MuteIt: Jaw Motion Based Unvoiced Command Recognition Using Earable

TANMAY SRIVASTAVA ⓘ, Stony Brook University, USA
PRERNA KHANNA ⓘ, Stony Brook University, USA
SHIJIA PAN ⓘ, University of California, Merced, USA
PHUC NGUYEN ⓘ, University of Texas at Arlington, USA
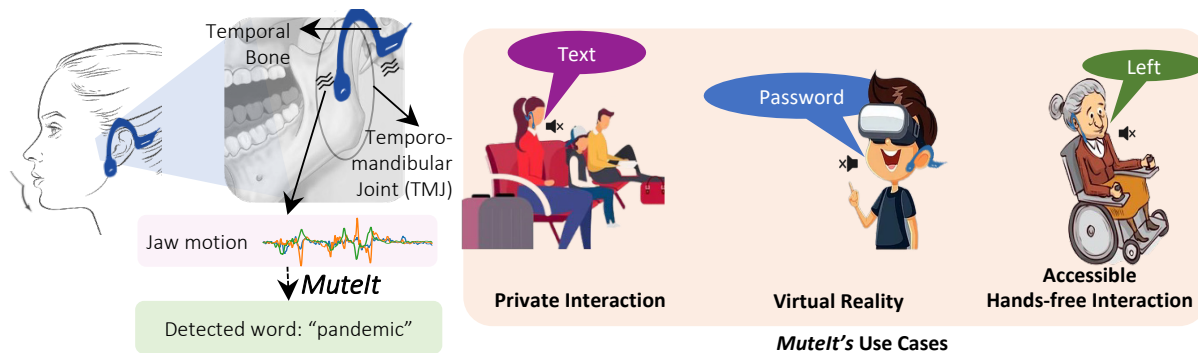SHUBHAM JAIN ⓘ, Stony Brook University, USA

Fig. 1. *MuteIt*: Concept and potential use cases

In this paper, we present *MuteIt*, an ear-worn system for recognizing unvoiced human commands. *MuteIt* presents an intuitive alternative to voice-based interactions that can be unreliable in noisy environments, disruptive to those around us, and compromise our privacy. We propose a twin-IMU set up to track the user's jaw motion and cancel motion artifacts caused by head and body movements. *MuteIt* processes jaw motion during word articulation to break each word signal into its constituent syllables, and further each syllable into phonemes (vowels, visemes, and plosives). Recognizing unvoiced commands by only tracking jaw motion is challenging. As a secondary articulator, jaw motion is not distinctive enough for unvoiced speech recognition. *MuteIt* combines IMU data with the anatomy of jaw movement as well as principles from linguistics, to model the task of word recognition as an estimation problem. Rather than employing machine learning to train a word classifier, we reconstruct each word as a sequence of phonemes using a bi-directional particle filter, enabling the system to be easily scaled to a large set of words. We validate *MuteIt* for 20 subjects with diverse speech accents to recognize 100 common command words. *MuteIt* achieves a mean word recognition accuracy of 94.8% in noise-free conditions. When compared with common

Authors' addresses: Tanmay Srivastava ⓘ, tsrivastava@cs.stonybrook.edu, Stony Brook University, New York, USA; Prerna Khanna ⓘ, pkhanna@cs.stonybrook.edu, Stony Brook University, New York, USA; Shijia Pan ⓘ, span24@ucmerced.edu, University of California, Merced, Merced, USA; Phuc Nguyen ⓘ, vp.nguyen@uta.edu, University of Texas at Arlington, Arlington, USA; Shubham Jain ⓘ, jain@cs.stonybrook.edu, Stony Brook University, New York, USA.

voice assistants, *MuteIt* outperforms them in noisy acoustic environments, achieving higher than 90% recognition accuracy. Even in the presence of motion artifacts, such as head movement, walking, and riding in a moving vehicle, *MuteIt* achieves mean word recognition accuracy of 91% over all scenarios.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**;

Additional Key Words and Phrases: Unvoiced Speech, IMU Sensing, Signal Processing

1 INTRODUCTION

The market for Internet of Things (IoT) devices has grown exponentially over the past decade, accounting for 12.3 billion devices by 2021 [4]. These devices offer a variety of interaction modes, ranging from touch to speech interactions [50]. However, with the emergence of speech recognition techniques, speech interactions have surpassed touch interactions owing to their ease of use, hands-free accessibility, as well as faster input (voice interaction is 3× faster than typing [74]). We use voice assistants on our phones and in our houses, querying for information ranging from weather to game scores and recipes. In fact, the speech and voice recognition market as a whole, which was valued at $6.9 billion in 2018, is projected to be worth $28.3 billion by the end of 2026 [24].

While intuitive and popular, voiced interactions can be inconvenient and unreliable in noisy environments, such as in a crowded gathering or with the TV playing in the background. It can also be a cause of disturbance to those around us, preventing us from using audible commands in the vicinity of someone studying or sleeping. Furthermore, audible speech interactions compromise our privacy when used in public places. More importantly, despite the ease of accessibility, voice-based systems do not serve people with speech disorders [38] or post-laryngectomy patients [44] who are unable to produce sound, even if their articulation skills (jaw and lip movements) are intact. Motivated by the limitations of voice-based interactions, we ask the following question: *"Can we develop a wearable system to recognize unvoiced (or inaudible) commands to enable voice-like intuitive interactions?"*

Unlike speech-based systems, using unvoiced speech is a novel hands-free interaction technique that is more convenient than touch gestures, while also being less disruptive to those around us. In fact, a recent study [66] shows that users prefer unvoiced speech over speech input, and are even willing to tolerate lower performance for maintaining their privacy. Research efforts on unvoiced speech recognition have focused on tracking primary articulators, such as lips or the vocal box, using vision [12, 30, 49, 65, 67, 82, 83], RFID [88], wireless [11, 47], or ultrasound [14, 17, 50, 78] sensing techniques. However, these techniques either capture the user's face [12, 67, 82], disregarding their privacy, or mount sensors on the tongue [33, 53, 75] or the face [32, 42, 59], making them obtrusive and socially awkward.

With the increasing social acceptance of headphones and earphones, ear-worn sensors (or *earables*) have emerged as a favorable sensing modality. Assistive interaction technologies involving earables, like teeth typing [61, 70] have been proposed, but they are not as intuitive and simple as language-based commands [42]. Recent developments around earables have shown that they can be used to capture jaw motion [5, 46]. In fact, *JawSense* [46] shows that skin deformities and muscle vibrations caused by unvoiced phoneme articulation can be captured using an IMU placed on the Temporomandibular Joint (TMJ). However, *JawSense* can only recognize individual sounds or *phonemes*, which is not sufficient for practical

MuteIt. In this paper, we present *MuteIt*, an earable system that recognizes unvoiced commands by tracking jaw motion. *MuteIt* is capable of recognizing entire words and is robust to body/head movements. In contrast to prior works [42, 73, 96] , *MuteIt* reconstructs a word from its components instead of training a word classifier,

enabling the system to easily scale to any word. *MuteIt* includes a pair of inertial measurement units (IMU), mounted on an off-the-shelf ear hook, to acquire the jaw motion effectively. One is placed behind the ear to be used as a reference sensor to capture head motion. The other is placed on the TMJ to track the user's jaw motion. Figure 1 demonstrates our system's concept and several use case scenarios for unvoiced command interactions. The twin-IMU setup can be retrofitted to most behind-the-ear or bone conduction earphones or even commonly used ear hooks used to hold ear-pods in place [20], enabling a socially acceptable form factor.

Challenges. We identify three research challenges for recognizing unvoiced command words using earables: (1) *Indirect speech sensing.* Speech production typically involves multiple articulators working together to produce sound. It is straightforward to interpret what a user is saying from primary articulators (such as lips and the vocal box), and many prior works have done so [88, 94], albeit using obtrusive [33, 76] or privacy-compromising techniques [67, 91]. The human jaw, on the other hand, is a secondary articulator with lower degrees of freedom compared to the lips, for example. As a result, jaw motion does not exhibit sufficiently distinctive movements for speech interpretation. As an exercise, imagine trying to interpret what a person is saying by looking at their lips versus looking at their jaw; unvoiced speech can be easily inferred by looking at the lips only. To the best of our knowledge, unvoiced command recognition by tracking a secondary articulator only is still an unsolved problem. (2) *Temporal phoneme overlap.* As words are enunciated, phonemes tend to overlap to produce compound sounds. For example, in the word *mat*, the phonemes */m/* and */æ/* combine to produce the first part of the word, and the combination does not resemble either phoneme. Accurate word recognition would require precise phoneme isolation and identification. Therefore, prior work on phoneme recognition, such as *JawSense*, cannot be effectively applied for unvoiced speech recognition; and (3) *Body motion artifacts.* Inertial sensors are susceptible to signal corruption by motion artifacts caused by body movements, such as head movements, walking, or even riding a bus. In our case, when tracking small jaw motions, large body movements lead to a lower signal-to-noise ratio. To accurately capture jaw movements, we must decouple them from head/body movements.

Our approach. To address these challenges, we synthesize principles from linguistics with signal processing techniques to accurately detect unvoiced command words. We leverage fundamental concepts involved in teaching the basics of the language to young children [92], i.e. each word is made up of several phonological components such as syllables, which further consist of vowels and consonants. The signals captured using our setup are dis-aggregated into these components. After segmenting each word into syllables, we localize and identify the vowel within each syllable. Further, we also determine the starting and ending phonemes in each syllable. We then model the task of word recognition as an estimation problem to continuously estimate state variables (phonemes) from noisy measurements. We reconstruct the word as a sequence of phonemes by fusing a forward and backward particle filter. *MuteIt* achieves command recognition without training on individual words, i.e., the system is scalable and not limited by the size of the vocabulary. The final output of the system is a list of phoneme sequences along with the posterior probability of each sequence. Lastly, we validate the system for 100 words that are commonly used in voice commands with real-world experiments with over 20 users with different speech accents.

To the best of our knowledge, *MuteIt* is the first system to recognize unvoiced words by tracking a single secondary articulator, without requiring to be trained on a predefined set of words. In summary, we make the following contributions in this paper:

- We design and develop *MuteIt*, a twin-IMU earable prototype that tracks a single secondary articulator, the user's jaw, for unvoiced command recognition.
- We devise algorithms for dis-aggregating a word signal into its phonological components, such as syllables, vowels, onset viseme, and terminal phoneme.

Table 1. Viseme Groups

Viseme Group	Phoneme	Word
$V_{m,b,p}$	/m/, /b/, /p/	move, backward
$V_{\int,s,n}$	/ʃ/,/s/,/n/	shoot, need
$V_{f,v}$	/f/, /v/	flip
$V_{t,d}$	/t/, /d/	time, document
V_w	/w/	weather
V_h	/h/	help

- We develop a word recognition algorithm that uses a particle-filter-inspired estimation technique for reconstructing a word from a partial phoneme sequence.
- Under an IRB-approved study, we validate the system for 20 users with diverse speech accents to recognize 100 words, commonly used in voice commands, with an average accuracy of 94.8%. *MuteIt* is also robust to user movements when evaluated under noisy conditions such as with head movements, walking, and even riding a bus. Further, we also compare the performance of *MuteIt* with state-of-the-art voice assistant systems. *MuteIt* achieves an accuracy of 94.3% as compared to 42% achieved by Automatic Speech Recognition (ASR) systems.

2 PRIMER TO WORD ARTICULATION

MuteIt relies on unique signatures of jaw motion during word articulation to detect unvoiced commands. In this section, we start by providing a primer on word articulation and the key components that makeup words.

2.1 Word Articulation

The human articulatory system shapes the airflow from the lungs to produce sound as we speak. The lips, teeth, tongue, alveolar ridge, hard palate, and soft palate are termed primary articulators since they play an important role in sound production and can be leveraged to interpret sounds. The jaw is often referred to as a secondary articulator [63] due to its limited role in sound production. It is therefore challenging to interpret speech from jaw motion only. The Temporomandibular Joint (TMJ) is a hinge-type joint that connects the jawbone to the skull allowing it to rotate [68]. The lower jaw moves up and down, about the TMJ, to facilitate the movement of lips and the tongue to articulate words [18].

2.2 Phonological Awareness

Phonological awareness refers to how young children learn the principles of speech articulation [10, 62, 81], specifically the ability to identify (1) syllables in a word, and (2) phonemic components in a syllable. A word can be broken into smaller chunks called *syllables* (σ), which are represented by the micro-pauses during word articulation. σ has three components: *onset, nucleus, and coda*. The *nucleus* or the core of the syllable is often a vowel sound, which is the most dominant sound in a syllable. The consonant(s) at the start of a syllable, before the nucleus, is called the *onset*. The consonant(s) at the end of a syllable, after the nucleus, are called *coda*. For example, *"fantastic"* is articulated with 3 micro pauses, **fan·tas·tic**; thus it is a tri-syllabic word. For the first syllable **fan**, /f/ is the onset, /æ/ is the nucleus, and /n/ is the coda. Consonant and vowel sounds are referred to as phonemes, which are the smallest unit of spoken language.

2.3 Understanding Visemes.

Visemes refer to groups of phonemes that *look* the same visually due to the shape made by the jaw and lips [23]. For example, the phonemes */m/, /b/, /p/* have similar articulation (produced by pressing the lips together). Similarly, the phonemes */f/* and */v/* (bringing the lips in contact with the teeth) belong to the same viseme group. Table 1
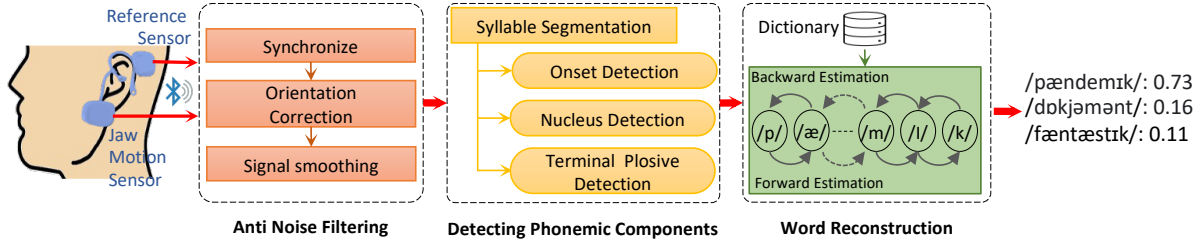
Fig. 2. *MuteIt* Overview.

shows a list of visemes and the corresponding phonemes in each group. The phonemes in the same viseme group have similar lip and jaw movements. Hence, it is not possible to distinguish the phonemes in the same viseme group just by using information from the secondary articulator. Recognizing phonemes within a viseme groups requires more information from the other articulators as well. In this project, we focus on detecting viseme groups, vowels, and phonemes for unvoiced command recognition.

3 SYSTEM OVERVIEW

To efficiently recognize unvoiced commands, *MuteIt* relies on three major modules as shown in Figure 2. The *first* module is designed to address the challenge of motion artifacts induced in movement data captured from the jaw. These artifacts are typically caused by body movements during head movements or even walking. To facilitate robust unvoiced command recognition while doing other daily activities, we leverage a twin-IMU sensing design to successfully decouple the overlapping movements of the jaw and the head. *MuteIt* avails a pair of IMUs - one placed on the temporal bone and the other on the TMJ. The IMU on the temporal bone can effectively capture head movements, while the IMU on the TMJ captures the mixed signal of the jaw and the head. We use signals from these two sensors to collaboratively remove the head movements induced noise via an anti-noise filtering technique (§ 4).

The *second* module undertakes the challenge resulting from the temporal overlap of phonemes by taking a top-down approach. *MuteIt* dis-aggregates the filtered jaw motion signal into phonological components and segments the IMU signal into its constituent *syllables* based on the characteristics of the jaw motions (§ 5.1). Next, we identify the phonemes in each syllable (§ 5.2). To do so, we start with localizing the core of the syllable – the vowel, followed by identifying the viseme group of the onset of the syllable (first phoneme). Finally, we detect if the last phoneme is a plosive. The output of this module is a partial phoneme sequence. Note that this phoneme sequence is incomplete, i.e. it does not identify all the phonemes in the word because not all phonemes can be identified by tracking jaw motion only.

In the *third* module, we build upon the previous modules to undertake the most significant challenge of command recognition by tracking a secondary articulator. *MuteIt* recognizes the unvoiced command by reconstructing the word from the partial phoneme sequence obtained from the previous module. We design a bi-directional particle filter that estimates the complete phoneme sequence using forward and backward estimation algorithms (§ 6). We leverage a dictionary, which contains the phonemic map of words, to ensure that only valid phoneme sequences are generated[1]. Finally, *MuteIt* outputs a list of words (as phoneme sequence outputs shown in Figure 2) along with the posterior probability of each sequence.

---

[1]Almost all standard dictionaries provide the phonemic map of each word [19].

This top-down approach enables us to reconstruct any word from its phonemic components. In contrast to machine learning-based word detection systems, *MuteIt* does not need to be retrained for recognizing new words. As long as the word is present in the dictionary, *MuteIt* is capable of identifying it.

4 TWIN-IMU SENSING DESIGN AND ANTI-NOISE FILTERING

Movement signals captured from the jaw are buried under noises caused by motion artifacts. Common overall body movements, such as walking, head movements, and riding in a vehicle can induce artifacts into the data captured by the jaw motion sensor. These body movements are much larger in magnitude than jaw motion, making it difficult to extract jaw motion from the corrupted data. To address this challenge, we propose a twin-IMU setup, with two inertial sensors as shown in Figure 2. The *jaw motion sensor* is placed at the lower TMJ to capture the jaw motion during word articulation. However, the signal from this sensor is easily corrupted by overall body movements. To remove these artifacts, we place a second inertial sensor, the *reference sensor*, behind the ear on the temporal bone. As shown in Figure 3, the *reference sensor* is able to capture the motion noise caused by human motion artifacts, and since the sensor is isolated from the jaw area, jaw motion signals do not appear in its readings. By applying differential measurements from the *jaw motion sensor* and *reference sensor* readings, *MuteIt* can recover the jaw movement. We leverage this observation to model the head and overall body movements (using data from the reference sensor), and then remove these artifacts from the jaw motion sensor.
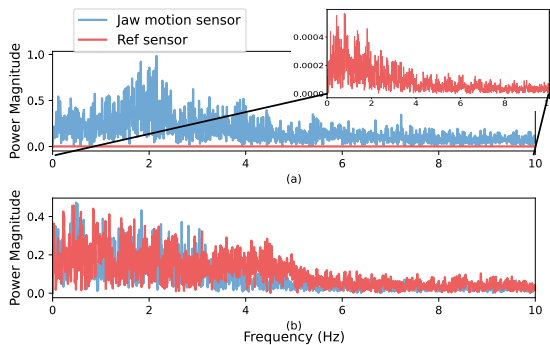
Fig. 3. Jaw signals captured using twin-IMU setup: (a) For jaw motion only, the jaw motion sensor captures the signal while the reference sensor does not. (b) When the user is walking, with no jaw motion, both jaw motion and reference sensor capture the equal magnitude of the body movements.
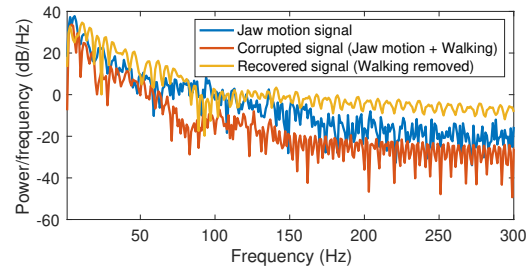
Fig. 4. Signal recovery using Anti-Noise Filtering results in higher SNR across multiple frequencies.

To synchronize data streams from the two sensors, the user lightly taps the reference sensor (once) at the beginning of a session when they wear the device. We detect the tap on each sensor to align the time-series signals.

We prefer this dynamic approach over a one-time offset measurement between the sensors because the twin-IMU setup may have a clock drift over a long period of time.

Next, we transform the orientation of the reference and jaw motion sensor to the North East Down global coordinate frame, so that both sensor signals are in the same coordinate space. This also makes our solution agnostic to sensor orientation. Lastly, the reference sensor data is used as the *anti-noise* signal and subtracted from the jaw motion signal to cancel the artifacts caused by body movements. Figure 4 shows that the SNR across multiple frequencies is higher when using an anti-noise filter, compared to corrupted signal with walking.
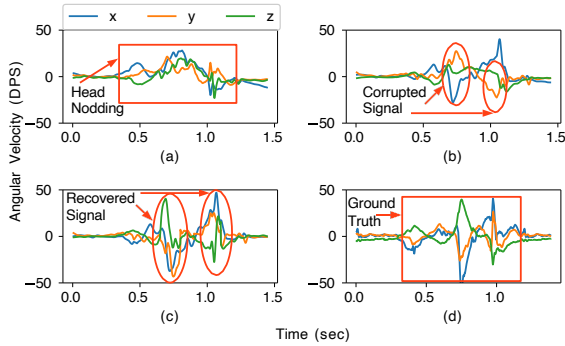
Fig. 5. Motion Artifacts. (a) Head movement data was captured by the reference sensor. (b) Unvoiced command word (captured by a jaw motion sensor) polluted by the head-nodding signal (SNR: -3.01 dB) (c) Recovered command word signal after applying anti-noise filtering (SNR: 6.43 dB). (d) Unpolluted command word signal captured by jaw motion sensor for comparison.

Fig. 6. Comparing sonority in audio data to falling and rising energy in IMU data for the word *"fantastic"*. The dashed vertical lines mark the areas of sonority in (a) voiced word, (b) its counterpart in jaw motion signal, (c) energy of the corresponding jaw motion signal.

Figure 5 demonstrates the noise removal process by applying the anti-noise filter to the example word *"map"*. Figure 5(a) shows the head-nodding signal captured by the referen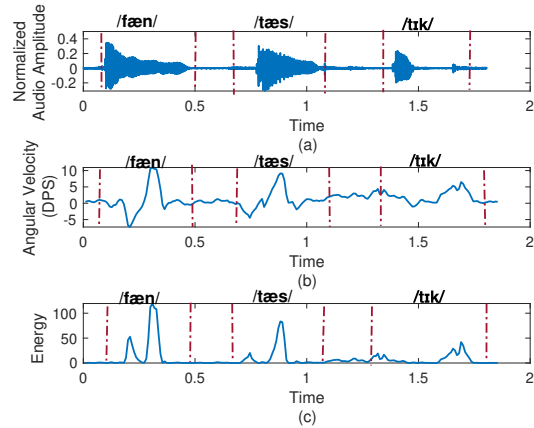ce sensor. Figure 5(b) shows the jaw signal while *"map"* was articulated, but corrupted by the head movement signal. The instances marked by red arrows are corrupted due to motion and lead to incorrect word recognition. Figure 5(c) shows the jaw motion output from the anti-noise filter; it is the unvoiced command *"map"* after the corrupted signal has been recovered. The red arrows indicate the recovered portions of the signal. The recovered signal is similar to the unvoiced command *"map"* when articulated without any head movement, as shown in Figure 5(d). We evaluate this anti-noise filtering in § 8.

5 PHONEMIC COMPONENTS DETECTION

To recognize command words, we take a top-down approach wherein the filtered jaw motion signal is disaggregated into its constituent phonemic components. To this end, we first segment the syllables from the filtered data (Section 5.1). Then, we devise an algorithm to identify the phonemic components in each syllable, such as the opening viseme (onset), the vowel (nucleus), and the terminal plosive. (Section 5.2). Breaking the signal down allows us to reconstruct it (Section 6) without requiring to train a word classifier.

5.1 Syllable Segmentation

Syllables are the primary building blocks of a word and are denoted by the micro pauses that occur within a word. To identify the syllables from the IMU data, we leverage insights from speech processing [9], where syllables are identified using acoustic sonority [64]. Acoustic sonority refers to the rising and falling cycles of acoustic energy. Figure 6(a) shows the audio signal for the voiced word *"fantastic"* and its acoustic sonority (dashed vertical lines). On the other hand, earlier studies have shown a positive correlation between acoustic sonority and jaw opening [21, 54]. In fact, a common technique for counting syllables is to place your hand under your chin and say a word; the number of times your chin touches your hand, i.e., the number of times your jaw drops, is the number of syllables in the word. Following this insight, we investigate the falling and rising energy of jaw opening and closing captured by the z-axis of the gyroscope, the axis about which the jaw rotates. Figure 6(b)
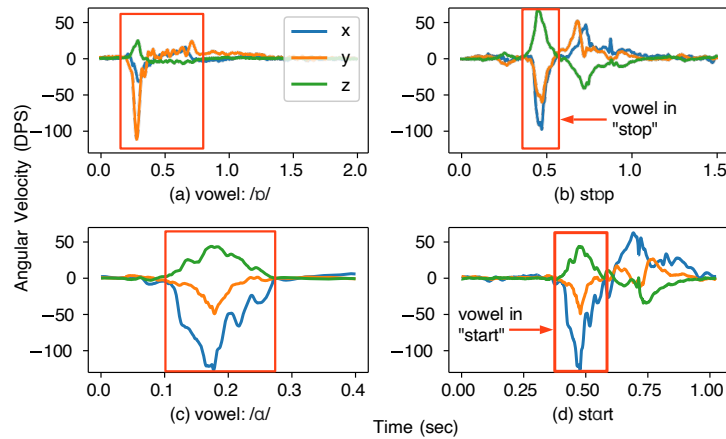
Fig. 7. Gyroscope signal for (a) vowel /ɒ/ and (b) its example word *stop*, (c) vowel /ɑ/ and (d) its example word *start*.

shows the measured rotation signal, where each cycle of falling and rising angular velocity corresponds to a syllable. We can see that syllables in the audio data are represented by the sonority, and syllables in the IMU data are represented by the falling and rising of the jaw.

Analogous to acoustic energy, we calculate the energy of jaw lowering motion from the z-axis of the gyroscope. The energy signal is shown in Figure 6(c). There are several potential approaches for segmenting a word into its constituent syllables based on the jaw motion energy. While we could employ a simple threshold-based algorithm to detect change points in either the magnitude of angular velocity or the signal energy, this would require choosing a threshold a priori. However, in practice, the threshold could be different for different users. Even the threshold for a user may vary depending on talking speed and fatigue.

To address this, we employ a dynamic approach that does not require manually defining a threshold. We use the Kullback Leibler (KL) distance for detecting syllable boundaries [39, 72, 80]. KL distance computes a measure of how distinct two input data windows are, based on their relative entropy. The larger this value, the greater the distance between the probability distribution functions (PDF) of the data segments. KL distance has been used for segmenting audio data of different speakers [16]. We divide the signal into 11ms windows (typical human speech rate is ≈ 9 phonemes/second) and compute the KL distance between consecutive windows. A distance higher than a threshold (θ) indicates sharp changes in the jaw motion energy —coinciding with syllable boundaries. This threshold, θ, is the operating point obtained from the ROC curve in Figure 18a. The syllable segmentation algorithm is evaluated in Section 8.2.3.

5.2 Identifying Phonemic Components

Within each syllable, we aim to identify as many phonemes as possible for efficient word reconstruction and recognition. Figure 8 presents an overview of how *MuteIt* detects various phonemic components, using the word *fantastic* as an example. For each syllable in the word, obtained from §5.1, first, the vowel is localized and identified (§5.2.1). After that, the part of the syllable right before the vowel is used to identify the onset viseme (§5.2.2), and the part right after the vowel is processed to determine whether or not a plosive is present (§5.2.3). We present the details of each of these components in the following subsections.

5.2.1 Vowel Localization and Recognition. A syllable may not always have an onset or a coda, but a nucleus is always present. For example, the word *ago* has two syllables **a·go** but the first syllable only has a vowel and
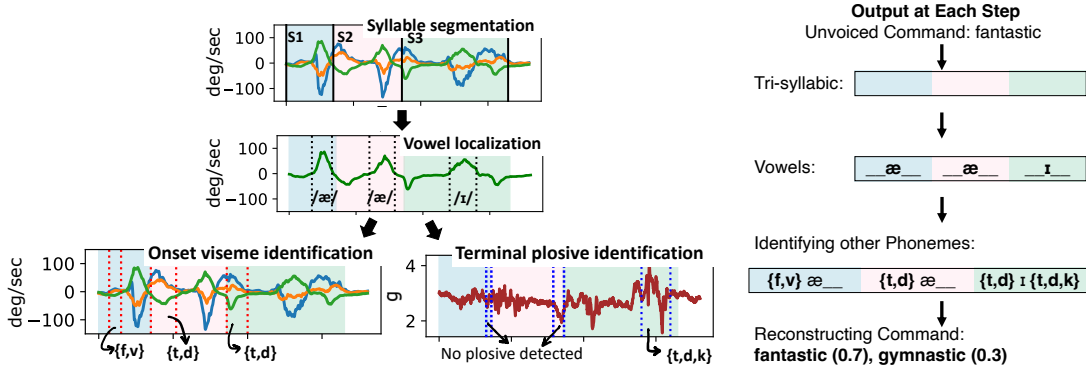
Fig. 8. Phonemic component detection for the word *"fantastic"*. The figure on the right shows the output generated at each step.

Table 2. Vowels in the English language.

Vowel	ə	ɪ	i	ɛ	æ	u:	ʌ	ɑː	aɪ	ɒ	ɔː	ʊ
Word	us	quick	speak	left	map	move	front	start	time	what	four	good

no consonants. Importantly, vowels are the most sonorous sounds in spoken language, i.e. they are typically enunciated with the mouth fairly open with lower jaw movement. Therefore, the jaw opening within a syllable indicates the beginning of the vowel. We compute the KL distance (discussed in §5.1) within each detected syllable to identify energy boundaries for rising and falling peaks - indicating jaw opening and closing. The segment with the highest energy change is identified as the vowel within each syllable.

Once we localize the vowel in each syllable, the next task is to recognize it. We observe that every vowel has a unique jaw opening signature and that a vowel retains its signature whether articulated as part of a word or in isolation, i.e. an isolated enunciation of a vowel exhibits a similar movement pattern to a vowel in a syllable. For example, Figure 7(a) shows the gyroscope data for the isolated vowel /ɒ/. For the word *"stop"*, the vowel /ɒ/ shows similar jaw motion as depicted in Figure 7(b). Similarly, Figure 7(c) shows another isolated vowel /ɑ/ and the same vowel in the word *"start"* is shown in Figure 7(d). The jaw opening phase of the vowel can be seen clearly in the word. We observe that when a vowel is present in a syllable, it always contains the jaw opening phase of that vowel, but the jaw-closing phase can be overlapped with the succeeding phonemes. Therefore, we use the jaw opening phase for recognizing the vowel.

To recognize vowels, we train a classifier for the 12 vowel sounds in English, as shown in Table 2. We extract two types of features from the time-domain IMU data for the jaw opening phase of each vowel: linguistic-based features and statistical features. Linguistic features capture vowel-specific information, whereas statistical features can capture user-specific characteristics. The linguistic features are based on insights derived from how jaw motion relates to the manner of articulation [79]. Since we do not have information about how other articulators are engaged in speaking, we indirectly compute this information from jaw motion. Specifically, we compute the jaw opening distance (correlated to tongue position), jaw motion smoothness (correlated with unstressed vowels and jerky motion of vowels), and the spread of energy (a measure of impulse like jaw motion), and jaw rotation speed along a different axis.

To calculate jaw opening distance, we first normalize the three-axis accelerometer signal and then double integrate to estimate jaw opening. Even though the accelerometer data is noisy, we are only interested in the relative jaw opening of different vowels and not the absolute values. Jaw motion smoothness captures if the jaw
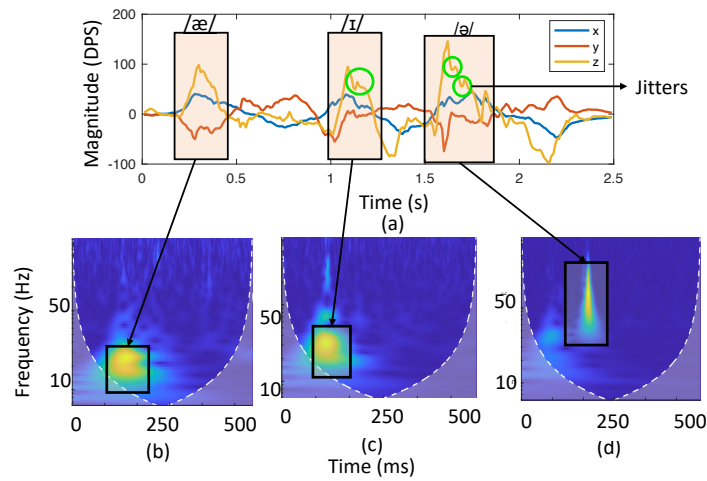
Fig. 9. Linguistic features for different vowels in the word *capital*. The jitters in the jaw motion are annotated in green circles in (a). The frequency spectrum ((b), (c), (d)) represents the spread of energy for the vowels. ɑ has no jitters and hence broader and smooth energy peak in the spectrogram. While, ɪ and ə have jitters and sharper energy peaks.

comes to the initial position in one smooth motion or is it irregular. For instance, as shown in Figure 9 (a) in vowel /ə /, there are jitters in the jaw opening phase. This is because /ə / is an unstressed vowel and hence has a very swift jaw motion which results in irregular spikes. We capture this by determining the location and magnitude of jitters in the gyroscope signal. To quantify the spread of energy, we perform a Continuous Wavelet Transform (CWT) [1] over the gyroscope data. We empirically choose the thresholds for calculating the duration of energy that yield the best classification accuracy. We find the duration of time for which the energy was greater than 3 times the floor value. We also take the ratio of angular speeds of all axis; this represents the jaw rotation speed in one axis relative to another and is different for different vowels.

We extract the following statistical features for the jaw opening phase of each vowel: mean, maximum, minimum, standard deviation, integral, kurtosis, skewness, and cross-correlation between the accelerometer and gyroscope axes. From the frequency domain, we add the first 8 DFT coefficients to our feature vector. We train a 14-class classifier using 4 different classification models: RBF Support Vector Machine, Gaussian Process, AdaBoost, and Random Forest. Of these, we use a Gaussian Process classifier (GPC) for classifying the vowels since it achieves the highest accuracy. We train a personal model for each user, by using a small portion (1̃ minute) of their data for training.

5.2.2 Onset Viseme Identification. After we localize the vowel within a syllable, we extract the signal from the start of the syllable to the start of the vowel —this is the onset for that syllable. Our goal is to identify the phonemes that form the onset. A major challenge in identifying consonant phonemes is that several phonemes lead to a similar movement of jaw and lips, and are therefore indistinguishable based on jaw or lip movement only [8]. In linguistics literature, these phoneme groups are called visemes. Visemes are the "visual equivalent" of a phoneme. Table 1 shows six viseme groups with their constituent phonemes. Visemes have been used in previous works on speech recognition using lip reading [7, 88].

Figure 10 shows the jaw motion signal for several words and the onset for the first syllable in each word is highlighted. In the word *"move"*, to produce the */m/* sound, the lips are pressed against each other, which causes a
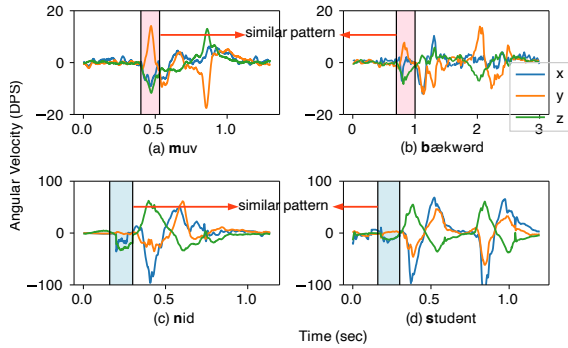
Fig. 10. Phonemes that belong to the same viseme group exhibit similar motion patterns.
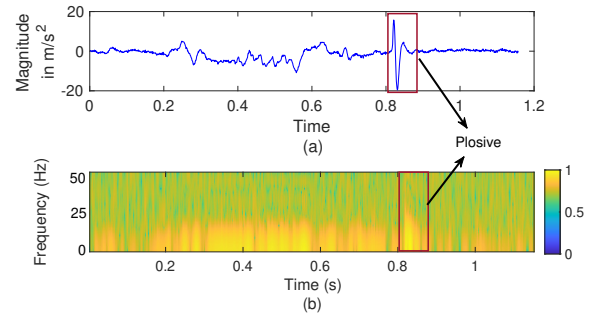

Fig. 11. (a) The plosive instance at the end of the word *flip*. (b) shows the high-frequency peak indicating the presence of a plosive.

change in the y-axis of the gyroscope data. The same jaw motion pattern is observed when another phoneme from the same viseme group is articulated. For example, */b/* in *"backward"* has the same pattern as */m/* in *"move"*.

As another example, notice that for the */n/* sound in the word *"need"*, the bottom lip creates friction with the top teeth, causing the lower jaw to slightly move upward. This causes a negative peak in the z-axis of the gyroscope data, similar to */s/* in *"student"*. We observe that the onset is not as prominent as the vowel, but different groups of phonemes involve different jaw motions, which can be leveraged to identify the viseme group each syllable onset belongs to. We train a Gaussian Process Classifier with the same features as that for vowel classification in § 5.2.1 for a 6-class onset viseme classification. The classifier outputs the viseme class with a probability. We use this probability estimate during word reconstruction in §6.

5.2.3 Terminal Plosive Detection. The last part of a syllable, from the end of the vowel to the end of the syllable, is the coda. Typically the last phoneme in a syllable is the hardest to identify because it is manipulated by the preceding phonemes. To our advantage, we observe that the most common terminal phonemes are plosives. Plosives are phoneme sounds represented by the blocking of air by the mouth and then releasing a puff of air. To aid in unvoiced command recognition, we are interested in the voiceless plosives */p/, /t/,* and */k/* [56].

We observe that unvoiced plosives cause muscle deformation at the TMJ when the puff of air is released, leading to a high-frequency component in the accelerometer signal. Figure 11 shows the accelerometer signal for the word *"flip"*. The sudden release of the puff of air while articulating the */p/* in *"flip"* at the end of the word causes a high-frequency component, which can be observed as the spike in the spectrogram in Figure 11(b). We detect this peak by using the change in energy for the frequency spectrum of the z-axis of the accelerometer data. If the energy change is greater than an empirically determined threshold, it is identified as a plosive instance in a syllable. This threshold is determined by analyzing the data and is evaluated in Section 8.2.4. When a plosive is detected, we do not know the exact phoneme, so we assign equal probability to the three voiceless plosives.

6 WORD RECONSTRUCTION

The phonemic components obtained by *MuteIt* so far are not a complete sequence, i.e. we have identified some phonemes (onset viseme, vowel, and terminal plosive) in the word, but not all. For example, when the terminal phoneme is not a plosive, we are unable to identify the phoneme candidates. Similarly, when a word has two consonants before a vowel, *MuteIt* is only able to identify the viseme group of the first phoneme. For example, for the word *"stop"*, *MuteIt* can identify the viseme group that */s/* belongs to and the vowel, but we cannot detect the phoneme */t/*. To bridge this knowledge gap we approach the phonemic map generation of the unvoiced

command as a process of traversing in space of phonemes. The analogy can be drawn as phonemes in the word are the checkpoints, we will use the phonemic components we can detect as guides and use the English language dictionary as a map.

Problem Formulation. We propose to model our problem as an estimation problem and use a particle filter-inspired technique to determine the complete phoneme sequence. Particle filters represent the belief state with a set of possible states (or *particles*) and assign a probability of being in each of the possible states.

- *State-space (X):* Each state in the state-space X is defined as a phoneme. Since there are 44 phonemes in the English language [57], the size of the state space is 44.
- *Motion model:* To model how phoneme states evolve over time, we use a subset of the English language dictionary, which contains the phonemic maps for 1100 words. A phoneme p_k at time k can only be followed by a subset of phonemes, depending on the words in the dictionary, i.e. some phonemes are more likely to appear consecutively in a word than others. To simplify, assume that /m/ is the state at time k, and the dictionary has only 2 words that start with /m/: *mat* and *movie*, then the motion model states that p_{k+1} can be /æ/ or /u:/. The motion model is given by $P(X_t|X_{t-1})$.
- *Measurement model:* The measurement model provides the probability that we will see certain observations in a particular state. In our formulation, we use the estimates Z for the phonemic components in a word, obtained from the previous subsection. The vowel and viseme classifiers from § 5.2.1 and § 5.2.2 output the probability for each class. We use these as measurements in our system. The measurement model is given by the likelihood $P(Z_t|X_t)$.

Sequential Estimation. Next, we discuss how we implement the estimation algorithm for identifying the word by constructing the phoneme sequence. At any given time, the state X of the system is represented by a number of particles, where each particle has a weight. The total number of particles that can exist in the system is the number of states, which is 44 in our case. Measurements from phonemic components are represented by Z. The state of the system at time t is given by:

$$P\left(X_t \mid Z_1, \ldots, Z_t\right) \propto P\left(Z_t \mid X_t\right) P\left(X_t \mid X_{t-1}\right) P\left(X_{t-1} \mid Z_1, \ldots, Z_{t-1}\right)$$

At each iteration, there are 3 key steps:
+ *Step 1:* prediction with motion model, or the motion update;
+ *Step 2:* measurement update;
+ *Step 3:* resampling.

Initially, at time t_1, during Step 1, one would expect the algorithm to initialize with a uniform distribution for all particles. However, since we have access to a dictionary, we assign weights to particles at t_1 based on the probability that the particular particle (in this case phoneme) will appear in the first place. We use the frequency of occurrence of a phoneme in the dictionary as the corresponding particle weight. This will automatically eliminate the phonemes that do not start a word. During Step 2, we obtain the measurement from Section 5.2, which is a list of phonemes (viseme, vowel, and whether or not there are plosives) with their probabilities. All phonemes within a viseme group are assigned equal probability. In Step 3, during resampling, we assign particle weights proportional to the observation likelihood from Step 2. We iterate over the entire word, and when we arrive at the last phoneme, we reconstruct the word using the particle sequences. The probability of each word is the joint probability of its candidates. The motion model ensures that all phoneme sequences are valid words.

However, this forward iteration has one drawback: If the first phoneme is incorrectly detected, the entire sequence can be incorrect. For example, if for the word *alarm* the first vowel ə is misclassified as ʌ , then the forward iteration will misclassify the word *alarm* as *uncharm*. This is to note that both of these words differ by only vowel sounds. To address this, we propose a particle filter with backward iterations. The backward estimation particle filter is defined as $P(X_k \mid X_{k+1}) = P(X_k \mid Z_t, Z_{t-1}, \ldots, Z_k)$. The backward iteration runs independently in

parallel with the forward iteration. The intuition is that forward and backward iterations are complementary in nature. The two iterations are intentionally kept independent as we do not want one to affect another's state estimation. For the same example of the word *alarm* discussed above, using the backward iteration the probability of *alarm's* phonemic sequence is more than *uncharm's*. This is because, when traversing from backward iteration the probability of the phonemic sequence of alarm is higher than uncharm based on our dictionary. The same is true when the vowel detected in the second and third syllable for di-, and tri-syllabic words is incorrect. In that case, the forward iteration complements the backward. The advantage of using a probabilistic approach, like particle filter, is that it can alleviate some errors that are likely to propagate in the top-down phonemic component detection module. Another probabilistic approach, like HMM, would only consider the previous state but not the entire sequence, and would therefore not be suitable. This is because the likelihood of a phoneme at time t is proportional to the joint distribution of all the phonemes till time $t - 1$. Our approach is built on the assumption that at least one of the vowels in di-, and tri-syllabic words are detected correctly.

Given that the two have complementary performance (evaluated in Section 8.2.5), we fuse them to develop a bi-directional filter that combines probability estimates from forward and backward iterations of the particle filter, given by:

$$p_c\left(X_k \mid Z_{1:T}\right) = f_{\text{comb}}\left(p_f\left(X_k \mid Z_{1:k}\right), p_b\left(X_k \mid Z_{T:k}\right)\right)_{k=1}^{T} \tag{1}$$

where p_c is the combined probability distribution, p_f is forward probability, and p_b is backward probability. We use conflation, a conjunction scheme for independent probabilities [31], to consolidate the distributions p_f and p_b into a single probability distribution. The combination function f_{comb} is therefore defined as the conflation of p_f and p_b, given by:

$$f_{comb} = \frac{p_f\left(X_k \mid Y_{1:k}\right).p_b(X_{k)} \mid Y_{T:k})}{p_f\left(X_k \mid Y_{1:k}\right) + p_b(X_{k)} \mid Y_{T:k})} \tag{2}$$

The bi-directional particle filter outputs candidate phoneme sequences, each with a probability. For the example discussed above, when we combine the forward and backward iteration using conflation probability for *alarm* is higher than *uncharm*. This is because the vowel detected in the second syllable is correct which shifts the estimate towards *alarm*.

7 *MUTEIT* IMPLEMENTATION

We prototype *MuteIt* and design experiments to validate unvoiced command recognition.

Prototype setup and data retrieval. To ensure a socially acceptable prototype, we design *MuteIt* with 1) a small form factor, and 2) non-intrusive placement. Our earable prototype consists of an silicon ear hook [20] and two IMUs [51]. Silicon ear hooks are commonly used to hold ear pods, specifically AirPods [20]. We attach two IMUs to the ear hook. We envision that in the future prototype could be miniaturized and integrated in bone conduction or around the ear earphones, or even hearing aids. The reference sensor is fixed behind the ear, so that it sits on the temporal bone behind the ear to capture larger body movements. The jaw motion sensor is attached to the ear hook using a flexible link. As opposed to a rigid system, the flexible link prevents vibrations captured by one sensor from being carried to the other sensor. To maintain contact with the jaw, we use a medical-grade gentle adhesive that allows the user to place the jaw motion sensor on the TMJ. Our experiment setup and the prototype are shown in Figure 12. The IMUs transmit data wirelessly to a mobile phone over Bluetooth low energy with a mobile application. We log timestamp, 3-axis accelerometer, and 3-axis gyroscope data at 100 Hz for both sensors.

Data collection. We invite 20 volunteers (7 female and 13 male) to collect data in an IRB-approved study. Our participants speak different native languages —English (5), Hindi (5), Telugu (4), Chinese (2), Spanish (1), Kannada (1), and Korean (1). Participants are undergraduate or graduate students aged 20-31 years. Before the
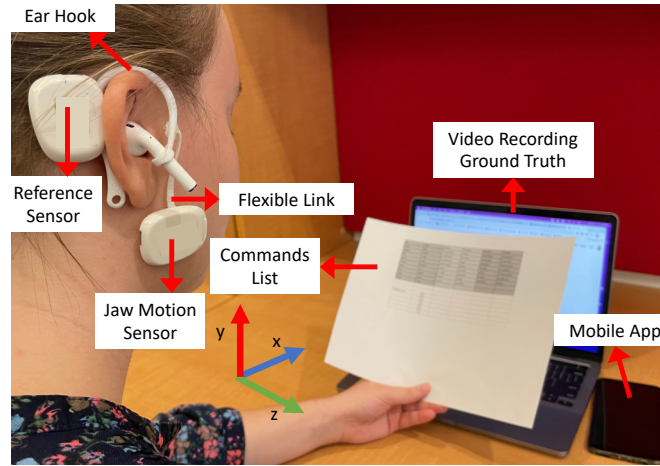
Fig. 12. *MuteIt*'s prototype and experiment setup.

data collection process we ask users to fill out a consent form to record their data using our prototype and their video. We hold an IRB protocol which allows us to attach the prototype on users' TMJ and temporal bone.

We curate a set of 100 command words that are commonly used in voice assistants, smart home, and gaming environments [55]. To obtain words from these commands, we select the words that occur commonly across similar commands. For example, for the commands "How is the weather today?" and "What does the weather today look like?", we select the words "weather" and "today". The words span across various combinations of all the vowel sounds in the English language. Of the 100 words, 50 are mono-syllabic, 35 are di-syllabic, and 15 are tri-syllabic.

We create a dictionary which contains the phonemic maps for 1100 words (100 test words and 1000 additional words). The 1000 additional words are chosen so that some of them have only one phoneme different from the 100 words set. For instance, for the word *freezer*, we add *breezer* to the dictionary; for *cancel*, we add *pencil*. We also include other common words used in the English language. While we only use a subset of the English language dictionary, our system could potentially work with a much larger dictionary.

In addition to the twin-IMU data, we also collect the audio and video data of the experiments with Google Pixel 3a and a laptop front camera, respectively. The audio and video data are used as ground truth to better understand jaw movements and unvoiced enunciations for each user. Saying the words out loud also gives the participants an opportunity to get familiar with the words before enunciating in an unvoiced manner. After the experiment, all volunteers fill out an exit survey about their experience. We collect data in 3 scenarios:

- *Noise-free environments:* From a randomized list of the command words, each user articulates the 100 words in an unvoiced manner 5 times and once inaudible manner, with as minimal body movements as possible.
- *In presence of motion noise:* We ask ten users to collect additional data where they articulate the word while (1) moving their head, (2) walking, and (3) riding a bus. For head movements, participants move their heads around freely. The experiment was designed to simulate the users' natural body language in a real-world work environment. For walking data, users walk in a large room at their normal walking speed. No instructions were given regarding the pace or direction of walking. Along with this, we ask users to ride a bus. The users are seated before the bus starts and are free to move around on the bus if they wanted to. This environment is most challenging as it involves multiple sources (user and bus) of overlapping motion noise.

- *In presence of external acoustic noise:* We ask users to articulate words when music was played on speakers and also over the Airpods worn by the user. This experiment was designed to evaluate if *MuteIt* an be integrated with off-the-shelf earphones without any interference from the music being played. We also play sounds in the background to simulate different noisy acoustic environments. We simulate five environments: quiet library, rock music playing in background, noisy restaurant, traffic noise, and machine noise. We play all these sounds via laptop at 60 dB. We specifically select these environments as they encompass a wide range of scenarios. For instance, a noisy restaurant has people chatting which is similar to attending a gathering or a conference while machine noise simulates environment of construction sites and factories. We ask 5 users to articulate 20 commands, 5 repetitions of each in voiced manner while the sound is played in the background. We then ask the volunteers to repeat same set of commands 5 times each in an unvoiced manner, with the same sounds playing.

8 EVALUATION

In this section, we first describe the evaluation metrics and baselines for *MuteIt* and its components (§ 8.1). Then, we evaluate the overall command recognition and various components of *MuteIt* (§ 8.2.1 - § 8.2.5). Next, we investigate the robustness of *MuteIt* in different environments (§ 8.2.6). Finally, we present results from our user study (§ 8.3).

8.1 Metrics and Baselines

8.1.1 Evaluation Metrics. We use Top-1 and Top-2 accuracy for word recognition. These metrics are commonly used in the Natural Language Processing (NLP) as well as the vision community [13, 15, 34, 45, 85, 86, 89] for models that output probabilities. Top-1 accuracy measures the proportion of instances for which the predicted word matches the target word. Top-2 accuracy measures the proportion of instances for which the target word is in the top two predicted words (with highest probabilities). Along the same lines, Top-1 error is the number of times the most probable outcome is not the same as the ground truth, and Top-2 error is the measure of number of times neither of the two most probable outcomes is the same as the ground truth. We report Top-1 and Top-2 accuracy as we envision that *MuteIt* can be integrated with a language model for continuous unvoiced speech recognition, which can select the most appropriate candidate from a list of words by taking the sentence context into account.

8.1.2 Baselines. We use the following four baselines for evaluating *MuteIt*.

- *Greedy matching.* In this baseline, we use a greedy matching approach for word recognition, instead of the proposed bi-directional particle filter. We match the phonemic components (from §5.2) with the words in our dictionary to find the closest match. If all the phonemic components are detected correctly, we are able to find the accurate target word from the dictionary. If no word is present with the detected phonemic sequence, the Top-1 accuracy is zero. However, when multiple words may be a match, for example a viseme group with multiple phonemes is detected, the greedy algorithm is not able to distinguish between words where only a single phoneme is different. For example, *map* and *back* are difficult to distinguish since the starting phonemes */m/* and */b/* belong to the same viseme group, and both words end with a plosive.
- *JawSense-based.* JawSense [46] senses the muscle deformation caused by unvoiced isolated phonemes spoken by the user. It is capable of recognizing six phonemes with 92% classification accuracy. In this baseline, we use the JawSense phoneme recognition model instead of the proposed phonemic component detection model to recognize command words. Since JawSense only accepts phoneme signals as input, we first split the jaw motion signal of a word into phonemes using heuristics from human speech rate (each phoneme lasts approximately one-ninth of a second), and extract statistical features proposed by JawSense. The features used by the classifier in JawSense are only statistical as compared to *MuteIt*'s pool of linguistic

and statistical features. The output phonemes from JawSense are put together to form the word. If the phonemic map generated for the word is incorrect (i.e. not a valid sequence), the Top-1 accuracy is zero. We also use JawSense as a baseline for vowel and viseme classification as well, as discussed in § 8.2.4.

- *Conventional particle filter approaches.* We use the conventional particle filter technique (without the bi-directional approach) as a baseline. The forward and backward PF are described in §6.
- *Voice assistant systems.* To compare *MuteIt* s performance in noisy acoustic conditions, we use two most popular voice assistants [2] Siri [6] and Alexa [3].

8.2 Results

(a) Command recognition accuracy across mono, di, and tri syllabic words.

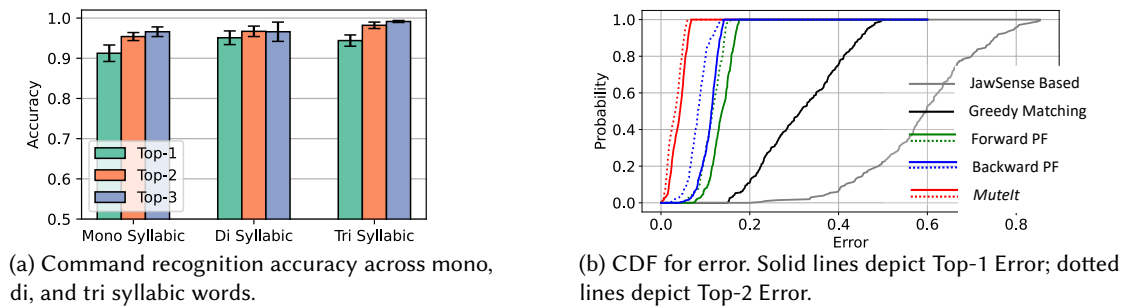(b) CDF for error. Solid lines depict Top-1 Error; dotted lines depict Top-2 Error.

Fig. 13. Command recognition accuracy and comparison with baselines .

8.2.1 Overall Command Recognition. We evaluate *MuteIt*'s overall command recognition performance under different conditions. Figure 13a shows the recognition accuracy for mono-, di- and tri-syllabic commands. We observe that *MuteIt* achieves a higher than 90% accuracy across all command words. Moreover, *MuteIt* achieves up to 96.7% and 97.5% on average for top-2 and top-3 accuracy for all words. Note that the top-3 accuracy is close to top-2 accuracy, implying that most of the time the target word is in the top 2 predicted words.

Figure 13b shows the cumulative distribution function (CDF) plot of top-1 and top-2 error for the baselines and *MuteIt*. We can see that *MuteIt* outperforms all the baseline approaches. JawSense based word recognition has an error of 68% for more than 75% of the command words, showing the importance of correctly localizing the phonemic components as well as our word reconstruction technique. Further, the 90-percentile error for greedy matching is as high as 47%, indicating that even for a moderately sized set of 100 command words, greedy matching is not able to accurately detect the unvoiced words from partial phoneme maps. Even by using the forward or backward particle filter approach, we significantly improve over the greedy matching based word recognition. Finally, the bi-directional particle filter in *MuteIt* overcomes the limitations presented by the baseline approaches, with top-1 error less than 7% for all data.

The top-1 accuracy for all users is shown in Figure 14. All the users have top-1 accuracy greater than 93% showing that our system is robust across users with different accents. Of the top ten users with the highest accuracy five were native speakers. This can be explained by the correctness of their enunciation (as compared to standard phonemic maps found in the dictionary), while Kannada and Chinese users had slightly lower accuracy as compared to other users.

8.2.2 Comparison with Voice Assistants. We evaluate how well *MuteIt* performs as compared to state-of-the-art voice assistants. The top-1 word recognition accuracy of Siri, Alexa, and *MuteIt* is shown in Figure 15. We compare *MuteIt* with voice assistants and not an acoustic speech recognizer as we want to emphasize the performance of
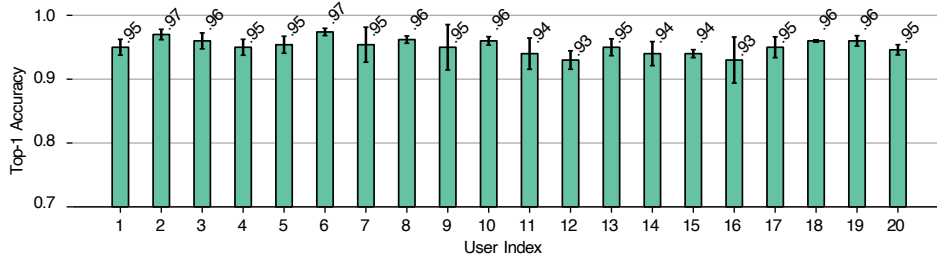
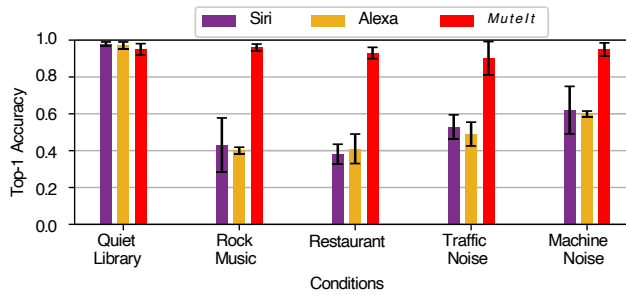Fig. 14. Median top-1 accuracy for all the users is greater 0.93.

Fig. 15. *MuteIt* performs better when compared with Siri and Alexa under external acoustic noise.

commercially available voice assistants in noisy acoustic conditions. To get the ground truth for voice assistants, users articulated command words like "weather". If the voice assistants recognize the command word correctly, they would respond accordingly and the accuracy would be 1. Some other example command words are timer, music, calendar, and map.

Though in a quiet environment (dB< 15) Alexa and Siri outperform *MuteIt* by a small margin (4%), our system is robust to external acoustic noise. We notice that the performance for Alexa and Siri drop sharply in a noisy restaurant and when rock music is playing in the background, with less than 40% top-1 accuracy. In contrast, *MuteIt* has accuracy higher than 90% for all the noisy acoustic environments. This demonstrates that the unvoiced nature of *MuteIt* makes it extremely robust to scenarios with acoustic noise (where popular voice assistants perform poorly), and therefore it can potentially be used as an alternate mode of interaction in noisy real-world scenarios.

8.2.3 Syllable Segmentation. In this section we evaluate the syllable segmentation module for both, the number of syllables detected as well as the segmentation accuracy. Figure 16 shows the confusion matrix for syllable counting. *MuteIt* can identify the number of syllables with a 97% weighted accuracy.

We notice that words with vowel /æ/ are among the most misclassified words. This is because /æ/ is a diphthong. In diphthongs, two vowel sounds are combined. This causes the jaw to move downwards twice in a syllable (instead of just once), leading to overcounting the syllables. Beyond syllable counting, we also evaluate for the accuracy of syllable segmentation. We consider a segmentation accurate (true positive) if the detected segment boundary lies within 10 ms of the manually labeled segment boundaries. This value is based on human word articulation speed. Syllable boundaries detected outside this limit are labeled as false positives. Figure 18a shows
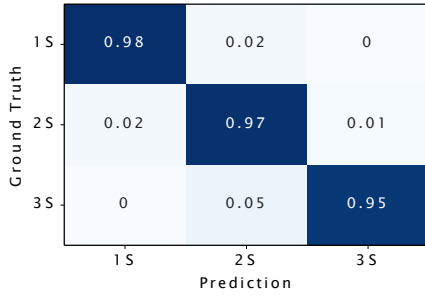
Fig. 16. Syllable counting confusion matrix for mono (1S), di (2S), and tri (3S) syllabic words.
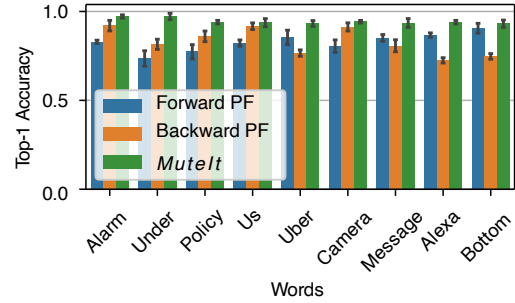

Fig. 17. Performance of *MuteIt* (bi-directional PF) compared with forward and backward particle filter.

the ROC for the syllable segmentation module. We achieve a 96.3% true positive rate and 3.1% false-positive rate at our chosen operating point. The value of this operating point is 1200 and does not need to be adjusted for every user.
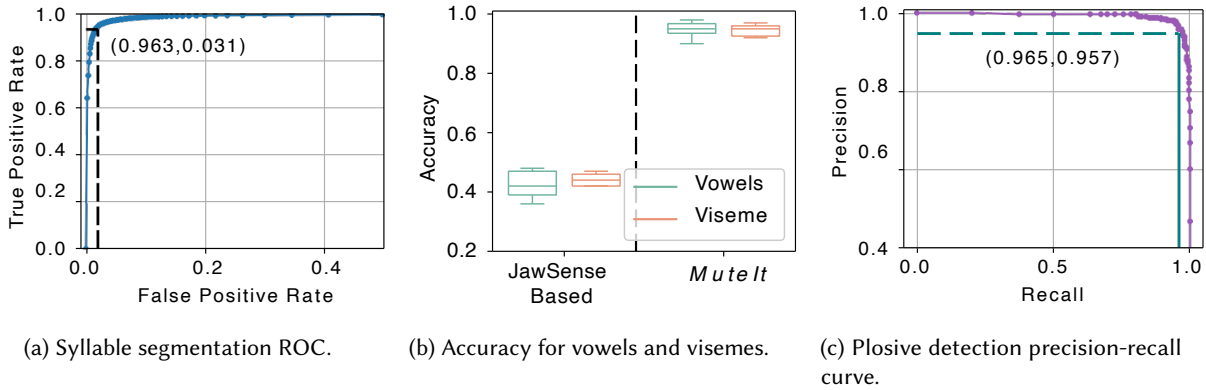

(a) Syllable segmentation ROC.

(b) Accuracy for vowels and visemes.

(c) Plosive detection precision–recall curve.

Fig. 18. Syllable segmentation and phonemic components detection.

8.2.4 Syllable-level Phoneme Identification. Here we evaluate the performance of vowel and viseme classification, and terminal plosive detection by *MuteIt*.

Vowel and onset viseme classification. To train the GPC classifier for onset vowel and viseme we first randomly shuffle the data and then perform a 70-30 split for training and testing respectively. Also, we perform a stratified split such that we have the same proportion of the samples from each mono, bi, and trisyllabic words. We leave out the training data in the evaluation. We test the vowel and onset viseme classification against JawSense based phoneme detection as the baseline. In the baseline approach, we divide a syllable in three equal parts and denote the first part to be the onset and the middle part to be the vowel. We extract the same statistical features as proposed in JawSense and use their model to classify vowels and visemes. Figure 18b shows the accuracy for the baseline approach vs *MuteIt*. We can see that in contrast to the baseline approach, *MuteIt* accurately localizes the vowel which significantly improves the vowel and viseme identification. The increase in accuracy is due to (1) correctly localizing the phonemic components, and (2) better representative features.

Terminal plosive detection. Figure 18c shows the precision-recall curve for terminal plosive detection. We achieve 96.5% recall and 95.7% precision at the threshold we choose for classifying the terminal phoneme as

plosive or not. This threshold is 0.6 after signal normalization and does not need to be adjusted for every user. The highest false negative rate (4.7%) is for phoneme */p/*. This is because the */p/* sound is produced by a puff of air which can be restricted due to phonemes surrounding them, which is not true for */t/* and */k/* which are produced by tongue rubbing on upper part of mouth. Detecting unvoiced plosives accurately also provides an anchor to the backward iteration of the particle filter, leading to higher overall word accuracy.

8.2.5 Comparing Particle Filter Variations. Figure 17 shows the top-1 accuracy for forward PF, backward PF, and *MuteIt* (bi-directional filter) for several words from our 100-word test set. For short words like *uber* and *alarm*, where the first syllable consists of a single phoneme, the forward PF has lower top-1 accuracy as compared to backward PF. This is because during the forward iteration of the PF, if the starting phoneme is detected incorrectly, the PF is unable to recover. Similarly, for words that end in a vowel, like *camera* or *policy*, the backward PF yields lower accuracy compared to the forward PF. We observe that the bi-directional PF implemented in *MuteIt* leads to higher accuracy for these words, even when the forward and backward PF do not provide accurate estimates separately.

8.2.6 Sensitivity Analysis. This section shows that *MuteIt* can be used successfully in various real-world scenarios. We evaluate *MuteIt* under the impact of body movements during daily activities. We also evaluate the system on different days to investigate for any mounting bias.

Impact of body motion. We evaluate *MuteIt* in scenarios where the person is performing larger body movements, such as head movements, walking, and riding a bus. Our twin-IMU setup plays an essential role in filtering these noisy movements.

First, we compare how *MuteIt*'s anti-noise filtering technique performs in scenarios with body movements. We compare the performance of *MuteIt*'s anti-noise filtering with three single sensor approaches as baselines (1) *S: Baseline.* we use the corrupted signal from jaw motion sensor and do not perform any noise reduction; (2) *S: High Pass.* we apply a 5 Hz filter to remove the body movements; and (3) *S: Deconvolution.* we perform deconvolution of the corrupted jaw motion signal and the body movement signal (with no jaw movement). Figure 19 shows the performance of *MuteIt*'s anti-noise filtering, compared with these three single-sensor approaches. We report significant improvement in word detection accuracy using the twin-IMU setup.
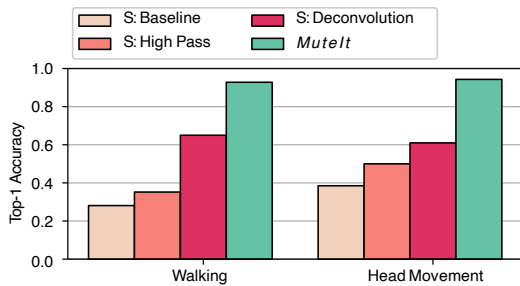
Fig. 19. Top-1 accuracy for different techniques to remove motion artifacts. S: single sensor
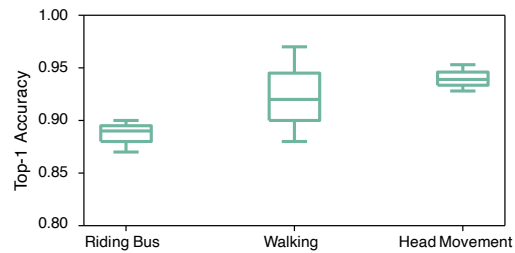
Fig. 20. Top-1 Accuracy of *MuteIt* in presence of different motion noise.

Next, we assess the performance of *MuteIt* in the presence of motion noise for the three scenarios presented in § 7. Figure 20 shows the top-1 word recognition accuracy in presence of different motion noises. Top-1 accuracy for word detection is 93.8% with head movements, 91.9% while the user is walking, and 87% when the user is riding a bus. The accuracy is lower when the user is riding a bus because the accelerometer signal's magnitude captured by reference sensor is significantly greater as compared to jaw motion sensor.

Impact of music playing on earables. We test *MuteIt* under the influence of music playing in the background and playing on earpods when a user wears them. We experiment with different dB levels: 45 dB and 75 dB for external music, and 20 dB and 60 dB for music playing on earpods. We measure the dB using Google Pixel 3a via Sound Meter app [69]. We observe that music does not impact the jaw movement signal and the accuracy of unvoiced command word recognition. This confirms that in the future, *MuteIt* can be integrated into commodity headphones without suffering interference from the music being played on the headphones.

Table 3. Performance of *MuteIt* across different sessions.

| User	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10
Top-1 Accuracy	0.939 ±0.012	0.963 ±0.094	0.965 ±0.057	0.948 ±0.062	0.881 ±0.113	0.926 ±0.034	0.915 ±0.091	0.922 ±0.018	0.937 ±0.056	0.948 ±0.061

Performance across different sessions. In this section, we evaluate how well *MuteIt* performs when data collected in one session is used to train vowel and viseme classifier and the data collected in a different session is used as a testing set. The sessions were at different times to ensure data diversity. We ask 10 volunteers to articulate 55 words, 5 repetitions of each word, in both sessions. Table 3 shows the Top-1 accuracy of the 10 users. As seen, the accuracy for different sessions is more than 90% with less than 0.1% variance across sessions for all the 55 commands. This demonstrates that the performance of *MuteIt* is repeatable across sessions without requiring any calibration.

8.3 Usability Analysis

We conduct a user study to validate the usability of *MuteIt*. Besides this, we also benchmark the latency and power consumption on Raspberry Pi 3 Model B+.

8.3.1 User Study. We conduct a user study to validate the usability of *MuteIt*. All participants are provided with the link to a Google Form that they filled anonymously. The form contains several questions on a Likert scale [41], designed after the SUS survey [52]. The feedback from the users is presented in Figure 21. When asked if the system *"was comfortable to use"*, all users either Agree or Strongly Agree. In response to *"would like to use this system frequently"*, 91% agreed. 80% of the users find *MuteIt* *"intuitive to use"*. All participants state that *MuteIt* is not complex. When asked how long they *" will be comfortable using the system for"*, 80% of users selected 2 hours or less. They further explain that this duration is determined generally by how long they're willing to wear their earpods, and does not bear on the comfort of *MuteIt*.
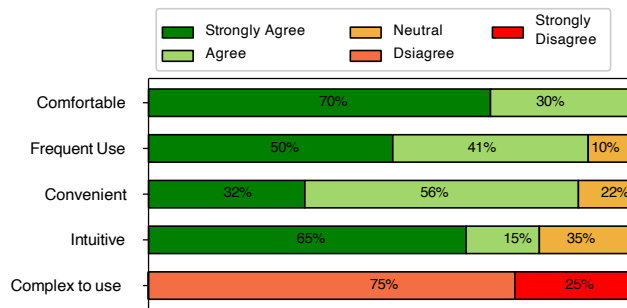
Fig. 21. User study responses for usability of *MuteIt*.

8.3.2 Latency and Power Consumption. We run *MuteIt* on Raspberry Pi 3 Model B+ and measure the latency using Python3, and power consumption using Monsoon Power monitor [37]. We run *MuteIt* for 1,000 unvoiced command words and report average results. *MuteIt* takes 2.72 seconds and consumes 1640 mW of power on an average to detect an unvoiced command. This includes streaming data over Bluetooth, anti-noise filtering, detecting the phonemic components, and a bi-directional particle filter for word reconstruction for each unvoiced command. The word reconstruction using a bi-directional particle filter takes the longest (0.8s) among all the steps. This is because it has to perform the mapping from an 1100 words dictionary, which can be reduced by using appropriate data structures. We aim to improve *MuteIt*'s response time in the future.

9 LIMITATIONS AND DISCUSSION

Continuous silent speech recognition: Currently, *MuteIt* can recognize isolated command words. We ask the users to speak slowly to obtain jaw movements during unvoiced speech articulation. In the future, we will explore continuous unvoiced speech recognition at a regular speaking pace. This can be performed by identifying word boundaries within a sentence and then performing recognition of individual words using *MuteIt*'s pipeline.
Impact of dictionary size: Currently we have a dictionary of 1100 words from which we estimate prior and transition probability for the bi-directional particle filter. However, if we increase the size of the dictionary to a larger number (>10M) there might be words with non-unique phonemic maps. This is due to ambiguity between phonemes within the same viseme group. For example, after identifying every phonemic component of the words "map" (/mæp/) and "back" (/bæk/), we do not have information about the onset phonemes as /m/ and /b/ belong to the same viseme group. These cases can be mitigated during continuous unvoiced speech recognition as we can select the appropriate word based on the context.
Integration with commodity earables: Currently, we have a prototype with a twin-IMU setup. While there are commercially available earables like eSense [43] and AirPods which are equipped with IMUs, the placement of our IMUs is different than the ones present in these devices. In the future, we envision that the prototype can be miniaturized and commercial headphones can be instrumented with it.
Scalability to larger user cohort: We evaluated *MuteIt* with 20 users. Currently, we have a personalized model for each user. Though the data collection time is less than 3 minutes for achieving >95% word recognition accuracy, in the future, we believe as we collect more data we can train a generalized model.

10 RELATED WORK

Unvoiced speech recognition (USR) has been explored as an efficient alternative mode of interaction, especially with the growth of wearable devices. In fact, users are more readily accepting silent speech interaction technologies [66]. Efforts in unvoiced speech recognition broadly fall into two categories: contact-based methods (e.g., wearable devices) and contactless methods.

10.1 Contact-based Methods

Contact-based methods often require that sensors have direct contact with users' head and neck area for signal acquisition. Sensors mounted in contact with the face are used to measure muscle activities for unvoiced speech recognition [58, 60, 71]. AlterEgo [42] uses EMG electrodes placed around cheeks, lips, and jaw to capture neuro-muscular signals in internal speech articulators. Approaches relying on muscle vibration sensing are obtrusive and often not socially acceptable as skin electrodes are attached to the user's face around the cheek and lips [59]. Sensors placed on the articulators [27, 33, 53, 73, 75, 88], or even sensors retrofitted to masks [32] can capture the articulators' motion and hence infer unvoiced speech. SilentSpeller [48] tracks tongue movements using a dental retainer equipped with a capacitive touch sensor, enabling the user to type by spelling unvoiced words.

However, some of these techniques are intrusive, wherein magnetic sensors are mounted on the tongue or inside the mouth [33, 53, 75]. Sahni et al. [76] captured tongue and jaw movements related to unvoiced speech using a magnet glued to users' tongues. Head-mounted sensors are also used to detect hotwords via head motions [29], but are limited in the number of words they can detect. RFID Tattoo [88] utilizes battery-less and flexible RFID tattoos placed around the lips. However, it requires an RFID reader in the vicinity and has poor performance on words out of their vocabulary dataset. JawSense [46] uses an accelerometer placed on TMJ to identify isolated phonemic sounds but does not recognize words. Cameras and microphones placed close to the mouth [25, 49] can be affected by external visual and acoustic noise. In contrast to existing wearable systems, *MuteIt* is non-intrusive, can be integrated into commodity headphones making them socially acceptable, and can recognize unseen words.

10.2 Contactless USR

: Efforts in developing contactless systems for USR typically employ technologies such as vision, wireless [87], microphone [26, 35, 36, 84, 94], or ultrasound [14, 17, 50, 78]. Vision-based systems [12, 22, 30, 49, 65, 67, 82, 83, 91] utilize a camera for tracking facial features from a primary articulator, such as lips. Lip-Interact [83] can detect 44 commands using the front-facing camera on a smartphone with a 95% accuracy. However, these systems have some limitations: their performance is impacted by lighting conditions, camera angle, and line of sight, and they are computationally expensive. Moreover, most camera-based systems approach word recognition as a classification task, and to support more commands over time, the system will need to be retrained with a large number of samples for the new words. Lastly, inaudible acoustic signals generated by lip movements are captured using microphones [26, 35, 36, 84, 95, 96], and the Doppler shift is leveraged for word identification. SoundLip [95] takes inspiration from sequential models like Connectionist Temporal Classification (CTC) [28] and successfully uses them for silent speech interfaces using inaudible acoustic signals. Like *MuteIt*, SoundLip can be generalized to unseen words. However, the CTC model is hard to train [93] because of the network initialization process. One way to circumvent this issue is to use a very large amount of training data [77], although it requires significant time and effort from the users and is not always feasible. SoundLip requires users to articulate 40 samples of each word, while we have five samples of each word and still achieve comparable accuracy. Another reason for not using CTC models is the requirement of a strong language model [90] for high accuracy. Since *MuteIt* focuses on isolated words, we cannot use a language model and context. EarCommand [40] recognizes silent speech commands from ear canal deformation using in-ear earphones for a limited dictionary of commands, but the performance drops for 3 syllable words as longer words introduce motion and unexpected noise in the system. In contrast, *MuteIt* achieves similar accuracy and can be easily scaled to new words because it reconstructs a word from its constituent components.

11 CONCLUSION AND FUTURE WORK

In this paper, we present *MuteIt*, an unvoiced command recognition technique that leverages jaw motion. To the best of our knowledge, *MuteIt* is the only system that achieves command recognition with an accuracy of 94.8% by tracking only a secondary articulator. *MuteIt* uses a twin-IMU earable setup that can cancel motion artifacts caused by walking and head movements. In contrast to prior approaches, *MuteIt* does not train a word classifier. Instead, we identify the components of a word and reconstruct the word by modeling it as an estimation problem. This enables our system to be scaled to a large number of words without the need for retraining.

In the future, we plan to explore miniaturization of the prototype and use *MuteIt* for continuous unvoiced speech recognition. We also plan to evaluate the system for day-to-day activities that involve non-speech-related jaw motions like chewing and yawning. These activities also induce jaw movement and cannot be removed via the twin-IMU design. Therefore, context-based approaches will be explored to further enhance the system's

robustness for day-to-day use. We believe that this work sets a new paradigm in non-intrusive unvoiced speech recognition.

ACKNOWLEDGMENTS

REFERENCES

[1] Luís Aguiar-Conraria and Maria Joana Soares. 2011. *The continuous wavelet transform: A primer.* Technical Report. NIPE-Universidade do Minho.

[2] Amazon. 2021. Most used voice assistants in the United States in 2021, by age group. https://www.statista.com/statistics/1274429/voice-assistants-use-by-age-group-united-states/

[3] Amazon. 2022. Amazon Alexa. https://developer.amazon.com/en-US/alexa

[4] IoT Analytics. 2021. State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion. https://iot-analytics.com/number-connected-iot-devices/

[5] Toshiyuki Ando, Yuki Kubo, Buntarou Shizuki, and Shin Takahashi. 2017. Canalsense: Face-related movement recognition system based on sensing air pressure in ear canals. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology.* 679–689.

[6] Apple. 2022. Siri Apple. https://www.apple.com/siri/

[7] Helen L Bear. 2017. Decoding visemes: improving machine lipreading. arXiv:1710.01288 [cs.CV]

[8] Helen L Bear, Gari Owen, Richard Harvey, and Barry-John Theobald. 2014. Some observations on computer lip-reading: moving from the dream to the reality. In *Optics and Photonics for Counterterrorism, Crime Fighting, and Defence X; and Optical Materials and Biomaterials in Security and Defence Systems Technology XI*, Vol. 9253. International Society for Optics and Photonics, 92530G.

[9] Štefan Beňuš and Marianne Pouplier. 2011. Jaw movement in vowels and liquids forming the syllable nucleus. In *Twelfth Annual Conference of the International Speech Communication Association.*

[10] J. Bird, D. V. M. Bishop, and N. H. Freeman. 1995. Phonological Awareness and Literacy Development in Children With Expressive Phonological Impairments. *Journal of Speech, Language, and Hearing Research* 38, 2 (April 1995), 446–462. https://doi.org/10.1044/jshr.3802.446 Publisher: American Speech-Language-Hearing Association.

[11] Peter Birkholz, Simon Stone, Klaus Wolf, and Dirk Plettemeier. 2018. Non-Invasive Silent Phoneme Recognition Using Microwave Signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26, 12 (Dec. 2018), 2404–2411. https://doi.org/10.1109/TASLP.2018.2865609 Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[12] Hang Chen, Jun Du, Yu Hu, Li-Rong Dai, Chin-Hui Lee, and Bao-Cai Yin. 2020. Lip-reading with Hierarchical Pyramidal Convolution and Self-Attention. arXiv:2012.14360 [cs.CV]

[13] Alberto Compagno, Mauro Conti, Daniele Lain, and Gene Tsudik. 2017. Don't Skype & Type! Acoustic Eavesdropping in Voice-Over-IP. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17).* Association for Computing Machinery, New York, NY, USA, 703–715. https://doi.org/10.1145/3052973.3053005

[14] Tamás Gábor Csapó, Csaba Zainkó, László Tóth, Gábor Gosztolya, and Alexandra Markó. 2020. Ultrasound-based articulatory-to-acoustic mapping with WaveGlow speech synthesis. *arXiv preprint arXiv:2008.03152* (2020).

[15] Na Le Dang, Tyler B Hughes, Varun Krishnamurthy, and S Joshua Swamidass. 2016. A simple model predicts UGT-mediated metabolism. *Bioinformatics* 32, 20 (2016), 3183–3189.

[16] P. Delacourt and C. Wellekens. 1999. Audio data indexing: Use of second-order statistics for speaker-based segmentation. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, Vol. 2. 959–963 vol.2. https://doi.org/10.1109/MMCS.1999.778619

[17] B. Denby, Y. Oussar, G. Dreyfus, and M. Stone. 2006. Prospects for a Silent Speech Interface using Ultrasound Imaging. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Vol. 1. I–I. https://doi.org/10.1109/ICASSP.2006.1660033

[18] Richard P Di Fabio. 1998. Physical therapy for patients with TMD: a descriptive study of treatment, disability, and health status. *Journal of orofacial pain* 12, 2 (1998).

[19] Collins Dictionary. 2021. Collins Dictionary. https://www.collinsdictionary.com/

[20] Elago. 2021. AirPods Pro EarHook. https://www.elago.com/new/airpods-pro-earhook-white-lkt4w

[21] Donna Erickson. 2002. Articulation of Extreme Formant Patterns for Emphasized Vowels. *Phonetica* 59, 2-3 (2002), 134–149. https://doi.org/10.1159/000066067

[22] Adriana Fernandez-Lopez and Federico M. Sukno. 2018. Survey on automatic lip-reading in the era of deep learning. *Image and Vision Computing* 78 (2018), 53–72. https://doi.org/10.1016/j.imavis.2018.07.002

[23] Cletus G Fisher. 1968. Confusions among visually perceived consonants. *Journal of speech and hearing research* 11, 4 (1968), 796–804.

[24] Fortune Business Insights. 2021. Speech and Voice Recognition Market Size. https://tinyurl.com/yyyxe4rk

[25] Masaaki Fukumoto. 2018. SilentVoice: Unnoticeable Voice Input by Ingressive Speech. In *Proceedings of UIST '18* (Berlin, Germany). Association for Computing Machinery, New York, NY, USA, 237–246. https://doi.org/10.1145/3242587.3242603

[26] Yang Gao, Yincheng Jin, Jiyang Li, Seokmin Choi, and Zhanpeng Jin. 2020. EchoWhisper: Exploring an Acoustic-Based Silent Speech Interface for Smartphone Users. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3, Article 80 (Sept. 2020), 27 pages. https://doi.org/10.1145/3411830

[27] Jose A. Gonzalez, Lam A. Cheah, Angel M. Gomez, Phil D. Green, James M. Gilbert, Stephen R. Ell, Roger K. Moore, and Ed Holdsworth. 2017. Direct Speech Reconstruction From Articulatory Sensor Data by Machine Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25, 12 (2017), 2362–2374. https://doi.org/10.1109/TASLP.2017.2757263

[28] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.

[29] Jiaxi Gu, Zhiwen Yu, and Kele Shen. 2020. Alohomora: Motion-Based Hotword Detection in Head-Mounted Displays. *IEEE Internet of Things Journal* 7, 1 (2020), 611–620. https://doi.org/10.1109/JIOT.2019.2946593

[30] J. Han, L. Shao, D. Xu, and J. Shotton. 2013. Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics* 43, 5 (2013), 1318–1334. https://doi.org/10.1109/TCYB.2013.2265378

[31] Theodore P. Hill. 2009. Conflations of Probability Distributions. arXiv:0808.1808 [math.PR]

[32] Hirotaka Hiraki and Jun Rekimoto. 2021. SilentMask: Mask-Type Silent Speech Interface with Measurement of Mouth Movement. In *Augmented Humans Conference 2021* (Rovaniemi, Finland) *(AHs'21)*. Association for Computing Machinery, New York, NY, USA, 86–90. https://doi.org/10.1145/3458709.3458985

[33] Robin Hofe, Stephen R. Ell, Michael J. Fagan, James M. Gilbert, Phil D. Green, Roger K. Moore, and Sergey I. Rybchenko. 2013. Small-vocabulary speech recognition using a silent speech interface based on magnetic sensing. *Speech Communication* 55, 1 (2013), 22–32. https://doi.org/10.1016/j.specom.2012.02.001

[34] Qiang Huang, Yongxiong Wang, and Zhong Yin. 2020. View-based weight network for 3D object recognition. *Image and Vision Computing* 93 (2020), 103828.

[35] Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. 2010. Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips. *Speech Communication* 52, 4 (2010), 288–300.

[36] Thomas Hueber, Gérard Chollet, Bruce Denby, and Maureen Stone. 2008. Acquisition of ultrasound, video and acoustic speech data for a silent speech interface application. *Proc. of ISSP* (2008), 365–369.

[37] Monsoon Solutions Inc. 2022. Monsoon Power Monitor. https://www.msoon.com/online-store

[38] Madeline Jefferson. 2019. Usability of Automatic Speech Recognition Systems for Individuals with Speech Disorders: Past, Present, Future, and A Proposed Model. *undefined* (2019). https://www.semanticscholar.org/paper/Usability-of-Automatic-Speech-Recognition-Systems-A-Jefferson/73eefd141f43750b3ae0648e6ef099597e24c6c9

[39] Frederick Jelinek. 1997. *Statistical methods for speech recognition.* MIT press.

[40] Yincheng Jin, Yang Gao, Xuhai Xu, Seokmin Choi, Jiyang Li, Feng Liu, Zhengxiong Li, and Zhanpeng Jin. 2022. EarCommand: " Hearing" Your Silent Speech Commands In Ear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–28.

[41] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *British Journal of Applied Science & Technology* 7, 4 (2015), 396.

[42] Arnav Kapur, Shreyas Kapur, and Pattie Maes. 2018. AlterEgo: A Personalized Wearable Silent Speech Interface. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. Association for Computing Machinery, New York, NY, USA, 43–53. https://doi.org/10.1145/3172944.3172977

[43] Fahim Kawsar, Chulhong Min, Akhil Mathur, Marc Van den Broeck, Utku Günay Acer, and Claudio Forlivesi. 2018. esense: Earable platform for human sensing. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services.* 541–541.

[44] Rachel Kaye, Christopher G Tang, and Catherine F Sinclair. 2017. The electrolarynx: voice restoration after total laryngectomy. *Medical Devices (Auckland, NZ)* 10 (2017), 133.

[45] Gokce Keskin, Tyler Lee, Cory Stephenson, and Oguz H Elibol. 2019. Measuring the effectiveness of voice conversion on speaker identification and automatic speech recognition systems. *arXiv preprint arXiv:1905.12531* (2019).

[46] Prerna Khanna, Tanmay Srivastava, Shijia Pan, Shubham Jain, and Phuc Nguyen. 2021. JawSense: recognizing unvoiced sound using a low-cost ear-worn system. In *Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications.* 44–49.

[47] Rohan Khanna, Daegun Oh, and Youngwook Kim. 2019. Through-Wall Remote Human Voice Recognition Using Doppler Radar With Transfer Learning. *IEEE Sensors Journal* 19, 12 (June 2019), 4571–4576. https://doi.org/10.1109/JSEN.2019.2901271 Conference Name: IEEE Sensors Journal.

[48] Naoki Kimura, Tan Gemicioglu, Jonathan Womack, Richard Li, Yuhui Zhao, Abdelkareem Bedri, Zixiong Su, Alex Olwal, Jun Rekimoto, and Thad Starner. 2022. SilentSpeller: Towards mobile, hands-free, silent speech text entry using electropalatography. In *CHI Conference on Human Factors in Computing Systems*. 1–19.

[49] Naoki Kimura, Kentaro Hayashi, and Jun Rekimoto. 2020. TieLent: A Casual Neck-Mounted Mouth Capturing Device for Silent Speech Interaction. In *Proceedings of the International Conference on Advanced Visual Interfaces* (Salerno, Italy) *(AVI '20)*. Association for Computing Machinery, New York, NY, USA, Article 33, 8 pages. https://doi.org/10.1145/3399715.3399852

[50] Naoki Kimura, Michinari Kono, and Jun Rekimoto. 2019. SottoVoce: An Ultrasound Imaging-Based Silent Speech Interaction Using Deep Neural Networks. In *Proceedings of CHI 2019* (Glasgow, Scotland Uk). Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300376

[51] Mbient Lab. 2020. Mbient IMU. https://mbientlab.com/metamotionr/

[52] James R Lewis. 2018. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction* 34, 7 (2018), 577–590.

[53] Richard Li, Jason Wu, and Thad Starner. 2019. TongueBoard: An Oral Interface for Subtle Input. In *Proceedings of the 10th Augmented Human International Conference 2019* (Reims, France) *(AH2019)*. Association for Computing Machinery, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3311823.3311831

[54] Rochelle Lieber. [n.d.]. Point and manner of articulation of English consonants and vowels. *Introducing Morphology* ([n. d.]), xii–xii. https://doi.org/10.1017/cbo9780511808845.003

[55] LifeWire. 2021. Top commands. https://www.lifewire.com/top-google-assistant-and-google-home-commands-4158256

[56] Ian Maddieson. 2013. Voicing and gaps in plosive systems. *The world atlas of language structures online* (2013).

[57] Magoosh. 2022. 44 Phonemes In English And Other Sound Blends. https://magoosh.com/english-speaking/44-phonemes-in-english-and-other-sound-blends/

[58] Hiroyuki Manabe, Akira Hiraiwa, and Toshiaki Sugimura. 2003. Unvoiced speech recognition using EMG-mime speech recognition. In *CHI'03 extended abstracts on Human factors in computing systems*. 794–795.

[59] Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering* 15, 4 (2018), 046031.

[60] Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering* 15, 4 (2018), 046031.

[61] Phuc Nguyen, Nam Bui, Anh Nguyen, Hoang Truong, Abhijit Suresh, Matt Whitlock, Duy Pham, Thang Dinh, and Tam Vu. 2018. TYTH-Typing On Your Teeth: Tongue-Teeth Localization for Human-Computer Interface. In *Proceedings of MobiSys 2018* (Munich, Germany). Association for Computing Machinery, New York, NY, USA, 269–282. https://doi.org/10.1145/3210240.3210322

[62] Diane Corcoran Nielsen and Barbara Luetke-Stahlman. 2002. Phonological Awareness: One Key to the Reading Proficiency of Deaf Children. *American Annals of the Deaf* 147, 3 (2002), 11–19. https://doi.org/10.1353/aad.2012.0213 Publisher: Gallaudet University Press.

[63] The University of Reading. 2021. The production of speech sounds. http://www.personal.rdg.ac.uk/~llsroach/phon2/artic-basics.htm

[64] John J. Ohala and Haruko Kawasaki-Fukumori. [n.d.]. Alternatives to the sonority hierarchy for explaining segmental sequential constraints. *Language and its Ecology* ([n. d.]). https://doi.org/10.1515/9783110805369.343

[65] Laxmi Pandey and Ahmed Sabbir Arif. 2021. *LipType: A Silent Speech Recognizer Augmented with an Independent Repair Model*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445565

[66] Laxmi Pandey, Khalad Hasan, and Ahmed Sabbir Arif. 2021. *Acceptability of Speech and Silent Speech Input Methods in Private and Public*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3411764.3445430

[67] Yuru Pei, Tae-Kyun Kim, and Hongbin Zha. 2013. Unsupervised random forest manifold alignment for lipreading. In *Proceedings of the IEEE International Conference on Computer Vision*. 129–136.

[68] Physiopedia. 2021. TMJ Anatomy. https://www.physio-pedia.com/TMJ_Anatomy

[69] PlayStore. [n.d.]. Sound Meter. https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=en_US&gl=US

[70] Jay Prakash, Zhijian Yang, Yu-Lin Wei, Haitham Hassanieh, and Romit Roy Choudhury. 2020. EarSense: Earphones as a Teeth Activity Sensor. In *Proceedings of MobiCom 2020* (London, United Kingdom). Association for Computing Machinery, New York, NY, USA, Article 40, 13 pages. https://doi.org/10.1145/3372224.3419197

[71] Amanda Purington, Jessie G. Taft, Shruti Sannon, Natalya N. Bazarova, and Samuel Hardman Taylor. 2017. "Alexa is My New BFF": Social Roles, User Satisfaction, and Personification of the Amazon Echo. In *Proceedings of CHI EA 2017* (Denver, Colorado, USA). Association for Computing Machinery, New York, NY, USA, 2853–2859. https://doi.org/10.1145/3027063.3053246

[72] L.R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286. https://doi.org/10.1109/5.18626

[73] Jun Rekimoto and Yu Nishimura. 2021. Derma: Silent Speech Interaction Using Transcutaneous Motion Sensing. In *Augmented Humans Conference 2021* (Rovaniemi, Finland) *(AHs'21)*. Association for Computing Machinery, New York, NY, USA, 91–100. https://doi.org/10.1145/3458709.3458941

[74] Sherry Ruan, Jacob O Wobbrock, Kenny Liou, Andrew Ng, and James Landay. 2016. Speech is 3x faster than typing for english and mandarin text entry on mobile devices. *arXiv preprint arXiv:1608.07323* (2016).

[75] Himanshu Sahni, Abdelkareem Bedri, Gabriel Reyes, Pavleen Thukral, Zehua Guo, Thad Starner, and Maysam Ghovanloo. 2014. The Tongue and Ear Interface: A Wearable System for Silent Speech Recognition. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers* (Seattle, Washington) *(ISWC '14)*. Association for Computing Machinery, New York, NY, USA, 47–54. https://doi.org/10.1145/2634317.2634322

[76] Himanshu Sahni, Abdelkareem Bedri, Gabriel Reyes, Pavleen Thukral, Zehua Guo, Thad Starner, and Maysam Ghovanloo. 2014. The Tongue and Ear Interface: A Wearable System for Silent Speech Recognition. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers* (Seattle, Washington) *(ISWC '14)*. Association for Computing Machinery, New York, NY, USA, 47–54. https://doi.org/10.1145/2634317.2634322

[77] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. 2015. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947* (2015).

[78] Amin Honarmandi Shandiz and László Tóth. 2021. Voice Activity Detection for Ultrasound-based Silent Speech Interfaces using Convolutional Neural Networks. *arXiv preprint arXiv:2105.13718* (2021).

[79] Noah H Silbert. 2012. Syllable structure and integration of voicing and manner of articulation information in labial consonant identification. *The Journal of the Acoustical Society of America* 131, 5 (2012), 4076–4086.

[80] Jorge Silva and Shrikanth Narayanan. 2008. Upper Bound Kullback–Leibler Divergence for Transient Hidden Markov Models. *IEEE Transactions on Signal Processing* 56, 9 (2008), 4176–4188. https://doi.org/10.1109/TSP.2008.924137

[81] Cheryl Smith Gabig. 2010. Phonological Awareness and Word Recognition in Reading by Children With Autism. *Communication Disorders Quarterly* 31, 2 (Feb. 2010), 67–85. https://doi.org/10.1177/1525740108328410 Publisher: SAGE Publications Inc.

[82] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6447–6456.

[83] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-interact: Improving mobile device interaction with silent speech commands. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 581–593.

[84] J. Tan, C. Nguyen, and X. Wang. 2017. SilentTalk: Lip reading through ultrasonic sensing on mobile phones. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2017.8057099

[85] Mohamed Trabelsi, Jin Cao, and Jeff Heflin. 2021. SeLaB: Semantic Labeling with BERT. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[86] Kwang-Hyun Uhm, Seung-Won Jung, Moon Hyung Choi, Hong-Kyu Shin, Jae-Ik Yoo, Se Won Oh, Jee Young Kim, Hyun Gi Kim, Young Joon Lee, Seo Yeon Youn, et al. 2021. Deep learning for end-to-end kidney cancer diagnosis on multi-phase abdominal computed tomography. *NPJ Precision Oncology* 5, 1 (2021), 1–6.

[87] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni. 2016. We Can Hear You with Wi-Fi! *IEEE Transactions on Mobile Computing* 15, 11 (2016), 2907–2920. https://doi.org/10.1109/TMC.2016.2517630

[88] Jingxian Wang, Chengfeng Pan, Haojian Jin, Vaibhav Singh, Yash Jain, Jason I. Hong, Carmel Majidi, and Swarun Kumar. 2019. RFID Tattoo: A Wireless Platform for Speech Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 155 (Dec. 2019), 24 pages. https://doi.org/10.1145/3369812

[89] Xinyuan Wang, Make Tao, Runpu Wang, and Likui Zhang. 2021. Reduce the medical burden: An automatic medical triage system using text classification BERT based on Transformer structure. In *2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*. IEEE, 679–685.

[90] Zhangyu Xiao, Zhijian Ou, Wei Chu, and Hui Lin. 2018. Hybrid CTC-Attention based End-to-End Speech Recognition using Subword Units. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. 146–150. https://doi.org/10.1109/ISCSLP.2018.8706675

[91] Wai Chee Yau, Sridhar Poosapadi Arjunan, and Dinesh Kant Kumar. 2008. Classification of voiceless speech using facial muscle activity and vision based techniques. In *TENCON 2008-2008 IEEE Region 10 Conference*. IEEE, 1–6.

[92] Hallie Kay Yopp. 1988. The validity and reliability of phonemic awareness tests. *Reading research quarterly* (1988), 159–177.

[93] Dong Yu and Jinyu Li. 2017. Recent progresses in deep learning based acoustic models. *IEEE/CAA Journal of automatica sinica* 4, 3 (2017), 396–409.

[94] Qian Zhang, Dong Wang, Run Zhao, and Yinggang Yu. 2021. SoundLip: Enabling Word and Sentence-Level Lip Interaction for Smart Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 1, Article 43 (March 2021), 28 pages. https://doi.org/10.1145/3448087

[95] Qian Zhang, Dong Wang, Run Zhao, and Yinggang Yu. 2021. SoundLip: Enabling Word and Sentence-level Lip Interaction for Smart Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–28.

[96] Yongzhao Zhang, Wei-Hsiang Huang, Chih-Yun Yang, Wen-Ping Wang, Yi-Chao Chen, Chuang-Wen You, Da-Yuan Huang, Guangtao Xue, and Jiadi Yu. 2020. Endophasia: Utilizing acoustic-based imaging for issuing contact-free silent speech commands. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–26.