and media playback, and manage tasks like email, to-do lists, and calendars.

Voice assistants are handy tools but also have significant security risks. One of the most critical challenges is ensuring the security of the personal information stored on these devices. Since voice-activated devices can read calendar events, emails, and other personal information, anyone with access to them can quickly access this sensitive data. Additionally, voice assistants are vulnerable to different types of attacks. Researchers have shown that voice assistants can respond to inaudible commands delivered at ultrasonic frequencies. Attackers can use this technique to gain control of the device, and this type of attack could be broadcast over the airwaves in the future [26, 27]. Therefore, voice-based HCI in crowded areas is not recommended for privacy considerations. Furthermore, voice-based HCI is inefficient in loud environments due to cross-talk and background noise.

Human-Computer Interaction has been rapidly growing and has introduced a new area of research known as tongue-based HCI systems. These systems offer a potential input method for individuals who have limited or no mobility in their limbs. However, creating and deploying such systems has significant challenges that must be carefully addressed.

Choosing suitable sensors and location is crucial for accurately detecting tongue gestures in HCI systems. Many proposed input devices require invasive procedures like tongue piercing or multiple electrodes, which could be unsuitable for public use due to hygiene and aesthetic concerns.

Moreover, deploying in-mouth devices, such as magnetic implants, poses significant concerns regarding oral hygiene. In this context, a wearable sensor device on the Genioglossus muscle has been devised to address these issues. This innovative approach eliminates the need for regular cleaning and minimizes the requirement for additional assistance during implantation.

In response to these limitations, we have developed "Ululate," an advanced hands-free HCI system designed as an alternative input method. The name "Ululate" draws inspiration from the Arabic term "Ululation," which signifies a long, wavering vocal sound produced by rapid tongue movements. This term is often associated with celebratory expressions of joy in African and Middle Eastern cultures, typically observed during communal ritual events, such as weddings. The name "Ululate" conveys the resemblance of tongue movement without implying the accompanying vocal sound.

In this section, we briefly introduced most of the aspects of HCI design. The following sections are organized as Literature Review, Methodology, Results and Discussion, and Conclusions.

2. LITERATURE REVIEW

Previous studies have explored various modes of human-computer interaction through wearable technology. HCI systems can be categorized according to the body parts involved. In this study, focus will be placed on gestures using the tongue and teeth and facial gestures.

Cheng et al. [28] investigated a non-invasive tongue gesture input device using an array of textile pressure sensors mounted to the user's cheek. Using magnetic tongue piercing and tapping on the teeth, Kim et al. [29] created an HCI tongue-based system for tetraplegic individuals. Goel et al. [30] have built wireless, non-intrusive tongue gesture recognition using X- band Doppler. The interplay between the tongue and teeth is the foundation of a unique system that Nguyen et al. [31] have created. EEG/EMG sensors enable users to tap on their teeth and recognize various movements. A device designed by Vega Gálvez et al. [32] uses an accelerometer and gyroscope placed on the lower mastoid and mandibular condyle to detect the clicking of four different tooth groups. Based on physical deformations in the ear canal caused by facial muscle movements, Amesaka et al. explored it as a new HCI input method [33]. Matthies et al. [34] studied the same concept and compared the performance level of several electric sensing technologies, i.e., EMG, CS, EFS, EAR FS. and applied noise cancellation to detect only low frequency of ear canal deformation.

While previous research has introduced numerous techniques for wearable HCI systems as input devices for daily use, most of these solutions rely on devices worn inside or outside the mouth, leading to several issues. Intrusive designs involving tongue piercing and dental retainers raise concerns about hygiene and discomfort. Conversely, external mouth devices may be obtrusive and challenging to wear in public settings due to their visibility. Additionally, these devices are often uncomfortable, making them impractical for everyday use.

Given these limitations, our study presents Ululate. This system utilizes accelerometer data to recognize tongue movement and categorize it as gestures for subtle, non-intrusive, non-invasive, and hand-free HCI system.

3. METHODOLOGY

One of the primary goals when developing the Ululate system was to make it acceptable for everyday use. As a result, when creating Ululate, we examined each approach in terms of non-intrusive, unobtrusive, non-invasive, and other design aspects.

Selecting the best sensing unit is one of the most significant aspects since it imposes constraints on the system. When selecting sensors for the Ululate, we investigated many sensors and methodologies for tongue detection, including textile pressure, X-band Doppler, and EEG/EMG sensors. Furthermore, due to its size, weight, accuracy, and low cost, we determined that an accelerometer sensor was the best match for the system.

Examining the optimum position for the sensing unit is one of the most critical and challenging aspects of the design process since it is a trade-off between what the user prefers and what is optimal for the sensing position. As a result, the sensors were placed on the Genioglossus muscle for maximum accuracy.

The gesture is a form of body language communication that may be described with or without spoken words. The primary goal of gesture interaction in HCI is to develop systems that can recognize specific human gestures and utilize them to communicate information or control devices [35, 36]. As a result, while choosing the best tongue gestures, we assessed numerous factors, including the intricacy of the gestures, whether they were short and direct, and if they were better if fewer organs were employed. Furthermore, we employed several tongue movements gestures to investigate the concepts of tongue movement as an input device and determine which gestures would be best suitable for consumers. As a result, four distinct tongue gestures are proposed.

As a result, a sensing device equipped with an accelerometer was designed in order to construct an HCI system based on tongue gesture recognition. The sensor unit was then placed on one test participant's lower jaw on the Genioglossus muscle, and we performed several tongue movements gestures to evaluate the concepts of tongue movement as an input device and discover which gestures would be most appropriate for users. As a consequence, four distinct tongue motions are proposed. Finally, measurements are collected, evaluated, and fed into supervised machine-learning algorithms to classify gestures. Furthermore, we computed and compared system accuracy with four different machine-learning classifiers and accuracy matrices.

3.1 Ululate system design

Ululate system comprises a compact sensing unit connected via USB to the computer. The sensing unit includes a 29×58 mm ESP32 micro controller and a 3×5 mm ADXL345 accelerometer sensor positioned below the lower jaw on the Genioglossus muscle, as shown in Figure 1. The accelerometer data were collected and saved on the computer.

As the accelerometer chip in Figure 2 has a modest profile (less than 1 cm^2), it could be effortlessly included in a scarf or shirt collar since the proposed solution must be undetectable.

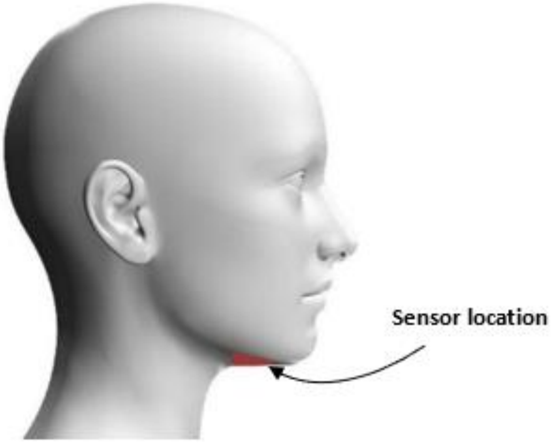


Figure 1. Sensing unit positioned on Genioglossus muscle

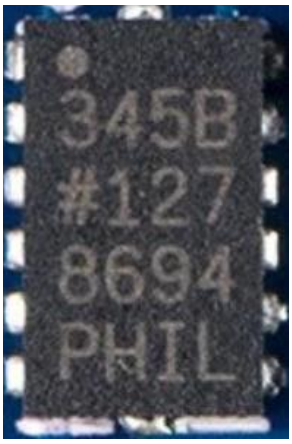


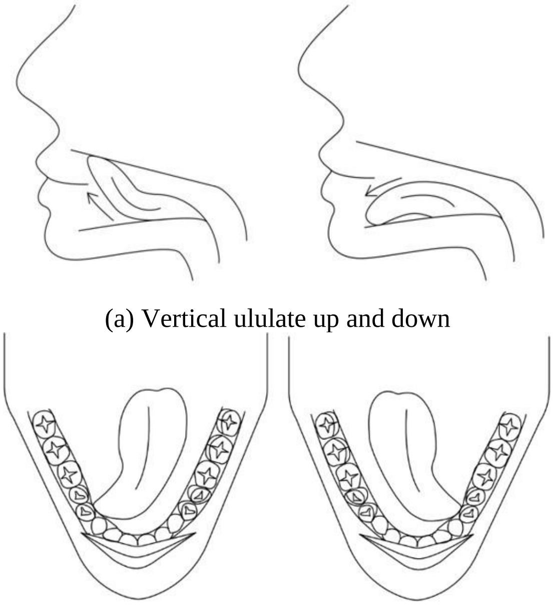
Figure 2. Accelerometer chip

3.2 Gesture selection

To design an intuitive HCI system, it’s essential to understand users’ preferences for these gestures, especially the if the study aims to create an everyday HCI system. Chen et al. [28] offered taxonomies to delineate the formation and application of micro-mouth gestures. They introduced a practical collection of 20 mouth micro-gestures, selected based on user preference, suitable for performing tasks in common software applications [35-37].

As the tongue-based HCI system is an unfamiliar mode of interaction, taking into consideration user preferences, the selected gestures should be short, direct, and batter if fewer organs are used. Furthermore, natural mouth and tongue gestures cannot be considered since their resemblance to everyday behavior may make them poor selections for gestures. Therefore, we propose four different gestures. The selected gestures are suitable for various users under different conditions since they are not complex and need just one moving organ, and because tongue movements work while the mouth is closed, they are more user-friendly.

Consequently, four tongue gestures were chosen: the first two are fast and slow vertical Ululate. During these two gestures, the tongue travels up and down, contacting the upper and lower jaws while the mouth is closed. The other two gestures are fast and slow Ululate, wherein these two gestures, the tongue moves left and right, contacting the inner side of the gums, as shown in Figure 3. Notice that the gestures do not require movement of the jaws or mouth opening, which is essential to make the proposed HCI system unobtrusive.



(a) Vertical ululate up and down

(b) Horizontal ululate left and right

Figure 3. Ululate gestures

3.3 Conducting experiments and feature selection

The experiment used data from one participant. The sensing unit was attached to the participant's Genioglossus muscle below the lower jaw using double-sided tape.

The sensing device was connected to the computer through a USB connection, and the experiment was conducted sitting, with participants requested to execute the specified gesture followed by no tongue movement.

In the experiment, a single test subject, a 23-year-old female, participated in the test phase. Prior to commencing the data collection, all gestures were thoroughly explained to the subject through both visual aids and in-person descriptions. The test subject was then instructed to practice the selected gestures during the initial phase of data collection.

Subsequently, the experimental process commenced, and each trial lasted for 80 seconds. Throughout the experiment, the sensor recorded 100 data points per second, resulting in approximately 8000 data records for each class of gestures.

In the feature selection process, we utilized raw coordinate data obtained from the sensor and further extracted additional properties, including mean, standard deviation, minimum, and maximum values. A time frame of ten consecutive data points was employed to capture temporal information to derive new characteristics. The resulting feature set comprises 12 distinct features with a corresponding label.

The chosen features are considered relevant for the time series classification process for several reasons. Firstly, extracting statistical properties such as mean, standard deviation, minimum, and maximum allows us to capture essential information about the distribution and variability of tongue movements, which could indicate different gestures. Secondly, the incorporation of temporal information by using a time frame of ten data points enables the model to consider the sequential patterns and dynamics in the tongue gestures, which can be crucial for accurate classification.

By including these specific features in our dataset, we aim to enhance the discriminatory power of the machine learning and deep learning models when differentiating between tongue-based HCI gestures. The relevance of these features lies in their ability to encapsulate essential characteristics of tongue movements, contributing to a more robust and accurate classification process.

3.4 Data classification

Classification is an essential step in putting the HCI Ululate system into action. Classification determines a new data class based on a training data set with known class membership. In this study, offline classification was done using four supervised machine learning algorithms, i.e., SVM, KNN, random forest, and Decision tree. Due to their proven effectiveness in previous studies, these machine-learning models have shown remarkable classification accuracy when utilized for the classification of tongue-based HCI systems. Next, we compared each algorithm's accuracy using five different accuracy matrices.

KNN is a supervised machine learning algorithm, i.e., it relies on labeled input data to learn a function and produces output according to the new unlabelled data. That is an instance-based learning method. It has been considered one of the simplest of all machine learning algorithms. The nearest Neighbour classification relies on two parameters, i.e., nearest distance metric (Euclidean Distance) and neighbors' number (k variable). Algorithm 1 shows the KNN classification pseudocode [38].

In a supervised machine learning algorithm, SVM is one of the most straightforward ways to classify data. Simply put, the main idea of classifying the data using SVM is to separate the two data sets with a hyperplane and maximize the margin between the two data classes. Algorithm 2 shows the SVM classification pseudocode [38].

Algorithm 1. KNN Algorithm

```
Step 1. Input:
Utilize a training dataset containing features X_train and corresponding labels y_train
Employ a test dataset with features X_test
Specify a value for K (number of neighbors to consider)
Step 2. For every test instance X_test_i in X_test:
Iterate over each training instance X_train_j in X_train:
Determine the distance between X_test_i and X_train_j using the Euclidean distance
Assign the computed distance to the training instance X_train_j
Step 3. Organize the training instances based on their distances in ascending order
Step 4. Choose the K nearest neighbors from the sorted training instances for each test instance X_test_i
Step 5. Tabulate the occurrences of each class label among the K nearest neighbors
Step 6. Designate the class label with the highest count as the predicted label for each test instance X_test_i
Step 7. Generate the predicted labels for the test instances
```

Algorithm 2. SVM algorithm

```
Step 1. Input:
  - Include a training dataset with features X_train and corresponding labels y_train
  - Incorporate a regularization parameter C
  - Integrate a kernel parameter gamma

Step 2. Compute the kernel matrix K based on the training data:
  - For each pair of training instances (X_train_i, X_train_j):
    - Calculate the Gaussian (RBF) kernel value K(X_train_i, X_train_j) = exp(-gamma × ||X_train_i - X_train_j||^2)

Step 3. Formulate the SVM optimization problem:
  - Initialize the weight vector w and bias b
  - Define the hinge loss function L(w, b) according to the SVM formulation
  - Define the regularization term R(w) in line with the SVM formulation
  - Define the objective function J(w, b) = L(w, b) + C × R(w)

Step 4. Solve the optimization problem to determine the optimal weight vector w and bias b:
  - Utilize an optimization algorithm (e.g., quadratic programming) to minimize J(w, b)
  - Iteratively update w and b until convergence

Step 5. Acquire the decision boundary and classify new instances:
  - For each test instance X_test:
    - Compute the decision function f(X_test) = sum(alpha_i × y_train_i × K(X_train_i, X_test)) + b
    - Assign the class label based on the sign of f(X_test)
      - If f(X_test) ⩾ 0, allocate the positive class label
      - If f(X_test) < 0, allocate the negative class label

Step 6. Generate the predicted labels for the test instances
```

Algorithm 3. Random forest algorithm

```
Step 1. Input:
  - Include a training dataset comprising features X_train and corresponding labels y_train
  - Specify the maximum depth of the decision tree as max_depth

Step 2. Establish a function for constructing a decision tree:
  - If the termination criteria are met:
    - Generate a leaf node and designate it the most frequent class label in the current subset of training instances
    - Alternatively:
    - Identify the optimal attribute for data split based on a criterion like information gain or Gini index
    - Create a new decision node for the selected attribute
    - Partition the training instances into subsets according to attribute values
    - Recursively invoke the function to construct a decision tree for each subset
    - Assign the decision nodes as children of the current node

Step 3. Assemble the decision tree using the training dataset and adhere to the maximum depth constraint:
  - Invoke the function outlined in step 2 to build the decision tree

Step 4. Formulate a function to classify new instances using the decision tree:
  - For each test instance X_test:
    - Commence at the root node of the decision tree
    - Traverse down the tree by assessing attribute conditions until reaching a leaf node
    - Designate the class label of the leaf node as the predicted label for the test instance

Step 5. Classify new instances using the decision tree constructed in step 3:
  - Utilize the function described in step 4 to classify new instances

Step 6. Produce the predicted labels for the test instances
```

Algorithm 4. Decision tree algorithm

```
Step 1. Input:
  - Incorporate a training dataset featuring features X_train and corresponding labels y_train
  - Specify the maximum depth of the decision tree, denoted as max_depth
Step 2. Establish a function for constructing a decision tree:
  - If the termination criteria are satisfied:
    - Formulate a leaf node and designate it the most frequent class label in the existing subset of training instances
    - Alternatively:
    - Identify the optimal attribute for splitting the data, based on a criterion such as information gain or Gini index
    - Generate a new decision node for the selected attribute
    - Partition the training instances into subsets based on the attribute values
    - Recursively invoke the function to construct a decision tree for each subset
    - Assign the decision nodes as children of the current node
Step 3. Assemble the decision tree using the training dataset and adhere to the maximum depth constraint:
  - Invoke the function outlined in step 2 to construct the decision tree
Step 4. Formulate a function to classify new instances using the decision tree:
  - For each test instance X_test:
    - Initiate at the root node of the decision tree
    - Traverse down the tree by assessing the attribute conditions until reaching a leaf node
    - Designate the class label of the leaf node as the predicted label for the test instance
Step 5. Employ the decision tree built in step 3 to classify new instances:
  - Utilize the function described in step 4 to classify new instances
Step 6. Produce the predicted labels for the test instances
```

Random forest is a supervised learning algorithm. It can be used for classification and is the most flexible and easy-to-use algorithm. Random forests build decision trees on randomly chosen data samples, obtaining predictions from each tree and selecting the optimal solution through voting. The process involves selecting random samples from a provided dataset and constructing a decision tree for each sampled subset.

Then, use each decision tree to generate a prediction result. Finally, choose the prediction result that receives the highest number of votes as the final prediction.

RF is composed of many binary decision trees and is used for diverse purposes such as regression, classification, and other applications via generating a large number of decision trees during the time of training [8]. Algorithm 3 [38] demonstrates the Random Forest classification pseudocode.

A Decision Tree is a supervised learning technique used for data categorization, and it has two nodes. Decision nodes are employed for decision-making, featuring multiple branches, while leaf nodes represent the outcomes of those decisions and lack additional branches. Algorithm 4 shows the Decision Tree classification pseudocode [38].

This research has analyzed different machine learning algorithms and their advantages and limitations. For instance, the K-Nearest Neighbors (KNN) algorithm works well in multi-class classification scenarios but needs help with datasets with many features due to dimensional complexity. On the other hand, Support Vector Machines (SVM) are proficient in handling linear and non-linear feature-to-target relationships using various kernel functions. However, selecting the proper kernel and fine-tuning the model parameters is complex. Decision trees are good at identifying non-linear patterns, but they can become unstable due to minor changes in the input data, resulting in significantly different tree structures. Conversely, Random Forests prevent overfitting issues by utilizing the power of multiple decision trees, but they can be biased towards dominant classes in unbalanced datasets. According to the results and discussion, the Random Forest algorithm has performed better in classification accuracy than other algorithms analyzed in this research.

4. RESULTS AND DISCUSSION

This section presents the results in two parts. First, Plot X coordinates in the time domain for each gesture to show the difference between gestures. Furthermore, we are showcasing the performed classification of the extracted features.

4.1 Time domain results

To clarify the difference between the gestures in the case of sitting, we can utilize the time domain analysis to identify the difference between the movements. Only one axis, the X-axis, was chosen to facilitate the procedure. Figure 4 shows the response difference in the time domain for different gestures, where we can distinguish between when the tongue moves and when it doesn't, as well as the distinction between each tongue gesture.

4.2 Classification

This section presents four distinct classification techniques for distinguishing tongue gestures. First, using a ten-value rolling window, we extracted four characteristics from each coordinate (mean, standard deviation (STD), minimum, and maximum). The newly retrieved characteristics were then utilized for training and evaluating four classification systems. The confusion matrix for the training, validation, and test phases employing the four classification methods, SVM, KNN, Decision Tree, and Random Forest, is shown in detail in Figure 5.

The results clearly illustrate each classification technique's successful discrimination of the four distinct gestures. However, a deeper analysis of the accuracy matrices and the performance of the different classification algorithms sheds light on their relative effectiveness and implications for this study.

Tables 1 to 4 present the accuracy rates achieved by each classification method, demonstrating that the Random Forest algorithm outperformed the other methods with an impressive accuracy of 97%. This high accuracy suggests that Random Forest excels in capturing complex patterns and relationships within the feature space, enabling it to make more accurate predictions than other classifiers.

On the other hand, while K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Decision Trees showed respectable accuracy rates, they did not match the performance of Random Forest. The observed differences in accuracy among these classifiers can be attributed to their respective strengths and weaknesses.

Table 1. Accuracy matrices for the KNN classification algorithm

Class	Precision	Recall	F1 score
Fast horizontal Ululate	0.95	0.94	0.94
Slow horizontal Ululate	0.96	0.97	0.97
Fast vatical Ululate	0.94	0.94	0.94
Slow vertical Ululate	0.94	0.93	0.94
Accuracy			0.95
Weighted Avg.	0.95	0.95	0.95

Table 2. Accuracy matrices for the SVM classification algorithm

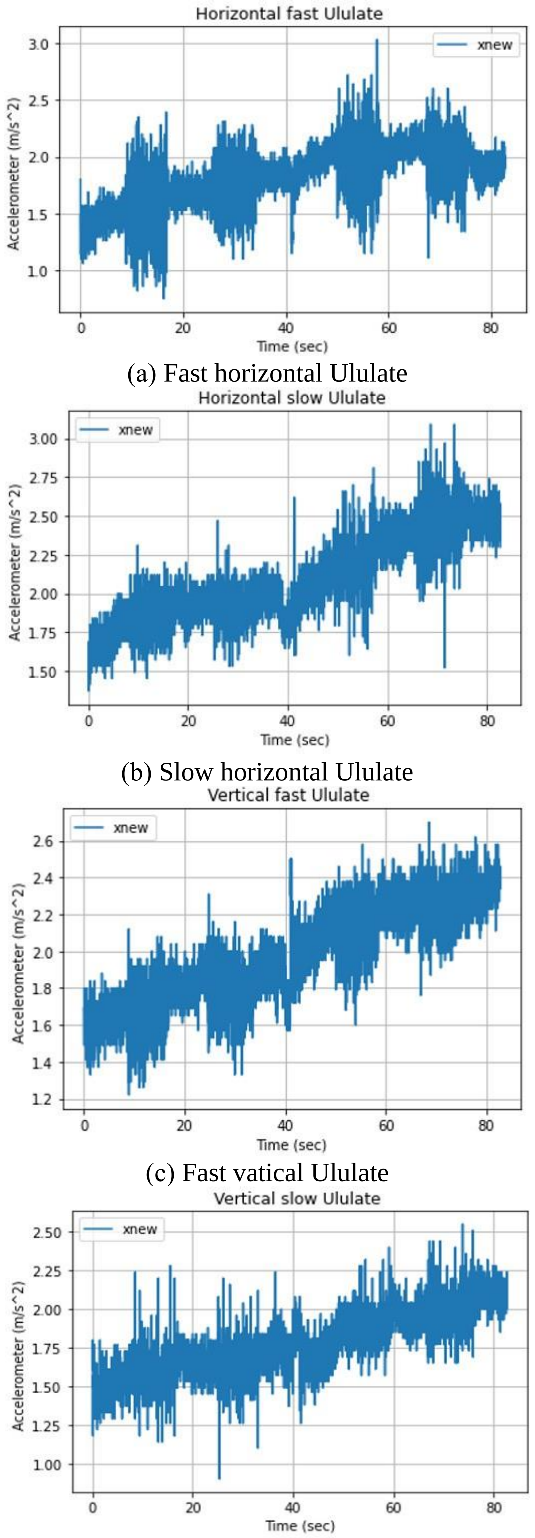
Class	Precision	Recall	F1 score
Fast horizontal Ululate	0.96	0.66	0.78
Slow horizontal Ululate	0.93	0.86	0.89
Fast vatical Ululate	0.79	0.85	0.82
Slow vertical Ululate	0.69	0.89	0.78
Accuracy			0.82
Weighted Avg	0.84	0.82	0.82

Table 3. Accuracy matrices for the Decision Tree classification algorithm

Class	Precision	Recall	F1 score
Fast horizontal Ululate	0.93	0.93	0.93
Slow horizontal Ululate	0.95	0.95	0.95
Fast vatical Ululate	0.92	0.91	0.91
Slow vertical Ululate	0.91	0.92	0.92
Accuracy			0.93
Weighted Avg.	0.93	0.93	0.93

Table 4. Accuracy matrices for the Random Forest classification algorithm

Class	Precision	Recall	F1 score
Fast horizontal Ululate	0.98	0.97	0.97
Slow horizontal Ululate	0.98	0.98	0.98
Fast vatical Ululate	0.96	0.96	0.96
Slow vertical Ululate	0.96	0.97	0.96
Accuracy			0.97
Weighted Avg.	0.97	0.97	0.97



(a) Fast horizontal Ululate

(b) Slow horizontal Ululate

(c) Fast vatical Ululate

(d) Slow vertical Ululate

Figure 4. X coordinate time response for sensing unit

K-Nearest Neighbors is known for its simplicity and ease of implementation, but it can be sensitive to noisy data and struggles with high-dimensional feature spaces. With its ability to handle linear and non-linear relationships through kernel functions, SVM is more flexible than KNN. However, selecting the appropriate kernel and tuning hyperparameters can significantly impact its performance. Decision Trees are proficient at capturing non-linear relationships. They are less sensitive to data noise but may create divergent tree structures when subjected to minor variations in the training data.

In contrast, Random Forest employs ensemble learning, aggregating multiple decision trees to reduce overfitting and enhance prediction accuracy. The robustness of this approach, coupled with its ability to handle diverse data characteristics, explains its superior performance in our study.

These findings have important implications for the development of tongue-based HCI systems. While Random Forest exhibited remarkable accuracy, its computational complexity should be considered for real-time applications. KNN and Decision Trees may be viable alternatives for scenarios where computational efficiency is crucial. Additionally, the choice of classifier may also depend on the specific requirements and constraints of the HCI system.

In conclusion, the results highlight the potential of different classification algorithms for tongue-based HCI systems. The superiority of Random Forest suggests its suitability for accurate gesture recognition. At the same time, the comparative analysis of other classifiers provides valuable insights for choosing appropriate models based on specific application scenarios. Further research can explore the combination of classifiers or hybrid approaches to achieve optimal performance and efficiency in real-world HCI implementations.

Algorithm 5 shows the Full feature extraction and data classification pseudocode.

Algorithm 5. Feature extraction and classification process

```
Input: Input Raw Dataset
Output: Four different accuracy matrices
Step 1: Input raw Dataset
Step 2: Time window feature extraction where T = 10 to create 12 new features from the original Dataset.
Step 3: Split Dataset into Training Dataset and Test Dataset.
Step 4: Use the Training Dataset to train each machine learning algorithm in Algorithms 1 to 4.
Step 5: Feed the Trained data and Test data to four different accuracy matrices.
```

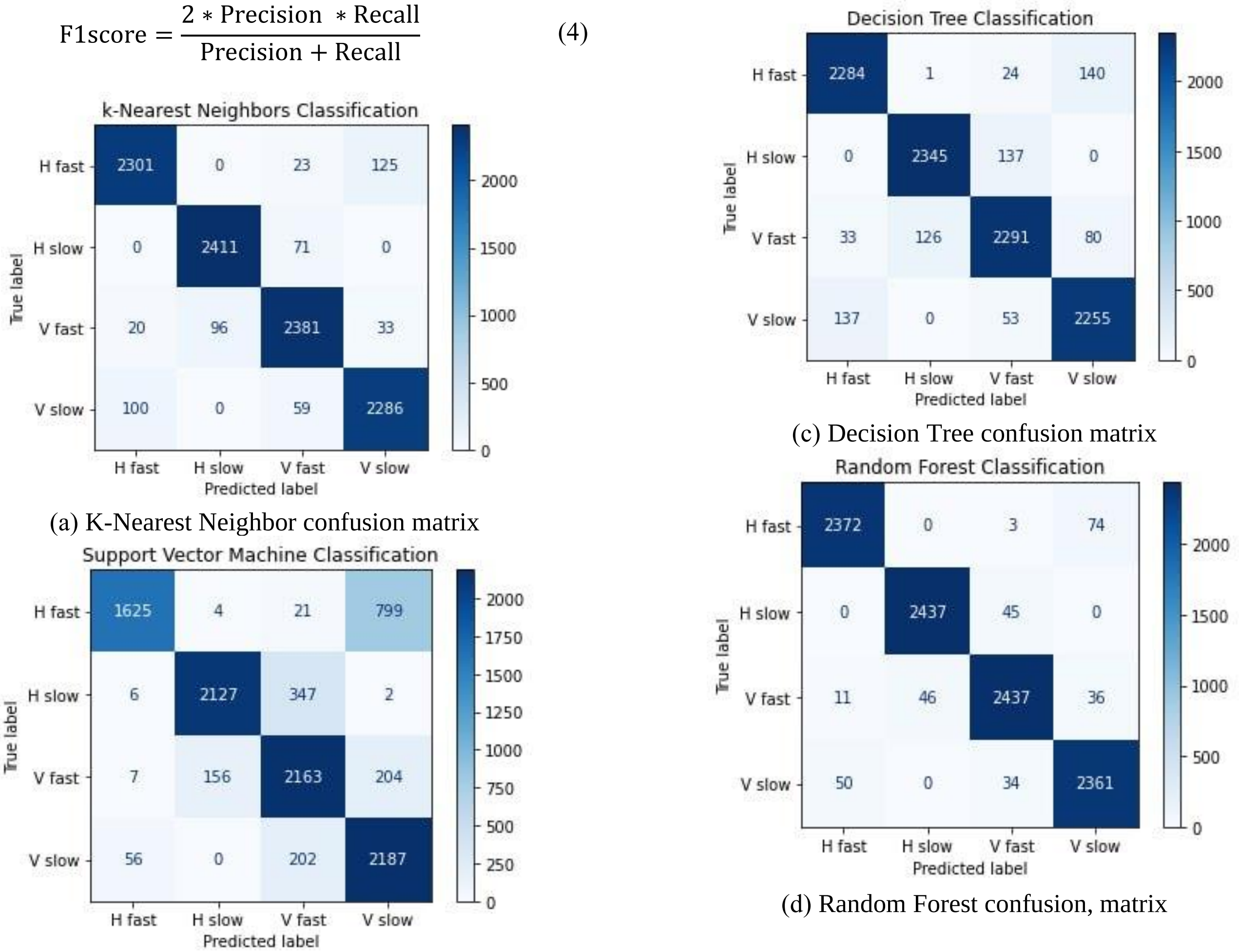
To understand the difference between accuracy matrices, we first need to understand the concept of the confusion matrix. The confusion matrix is a table that demonstrates how well the classification model predicts instances from different classes. The confusion matrix has two axes: one for the expected label and one for the actual label. When comparing multiple models, we could use the confusion matrix to see how well they predicted a true positive (TP) and true negative (TN). If one model predicts a TP and TN better than another, we use that model as our base model. There are four parameters in the confusion matrix: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The accuracy is calculated as the Eqs. (1)-(4). A comparison can be made between the proposed system and existing work in different categories, including sensor location and type, methodology used, system application, classification technique, and the level of intrusiveness, invasiveness, and obtrusiveness. The comparison is presented in Table 5.

$$\text{Accuracy} = \frac{\text{TP} + \text{T}N}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1score} = \frac{2 * \text{Precision} \ * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$



(a) K-Nearest Neighbor confusion matrix

(b) Support Vector Machine (SVM) confusion matrix

(c) Decision Tree confusion matrix

(d) Random Forest confusion, matrix

Figure 5. Training, validation, and test confusion matrices

Table 5. Comparison between the performance of the proposed system and the performance of the previous works

Author	Year	Used Sensor	Sensor Location	Method	Used Machine Learning	Intrusive	Invasive	Obtrusive	Accuracy
Cheng et al. [28]	2014	array of textile pressure sensors	attached to the user's cheek	Controlled through tongue	KNN	No	No	Yes	98%
Kim et al. [29]	2015	magnetic tongue piercing	magnetic tongue piercing	Controlled through tongue	KNN	No	No	Yes	95%
Goel et al. [30]	2015	X- band Doppler	Headset mounted on the ears attached to three Ex Doblar sensors	Facial gesture detection	SVM	No	No	Yes	94.30%
Matthies et al. [34]	2017	electric field sensing	the ear canal	Physical deformations in the ear canal caused by facial muscle movements	SVM	No	No	No	90.00%
Nguyen et al. [31]	2018	EEG/EMG	the back of the ear	Tap on their teeth	SVM	No	No	Yes	88.61%
Vega Gálvez et al. [32]	2019	accelerometer and gyroscope	on the lower mastoid touching constantly the mandibular condyle	Teeth-clicking	KNN	No	No	Yes	89%.
Amesaka et al. [33]	2019	electric field sensing	the ear canal	Physical deformations in the ear canal caused by facial muscle movements	Random Forest	No	No	No	90%
The proposed system (Ululate)	2023	An accelerometer	below the lower jaw on the Genioglossus muscle	Detects tongue movement by measuring vibration	Random Forest	No	No	No	97%

5. CONCLUSIONS

In this paper, we envision a future of hands-free computer interaction, where users can seamlessly interact with smart devices using a non-intrusive and non-invasive wearable system, facilitating everyday usage. Our study takes the first step in realizing this vision by developing the Ululate system, which is comfortably attached to the user's lower jaw on the Genioglossus muscle.

The design of Ululate prioritizes user comfort, minimizing obtrusiveness and social awkwardness. Ululate showcases its potential as a robust and reliable input device by successfully detecting four distinct, basic tongue gestures with an impressive 97% accuracy rate in our testing on one subject.

Moreover, the versatility of the Ululate system makes it accessible to various user groups, offering potential benefits to populations such as quadriplegic patients and individuals with mutism. Ululate can empower users with enhanced control and communication capabilities in settings where conventional devices may pose limitations, such as crowded and noisy environments.

Future work can address the limitations of our current study, such as testing only one subject and expanding the study to include a diverse group of subjects. Additionally, it is important to highlight that the information utilized for this research was collected in a supervised laboratory setup. In upcoming studies, it would be beneficial to tackle potential obstacles in real-life situations, such as disturbances caused by moving users.

REFERENCES

[1] Umer, W., Li, H., Yantao, Y., Antwi-Afari, M. F., Anwer, S., Luo, X. (2020). Physical exertion modeling for construction tasks using combined cardiorespiratory and thermoregulatory measures. Automation in Construction, 112: 103079. https://doi.org/10.1016/j.autcon.2020.103079

[2] Meen, T.H. (2019). Biomedical Engineering, Healthcare and Sustainability. In 2019 IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS), Okinawa, Japan. https://doi.org/10.1109/ECBIOS.2019.8807850

[3] Maman, Z.S., Chen, Y.J., Baghdadi, A., Lombardo, S., Cavuoto, L.A., Megahed, F.M. (2020). A data analytic framework for physical fatigue management using wearable sensors. Expert Systems with Applications, 155: 113405. https://doi.org/10.1016/j.eswa.2020.113405

[4] Chen, J., Wang, H., Hua, C. (2018). Electroencephalography based fatigue detection using a novel feature fusion and extreme learning machine. Cognitive Systems Research, 52: 715-728. https://doi.org/10.1016/j.cogsys.2018.08.018

[5] Keshan, N., Parimi, P.V., Bichindaritz, I. (2015). Machine learning for stress detection from ECG signals in automobile drivers. In 2015 IEEE International conference on big data (Big Data), Santa Clara, CA, USA, pp. 2661-2669. https://doi.org/10.1109/BigData.2015.7364066

[6] Nasser, A.R., Hasan, A.M., Humaidi, A.J., Alkhayyat, A., Alzubaidi, L., Fadhel, M.A., Santamaría, J., Duan, Y. (2021). IOT and cloud computing in health-care: A new wearable device and cloud-based deep learning algorithm for monitoring of diabetes. Electronics, 10(21): 2719. https://doi.org/10. 3390/electronics10212719

[7] Wali, S.S., Abdullah, M.N. (2021). Integrating wearable devices for intelligent health monitoring system. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 21(4): 1-14. https://doi.org/10.33103/uot.ijccce.21.4.1

[8] Nasser, A.R., Mahmood, A.M. (2021). Cloud-based Parkinson's disease diagnosis using machine learning. Mathematical Modelling of Engineering Problems, 8(6): 915-922. https://doi.org/10.18280/mmep.080610

[9] Croock, M.S. (2014). LTE based E-health monitoring system. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 14(2): 37-45.

[10] Mahmoud, A.G., Hasan, A.M., Hassan, N.M. (2021). Convolutional neural networks framework for human hand gesture recognition. Bulletin of Electrical Engineering and Informatics, 10(4): 2223-2230. https://doi.org/10.11591/ EEI.V10I4.2926

[11] Hamoodi, A.S. (2021). Logistic regression model to investigate the risk factors for glaucoma. Mathematical Modelling of Engineering Problems, 8(6): 881-887. https://doi.org/10.18280/ mmep.080606

[12] Ali, M.J., Ali, A.H., Mahmood, A.I. (2020). The design and simulation of FBG sensors for medical application. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 20(4): 1-8. https://doi.org/10.33103/uot.ijccce.20.4.1

[13] Søgaard Neilsen, A., Wilson, R.L. (2019). Combining e-mental health intervention development with human computer interaction (HCI) design to enhance technology-facilitated recovery for people with depression and/or anxiety conditions: An integrative literature review. International journal of mental health nursing, 28(1): 22-39. https://doi.org/10.1111/inm.12527

[14] Lin, F.R., Kao, C.M. (2018). Mental effort detection using EEG data in E-learning contexts. Computers & Education, 122: 63-79. https://doi.org/10.1016/j.compedu. 2018.03.020

[15] McKay, E. (2007). Planning effective HCI to enhance access to educational applications. Universal Access in the Information Society, 6: 77-85. https://doi.org/10.1007/s10209-007-0070-3

[16] Kolidakis, S.Z., Kotoula, K.M.A., Botzoris, G.N. (2022). School mode choice classification model exploitation though artificial intelligence classification application. Mathematical Modelling of Engineering Problems, 9(6): 1441-1450. https://doi.org/10.18280/mmep.090601 .

[17] Yamasari, Y., Qoiriah, A., Rochmawati, N., Yoshimoto, K., Ahmad, R.A., Putra, O.V. (2023). Detecting students' behavior on the E-learning system using SVM kernels-based ensemble learning algorithm. International Journal of Intelligent Engineering & Systems, 16(1): 142-153. https://doi.org/10.22266/ijies2023.0228.13.

[18] Amanuel, O., Alazzawi, Y. (2023). Design and implementation of EEG-based smart structure. International Journal of Intelligent Engineering & Systems, 16(1): 314-327. https://doi.org/10.22266/ijies2023.0228.28

[19] Mohamed, H., Hamza, A., Hefny, H. (2023). An efficient intrusion detection approach using ensemble deep learning models for IoT. International Journal of Intelligent Engineering & Systems, 16(1): 350-363. https://doi.org/10.22266/ijies2023.0228.31

[20] Sasse, M.A., Brostoff, S., Weirich, D. (2001). Transforming the 'weakest link'—A human/computer interaction approach to usable and effective security. BT Technology Journal, 19(3): 122-131. https://doi.org/10.1023/A:1011902718709

[21] Hussein, M.A., Hamza, E.K. (2022). Secure mechanism applied to big data for IIoT by using security event and information management system (SIEM). International Journal of Intelligent Engineering & Systems, 15(6): 667-681. https://doi.org/10.22266/ijies2022.1231.59

[22] Nasser, T.H., Hamza, E.K., Hasan, A.M. (2023). MOCAB/HEFT algorithm of multi radio wireless communication improved achievement assessment. Bulletin of Electrical Engineering and Informatics, 12(1): 224-231. https://doi.org/10.11591/eei.v12i1.4078

[23] Xu, W. (2020). From automation to autonomy and autonomous vehicles: Challenges and opportunities for human-computer interaction. Interactions, 28(1): 48-53. https://doi.org/10.1145/3434580

[24] Wang, X., Zheng, X., Chen, W., Wang, F.Y. (2020). Visual human–computer interactions for intelligent vehicles and intelligent transportation systems: The state of the art and future directions. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51(1): 253-265. https://doi.org/10.1109/TSMC. 2020.3040262

[25] Sadik Croock, M., Salman Mahmood, S. (2022). Management system of smart electric vehicles using software engineering model. International Journal of Electrical and Computer Engineering Systems, 13(5): 369-377. https://doi.org/10.32985/ijeces.13.5.5

[26] Nasser, A.R., Azar, A.T., Humaidi, A.J., Al-Mhdawi, A.K., Ibraheem, I.K. (2021). Intelligent fault detection and identification approach for analog electronic circuits based on fuzzy logic classifier. Electronics, 10(23): 2888. https://doi.org/10. 3390/electronics10232888

[27] Yong, W., Zhang, J., Chen, S., Yang, B., Zhang, J. (2020). Design of a fault diagnosis equipment. In IOP Conference Series: Earth and Environmental Science, 042070. https://doi.org/10.1088/1755-1315/440/4/042070

[28] Cheng, J., Okoso, A., Kunze, K., Henze, N., Schmidt, A., Lukowicz, P., Kise, K. (2014). On the tip of my tongue: A non-invasive pressure-based tongue interface. In Proceedings of the 5th Augmented Human International Conference, Kobe Japan, pp. 1-4. https://doi.org/10.1145/2582051.2582063

[29] Kim, J., Park, H., Bruce, J., Rowles, D., Holbrook, J., Nardone, B., West, D.P., Laumann, A., Roth, E.J., Ghovanloo, M. (2015). Assessment of the tongue-drive system using a computer, a smartphone, and a powered-wheelchair by people with tetraplegia. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 24(1): 68-78. https://doi.org/10.1109/TNSRE.2015.2405072

[30] Goel, M., Zhao, C., Vinisha, R., Patel, S.N. (2015). Tongue-in-cheek: Using wireless signals to enable non-intrusive and flexible facial gestures detection. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Republic of Korea, pp. 255-258. https://doi.org/10.1145/2702123.2702591

[31] Nguyen, P., Bui, N., Nguyen, A., Truong, H., Suresh, A., Whitlock, M., Pham, D., Dinh, T., Vu, T. (2018). Tyth-typing on your teeth: Tongue-teeth localization for human-computer interface. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, pp. 269-282. https://doi.org/10.1145/3210240.3210322

[32] Vega Gálvez, T., Sapkota, S., Dancu, A., Maes, P. (2019). Byte. it: Discreet teeth gestures for mobile device interaction. In Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland, UK, pp. 1-6. https://doi.org/10.1145/3290607.3312925

[33] Amesaka, T., Watanabe, H., Sugimoto, M. (2019). Facial expression recognition using ear canal transfer function. In Proceedings of the 2019 ACM International Symposium on Wearable Computers, London, United Kingdom, pp. 1-9. https://doi.org/10.1145/3341163.3347747

[34] Matthies, D.J., Strecker, B.A., Urban, B. (2017). Earfieldsensing: A novel in-ear electric field sensing to enrich wearable gesture input through facial expressions. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, Colorado, USA, pp. 1911-1922. https://doi.org/10.1145/3025453.3025692

[35] Hornbæk, K., Mottelson, A., Knibbe, J., Vogel, D. (2019). What do we mean by "interaction"? An analysis of 35 years of CHI. ACM Transactions on Computer-Human Interaction (TOCHI), 26(4): 1-30. https://doi.org/10.1145/3325285

[36] Satchell, C., Dourish, P. (2009). Beyond the user: use and non-use in HCI. In Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7, Melbourne Australia, pp. 9-16. https://doi.org/10.1145/1738826.1738829

[37] Chen, V., Xu, X., Li, R., Shi, Y., Patel, S., Wang, Y. (2021). Understanding the design space of mouth microgestures. In Designing Interactive Systems Conference 2021, USA, pp. 1068-1081. https://doi.org/10.1145/3461778.3462004

[38] Jasim, D.F., Shareef, W.F. (2023). Non-invasive tongue-based HCI system using deep learning for microgesture detection. Revue d'Intelligence Artificielle, 37(4): 985-995. https://doi.org/10.18280/ria.370420