MELDER: The Design and Evaluation of a Real-time Silent Speech Recognizer for Mobile Devices

Laxmi Pandey
Inclusive Interaction Lab
University of California, Merced
Merced, California, United States
lpandey@ucmerced.edu

Ahmed Sabbir Arif
Inclusive Interaction Lab
University of California, Merced
Merced, California, United States
asarif@ucmerced.edu

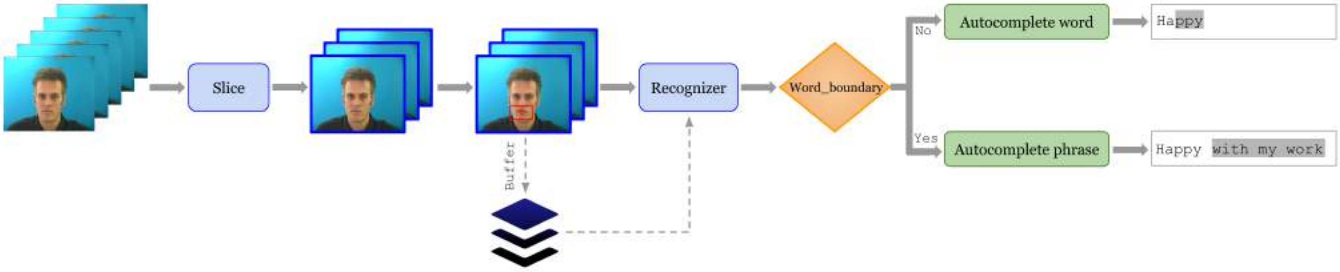


Figure 1: A high-level overview of MELDER. Short overlapping video clips are sliced to extract lip landmarks for character prediction, which are appended to a buffer to reduce the processing time by simultaneously slicing and recognizing them. As character predictions are made, the strings are auto-completed with the most probable words, then eventually with the most probable phrases.

ABSTRACT

Silent speech is unaffected by ambient noise, increases accessibility, and enhances privacy and security. Yet current silent speech recognizers operate in a phrase-in/phrase-out manner, thus are slow, error prone, and impractical for mobile devices. We present MELDER, a Mobile Lip Reader that operates in real-time by splitting the input video into smaller temporal segments to process them individually. An experiment revealed that this substantially improves computation time, making it suitable for mobile devices. We further optimize the model for everyday use by exploiting the knowledge from a high-resource vocabulary using a transfer learning model. We then compare MELDER in both stationary and mobile settings with two state-of-the-art silent speech recognizers, where MELDER demonstrated superior overall performance. Finally, we compare two visual feedback methods of MELDER with the visual feedback method of Google Assistant. The outcomes shed light on how these proposed feedback methods influence users' perceptions of the model's performance.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; **User studies**; *Interaction paradigms*; *Text input*.

CHI '24, May 11–16, 2024, Honolulu, HI, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642348

KEYWORDS

Silent speech, digital lip reading, image processing, deep learning, transfer learning, language modeling, visual feedback, text input

ACM Reference Format:
Laxmi Pandey and Ahmed Sabbir Arif. 2024. MELDER: The Design and Evaluation of a Real-time Silent Speech Recognizer for Mobile Devices. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/3613904.3642348

1 INTRODUCTION

Speech input, an auditory-based language processing technique that transcribes acoustic signals into text, stands out as one of the most intuitive and efficient means of engaging with mobile devices. It holds the potential to enhance user comfort and productivity, particularly when conventional input methods like touchscreens and physical keyboards prove inefficient, cumbersome, or inconvenient [89]. Moreover, it serves as a crucial accessibility feature, empowering individuals with limited motor skills to seamlessly interact with mobile technology without reliance on manual dexterity. This functionality also proves invaluable for those experiencing situationally-induced impairments and disabilities (SIID), a category encompassing instances where hand use is restricted due to concurrent tasks, glove-wearing, or minor injuries [92]. However, it is important to acknowledge that while speech input excels in numerous interaction scenarios, its suitability may be compromised in situations characterized by high ambient noise levels, privacy and security considerations, or pre-existing speech impairments [26, 27].

Silent speech input, an image-based language processing method that translates users' lip movements into textual content, presents a promising solution to address a multitude of challenges [81]. Its independence from acoustic cues allows for versatile application, thriving even in noisy or sensitive environments like libraries or museums. Moreover, it significantly bolsters privacy and security, given the limited number of individuals skilled in lip reading. In fact, studies indicate that even those skilled in lip reading can typically comprehend only about 30-45% of spoken English [65], further underscoring the privacy and confidentiality advantages of silent speech. Silent speech further promotes inclusivity by accommodating individuals who are unable to vocalize or have speech disorders, thereby making communication with computers more accessible.

In pursuit of optimal silent speech recognition, researchers have explored various sensor-based techniques, achieving high accuracy in speech transcription [73, 78, 83, 88, 107]. However, these approaches often entail invasive, unwieldy, and non-portable setups, rendering them impractical in real-world scenarios. Recent endeavors have aimed to harness video-based recognition, commonly referred to as digital lip reading, to facilitate silent speech communication [3, 9, 16, 18]. Yet, many of these models are primarily tailored for high-performance computing devices, like desktop computers [3, 9, 77]. Even with ample computational resources, these models exhibit sluggish response times, susceptibility to errors, and a lack of real-time functionality, making them unsuitable for mobile devices. Additionally, existing models tend to support only a limited, pre-determined vocabulary, hampering their applicability in everyday conversational interactions.

A well-known challenge in developing deep learning models is the demand for substantial data for training, particularly datasets tailored to specific vocabularies, a time-consuming and arduous process. Thus, the imperative arises to design models robust enough to operate effectively with modest data quantities, without compromising performance. Furthermore, the exploration of interface and feedback mechanisms tailored for silent speech interaction on mobile devices has remained ignored in the literature. A mobile-optimized interface is of paramount importance, as it directly impacts user experience and usability. The creation of a swifter, more accurate, and real-time silent speech recognition system optimized for mobile devices, thus, holds the potential to serve as a versatile medium for input and interaction tasks, seamlessly integrating into daily routines.

This paper presents MELDER, a Mobile Lip Reader optimized for performance and usability on mobile devices. The contribution of the work is five-fold. First, it develops a new real-time silent speech recognizer that improves recognition performance on mobile devices by splitting the input video into smaller temporal segments, then processing them individually. Second, it introduces a transfer learning approach aimed at enhancing the performance of silent speech recognition models in everyday conversational contexts. Through a study, we validate the applicability of this approach, demonstrating its effectiveness not only with MELDER but also

Table 1: A summary of experiments conducted in this work, including conditions, number of phrases, sample size, and total number of videos used in the experiments.

Experiment	Conditions		Phrases	Sample Size	Total Videos
1	Windowing Functions (**4**)	Models (**3**)	30 × 3 = 90 random [77]	$N = 12$	90 × 12 = 1,080
	Linear: $y = x + 5$ *Linear:* $y = 2x + 5$ *Non-linear:* $y = x^3$ *Non-linear:* $y = 2^x$	*LipNet* *Transformer* *LipType*			
2	Transfer Learning Strategies (**3**)	Models (**3**)	30 random [69]	$N = 12$	30 × 12 = 360
	Finetune_Last *Finetune_Visual_Frontend* *Finetune_Sequence*	*LipNet* *Transformer* *LipType*			
3	Models in Stationary Condition (**3**)		30 random [110]	$N = 20$	30 × 20 = 600
	RT-LipNet *RT-Transformer* *MELDER*				
4	Models in Mobile Condition (**3**)		30 random [110]	$N = 6$	30 × 6 = 180
	RT-LipNet *RT-Transformer* *MELDER*				
5	Method + Visual Feedback (**3**)		30 random [110]	$N = 12$	30 × 3 × 12 = 1,080
	Google Voice Assistant *MELDER + Word-level Feedback* *MELDER + Phrase-level Feedback*				

with other pre-trained models. Third, a comparative evaluation of MELDER against two state-of-the-art silent speech recognition models, assessing their performance in both stationary (seated position) and mobile settings (while walking). Fourth, it introduces two visual feedback methods designed for silent speech recognition systems to keep the users informed about the ongoing recognition process. These methods are compared with the feedback method of Google Assistant in a qualitative study. Fifth, the dataset[1] and the source code and other material produced in this study[2] are freely available to download for research and development, encouraging replication and further investigations in the area. Table 1 provides a summary of all the experiments conducted in this work, detailing aspects such as the conditions under which each experiment was carried out, the number of phrases used, the sample size of participants, and the total number of videos utilized in these experiments.

2 RELATED WORK

Silent speech input is a form of unspoken communication that allows users to interact with mobile devices without making any audible sounds. As opposed to speech, this method allows users to communicate effectively with mobile devices without invading their privacy and security or disrupting the environment. There have been several previous attempts at achieving silent speech communication through sensor-based recognition and video-based recognition.

2.1 Sensor-Based Recognition

Speech production mechanism is composed of several stages, starting from the conceptual idea, followed by brain signals, muscular activity, and, finally, sound waves. In order to develop silent speech interfaces, researchers acquire and process information from different stages of speech production. Some of them have utilized ultrasonic imaging to achieve silent speech interaction by measuring mouth and tongue movements through a sensor attached under the chin [24, 25, 29, 32, 38, 43, 44, 57, 117]. However, the technique requires applying gel to the skin to obtain the echo images, which is a complicated and expensive process.

Several studies have attempted to estimate speech by using electromyography (EMG) to measure muscle movement around the mouth [45, 48–50, 52, 70, 93, 112]. It is, however, difficult to estimate speech with EMG because it uses movement of the oral cavity as a basis for gesture recognition. As a result, there are fewer detectable commands and the user must learn new gestures instead of using existing speaking abilities. Another study recognizes tongue gestures with an electrostatic sensor array installed in the mouth [64]. Since the sensor must be placed in the mouth, it interferes with normal activities like eating and conversing. A recent work employs electropalatography (EPG) to observe tongue movements as users spell out a word to detect individual letters within the word [55]. The method uses a hidden Markov model (HMM) to decode 100Hz 16-dimensional signal from the EPG. Research has indicated that EPG is an effective approach for detecting individual letters in spelling (97% character accuracy), but it is considered intrusive and not very user-friendly due to its reliance on an artificial palate equipped with embedded sensors.

Fukumoto [30] propose the "ingressive speech" method, where a microphone is placed very close to the front of the mouth to capture soft speech sounds. However, placing the device in front of the mouth is obtrusive and hinders social interactions. Several studies have also attempted to achieve silent communication with non-audible murmurs (NAM) [39–41, 72] by using a microphone worn on the skin or throat to recognize speech. In this case, the user uses articulate respiratory sounds without vibrating their vocal folds (i.e., whispering). Whispers are, however, evident to bystanders, and a long-term use of whispers could negatively effect the vocal cords [90].

A few researchers have developed intracortical microelectrode Brain-Computer Interfaces (BCI) to predict users' intended speech data directly from the brain activity during speech production [13, 22, 86, 104, 105]. Several multimodal imaging systems have also been employed for speech recognition, mainly focused on tongue visualization [44]. Some have employed electromagnetic articulography (EMA) [28, 32, 38], electroencephalogram (EEG) [86], vibration sensors of glottal activity [73, 83, 88, 107], and speech motor cortex implants [10] to recover the speech produced without vibration of the vocal folds, by detecting tongue, facial, and throat movements. A recent study developed a wearable interface for detecting silent speech from neural signals captured by electrodes placed above the face [51]. However, the majority of these studies employ invasive, impractical, and non-portable setups, rendering them unsuitable for real-world applications.

Recent research has investigated the innovative approach of capturing vocal cord vibrations through millimeter-wave (mmWave) sensing [66, 114, 116] and using smartphones' acoustic sensors to detect continuous wave ultrasound signals for analyzing lip movements [31, 118]. While these methods are computationally lighter than image-based approaches and can be accurate in ideal scenarios, the recognition results can be influenced by both static environmental objects and subtle movements of the body or hand. Besides, these methods necessitate the device being in close proximity to the mouth, sometimes even requiring the user to hold the device near their mouth. This requirement could potentially impact usability, as it may be inconvenient or uncomfortable for users to maintain such close interaction with the device for extended periods or in various settings.

2.2 Video-Based Recognition

Recently, attempts have been made to enable silent speech communication using video-based recognition, referred to as lip reading or silent speech recognition [103]. It captures lip movements with a camera, then recognizes silently spoken words using image processing and language models [3, 6, 9, 11, 17, 18, 84, 100]. Initially, lip reading methods relied on handcrafted pipelines and statistical models for visual feature extraction and temporal modelling, limiting their generalizability.[34, 67, 75, 82, 87] (refer to [120] for a comprehensive review). However, with the advent of deep learning and the availability of large-scale lip-reading datasets, such as GRID [21], lip reading in-the-wild (LRW) [16], and lip reading sentences in-the-wild [4, 97], this field has been revitalized. Researchers initially

[1] MELDER Dataset: https://www.theiilab.com/resources/MELDER_Data.zip
[2] MELDER Source Code: https://github.com/theiilab/MELDER

Table 2: A high-level overview of recent silent speech recognition methods and their reported performances.

Research	Camera	Vocabulary	Size	WER (%)	Mode
TieLent, Kimura et al. [56]	Wearable	Command	15 commands	6.0	Offline
C-Face, Chen et al. [14]	Wearable	Command	8 commands	15.3	Offline
SpeeChin, Zhang et al. [117]	Wearable	Command	54 commands	9.5	Offline
Lip-Interact, Sun et al. [103]	Smartphone	Command	44 commands	4.6	Offline
LipType, Pandey and Arif [77]	Smartphone	Sentence	30 phrases/41 words	40.9	Offline
LipLearner, Su et al. [101]	Smartphone	Command	30 commands	1.2	Offline
MELDER	**Smartphone**	**Sentence**	**30 phrases/122 words**	**19.7**	**Real-time**

focused on estimating phoneme-level and word-level recognition [16, 100]. Koller et al. [60] trained a convolutional neural network (CNN) to differentiate between visemes[3] on a sign language dataset of signers mouthing words. Similarly, Noda et al. [74] used CNN to predict phonemes in spoken Japanese. Tamura et al. [106] used deep bottleneck features (DBF) to encode shallow input features, such as latent dirichlet allocation (LDA) and GA-based informative feature (GIF) [108] for word recognition. Petridis and Pantic [84] also utilized DBF to encode every video frame and trained a long short-term memory (LSTM) classifier for word-level classification. In contrast, Wand et al. [111] used an LSTM with histograms of oriented gradients (HoG) input features to recognize words. Another work developed CNN architectures for classifying multi-frame time series of lip movements [16]. Kashiwagi et al. [53] introduced a method that places emphasis on identifying shared viseme representations between normal and silent speech. This is achieved by employing metric learning techniques to acquire knowledge about visemes across different speech instances and within the same speech type. This approach enables the efficient utilization of silent speech data while accommodating variations within specific speech types. These approaches still cannot be adapted to make sentence-level sequence predictions due to their inability to handle variable sequence lengths.

More recently, researchers have focused their attention on adapting sentence-level recognition by modifying models for automatic speech recognition using LSTM-based sequence-to-sequence models [97] or connectionist temporal classification (CTC) approach [9, 94]. Another work has taken a hybrid approach, training an long short-term memory (LSTM)-based sequence-to-sequence model with an auxiliary CTC loss [85]. Researchers have also explored transformer-based architectures [2], convolution block variants [119], or hybrid architectures such as conformers [36]. Most of these models either make use of spatiotemporal convolutional neural networks (CNNs) with multiple 3D convolution layers [9, 94] or use lightweight approaches that combine a 3D layer applied frame-by-frame with a 2D one for visual feature extraction and short-term dynamic modeling [2, 16, 100]. LipType [77], on the other hand, use a hybrid approach by combining a shallow 3D-CNN and a deep squeeze and excited 2D-CNN [42], thereby modeling spatial and temporal interdependencies between channels, which led to a reduction of 57% in word errors compared to existing methods. Some have focused on audiovisual speech recognition that uses both acoustic and video channels to recognize speech using deep learning models [61].

Despite these improvements, existing video-based recognition models remain slow (refer to [95] for a comprehensive review). They take fourteen seconds or more to process a short English phrase, which makes them ineffective for everyday usage. In addition, these models do not operate in real-time, instead require the user to perform an action (e.g., pressing a button) or wait for a time-out period after speaking a phrase for the system to start processing it. This additional waiting time negatively impacts the user's perception of the model. We address these issues by automatically slicing the input video into shorter clips, processing the clips character-by-character in real-time, leveraging the insights gained from a high-resource vocabulary through a transfer learning model, and providing real-time visual feedback on the progress of speech recognition. Table 2 presents a summary of recent advancements in silent speech recognition, with a particular focus on its application in human-computer interaction (HCI) through a camera-based approach. This table emphasizes the efforts to utilize camera technologies, whether incorporated into wearable devices or smartphones, for capturing and interpreting silent speech cues.

2.3 Silent Command Recognition

A new research direction is centered around optimizing both sensor-based and image-based silent speech recognition models specifically for commands. Pandey and Arif [79], for example, proposed a stripped-down version of LipType [77] that can recognize silent commands almost as fast as state-of-the-art speech recognition models. Su et al. [101] proposed a semi-supervised model trained on public datasets that enables customizing commands using a few-shot silent speech customization framework. Kunimi et al. [62], on the other hand, designed a mask-shaped interface containing eight flexible and highly sensitive strain sensors to recognize commands with an existing EMG-based model [52]. Zhang et al. [118] designed EchoSpeech, which utilizes a glass-frame configuration featuring integrated speakers and microphones to project inaudible sound waves toward the skin. It employs a deep learning pipeline with connectionist temporal classification (CTC) loss to discern speech by capturing and analyzing the subtle skin deformations arising from silent utterances. Su et al. [102] used a spatiotemporal convolution network to enable rapid and precise interacting with big displays using gaze and silent commands. Jin et al. [47], in contrast, developed a earphone-based model that recognize commands using the relationship between the deformation of the ear canal and

[3] A viseme is the visual equivalent of a phoneme that represents the position of the face and the mouth when making a sound.

the movements of the articulator. [99] also used an ear-worn system to process jaw motion during word articulation to break each word signal into its constituent syllables, then each syllable into phonemes. These methods are often faster and more accurate than general silent speech models, primarily because they are tailored to recognize a limited set of specific commands and are not intended for everyday communication.

2.4 Applications of Silent Speech in HCI

Silent speech recognition technology [77, 81] holds significant promise for diverse applications in the field of HCI. This innovation provides a hands-free communication solution, particularly valuable in environments where vocalization may be impractical or socially discouraged, such as libraries or quiet workspaces. Users can seamlessly navigate applications, compose text messages, make calls, control smart devices, and perform online searches, all achieved through the simple act of forming words silently [101, 118]. Its ability to interpret silent lip movements makes it an invaluable assistive technology for individuals with speech impairments, enabling a more accessible mode of communication. In addition, silent speech recognition's multimodal capabilities find practical use in wearable devices, enabling users to interact through silent speech with devices like smartwatches and desktop computers [103]. Silent speech has also been used for hands-free selection with eye-gaze pointing [79, 102], offering performance, usability, and privacy benefits over conventional methods such as speech and dwell.

Silent speech also holds significant potential for emotion recognition in HCI applications [78]. By analyzing facial expressions and lip movements associated with silent speech, the technology could infer emotional states in various contexts. For instance, in adaptive user interfaces, if frustration is detected during a task, the system could offer additional assistance. This enables virtual agents and avatars to express empathy by responding to users with appropriate emotional cues. Emotion-aware assistive technologies benefit from recognizing emotional nuances, aiding individuals with autism in more effective communication. Educational applications could create personalized learning environments, adjusting content based on the student's emotional engagement. Silent speech recognition could also enhance gaming experiences by dynamically adjusting game elements according to the player's emotional responses.

There are various other interesting directions one could explore with silent speech recognition. For example, in robotics, silent speech recognition could facilitate more intuitive human-robot interaction, offering users the ability to convey commands without audible speech. Additionally, the technology holds promise in security applications, serving as a unique identifier for biometric authentication. In virtual and augmented reality settings, silent speech recognition could enhance user experiences by allowing silent communication with virtual characters and interfaces. Finally, in training scenarios, silent speech recognition could provide a platform for individuals to practice and refine their communication skills without the need for vocalization, making it a versatile tool in education and skill development. As this technology continues to evolve, its integration into HCI promises more inclusive, adaptable, and natural interaction paradigms.

3 MELDER: A MOBILE LIP READER

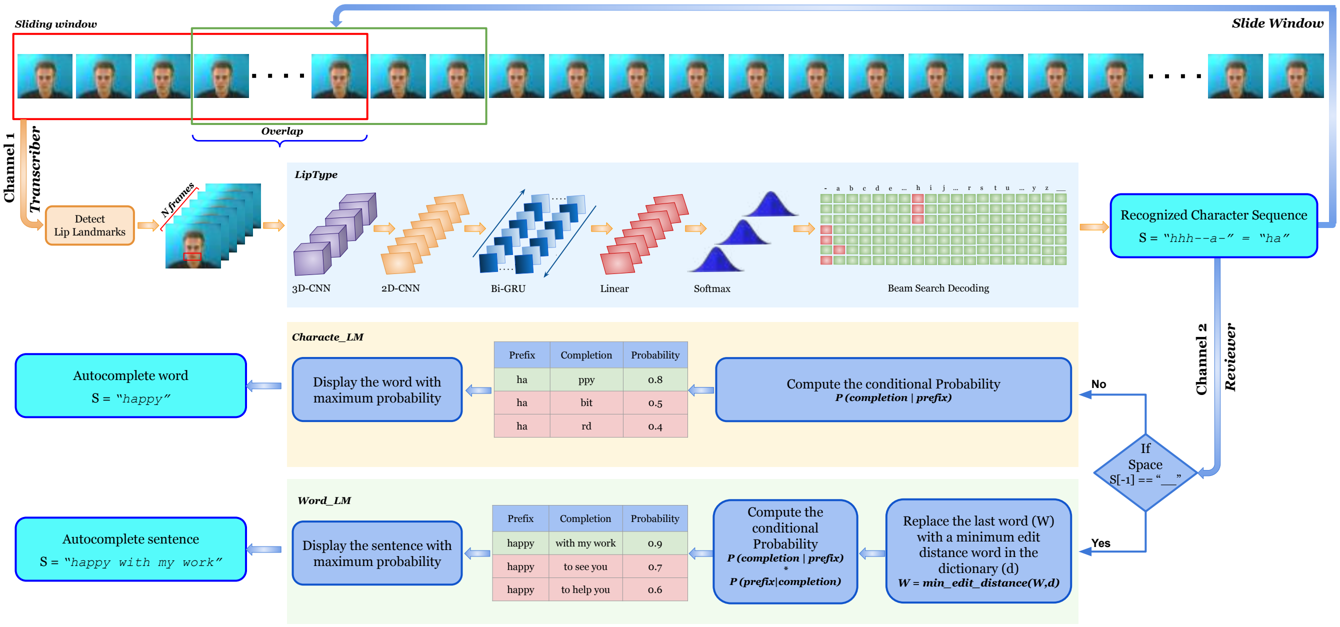
MELDER leverages LipType as its foundational model. LipType [77], an established end-to-end sentence-level model, translates a variable-length sequence of video frames into text. It achieves this through the integration of a shallow 3D-CNN (1-layer) with a deep 2D-CNN (34-layer ResNet [37]), enhanced by squeeze and excitation (SE) blocks (SE-ResNet). This configuration effectively captures both spatial and temporal information. The choice of SE-ResNet is strategic, as it adaptively recalibrates channel-wise feature responses by explicitly modeling the inter-dependencies between channels, thereby refining the quality of feature representations. Moreover, SE-ResNet is notable for its computational efficiency, adding only a minimal increase in model complexity and computational demands. For additional details, please refer to Section 4.1. First, MELDER enhances the model by introducing innovative transcriber and reviewer channels that run in parallel. This structure not only enables real-time processing but also provides users with continuous visual feedback during the recognition of silent speech.

3.1 The Transcriber Channel

The proposed transcriber channel consists of three sub-modules: a *windowing* frontend that splits the input video into smaller temporal segments, a *spatiotemporal feature extraction* module that takes a sequence of frames and outputs one feature vector per frame, and a *sequence modeling* module that inputs the sequence of per-frame feature vectors and outputs a sentence character by character. The model appends the sliced clip to the buffer for parallel processing. This cycle continues until the end of a video clip is detected. We must emphasize that the transcriber channel operates on a server, as modern smartphones do not possess the necessary storage and processing capacity to work with large datasets. Consequently, the results presented in this study may not be directly comparable to models that were tested exclusively on smartphones. Additionally, we acknowledge the concerns some users might have regarding the security of sending video clips to a server. Nonetheless, it is pertinent to point out that nearly all sophisticated real-time recognition systems, including Google Lens, Google Speech, Google Home, and Amazon Alexa, employ a similar server-based approach for processing significant volumes of data [33].

3.1.1 *Windowing.*

The channel slices video input into smaller segments. In order to determine the best windowing function in the defined context, we studied two linear ($y = x + 5$, $y = 2x + 5$) and two non-linear ($y = x^3$, $y = 2^x$) windowing functions, where x = window start frame and y = window end frame. Each function has an overlapping window of two frames. This was decided in lab trials with an existing silent speech recognition model [77], where we compared speed-accuracy trade-offs between 1–4 overlapping windows. We did not examine more than four frames because the average time per phoneme with silent speech is 176 ms [80], which corresponds to four frames (video frame rate is 25 frames per second). Since the model was already slicing a video into small chunks (~5 frames), reprocessing a large overlapped window increased the processing time without improving accuracy. However, using two frames as an overlapped window improved accuracy without substantially slowing the processing time (Table 3). The overlap



Prefix	Completion	Probability
ha	ppy	0.8
ha	bit	0.5
ha	rd	0.4

Prefix	Completion	Probability
happy	with my work	0.9
happy	to see you	0.7
happy	to help you	0.6

Figure 2: The architecture of MELDER. It consists of a transcriber channel and a reviewer channel, which run simultaneously. The transcriber channel slices a video and passes it to a 1-layer 3D CNN, followed by a 34-layer 2D SE-ResNet for spatiotemporal feature extraction. The features are then processed by two Bi-GRUs, a linear layer, and a softmax. Finally, the softmax output is decoded with a left-to-right beam search. The reviewer channel corrects both character-level and word-level errors and provides real-time visual feedback on the system's silent speech recognition process.

between the clips assures that any lost phonemes due to the slicing process are recovered using the information in the overlap frames.

Table 3: Performance of a silent speech recognition model with varying windowing size.

LipType	Windowing Size			
	1	2	3	4
Word Error Rate (WER)	28.9%	22.6%	22.5%	22.1%
Computation Time (CT)	0.4s	0.6s	1.1s	1.4s

We selected the windowing function based on certain assumptions. We chose linear functions because they have constant window sizes, possibly resulting in faster computations. For instance, $y = x + 5$ has a fixed length, thus likely to have a faster processing time, but the accuracy can suffer due to limited context. While larger window sizes, such as those used in $y = 2x + 5$, may increase accuracy, but may lead to extended processing times. Alternatively, for non-linear functions, the window size increases gradually rather than being constant. They may initially have a faster processing time with a lower accuracy. However, as the window size increases, the processing time will slow down and the accuracy is likely to rise. For this work, we selected non-linear functions based on their window interval size. For instance, $y = 2 * x$ has a gradual increase in the window size, while $y = x * 3$ has a steeper increase in the window size. Because the optimal windowing function for real-time processing within this context is unclear, we validated our choice in an experiment described in Section 4.

3.1.2 Spatiotemporal Feature Extraction. This module takes the sliced video chunk and extracts the mouth-centred cropped image of size H:100 × W:50 pixels per video frame. For this, videos are first pre-processed using the DLib face detector [58] and the iBug face landmark predictor [91] with 68 facial landmarks combined with Kalman filtering. Then, a mouth-centred cropped image is extracted by applying affine transformations. The sequence of T mouth-cropped frames are then passed to 3D-CNN, with a kernel dimension of T:5 × W:7 × H:7, followed by Batch Normalization (BN) [46] and Rectified Linear Units (ReLU) [5]. The extracted feature maps are then passed through a 34-layer 2D SE-ResNet that gradually decreases the spatial dimensions with depth, until the feature becomes a single dimensional tensor per time step.

3.1.3 Sequence Modeling. The extracted features are processed by 2-Bidirectional Gated Recurrent Units (Bi-GRUs) [15]. Each time-step of the GRU output is processed by a linear layer, followed by a softmax layer over the vocabulary and an end-to-end model is trained with connectionist temporal classification (CTC) loss [35]. The softmax output is then decoded with a left-to-right beam search [20] using the Stanford-CTC decoder [68] to recognize the spoken utterance. The model appends the recognized character to the buffer for post-processing. This cycle continues until the end of a phrase is detected. The model predicts the end of phrase when the newline character is detected.

3.2 The Reviewer Channel

The proposed reviewer channel corrects both character-level and word-level errors and provides real-time feedback by displaying the most probable candidate words and phrases for auto-completion. The process comprises of the following two steps.

3.2.1 Character-Level Corrector. The character-level model enables real-time word completion based on the sequence of characters or a prefix string obtained from the transcriber channel. As soon as the transcriber channel recognizes a character S, the model auto-completes the string with its most probable word ($\hat{S}$). The conditional probability can be formulated as:

$$P(S_1^n) = P(\hat{S}|S) = P(\text{completion} \mid \text{prefix}) \tag{1}$$

Consider, $S_{1:m}$ as the first m characters in string S and all completions must contain the prefix exactly, i.e.,

$$\begin{aligned} \hat{S}_{1:m} = S_{1:m} \quad \text{and} \quad P(\hat{S}_{1:n}|S_{1:m}) &= \\ P(\hat{S}_{m+1:n}|S_{1:m}) &= \\ P(\hat{S}_{m+1:n}|\hat{S}_{1:m}) & \end{aligned} \tag{2}$$

where n is the total length of a completion. As probabilities in the sequence domain contain exponentially many candidate strings, we simplified the model by calculating conditional probabilities recursively:

$$P(S_1^n) = P(\hat{S}_{m+1:n}|\hat{S}_{1:m}) = \underset{S_1,\ldots,S_n}{argmax} \prod_{t=m}^{n-1} P(\hat{S}_{t+1}|\hat{S}_{1:t}) \tag{3}$$

This requires modelling only $P(\hat{S}_{t+1}|\hat{S}_{1:t})$, which is the probability of the next character under the current prefix. For this, it computes $argmaxP(\hat{S}_{t+1}|\hat{S}_{1:t})$ using the prefix tree (Trie) data structure. Upon finding the most probable completion for the current prefix, the model automatically displays the auto-completion.

3.2.2 Word-Level Corrector. The module is activated only when a space character is detected. Upon detection, the sequence recognized so far is passed to the word-level n-gram language model (LM). First, it extracts the last word W from the recognized text, calculates edit distances [63] between W and each dictionary word d, then replaces W with a word that has the minimum edit distance. Second, it auto-completes the sentence by modelling the joint probability distribution of the given words and future words.

Formally, we consider a given string of t words, $W = W_1, W_2, \ldots, W_t$ and our goal to predict the future word sequence $(W_{t+1}, W_{t+2}, \ldots, W_{t+T})$. The conditional probability can be formulated as:

$$P_{combined}(W_1^T) = P(\text{completion} \mid \text{prefix}) \tag{4}$$

This model uses bidirectional n-grams to account for both forward and reverse directions. The combined probability of a sentence, thus, is computed by multiplying the forward and backward n-gram probability of each word:

$$\begin{aligned} P_{combined}(W_1^T) &= \\ P(\text{completion} \mid \text{prefix}) * P(\text{prefix} \mid \text{completion}) &= \\ P_{forward}(W_1^T) * P_{backward}(W_1^T) & \end{aligned} \tag{5}$$

In a forward n-gram, the conditional probability is estimated depending on the preceding words:

$$\begin{aligned} P_{forward}(W_1^T) &= \\ P_{forward}((W_{t+1}, W_{t+2}, \ldots, W_{t+T})|W_1, W_2, \ldots, W_t) &= \\ \underset{W_{t+1},\ldots,W_{t+T}}{argmax} \prod_{j=1}^{T} P(W_{t+j}|W_1, \ldots, W_{t+j-1}) & \end{aligned} \tag{6}$$

In contrast, in a backward n-gram, the probability of each word is estimated depending on the succeeding words:

$$\begin{aligned} P_{backward}(W_1^T) &= \\ P_{backward}(W_1, W_2, \ldots, W_t|(W_{t+1}, W_{t+2}, \ldots, W_{t+T})) &= \\ \underset{W_1,\ldots,W_t}{argmax} \prod_{j=1}^{T} P(W_1, \ldots, W_{t+j-1}|W_{t+j}) & \end{aligned} \tag{7}$$

Applying the values from Eq. 6 and Eq. 7, we get:

$$\begin{aligned} P_{combined}(W_1^T) &= \\ (P(W_1| < start >) * P(< start > |W_1))* & \\ (P(W_2|W_1^1) * P(W_1^1|W_2))* & \\ (P(W_3|W_1^2) * P(W_1^2|W_3))* & \\ \ldots* & \\ (P(< end > |W_T) * (W_T| < end >)) & \end{aligned} \tag{8}$$

Finally, the model predicts the most probable auto-completion of the given words and automatically adds it to the input text. We used COCA corpus [23], one of the largest publicly available and genre-balanced corpus of English, to train the reviewer modules. The dataset contains approximately 1 billion words, however, we extracted the top 200,000 sentences as vocabulary to reduce the computation time. The average perplexity[4] score for the model is 42.6, indicating that it is well-trained and can anticipate words accurately.

4 EXPERIMENT 1: SELECTION OF WINDOWING FUNCTION

We conducted an experiment to evaluate the performance of the four windowing functions proposed in Section 3.1.1 with three state-of-the-art silent speech recognizers. In video processing, windowing functions play a critical role by isolating specific sections of video frames for detailed analysis. This method significantly improves noise reduction and emphasizes important features within the chosen segments. In the context of MELDER, we specifically explored windowing functions to enable real-time silent speech

[4] Perplexity is the multiplicative inverse of the probability assigned to the sentence by the language model, normalized by the number of words in the sentence. The lower the perplexity the better the language model.

processing. This approach is designed to facilitate continuous processing without the need for a "stop" cue, such as pausing after completing a phrase spoken silently.

4.1 Silent Speech Recognition Models

We selected the following three pre-trained silent speech recognition models for this study.

(1) **LipNet** [9] model uses a neural network architecture for lip reading that maps variable-length sequences of video frames to text sequences, making use of deep 3-dimensional convolutions, a recurrent network, and the connectionist temporal classification loss [35], trained entirely end-to-end. It was trained on the GRID dataset (21,635 videos) [21], which comprises of short and formulaic videos that show a well-lit person's face while uttering a highly constrained vocabulary in a specific order.

(2) **Transformer** [3] model comprises of two sub-modules: a *spatio-temporal visual frontend* that takes a sequence of video frames to extract one feature vector per frame and a *sequence processing backend* comprised of encoder-decoder structure with multi-head attention layers [109] that generates character probabilities over the vocabulary. It was trained on Lip Reading in the Wild (LRW: 500 videos) [16] and the Lip Reading Sentences 2 (LRS2: 41,000 videos) [3] datasets.

(3) **LipType** [77] model follows the same architecture as LipNet except it replaces deep 3-dimensional convolutions with a combination of shallow 3-dimensional convolutions (1-layer) and deep 2-dimensional convolutions (34-layer ResNet) integrated with squeeze and excitation (SE) blocks (SE-ResNet). It was also trained on the GRID dataset (21,635 videos).

To ensure a fair comparison, we utilized an openly accessible dataset consisting of thirty randomly selected phrases from each model's training dataset [77].

4.2 Performance Metrics

We used the following metrics to benchmark the proposed framework.

- **Word error rate** is the minimum number of operations required to transform the predicted text to the ground truth, divided by the number of words in the ground truth. It is calculated using the following equation, where S is the number of substitutions, D is the number of deletions, I is the number of insertions, N is the number of words in the ground truth.

$$\textit{Word error rate} = \frac{S + D + I}{N} \tag{9}$$

- **Words per minute (wpm)** is a commonly used text entry metric that signifies the rate in which words (= 5 chars) are entered [8]. It is calculated using the following equation, where T is the number of recognized words, t is the sum of speaking time and computation time in seconds, the constant 60 is the number of seconds per minute, and the factor of one fifth accounts for the average length of a word in the English language.

$$WPM = \frac{|T| - 1}{t} \times 60 \times \frac{1}{5} \tag{10}$$

- **Computation time** (s) is the total time required by the model to process each window. It does not include the time users took to silently speak a phrase.

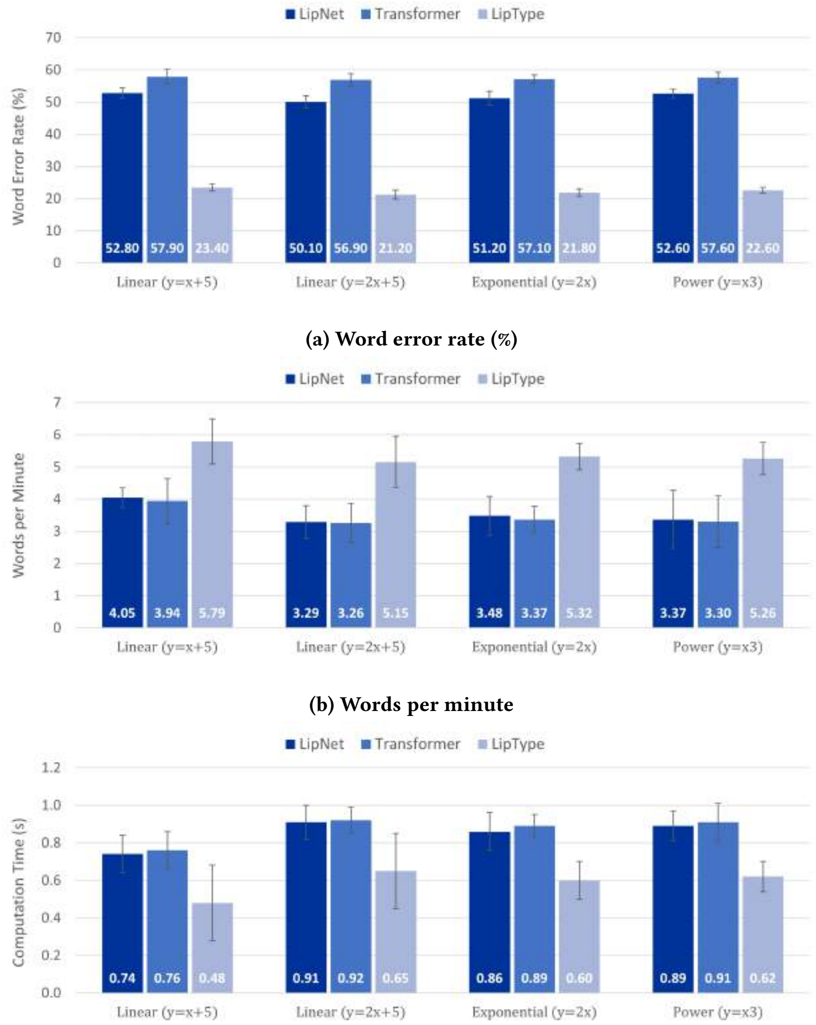
4.3 Results

We evaluated all models on NVIDIA GeForce 1080Ti GPU board. Based on the results, $y = x + 5$ results in less computation time for processing each sliced clip, thereby resulting in a faster input speed. The function, however, is slightly more erroneous than others, but since our aim is to show recognition as quickly as possible in order to mimic the real-time recognition, we considered it the most effective method. Fig. 3 shows the performance of each windowing function on the three examined silent speech recognition models. It can be seen that all pre-trained models performed much better with $y = x + 5$ functions in terms of input speed and computation time. With LipNet, $y = x + 5$ shows 2.5% increase in word error rate, 19.8% increase in words per minute, and 15.9% reduction in the computation time than the remaining three windowing functions. With Transformer, $y = x + 5$ shows 1.4% increase in word error rate, 19.1% increase in words per minute, and 15.5% reduction in the computation time than the remaining three windowing functions. With LipType, $y = x + 5$ shows 7.3% increase in word error rate, 10.5% increase in words per minute, and 22.5% reduction in the computation time than the remaining three windowing functions. Regardless of windowing function, LipType performed better. This further strengthens the decision to use LipType as the base model for this work. Note that in the proposed model, repetitions of blank tokens (> 3) in the recognized sequence are used to determine the end of the sentence. The buffer is cleared if the following sequence is detected and buffering will begin from scratch. However, since we focus on text entry on mobile devices, we did not optimize the model on very long sentences.

5 ADOPTING A TRANSFER LEARNING STRATEGY

Most lip reading datasets contain limited vocabulary and do not support vocabulary relevant to everyday conversation. A model trained on a dataset with specific vocabulary performs poorly when applied to a dataset other than the training vocabulary words. Furthermore, training a deep learning model requires an enormous amount of data. Developing large-scale datasets tailored to particular vocabularies is extremely challenging, expensive, and time-consuming. To overcome this, we leverage the effectiveness of transfer learning, which exploits existing features (or knowledge) from a model trained on a high-resource vocabulary, *source model*, and generalizes it to a new low-resource vocabulary, *target model* [76, 121].

Generally, features transition from general to specific characteristics by the last layer of the network, but this transition has not been extensively investigated in the context of lip reading. Research in deep learning research showed that standard features learned on the first layer appear regardless of the dataset and the task [96, 115], thus are called general features. In contrast, features calculated by the last layer of a trained network is highly dependent on the dataset and the task. It is unclear, however, how this transition can be generalized to lip reading, that is, to what extent features within a network could be generalized and used for



(a) Word error rate (%)

(b) Words per minute

(c) Computation time (s) per window

Figure 3: Performance comparison of the three investigated silent speech recognition model with different windowing functions in terms of a) word error rate, b) words per minute, and c) computation time per window. Reported values are the average of all values. Error bars represent ±1 standard deviation.

transfer learning. Towards this, we investigated three strategies to transfer learning (Fig. 4). Consider a *source model* composed of N layers, with V layers representing *visual_frontend* and S layers representing *sequence_processing*.

(1) **Finetune_Last:** The network is first initialized with the weights from the source model, then the top layers $(N-1)$ are frozen, and only the last layer is allowed to modify its weights. The model is then trained to fine-tune the last layer for the target vocabulary. During the training process, only the weights associated with last layer are changed until they converge. Using this method, fine-tuning only the final layer is needed to work more effectively with the target dataset and it makes use of the features learned from the source model.

(2) **Finetune_Visual_Frontend:** The network is first initialized with the weights from the source model, then the *sequence_processing* layers $(N-V)$ are frozen and only the *visual_frontend* layers are allowed to modify their weights. Afterwards, the model is trained to fine-tune the *visual_frontend* for the target vocabulary. During the training process, only the weights associated with the *visual_frontend* are changed until they converge.

(3) **Finetune_Sequence:** The network is first initialized with the weights from the source model, then the *visual_frontend* layers (V) are frozen and only the *sequence_processing* layers are allowed to modify their weights. Afterwards, the model

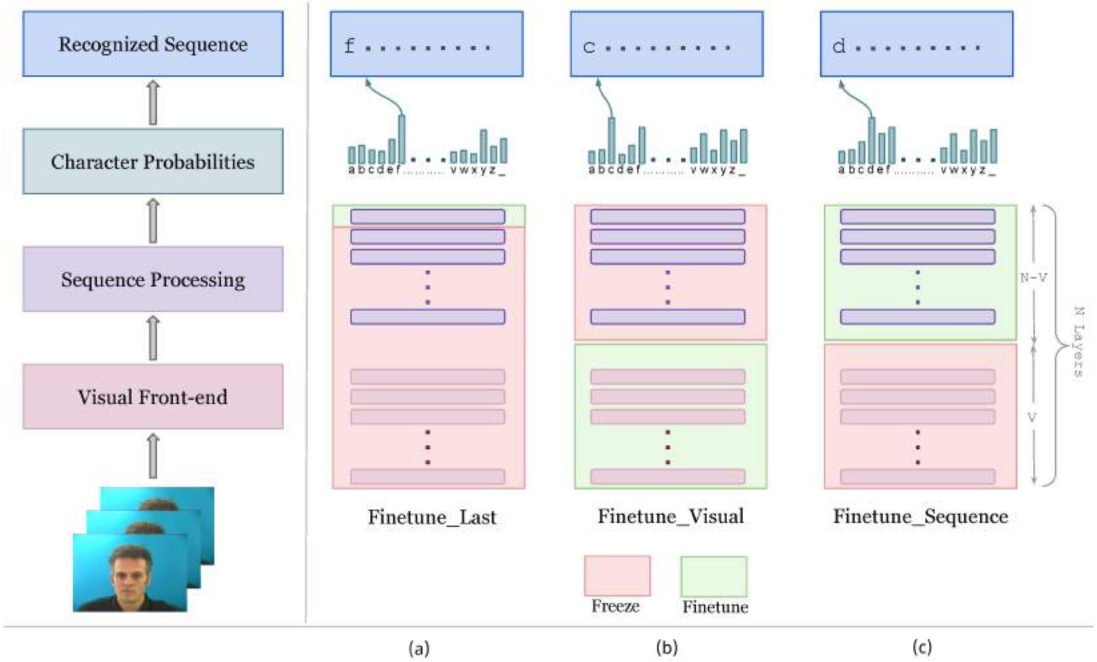


Figure 4: Transfer learning strategies: a) freeze $N - 1$ layers and fine-tune last layer, b) freeze sequence modeling layers and fine-tune visual front-end, and c) freeze visual front-end layers and fine-tune sequence modeling.

is trained to fine-tune the *sequence_processing* for the target vocabulary. During the training process, only the weights associated with the *sequence_processing* are changed until they converge.

6 EXPERIMENT 2: EFFECTS OF TRANSFER LEARNING

In this experiment, we examined how different strategies of transfer learning affect the performance of silent speech recognition models. For the source models, we used the same pre-trained silent speech models as described in Section 4.1. For target models, we trained these source models from scratch with a low-resource target dataset.

The experiment calculated the same word error rate and words per minute performance metrics as described in Section 4.2. However, regarding computation time, this experiment specifically measured the average time required by the model to process a phrase.

6.1 Transfer Learning Dataset

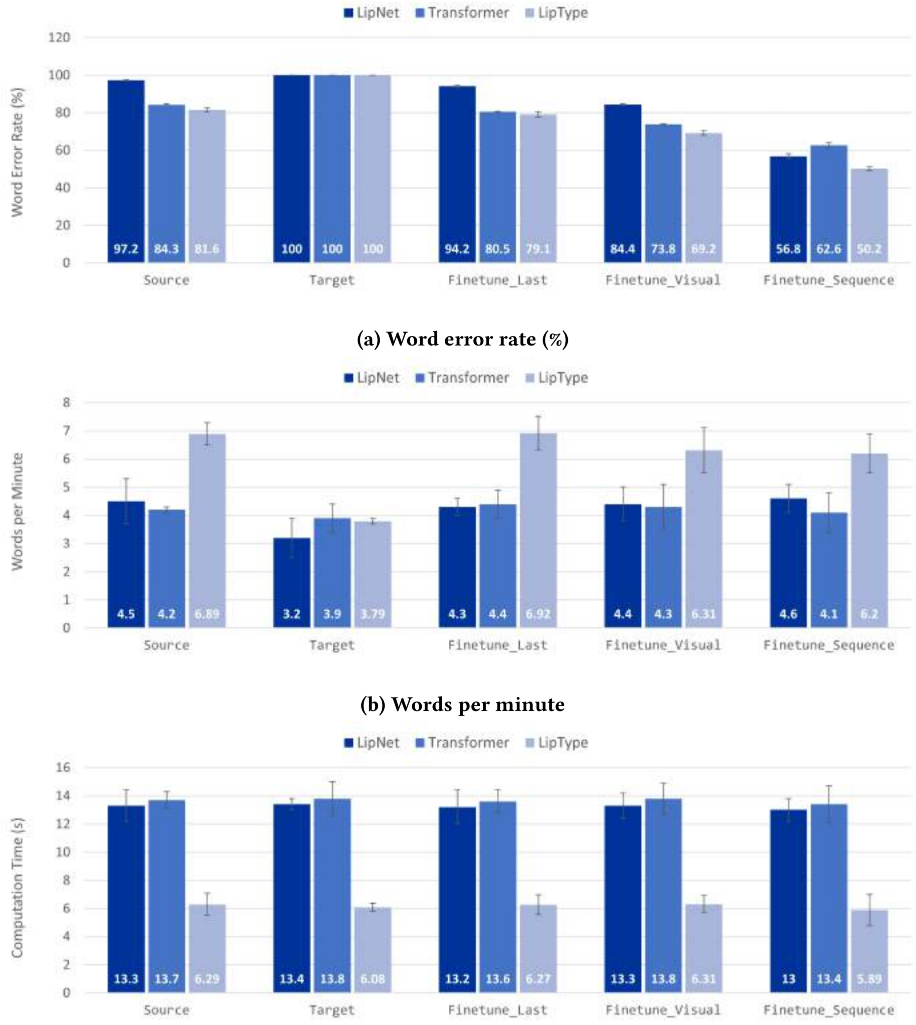
All source models were trained on their respective training datasets (Section 4.1). For target models, we used the publicly available dataset [77], which consists of thirty randomly selected phrases from the MacKenzie & Soukoref dataset [69]. It comprises of short and formulaic video clips of a person's face when uttering the phrases. The selected phrases are a good representation of the English language and is highly correlated with Mayzner & Tresselt's letter frequencies [71], thus are more generalizable. Target dataset contains 1,080 video clips of twelve speakers uttering thirty phrases. For the experiment, we employed a random selection of 720 videos for the fine-tuning phase and 360 videos for the evaluation phase. The same evaluation dataset was consistently used for all models (Table 4).

6.2 Implementation

To avoid any potential confounding factor, we trained all models from scratch with the same training parameters used in their respective source model. For target model, we did not apply any transfer learning at all, and let the model train on the given low-resource training data. The number of frames was fixed to 75. Since all videos are 25 fps with a length of ~3 seconds, they have 75 frames in total (25 fps × 3 seconds = 75 frames). Longer image sequences were

Table 4: Statistics of dataset used for training and fine-tuning the models. The values are the total number of video samples.

Model	Training		Fine-tuning		
	Source	Target	Finetune_Last	Finetune_Visual	Finetune_Sequence
LipNet	21,635	720	720	720	720
LipType	21,635	720	720	720	720
Transformer	41,500	720	720	720	720



(a) Word error rate (%)

(b) Words per minute

(c) Computation time (s) per phrase

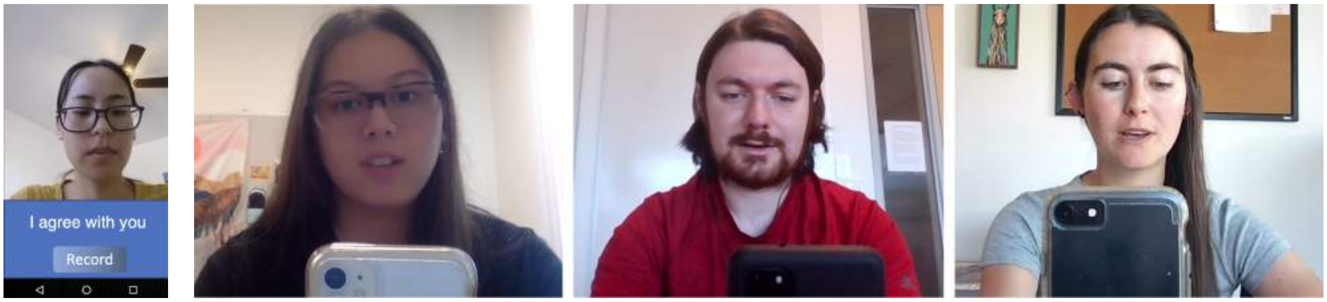
Figure 5: Performance comparison of the three investigated silent speech recognition model with different transfer learning approaches in terms of a) word error rate, b) words per minute, and c) computation time per phrase. Reported values are the average of all values. Error bars represent ±1 standard deviation.

truncated and shorter sequences were padded with zeros. We applied a channel-wise dropout [98] of 0.5. The model was trained end-to-end by the Adam optimizer [59] for 60 epochs with a batch size of 50. The network was implemented based on the Keras deep-learning platform with TensorFlow [1] as the backend. We trained and tested both models on NVIDIA GeForce 1080Ti GPU board.

6.3 Results

Results showed that the target model performed the worst, while the model that kept visual front-end frozen and sequential module fine-tuned performed the best. With LipNet, *Finetune_Sequence* shows 39.5% decrease in word error rate, 12.1% increase in words per minute, and 2.2% reduction in the computation time than the other models. With Transformer, *Finetune_Sequence* shows 26.1% decrease in word error rate, 2.3% increase in words per minute, and 2.1% reduction in the computation time than the other models. With LipType, *Finetune_Sequence* shows 39.1% decrease in word error rate, 5.1% increase in words per minute, and 5.4% reduction in the computation time than the other models. Fig. 5 presents the findings of this experiment.

This means that performance worsens as we keep bottom layers fixed when transferring parameters from the source task. We speculate that this is because the top layer features are not specific to particular datasets or tasks, but are general in that they can be applied to a wide range of datasets and tasks. On the other hand, the features computed by the bottom layer of a network are highly dependent on the dataset and the task chosen. In addition,



(a) The app

(b) Three participants taking part in the study

Figure 6: The custom app (a) and three participants (b) of Experiment 3.

fine-tuning only the last layer is not sufficient since the sequential module learns transitional probability of characters based on context. Therefore, fine-tuning only the last layer will not be able to model the transition of characters that were not part of the source model's training vocabulary.

7 EXPERIMENT 3: STATIONARY PERFORMANCE

We conducted a user study to compare MELDER with two state-of-the-art, pre-trained silent speech models LipNet [9] and Transformer [3] with unseen data (data that has not been used to train the models) in a stationary setting (in a seated position). Since these models do not work in real-time (computes one phrase at a time), we equipped these with the $y = x + 5$ windowing function and the Finetune_Sequence transfer learning strategy, as MELDER, for a fair comparison between the models (henceforth referred to as RT-LipNet and RT-Transformer) and to demonstrate that these approaches could be used independently with other silent speech models to make them real-time. We also disabled the visual feedback component of the reviewer channel (described in Section 3.2) in the study to eliminate a confounding factor (to remove any potential effects of feedback on performance).

7.1 Experimental Dataset

We used the Enron Mobile Email dataset [110] in this study. It contains genuine mobile emails, thus is better suited to evaluate mobile text entry. Towards this, first, we filtered the dataset using the following rules: 1) exclude phrases with lengths less than three or greater than ten, 2) exclude phrases containing common nouns, such as general names, places, and things, and 3) exclude phrases containing contractions or numeric values. After filtering, we randomly selected thirty phrases and removed all punctuation and non-alphanumeric tokens, and replaced all uppercase letters with lowercase letters. The selected phrases are presented in Appendix A.

7.2 Participants

Twenty volunteers took part in the study (Fig. 6b). Their age ranged from 18 to 41 years (M = 25.55 years, SD = 6.2). Ten of them identified as women, nine as men, and one as non-binary. They all owned a smartphone for at least five years (M = 8.4 years, SD = 2.1). Sixteen of them were frequent users of a voice assistant system on their smartphones (M = 3 years, SD = 2.3), while the remaining four were infrequent or nonusers. They all received U.S. $10 for volunteering.

7.3 Apparatus

We developed a custom application for smartphones running on Android OS using the default Google Android API (Fig. 6a). The application enabled users to record videos of them silently speaking the presented phrases using the front camera of a smartphone. In the study, we enabled participants to record videos using the front camera of their own smartphones to increase the variability of the dataset.

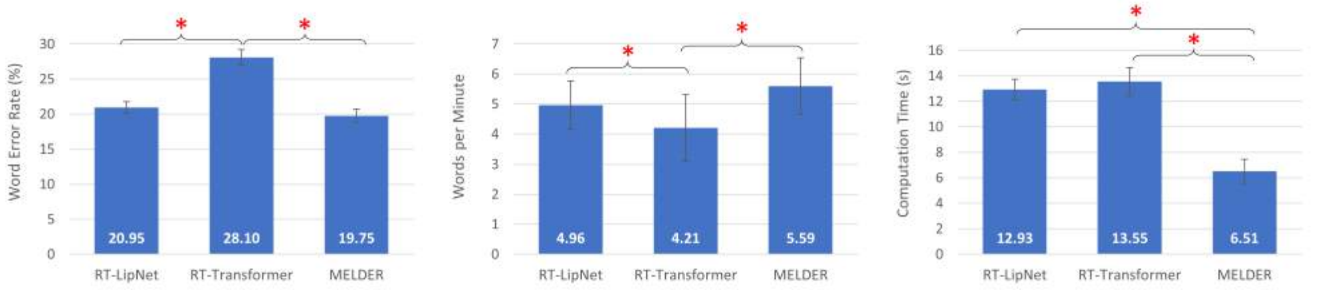
7.4 Design

The study used a within-subjects design with one independent variable "model" with three levels: RT-LipNet, RT-Transformer, and MELDER. In total, we collected (20 participants × 30 phrases) = 600 samples. The dependent variables were the same word error rate and words per minute performance metrics as described in Section 4.2. However, regarding computation time, this experiment measured the average time required by the model to process a phrase.

7.5 Procedure

The data collection process occurred remotely. We explained the purpose of the study and scheduled individual Zoom video calls with each participant ahead of time. We instructed them to join the call from a quiet room to avoid any interruptions during the study. First, we demonstrated the application and collected their consents and demographics using electronic forms. We then shared the application (APK file) with them and guided them through the installation process on their smartphones.

Participants were instructed to sit at a desk during the study. The application displayed one phrase at a time. Participants pressed the "Record/Stop" toggle button, silently spoke the phrase (uttered the phrase without vocalizing sound), then pressed the same button to see the next phrase. We did not instruct them about how to hold the device. But most of them held the device with their non-dominant hand and pressed the button with their dominant hand. Upon completion of the study, participants shared the logged data with us by uploading those to a cloud storage under our supervision. For evaluation, we passed the recorded video through



(a) Word error rate (stationary)

(b) Words per minute (stationary)

(c) Computation time (stationary)

Figure 7: Performance comparisons between RT-LipNet, RT-Transformer, and MELDER models in a stationary setting in terms of a) word error rate, b) words per minute, and c) computation time per phrase. Reported values are the average of all values. Error bars represent ±1 standard deviation. Red asterisks represent statistically significant differences.

the transcriber channel to obtain recognition, then post-processed the recognized text through the reviewer channel to auto-correct errors and present the most probable auto-completion of text at both word and phrase-level.

7.6 Results

A Martinez-Iglewicz test revealed that the response variable residuals were normally distributed. A Mauchly's test indicated that the variances of populations were equal. Therefore, we used a repeated-measures ANOVA and a post-hoc Tukey-Kramer multiple-comparison test for all analysis. We also report effect sizes in eta-squared (η^2) for all statistically significant results.

7.6.1 Word Error Rate. An ANOVA identified a significant effect of model on word error rate ($F_{2,19} = 3632.67, p < .00001, \eta^2 = 0.94$). On average, RT-LipNet, RT-Transformer, and MELDER yielded 20.95% (SD = 0.8), 28.1% (SD = 1.1), and 19.75% (SD = 0.9) word error rates, respectively. A Tukey-Kramer test revealed that RT-Transformer was significantly more error prone than RT-LipNet and MELDER. Fig. 7a illustrates this.

7.6.2 Words per Minute. An ANOVA identified a significant effect of model on word error rate ($F_{2,19} = 557.08, p < .00001, \eta^2 = 0.89$). On average, RT-LipNet, RT-Transformer, and MELDER yielded 4.96 wpm (SD = 0.3), 4.21 wpm (SD = 0.2), and 5.59 wpm (SD = 0.1), respectively. A Tukey-Kramer test revealed that RT-Transformer was significantly slower than RT-LipNet and MELDER. Fig. 7b illustrates this.

7.6.3 Computation Time. An ANOVA identified a significant effect of model on word error rate ($F_{2,19} = 11085.33, p < .00001, \eta^2 = 0.99$). On average, RT-LipNet, RT-Transformer, and MELDER required 12.93s (SD = 0.4), 13.55s (SD = 0.2), and 6.51s (SD = 0.2) to compute a phrase, respectively. A Tukey-Kramer test revealed that MELDER was significantly faster in computing the phrases than RT-LipNet and RT-Transformer. Fig. 7c illustrates this.

7.7 Discussion

MELDER outperformed RT-LipNet and RT-Transformer both in terms of speed and accuracy. MELDER took 50% less time than RT-LipNet and 52% less time than RT-Transformer to compute a phrase. These effects were statistically significant, and resulted in a 13% and a significantly 33% faster text entry speed than RT-LipNet and RT-Transformer, respectively. MELDER was also the most accurate. It committed 6% fewer errors than RT-LipNet and a significantly 30% fewer errors than RT-Transformer. The statistically significant differences, accompanied by large effect sizes ($\eta^2 \geq 0.1$ constitutes a large effect size [7, 19]), indicate their potential generalizability to a broader population. These results strengthen our argument that MELDER is better suited for use on mobile devices than existing models.

We also compared the original LipNet and Transformer models with RT-LipNet and RT-Transformer in an ablation study[5]. In the study, LipNet yielded 97.3% word error rate, 4.6 wpm entry speed, and 14.2s computation time. The addition of windowing and transfer learning approaches reduced word error rate by 78%, improved entry speed by 7%, and reduced computation time by 9%. Transformer also demonstrated substantial improvements in performance when empowered with the proposed windowing and transfer learning approaches. The original Transformer yielded 81.2% word error rate, 5.2 wpm entry speed, and 14.7s computation time. RT-Transformer, conversely, demonstrated a 65% reduction in word error rate, 19% improvement in entry speed, and 14% reduction in computation time. These findings validate that the suggested windowing and transfer learning methods can be employed separately with existing silent speech recognizers, not only enabling real-time capabilities but also enhancing their overall performance.

8 EXPERIMENT 4: MOBILE PERFORMANCE

We conducted a follow-up pilot study to compare MELDER with RT-LipNet and RT-Transformer with unseen data in a mobile setting (while walking). The study used the same dataset as the previous experiment (Appendix A).

[5] An ablation study *"investigates the performance of an AI system by removing certain components to understand the contribution of the component to the overall system"* [113].

8.1 Participants

Six new volunteers took part in the study. Their age ranged from 22 to 31 years (M = 26.33 years, SD = 3.1). Three of them identified as women and three as men. They all owned a smartphone for at least four years (M = 6.67 years, SD = 2.4). All of them were frequent users of a voice assistant system on their smartphones (M = 2.17 years, SD = 1.2). They all received U.S. $10 for volunteering.

8.2 Apparatus, Design, and Procedure

The study used the same apparatus, design, and procedure as the previous experiment (Section 7). However, unlike the previous study, participants were instructed to silently speak the phrases while walking indoors. They were instructed to walk at a pace they would usually walk while using a smartphone. We did not collect outdoor data because the risk of slips, trips, and falls is higher outdoors, which could have subjected participants to unnecessary risks. The study computed the same word error rate, words per minute, and computation time performance metrics as outlined in Section 7.4.

8.3 Results

A Martinez-Iglewicz test revealed that the response variable residuals were normally distributed. A Mauchly's test indicated that the variances of populations were equal. Therefore, we used a repeated-measures ANOVA and a post-hoc Tukey-Kramer multiple-comparison test for all analysis. We also report effect sizes in eta-squared (η^2) for all statistically significant results.

8.3.1 Word Error Rate. An ANOVA identified a significant effect of model on word error rate ($F_{2,5} = 32.25, p < .00005, \eta^2 = 0.78$). On average, RT-LipNet, RT-Transformer, and MELDER yielded 27.01% (SD = 3.1), 34.24% (SD = 1.8), and 25.34% (SD = 1.3) word error rates, respectively. A Tukey-Kramer test revealed that RT-Transformer was significantly more error prone than RT-LipNet and MELDER. Fig. 8a illustrates this.

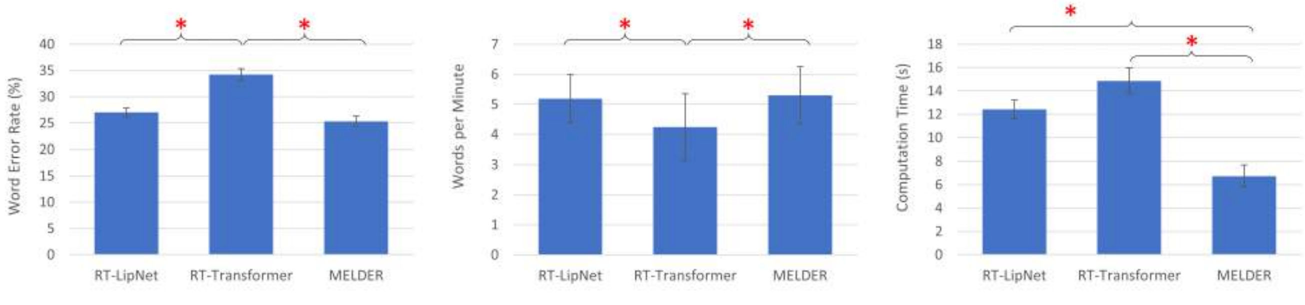
8.3.2 Words per Minute. An ANOVA identified a significant effect of model on word error rate ($F_{2,5} = 25.76, p < .0005, \eta^2 = 0.68$). On average, RT-LipNet, RT-Transformer, and MELDER yielded 5.19 wpm (SD = 3.1), 4.24 wpm (SD = 1.8), and 5.31 wpm (SD = 1.3), respectively. A Tukey-Kramer test revealed that RT-Transformer was significantly slower than RT-LipNet and MELDER. Fig. 8b illustrates this.

8.3.3 Computation Time. An ANOVA identified a significant effect of model on word error rate ($F_{2,5} = 385.09, p < .00001, \eta^2 = 0.97$). On average, RT-LipNet, RT-Transformer, and MELDER required 12.42s (SD = 3.1), 14.85s (SD = 1.8), and 6.73s (SD = 1.3) to compute a phrase, respectively. A Tukey-Kramer test revealed that MELDER was significantly faster in computing the phrases than RT-LipNet and RT-Transformer. Fig. 8c illustrates this.

8.4 Discussion

The findings of this study parallel those of the previous study, which evaluated the models' performance in a stationary setting. MELDER outperformed RT-LipNet and RT-Transformer both in terms of speed and accuracy. MELDER was significantly faster in computing the phrases than RT-LipNet (46% faster) and RT-Transformer (55% faster). It also demonstrated a 2% faster text entry speed than RT-LipNet and a significantly 25% faster entry speed than RT-Transformer. Further, MELDER yielded a 6% lower word error rate than RT-LipNet and a significantly 26% lower word error rate than RT-Transformer. Most importantly, despite the small sample size ($N = 6$), the statistically significant results yielded large effect sizes ($\eta^2 \geq 0.1$ constitutes a large effect size [7, 19]), which suggests the potential for these findings to generalize to a wider population.

We conducted an independent-samples t-test to compare the results of Experiment 3 and Experiment 4. Table 5 presents the findings. As anticipated, there were some performance differences between the experiments not only because they were conducted in different settings but also with different samples and sample sizes ($N = 20$, $N = 6$). A t-test revealed that both RT-LipNet ($t(24) = -8.16, p < .00001, d = 1.6$), RT-Transformer ($t(24) = -10.16, p < .00001, d = 1.3$), and MELDER ($t(24) = -11.69, p < .00001, d = 1.03$) committed significantly more errors in the mobile setting than in the stationary setting. This is not surprising, since the videos were shakier and more jittery in the mobile condition, which affects



(a) Word error rate (mobile) **(b) Words per minute (mobile)** **(c) Computation time (mobile)**

Figure 8: Performance comparisons between RT-LipNet, RT-Transformer, and MELDER models in a mobile setting (when walking) in terms of a) word error rate, b) words per minute, and c) computation time. Reported values are the average of all values. Error bars represent ±1 standard deviation. Red asterisks represent statistically significant differences.

Table 5: Performance differences between the three silent speech recognition models in stationary and mobile settings. The up and down arrows indicate increments and decrements in the respective values, the colors green and red indicate whether a difference is an improvement or a decline, respectively, in performance.

Metric	Model	Stationary	Mobile	Difference	Significance ($\alpha = 0.05$)
Word error rate	*RT-LipNet*	20.95	27.01	29% ↑	Significant
	RT-Transformer	28.10	34.24	22% ↑	Significant
	MELDER	19.75	25.34	28% ↑	Significant
Words per minute	*RT-LipNet*	4.96	5.19	5% ↑	Not significant
	RT-Transformer	4.21	4.24	1% ↑	Not significant
	MELDER	5.59	5.31	5% ↓	Significant
Computation time	*RT-LipNet*	12.93	12.42	4% ↓	Significant
	RT-Transformer	13.55	14.85	10% ↑	Significant
	MELDER	6.51	6.73	3% ↑	Not significant

video processing. Nevertheless, MELDER yielded the lowest average word error rate than the other models when mobile. Text entry speed with MELDER was significantly slower in the mobile setting compared to the stationary setting ($t(24) = 2.3, p < .05, d = 0.3$), while RT-LipNet and RT-Transformer had relatively similar speeds. Likewise, RT-LipNet yielded a significantly faster computation time ($t(24) = 2.37, p < .05, d = 0.5$), while RT-Transformer yielded a significantly slower computation time ($t(24) = -7.05, p < .00001, d = 0.4$) in the mobile setting than in the stationary setting. MELDER's computation time in the two settings were comparable. These significant differences are likely caused by the differences in the samples or by chance, as we did not identify any other reasons through data analysis. Relevantly, these relationships produced small–medium effect sizes (Cohen's $d \leq 0.2$ constitutes a small effect size and $d \geq 0.5$ constitutes a medium effect size [19]), indicating to the possibility that these outcomes are likely due to chance. However, further investigations are needed to confirm this assumption. The results of this study further solidify our claim that MELDER is effective not only in stationary settings but also in mobile ones.

9 EXPERIMENT 5: VISUAL FEEDBACK

We conducted a final study to compare the visual feedback methods of MELDER with the visual feedback method of Google Assistant. Note that the feedback methods were not included in Experiments 3 and 4 to eliminate a potential confounding factor. This study focuses on assessing the perceived performance of visual feedback methods in MELDER and Google Assistant, rather than directly comparing their actual performance. Such a comparison would be unfair due to the inherent differences between the two systems: MELDER is an image-based silent speech recognizer, while Google Assistant's speech-to-text relies on audio processing. These disparities stem from the distinct data types they handle (visual for images, auditory for audio), resulting in varying complexities in operations and feature extraction.

9.1 Apparatus

We developed a custom Web application with HTML5, CSS, PHP, JavaScript, and Node.js. We hosted the application on GitHub. The application was loaded on a Chrome web browser v71.0.3578.98 running on a Motorola Moto G^5 Plus smartphone (150.2x74x7.7 mm, 155 g). Its built-in front camera (12 megapixel with 1080×1920 pixel resolution) was used to track lip movements. Through an IP webcam Android application [54], we connected the smartphone's camera to the server, which ran the silent speech recognition model. The server was running on a MacBook Pro 16" laptop with 2.6 GHz Intel Core i7 processor, 16 GB RAM, 3072×1920 at 226 ppi. The laptop and the smartphone were connected to a fast and reliable Wi-Fi network. There were no network dropouts during the study.

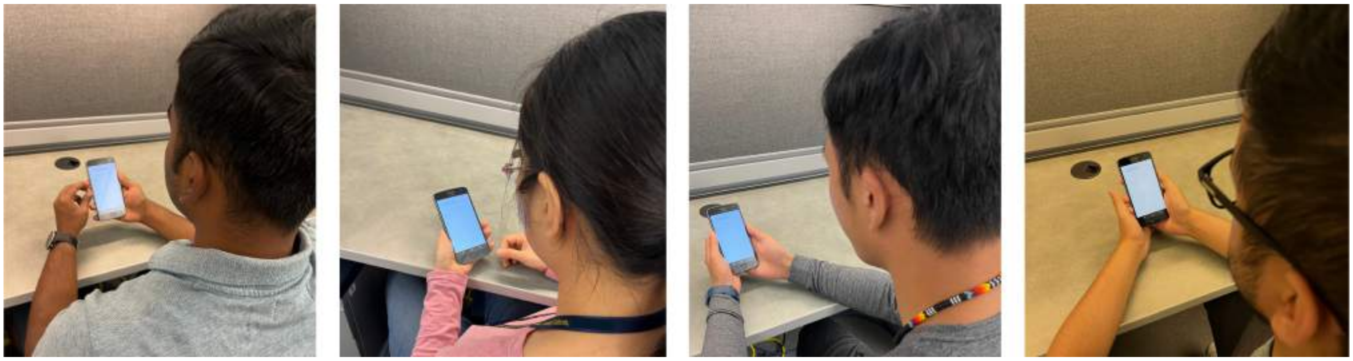
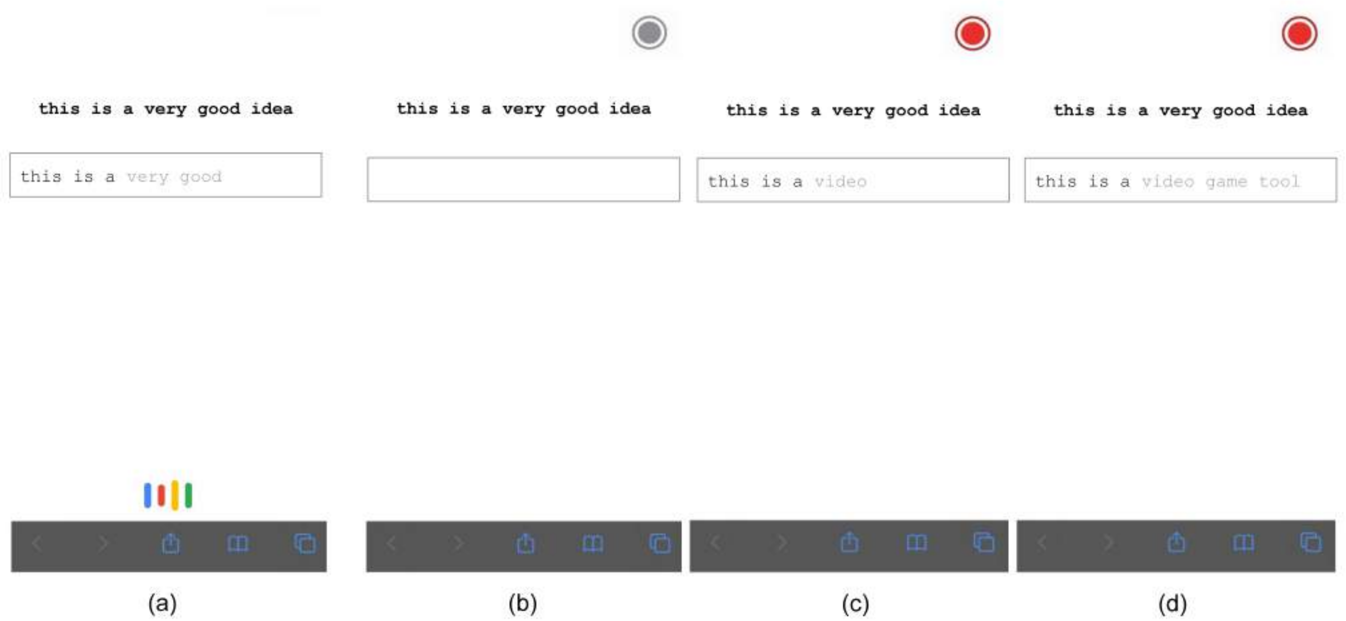


Figure 9: Four participants taking part in the final user study.



this is a very good idea

this is a very good

this is a very good idea

this is a very good idea

this is a video

this is a very good idea

this is a video game tool

Figure 10: Real-time visual feedback provided by (a) the default Google Speech recognizer and (b–c) MELDER. In (b), the gray circle in the top-right corner indicates that MELDER is unable to track the lips, passively prompting the user to reposition the device for a better camera view. In (c), MELDER is providing word-level feedback and in (d) it is providing phrase-level feedback. The suggestions are in a greyed-out font. In both (c) and (d), the red blinking circle indicates that MELDER is able to track the lips.

9.2 Participants

Twelve volunteers participated in the user study (Fig. 9). Their age ranged from 21 to 41 years (M = 27.8 years, SD = 5). Eight of them identified as women and four as men. They all owned a smartphone for at least five years (M = 8.2 years, SD = 2.2). Eleven of them were frequent users of a voice assistant system on their smartphones (M = 3 years, SD = 2.4), while one was an infrequent user. They all received U.S. $15 for volunteering.

9.3 Design

We used a within-subjects design for the user study with one independent variable "feedback" with three levels: Google, word-level MELDER, and phrase-level MELDER. In each condition, participants entered thirty short English phrases from a subset of the Enron Mobile Email corpus, presented in Appendix A. In summary, the design was 12 participants × 3 conditions × 30 phrases = 1,080 input tasks in total. The dependent variables were the eight items on a questionnaire. The study gathered qualitative data through the utilization of a custom questionnaire inspired by the System Usability Scale (SUS) [12]. The questionnaire asked participants to rate eight statements on the examined methods' speed (*"The technique was fast"*), accuracy (*"The technique was accurate"*), effectiveness (*"The feedback method used in the technique was effective and useful"*), willingness-to-use (*"I think that I would like to use this system frequently"*), ease-of-use (*"I thought the system was easy to use"*), learnability (*"I would imagine that most people would learn to use this system very quickly"*), confidence (*"I felt very confident using the system"*), and privacy and security (*"I think the system will be private and secure when using in public places"*) on a 5-point Likert scale.

9.4 Feedback Approaches

We created two real-time visual feedback methods for silent speech recognition models, drawing inspiration from Google Assistant's feedback approach. In Google Assistant, the system starts displaying likely letters and words as soon as it detects speech, refining the output as the speaker continues. These initial predictions are presented in a greyed-out font (Fig. 10a) to signify their potential for correction as more information becomes available. Unlike suggestions on a virtual keyboard, these predictions in Google Assistant are automatically managed by the system and cannot be manually selected, discarded, or updated by users. Additionally, the system offers feedback for sound detection, resembling oscilloscope traces or sound waves, presented as four colored vertical lines (Fig. 10a, bottom of the display). These lines dynamically change in height to indicate when the system detects sound and come to a halt when sound detection ceases.

MELDER also offers feedback on lip detection and speech recognition. When the front camera detects the user's lips, it displays a red blinking circle, similar to the video recording indicator on mobile devices. The red circle ceases blinking and changes to grey when the lips are no longer visible (Fig. 10b). To keep users informed about the speech recognition process, we developed two feedback methods:

- **Word-level feedback**: This method offers real-time feedback on a word-by-word basis. It presents the most probable word based on the recognized input. The text remains gray until the confidence level of the word surpasses a specified threshold (empirically set at 0.75). Once this condition is met, the word turns black, signifying that it is considered finalized and will not be corrected (Fig. 10c).
- **Phrase-level feedback**: In this approach, real-time feedback is provided by displaying the most likely phrase based on the recognized prefix string. Each word within the phrase starts in gray and transitions to black when its confidence level exceeds a specific threshold (empirically set at 0.87). This change to black indicates that the word is considered fixed and will not undergo further correction (Fig. 10d).

The threshold values were determined empirically through multiple lab trials. During these trials, we tested thresholds ranging from 0.65 to 1.0 for both feedback methods. We selected the threshold values that proved most effective in delivering real-time feedback based on the experimental results. Similar to Google Assistant, neither of these feedback methods allowed users to proactively select, dismiss, or modify the suggestions; they were merely provided to inform users about the recognition process.

9.5 Procedure

The study was conducted in a quiet computer laboratory. First, we provided the participants with a brief overview of the functioning principles behind both speech and silent speech recognition. Subsequently, we offered practical demonstrations of the three distinct feedback methods employed in the study. We then collected their informed consent forms, and enabled them to practice with the three methods for about five minutes. They could extend the duration of the practice an extra two minutes upon request.

The main study started after that. In the study, participants entered thirty short English phrases from the Enron set [110] by either speaking or silently speaking on a smartphone. All participants were seated at a desk. The three conditions (Google Assistant, MELDER with word-level feedback, and MELDER with phrase-level feedback) were counterbalanced to eliminate any potential effect of practice. As each phrase was recognized, the application automatically displayed the next phrase, continuing in this manner until all phrases within the given condition had been successfully completed. Participants were not required to re-speak a phrase in the event that it was not accurately recognized by the system.

Upon the completion of all conditions, participants completed a questionnaire that asked them to rate the three methods' speed, accuracy, effectiveness, willingness-to-use, ease-of-use, learnability, confidence, and privacy and security on a five-point Likert scale (Section 9.3). Finally, we concluded the study with a debrief session, where participants were given a chance to share their thoughts and comments regarding their responses to the questionnaire.

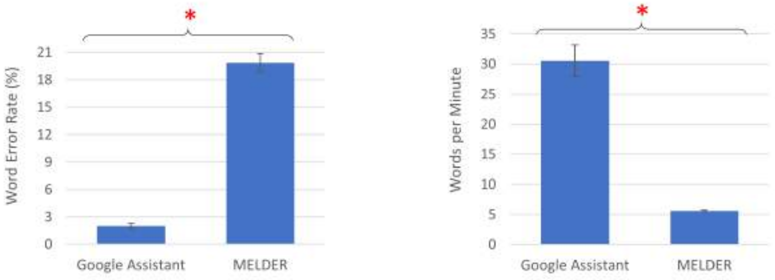
9.6 Speed and Accuracy

As discussed in Section 9, the primary aim of this qualitative study was not to conduct a direct comparison of the actual speed and accuracy of the models. However, it is noteworthy that we did carry out a separate study comparing Google Assistant and MELDER. In this between-subjects study, 24 participants (average age = 26.25 years, SD = 5.9, comprising 12 females, 11 males, and 1 non-binary) were evenly distributed into two groups: one using Google Assistant and the other using MELDER. Each group employed their designated input method in a seated position. A between-subjects ANOVA analysis revealed a statistically significant impact of the input method on both entry speed ($F_{1,22} = 1083.35, p < .00001, \eta^2 = 0.98$) and accuracy ($F_{1,22} = 1219.38, p < .00001, \eta^2 = 0.99$).

As expected, participants using Google Assistant achieved an average entry speed of 30.54 wpm (SD = 2.6) and a remarkably low word error rate of 2.01% (SD = 0.3). In contrast, those using MELDER exhibited significantly slower input speeds, averaging 5.62 wpm (SD = 0.1), along with a much higher word error rate of 19.86% (SD = 1.0). Fig. 11 summarizes these findings. It is important to highlight that both the word-level and phrase-level versions of MELDER utilize the same recognition model and do not necessitate users to actively choose suggestions from the feedback. Consequently, they are indistinguishable in terms of actual speed and accuracy.

9.7 Results

We used a Friedman test and a post-hoc Games-Howell multiple-comparison test for analysing all non-parametric study data. We



(a) Word error rate **(b) Words per minute**

Figure 11: Performance comparisons between Google Assistant and MELDER in terms of a) word error rate and b) words per minute. Reported values are the average of all values. Error bars represent ±1 standard deviation. Red asterisks represent statistically significant differences.

also report effect sizes in Kendall's W for all statistically significant results. Kendall's W uses the Cohen's interpretation guidelines [19] of $W < 0.3$ as small, $W \geq 0.3$ as medium, and $W \geq 0.5$ as large effect sizes. Fig. 12 summarizes the findings of the study.

9.7.1 Perceived Speed and Accuracy. A Friedman test identified a significant effect of feedback on perceived speed ($\chi^2 = 9.83, df = 2, p < .01, W = 0.4$) and accuracy ($\chi^2 = 6.4, df = 2, p < .05, W = 0.3$). A Games-Howell test revealed that participants found Google Assistant to be significantly faster than both word-level and phrase-level MELDER. But interestingly, the pairwise test was unable to identify any significant difference between the three methods in terms of accuracy.

9.7.2 Effectiveness. A Friedman test failed to identify a significant effect of feedback on effectiveness ($\chi^2 = 4.69, df = 2, p = .09$). Additionally, a Games-Howell test confirmed that participants perceived all three examined feedback approaches to be relatively equally effective.

9.7.3 Willingness-to-Use. A Friedman test identified a significant effect of feedback on willingness-to-use ($\chi^2 = 7.0, df = 2, p < .05, W = 0.3$). An analysis using the Games-Howell test demonstrated that participants expressed a significantly stronger preference for phrase-level feedback over word-level feedback. However, there was no statistically significant difference in their preference between these feedback types and Google Assistant.

9.7.4 Ease-of-Use and Learnability. A Friedman test failed to identify a significant effect of feedback on either ease-of-use ($\chi^2 = 6.0, df = 2, p = .05$) or learnability ($\chi^2 = 6.0, df = 2, p = .05$). A Games-Howell test also confirmed that participants found the three examined methods relatively comparable in terms of ease-of-use and learnability.

9.7.5 Confidence. A Friedman test identified a significant effect of feedback on confidence ($\chi^2 = 12.56, df = 2, p < .01, W = 0.5$). A Games-Howell test indicated that participants exhibited a notably higher level of confidence when utilizing Google Assistant compared to both work-level and phrase-level MELDER. Their confidence levels in using the two variations of MELDER appeared to be relatively similar.

9.7.6 Privacy and Security. A Friedman test identified a significant effect of feedback on privacy and security ($\chi^2 = 24.0, df = 2, p < .0001, W = 1.0$). A Games-Howell test revealed that participants found both word-level and phrase-level MELDER to be significantly more secure and private than Google Assistant.

9.8 Discussion

MELDER was notably slower and displayed a higher error rate compared to Google Assistant. The discrepancy in text entry speed between the two methods was readily observed by all participants. They universally perceived MELDER, regardless of the feedback method, to be slower than Google Assistant. This affected their confidence in both variants of MELDER. This notably influenced participants' confidence levels. Participants reported feeling significantly more confident when using Google Assistant compared to both word-level and phrase-level MELDER. One participant (male, 26 years) commented, *"'I think silent speech is slower, and speed is really important in some cases. Apart from this, I think it is going to be an extremely cool piece of technology."*

Interestingly, participants found MELDER with phrase-level feedback to be relatively faster than MELDER with word-level feedback, even though both variants used the same underlying model. The majority of participants agreed with the statement that MELDER with phrase-level feedback is fast (N = 8), while a few remained neutral (N = 3), and only one participant disagreed with the statement. These results indicate that phrase-level feedback enhanced users' perception of the method's speed, despite the actual performance being similar.

Participants' perception of the accuracy of the examined methods yielded surprising results. Despite the fact that both variants of MELDER, with either word-level or phrase-level feedback, displayed significantly higher error rates compared to Google Assistant, participants did not perceive them as notably error-prone. In

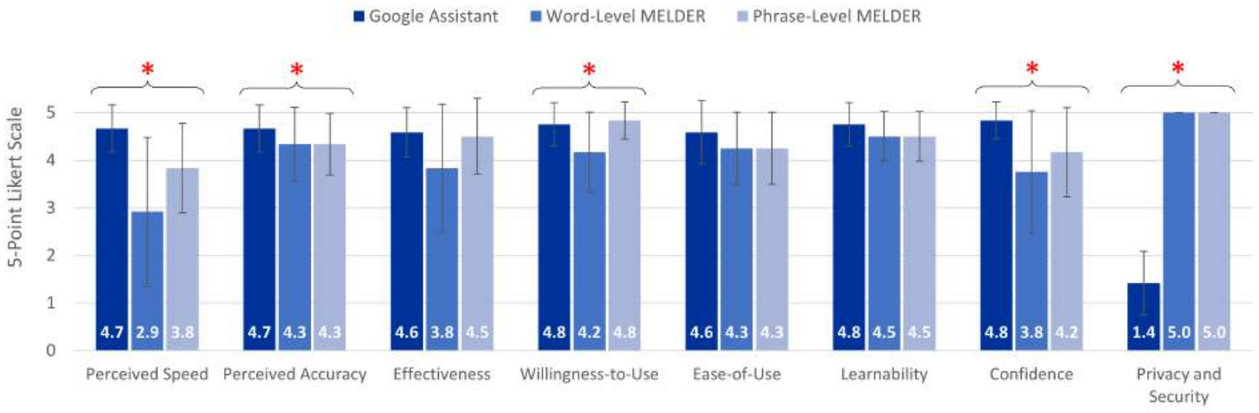


Figure 12: Average user ratings of the three methods on a 5-point Likert scale, where 1–5 signifies disagree–agree. Error bars represent ±1 standard deviation (SD). Red asterisks represent statistically significant differences.

fact, the vast majority of participants agreed with the statement that the method is accurate (N = 11), with only one participant expressing a neutral opinion on the matter. It is important to note that while a Friedman test identified a statistically significant difference in error rates between the methods, a post-hoc multiple-comparison analysis did not confirm this significance. This suggests that participants' perceptions of accuracy may not align with the quantitative error rates, highlighting an interesting aspect of user perception in human-computer interaction studies.

Participants' perception of the performance of MELDER with phrase-level feedback had a clear impact on their willingness to use the different methods. They expressed a significantly higher willingness to use both Google Assistant and MELDER with phrase-level feedback compared to MELDER with word-level feedback. This observation underscores the potential effectiveness of the proposed methods and the feedback approaches employed in the study. Participants' willingness to use MELDER with phrase-level feedback was also positively influenced by their perception of the method's security and privacy features. They viewed both variants of MELDER as significantly more private and secure compared to Google Assistant, primarily because bystanders could not overhear their interactions. Some participants even indicated that they would consider using the method primarily for its privacy and security benefits. For instance, one participant (female, 21 years) stated, *"Due to its privacy benefits, it is extremely useful."* These findings align with prior research on the perceived privacy and security advantages of speech and silent speech-based input methods [81].

The results showed that participants found both Google Assistant and the two variations of MELDER to be relatively comparable in terms of effectiveness, ease of use, and learnability. While there were slight variations in the ratings for these three methods, a Friedman test did not detect any statistically significant differences in these aspects. Furthermore, participants expressed that both variants of MELDER were easy to use, and they believed that their performance would improve with practice. As one participant (female, 21 years) noted, *"Adapting to silent speech was challenging at first, but became easier as I progressed."* This feedback suggests that users may require some time to acclimate to silent speech input but can become more proficient with practice.

10 CONCLUSION

In this comprehensive work, we have successfully developed a real-time silent speech recognition system tailored for mobile devices. Our approach involves breaking down the input video into smaller temporal segments, processing them individually, and utilizing advanced language models to auto-correct output at both character and word-levels. Additionally, our system offers users valuable feedback on the silent speech recognition process.

The work began with an experiment where we explored four different windowing functions for segmenting video lips, ultimately determining that a linear function ($y = x + 5$) yielded the best performance. Building upon this, we introduced a transfer learning approach aimed at enhancing the capabilities of silent speech recognition models for everyday conversational contexts. We investigated three strategies for transferring learning with three existing silent speech models, with the Finetune_Sequence strategy emerging as the most effective, showcasing its potential for improving the performance of existing pre-trained models. Equipped with the linear slicing function and the Finetune_Sequence transfer learning approach, we compared our system, MELDER, with two state-of-the-art silent speech models in two user studies–one in a stationary (seated position) and another in a mobile setting (while walking). The results demonstrated that MELDER outperformed both methods, establishing its feasibility for mobile device use. Furthermore, we conducted a qualitative study comparing our proposed word-level and phrase-level visual feedback methods with Google Assistant's feedback mechanism. Interestingly, the study revealed that users' perceived performance did not always align with actual performance. Notably, the phrase-level feedback significantly enhanced users' perception of the silent speech model.

In conclusion, this work firmly establishes silent speech as a viable and effective method for interacting with mobile devices. As part of our commitment to advancing research in this field, we have made the dataset, source code, and other materials generated during this study freely available for download. We hope that this will encourage further investigations and replication efforts in this promising area of study.

11 FUTURE WORK

In future work, we plan to investigate various manual error correction strategies, empowering users to effectively correct recognition errors. Additionally, our aim is to further optimize the algorithm, enhancing its speed, accuracy, and adaptability, especially for individuals with diverse speech disorders. We also intend to conduct more in-depth studies to thoroughly examine the usability, adaptiveness, and robustness of the model. Moreover, testing the method in varied settings, such as under different lighting conditions and noise levels, is also part of our future research agenda.

ACKNOWLEDGMENTS

This work has been funded in part by a National Science Foundation (NSF) CAREER grant, Award # 2239633.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. https://doi.org/10.48550/arXiv.1603.04467 arXiv:1603.04467 [cs].

[2] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2022. Deep Audio-Visual Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (Dec. 2022), 8717–8727. https://doi.org/10.1109/TPAMI.2018.2889052 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. Deep Lip Reading: A Comparison of Models and an Online Application. https://doi.org/10.48550/arXiv.1806.06053 arXiv:1806.06053 [cs].

[4] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. LRS3-TED: A Large-Scale Dataset for Visual Speech Recognition. https://doi.org/10.48550/arXiv.1809.00496 arXiv:1809.00496 [cs].

[5] Abien Fred Agarap. 2019. Deep Learning using Rectified Linear Units (ReLU). https://doi.org/10.48550/arXiv.1803.08375 arXiv:1803.08375 [cs, stat].

[6] Ibrahim Almajai, Stephen Cox, Richard Harvey, and Yuxuan Lan. 2016. Improved Speaker Independent Lip Reading Using Speaker Adaptive Training and Deep

Neural Networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2722–2726. https://doi.org/10.1109/ICASSP.2016.7472172 ISSN: 2379-190X.

[7] Ahmed Sabbir Arif. 2021. Statistical Grounding. In *Intelligent Computing for Interactive System Design: Statistics, Digital Signal Processing, and Machine Learning in Practice* (1 ed.). Association for Computing Machinery, New York, NY, USA, 59–99. https://doi.org/10.1145/3447404.3447410

[8] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of Text Entry Performance Metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[9] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, and Nando de Freitas. 2016. LipNet: End-to-End Sentence-level Lipreading. *arXiv:1611.01599 [cs]* (Dec. 2016). http://arxiv.org/abs/1611.01599 arXiv: 1611.01599.

[10] Jess Bartels, Dinal Andreasen, Princewill Ehirim, Hui Mao, Steven Seibert, E. Joe Wright, and Philip Kennedy. 2008. Neurotrophic Electrode: Method of Assembly and Implantation into Human Motor Speech Cortex. *Journal of Neuroscience Methods* 174, 2 (Sept. 2008), 168–176. https://doi.org/10.1016/j.jneumeth.2008.06.030

[11] Helen L. Bear and Richard Harvey. 2019. Alternative Visual Units for an Optimized Phoneme-Based Lipreading System. *Applied Sciences* 9, 18 (Jan. 2019), 3870. https://doi.org/10.3390/app9183870 Number: 18 Publisher: Multidisciplinary Digital Publishing Institute.

[12] John Brooke. 1996. SUS - a Quick and Dirty Usability Scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7. https://doi.org/10.1371/journal.pone.0170531

[13] Jonathan S. Brumberg, Alfonso Nieto-Castanon, Philip R. Kennedy, and Frank H. Guenther. 2010. Brain–Computer Interfaces for Speech Communication. *Speech Communication* 52, 4 (April 2010), 367–379. https://doi.org/10.1016/j.specom.2010.01.001

[14] Tuochao Chen, Benjamin Steeper, Kinan Alsheikh, Songyun Tao, François Guimbretière, and Cheng Zhang. 2020. C-Face: Continuously Reconstructing Facial Expressions by Deep Learning Contours of the Face with Ear-mounted Miniature Cameras. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 112–125. https://doi.org/10.1145/3379337.3415879

[15] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. https://doi.org/10.48550/arXiv.1412.3555 arXiv:1412.3555 [cs].

[16] Joon Son Chung and Andrew Zisserman. 2017. Lip Reading in the Wild. In *Computer Vision – ACCV 2016 (Lecture Notes in Computer Science)*, Shang-Hong Lai, Vincent Lepetit, Ko Nishino, and Yoichi Sato (Eds.). Springer International Publishing, Cham, 87–103. https://doi.org/10.1007/978-3-319-54184-6_6

[17] Joon Son Chung and Andrew Zisserman. 2017. Out of Time: Automated Lip Sync in the Wild. In *Computer Vision – ACCV 2016 Workshops (Lecture Notes in Computer Science)*, Chu-Song Chen, Jiwen Lu, and Kai-Kuang Ma (Eds.). Springer International Publishing, Cham, 251–263. https://doi.org/10.1007/978-3-319-54427-4_19

[18] Joon Son Chung and Andrew Zisserman. 2018. Learning to Lip Read Words by Watching Videos. *Computer Vision and Image Understanding* 173 (Aug. 2018), 76–85. https://doi.org/10.1016/j.cviu.2018.02.001

[19] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences* (2 ed.). Routledge, New York, NY, USA. https://doi.org/10.4324/9780203771587

[20] Ronan Collobert, Awni Hannun, and Gabriel Synnaeve. 2019. A Fully Differentiable Beam Search Decoder. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 1341–1350. https://proceedings.mlr.press/v97/collobert19a.html ISSN: 2640-3498.

[21] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. 2006. An Audio-Visual Corpus for Speech Perception and Automatic Speech Recognition. *The Journal of the Acoustical Society of America* 120, 5 (Nov. 2006), 2421–2424. https://doi.org/10.1121/1.2229005

[22] Charles S. DaSalla, Hiroyuki Kambara, Yasuharu Koike, and Makoto Sato. 2009. Spatial Filtering and Single-Trial Classification of EEG During Vowel Speech Imagery. In *Proceedings of the 3rd International Convention on Rehabilitation Engineering & Assistive Technology (i-CREATe '09)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/1592700.1592731

[23] Mark Davies. 2022. N-Grams Based on 520 Million Word COCA Corpus. https://www.ngrams.info/coca2020.asp

[24] 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 ISSN: 2379-190X.

[25] B. Denby and M. Stone. 2004. Speech Synthesis from Real Time Ultrasound Images of the Tongue. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. I–685. https://doi.org/10.1109/ICASSP.2004.1326078 ISSN: 1520-6149.

[26] Aarthi Easwara Moorthy and Kim-Phuong L. Vu. 2014. Voice Activated Personal Assistant: Acceptability of Use in the Public Space. In *Human Interface and the Management of Information. Information and Knowledge in Applications and Services (Lecture Notes in Computer Science)*, Sakae Yamamoto (Ed.). Springer International Publishing, Cham, 324–334. https://doi.org/10.1007/978-3-319-07863-2_32

[27] Aarthi Easwara Moorthy and Kim-Phuong L. Vu. 2015. Privacy Concerns for Use of Voice Activated Personal Assistant in the Public Space. *International Journal of Human–Computer Interaction* 31, 4 (April 2015), 307–335. https://doi.org/10.1080/10447318.2014.986642 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2014.986642.

[28] M. J. Fagan, S. R. Ell, J. M. Gilbert, E. Sarrazin, and P. M. Chapman. 2008. Development of a (silent) Speech Recognition System for Patients Following Laryngectomy. *Medical Engineering & Physics* 30, 4 (May 2008), 419–425. https://doi.org/10.1016/j.medengphy.2007.05.003

[29] Victoria M. Florescu, Lise Crevier-Buchman, Bruce Denby, Thomas Hueber, Antonia Colazo-Simon, Claire Pillot-Loiseau, Pierre Roussel, Cédric Gendrot, and Sophie Quattrocchi. 2010. Silent vs Vocalized Articulation for a Portable Ultrasound-Based Silent Speech Interface. In *Interspeech 2010*. ISCA, 450–453. https://doi.org/10.21437/Interspeech.2010-195

[30] Masaaki Fukumoto. 2018. SilentVoice: Unnoticeable Voice Input by Ingressive Speech. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 237–246. https://doi.org/10.1145/3242587.3242603

[31] Yang Gao, Yincheng Jin, Jiyang Li, Seokmin Choi, and Zhanpeng Jin. 2020. EchoWhisper: Exploring an Acoustic-based Silent Speech Interface for Smartphone Users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (Sept. 2020), 80:1–80:27. https://doi.org/10.1145/3411830

[32] J. M. Gilbert, S. I. Rybchenko, R. Hofe, S. R. Ell, M. J. Fagan, R. K. Moore, and P. Green. 2010. Isolated Word Recognition of Silent Speech Using Magnetic Implants and Sensors. *Medical Engineering & Physics* 32, 10 (Dec. 2010), 1189–1197. https://doi.org/10.1016/j.medengphy.2010.08.011

[33] Alexandre Gonfalonieri. 2018. How Amazon Alexa Works? Your Guide to Natural Language Processing (AI). https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3

[34] J.N. Gowdy, A. Subramanya, C. Bartels, and J. Bilmes. 2004. DBN Based Multi-Stream Models for Audio-Visual Speech Recognition. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1. I–993. https://doi.org/10.1109/ICASSP.2004.1326155 ISSN: 1520-6149.

[35] 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 (ICML '06)*. Association for Computing Machinery, New York, NY, USA, 369–376. https://doi.org/10.1145/1143844.1143891

[36] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-Augmented Transformer for Speech Recognition. https://doi.org/10.48550/arXiv.2005.08100 arXiv:2005.08100 [cs, eess].

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 770–778. https://doi.org/10.1109/CVPR.2016.90

[38] Panikos Heracleous and Norihiro Hagita. 2011. Automatic Recognition of Speech Without Any Audio Information. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2392–2395. https://doi.org/10.1109/ICASSP.2011.5946965 ISSN: 2379-190X.

[39] Panikos Heracleous, Tomomi Kaino, Hiroshi Saruwatari, and Kiyohiro Shikano. 2006. Unvoiced Speech Recognition Using Tissue-Conductive Acoustic Sensor. *EURASIP Journal on Advances in Signal Processing* 2007, 1 (Dec. 2006), 1–11. https://doi.org/10.1155/2007/94068 Number: 1 Publisher: SpringerOpen.

[40] Tatsuya Hirahara, Makoto Otani, Shota Shimizu, Tomoki Toda, Keigo Nakamura, Yoshitaka Nakajima, and Kiyohiro Shikano. 2010. Silent-Speech Enhancement Using Body-Conducted Vocal-Tract Resonance Signals. *Speech Communication* 52, 4 (April 2010), 301–313. https://doi.org/10.1016/j.specom.2009.12.001

[41] Hirotaka Hiraki and Jun Rekimoto. 2022. SilentWhisper: Faint Whisper Speech Using Wearable Microphone. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 1–3. https://doi.org/10.1145/3526114.3558715

[42] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. 7132–7141. https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html

[43] T. Hueber, G. Aversano, G. Chollet, B. Denby, G. Dreyfus, Y. Oussar, P. Roussel, and M. Stone. 2007. Eigentongue Feature Extraction for an Ultrasound-Based Silent Speech Interface. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Vol. 1. I–1245–I–1248. https://doi.org/10.1109/ICASSP.2007.366140 ISSN: 2379-190X.

[44] 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 (April 2010), 288–300. https://doi.org/10.1016/j.specom.

2009.11.004

[45] Hiroki Ikeda, Takashi Ohhira, and Hideki Hashimoto. 2023. Classification of Silent Speech Words Considering Walking Using VMD Applied Facial EMG. *International Symposium on Affective Science and Engineering* ISASE2023 (2023), 1–4. https://doi.org/10.5057/isase.2023-C000013

[46] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 448–456. https://proceedings.mlr.press/v37/ioffe15.html ISSN: 1938-7228.

[47] 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 (July 2022), 57:1–57:28. https://doi.org/10.1145/3534613

[48] Charles Jorgensen and Sorin Dusan. 2010. Speech Interfaces Based Upon Surface Electromyography. *Speech Communication* 52, 4 (April 2010), 354–366. https://doi.org/10.1016/j.specom.2009.11.003

[49] C. Jorgensen, D.D. Lee, and S. Agabont. 2003. Sub Auditory Speech Recognition Based on EMG Signals. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, Vol. 4. 3128–3133 vol.4. https://doi.org/10.1109/IJCNN.2003.1224072 ISSN: 1098-7576.

[50] Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. 2006. Towards Continuous Speech Recognition Using Surface Electromyography. (2006).

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

[52] Arnav Kapur, Utkarsh Sarawgi, Eric Wadkins, Matthew Wu, Nora Hollenstein, and Pattie Maes. 2020. Non-Invasive Silent Speech Recognition in Multiple Sclerosis with Dysphonia. In *Proceedings of the Machine Learning for Health NeurIPS Workshop*. PMLR, 25–38. https://proceedings.mlr.press/v116/kapur20a.html ISSN: 2640-3498.

[53] Sara Kashiwagi, Keitaro Tanaka, Qi Feng, and Shigeo Morishima. 2023. Improving the Gap in Visual Speech Recognition Between Normal and Silent Speech Based on Metric Learning. https://doi.org/10.48550/arXiv.2305.14203 arXiv:2305.14203 [cs, eess].

[54] Pavel Khlebovich. 2023. IP Webcam - Apps on Google Play. https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en_US

[55] 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 *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3491102.3502015

[56] 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 (AVI '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3399715.3399852

[57] Naoki Kimura, Michinari Kono, and Jun Rekimoto. 2019. SottoVoce: An Ultrasound Imaging-Based Silent Speech Interaction Using Deep Neural Networks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3290605.3300376

[58] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *The Journal of Machine Learning Research* 10 (2009), 1755–1758. Publisher: JMLR. org.

[59] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980 arXiv:1412.6980 [cs].

[60] Oscar Koller, Hermann Ney, and Richard Bowden. 2015. Deep Learning of Mouth Shapes for Sign Language. 85–91. https://www.cv-foundation.org/openaccess/content_iccv_2015_workshops/w12/html/Koller_Deep_Learning_of_ICCV_2015_paper.html

[61] L Ashok Kumar, D Karthika Renuka, S Lovelyn Rose, M C Shunmuga priya, and I Made Wartana. 2022. Deep Learning Based Assistive Technology on Audio Visual Speech Recognition for Hearing Impaired. *International Journal of Cognitive Computing in Engineering* 3 (June 2022), 24–30. https://doi.org/10.1016/j.ijcce.2022.01.003

[62] Yusuke Kunimi, Masa Ogata, Hirotaka Hiraki, Motoshi Itagaki, Shusuke Kanazawa, and Masaaki Mochimaru. 2022. E-MASK: A Mask-Shaped Interface for Silent Speech Interaction with Flexible Strain Sensors. In *Proceedings of the Augmented Humans International Conference 2022 (AHs '22)*. Association for Computing Machinery, New York, NY, USA, 26–34. https://doi.org/10.1145/3519391.3519399

[63] Vladimir I Levenshtein and others. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. In *Soviet physics doklady*, Vol. 10. MAIK Nauka/Interperiodica, Soviet Union, 707–710. Issue: 8.

[64] 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 (AH2019)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3311823.3311831

[65] Christine Chong-hee Lieu, Georgia Robins Sadler, Judith T Fullerton, and Paulette Deyo Stohlmann. 2007. Communication Strategies for Nurses Interacting with Patients Who Are Deaf. *Dermatology Nursing* 19, 6 (2007), 541. Publisher: Anthony J. Jannetti, Inc..

[66] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenyao Xu, and Kui Ren. 2021. Wavoice: A Noise-resistant Multi-modal Speech Recognition System Fusing mmWave and Audio Signals. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21)*. Association for Computing Machinery, New York, NY, USA, 97–110. https://doi.org/10.1145/3485730.3485945

[67] Karen Livescu, Ozgur Cetin, Mark Hasegawa-Johnson, Simon King, Chris Bartels, Nash Borges, Arthur Kantor, Partha Lal, Lisa Yung, Ari Bezman, Stephen Dawson-Haggerty, Bronwyn Woods, Joe Frankel, Mathew Magimai-Doss, and Kate Saenko. 2007. Articulatory Feature-Based Methods for Acoustic and Audio-Visual Speech Recognition: Summary from the 2006 JHU Summer workshop. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Vol. 4. IV–621–IV–624. https://doi.org/10.1109/ICASSP.2007.366989 ISSN: 2379-190X.

[68] Andrew Maas, Ziang Xie, Dan Jurafsky, and Andrew Ng. 2015. Lexicon-Free Conversational Speech Recognition with Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, 345–354. https://doi.org/10.3115/v1/N15-1038

[69] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971 event-place: Ft. Lauderdale, Florida, USA.

[70] L. Maier-Hein, F. Metze, T. Schultz, and A. Waibel. 2005. Session Independent Non-Audible Speech Recognition Using Surface Electromyography. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* 331–336. https://doi.org/10.1109/ASRU.2005.1566521

[71] M. S. Mayzner and M. E. Tresselt. 1965. Tables of Single-Letter and Digram Frequency Counts for Various Word-Length and Letter-Position Combinations. *Psychonomic Monograph Supplements* 1, 2 (1965), 13–32.

[72] Y. Nakajima, H. Kashioka, K. Shikano, and N. Campbell. 2003. Non-Audible Murmur Recognition Input Interface Using Stethoscopic Microphone Attached to the Skin. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, Vol. 5. V–708. https://doi.org/10.1109/ICASSP.2003.1200069 ISSN: 1520-6149.

[73] L.C. Ng, G.C. Burnett, J.F. Holzrichter, and T.J. Gable. 2000. Denoising of Human Speech Using Combined Acoustic and EM Sensor Signal Processing. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, Vol. 1. 229–232 vol.1. https://doi.org/10.1109/ICASSP.2000.861925 ISSN: 1520-6149.

[74] Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. 2014. Lipreading Using Convolutional Neural Network. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Singapore, 1149–1153.

[75] Eng-Jon Ong and Richard Bowden. 2011. Learning Temporal Signatures for Lip Reading. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 958–965. https://doi.org/10.1109/ICCVW.2011.6130355

[76] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (Oct. 2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[77] Laxmi Pandey and Ahmed Sabbir Arif. 2021. LipType: A Silent Speech Recognizer Augmented with an Independent Repair Model. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3411764.3445565

[78] Laxmi Pandey and Ahmed Sabbir Arif. 2021. Silent Speech and Emotion Recognition from Vocal Tract Shape Dynamics in Real-Time MRI. *arXiv:2106.08706 [cs, eess]* (June 2021). http://arxiv.org/abs/2106.08706 arXiv: 2106.08706.

[79] Laxmi Pandey and Ahmed Sabbir Arif. 2022. Design and Evaluation of a Silent Speech-Based Selection Method for Eye-Gaze Pointing. *Proceedings of the ACM on Human-Computer Interaction* 6, ISS (Nov. 2022), 570:328–570:353. https://doi.org/10.1145/3567723

[80] Laxmi Pandey and Ahmed Sabbir Arif. 2022. Effects of Speaking Rate on Speech and Silent Speech Recognition. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3491101.3519611

[81] Laxmi Pandey, Khalad Hasan, and Ahmed Sabbir Arif. 2021. Acceptability of Speech and Silent Speech Input Methods in Private and Public. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM, New

York, NY, USA, Yokohama, Japan, 13 pages. https://doi.org/10.1145/3411764.3445430

[82] George Papandreou, Athanassios Katsamanis, Vassilis Pitsikalis, and Petros Maragos. 2009. Adaptive Multimodal Fusion by Uncertainty Compensation With Application to Audiovisual Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 17, 3 (March 2009), 423–435. https://doi.org/10.1109/TASL.2008.2011515 Conference Name: IEEE Transactions on Audio, Speech, and Language Processing.

[83] Sanjay A. Patil and John H. L. Hansen. 2010. The Physiological Microphone (PMIC): A Competitive Alternative for Speaker Assessment in Stress Detection and Speaker Verification. *Speech Communication* 52, 4 (April 2010), 327–340. https://doi.org/10.1016/j.specom.2009.11.006

[84] Stavros Petridis and Maja Pantic. 2016. Deep Complementary Bottleneck Features for Visual Speech Recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2304–2308. https://doi.org/10.1109/ICASSP.2016.7472088 ISSN: 2379-190X.

[85] Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic. 2018. Audio-Visual Speech Recognition with a Hybrid CTC/Attention Architecture. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. 513–520. https://doi.org/10.1109/SLT.2018.8639643

[86] Anne Porbadnigk, Marek Wester, Jan Calliess, and Tanja Schultz. 2009. EEG-Based Speech Recognition - Impact of Temporal Effects, Vol. 1. SCITEPRESS, 376–381. https://doi.org/10.5220/0001554303760381

[87] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior. 2003. Recent Advances in the Automatic Recognition of Audiovisual Speech. *Proc. IEEE* 91, 9 (Sept. 2003), 1306–1326. https://doi.org/10.1109/JPROC.2003.817150 Conference Name: Proceedings of the IEEE.

[88] T.F. Quatieri, K. Brady, D. Messing, J.P. Campbell, W.M. Campbell, M.S. Brandstein, C.J. Weinstein, J.D. Tardelli, and P.D. Gatewood. 2006. Exploiting Nonacoustic Sensors for Speech Encoding. *IEEE Transactions on Audio, Speech, and Language Processing* 14, 2 (March 2006), 533–544. https://doi.org/10.1109/TSA.2005.855838 Conference Name: IEEE Transactions on Audio, Speech, and Language Processing.

[89] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James A. Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (Jan. 2018), 159:1–159:23. https://doi.org/10.1145/3161187

[90] Adam D. Rubin, Veeraphol Praneetvatakul, Shirley Gherson, Cheryl A. Moyer, and Robert T. Sataloff. 2006. Laryngeal Hyperfunction During Whispering: Reality or Myth? *Journal of Voice* 20, 1 (March 2006), 121–127. https://doi.org/10.1016/j.jvoice.2004.10.007

[91] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2013. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. 397–403. https://www.cv-foundation.org/openaccess/content_iccv_workshops_2013/W11/html/Sagonas_300_Faces_in-the-Wild_2013_ICCV_paper.html

[92] Zhanna Sarsenbayeva, Vassilis Kostakos, and Jorge Goncalves. 2019. Situationally-Induced Impairments and Disabilities Research. *arXiv:1904.06128 [cs]* (April 2019). http://arxiv.org/abs/1904.06128 arXiv: 1904.06128.

[93] Tanja Schultz and Michael Wand. 2010. Modeling Coarticulation in EMG-Based Continuous Speech Recognition. *Speech Communication* 52, 4 (April 2010), 341–353. https://doi.org/10.1016/j.specom.2009.12.002

[94] Brendan Shillingford, Yannis Assael, Matthew W. Hoffman, Thomas Paine, Cían Hughes, Utsav Prabhu, Hank Liao, Hasim Sak, Kanishka Rao, Lorrayne Bennett, Marie Mulville, Ben Coppin, Ben Laurie, Andrew Senior, and Nando de Freitas. 2018. Large-Scale Visual Speech Recognition. https://doi.org/10.48550/arXiv.1807.05162 arXiv:1807.05162 [cs].

[95] Preethi S.j. and Niranjana Krupa B. 2023. Analyzing Lower Half Facial Gestures for Lip Reading Applications: Survey on Vision Techniques. *Computer Vision and Image Understanding* 233 (Aug. 2023), 103738. https://doi.org/10.1016/j.cviu.2023.103738

[96] Deepak Soekhoe, Peter van der Putten, and Aske Plaat. 2016. On the Impact of Data Set Size in Transfer Learning Using Deep Neural Networks. In *Advances in Intelligent Data Analysis XV (Lecture Notes in Computer Science)*, Henrik Boström, Arno Knobbe, Carlos Soares, and Panagiotis Papapetrou (Eds.). Springer International Publishing, Cham, 50–60. https://doi.org/10.1007/978-3-319-46349-0_5

[97] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip Reading Sentences in the Wild. 6447–6456. https://openaccess.thecvf.com/content_cvpr_2017/html/Chung_Lip_Reading_Sentences_CVPR_2017_paper.html

[98] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958. Publisher: JMLR. org.

[99] Tanmay Srivastava, Prerna Khanna, Shijia Pan, Phuc Nguyen, and Shubham Jain. 2022. MuteIt: Jaw Motion Based Unvoiced Command Recognition Using Earable. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (Sept. 2022), 140:1–140:26. https://doi.org/10.1145/3550281

[100] Themos Stafylakis and Georgios Tzimiropoulos. 2017. Combining Residual Networks with LSTMs for Lipreading. https://doi.org/10.48550/arXiv.1703.04105 arXiv:1703.04105 [cs].

[101] Zixiong Su, Shitao Fang, and Jun Rekimoto. 2023. LipLearner: Customizable Silent Speech Interactions on Mobile Devices. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–21. https://doi.org/10.1145/3544548.3581465

[102] Zixiong Su, Xinlei Zhang, Naoki Kimura, and Jun Rekimoto. 2021. Gaze+Lip: Rapid, Precise and Expressive Interactions Combining Gaze Input and Silent Speech Commands for Hands-free Smart TV Control. In *ACM Symposium on Eye Tracking Research and Applications (ETRA '21 Short Papers)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3448018.3458011

[103] 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 (UIST '18)*. Association for Computing Machinery, New York, NY, USA, 581–593. https://doi.org/10.1145/3242587.3242599

[104] Patrick Suppes, Bing Han, and Zhong-Lin Lu. 1998. Brain-Wave Recognition of Sentences. *Proceedings of the National Academy of Sciences* 95, 26 (Dec. 1998), 15861–15866. https://doi.org/10.1073/pnas.95.26.15861 Publisher: Proceedings of the National Academy of Sciences.

[105] Patrick Suppes, Zhong-Lin Lu, and Bing Han. 1997. Brain Wave Recognition of Words. *Proceedings of the National Academy of Sciences* 94, 26 (Dec. 1997), 14965–14969. https://doi.org/10.1073/pnas.94.26.14965 Publisher: Proceedings of the National Academy of Sciences.

[106] Satoshi Tamura, Hiroshi Ninomiya, Norihide Kitaoka, Shin Osuga, Yurie Iribe, Kazuya Takeda, and Satoru Hayamizu. 2015. Audio-Visual Speech Recognition Using Deep Bottleneck Features and High-Performance Lipreading. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. 575–582. https://doi.org/10.1109/APSIPA.2015.7415335

[107] Ingo R. Titze, Brad H. Story, Gregory C. Burnett, John F. Holzrichter, Lawrence C. Ng, and Wayne A. Lea. 2000. Comparison Between Electroglottography and Electromagnetic Glottography. *The Journal of the Acoustical Society of America* 107, 1 (Jan. 2000), 581–588. https://doi.org/10.1121/1.428324

[108] Naoya Ukai, Takumi Seko, Satoshi Tamura, and Satoru Hayamizu. 2012. GIF-LR:GA-Based Informative Feature for Lipreading. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*. 1–4.

[109] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[110] Keith Vertanen and Per Ola Kristensson. 2011. A Versatile Dataset for Text Entry Evaluations Based on Genuine Mobile Emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. Association for Computing Machinery, New York, NY, USA, 295–298. https://doi.org/10.1145/2037373.2037418

[111] Michael Wand, Jan Koutník, and Jürgen Schmidhuber. 2016. Lipreading with Long Short-Term Memory. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6115–6119. https://doi.org/10.1109/ICASSP.2016.7472852 ISSN: 2379-190X.

[112] Michael Wand and Tanja Schultz. 2011. Session-Independent EMG-Based Speech Recognition:. In *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*. SciTePress - Science and and Technology Publications, Rome, Italy, 295–300. https://doi.org/10.5220/0003169702950300

[113] Wikipedia. 2022. Ablation (artificial Intelligence). https://en.wikipedia.org/w/index.php?title=Ablation_(artificial_intelligence)&oldid=1097614343 Page Version ID: 1097614343.

[114] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. 2019. WaveEar: Exploring a mmWave-based Noise-resistant Speech Sensing for Voice-User Interface. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '19)*. Association for Computing Machinery, New York, NY, USA, 14–26. https://doi.org/10.1145/3307334.3326073

[115] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How Transferable Are Features in Deep Neural Networks?. In *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/hash/375c71349b295fbe2dcdca9206f20a06-Abstract.html

[116] Shang Zeng, Haoran Wan, Shuyu Shi, and Wei Wang. 2023. mSilent: Towards General Corpus Silent Speech Recognition Using COTS mmWave Radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (March 2023), 39:1–39:28. https://doi.org/10.1145/3580838

[117] Ruidong Zhang, Mingyang Chen, Benjamin Steeper, Yaxuan Li, Zihan Yan, Yizhuo Chen, Songyun Tao, Tuochao Chen, Hyunchul Lim, and Cheng Zhang. 2022. SpeeChin: A Smart Necklace for Silent Speech Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (Dec. 2022), 192:1–192:23. https://doi.org/10.1145/3494987

[118] Ruidong Zhang, Ke Li, Yihong Hao, Yufan Wang, Zhengnan Lai, François Guimbretière, and Cheng Zhang. 2023. EchoSpeech: Continuous Silent Speech Recognition on Minimally-obtrusive Eyewear Powered by Acoustic Sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. https://doi.org/10.1145/3544548.3580801

[119] Xingxuan Zhang, Feng Cheng, and Shilin Wang. 2019. Spatio-Temporal Fusion Based Convolutional Sequence Learning for Lip Reading. 713–722. https://openaccess.thecvf.com/content_ICCV_2019/html/Zhang_Spatio-Temporal_Fusion_Based_Convolutional_Sequence_Learning_for_Lip_Reading_ICCV_2019_paper.html

[120] Ziheng Zhou, Guoying Zhao, Xiaopeng Hong, and Matti Pietikäinen. 2014. A Review of Recent Advances in Visual Speech Decoding. *Image and Vision Computing* 32, 9 (Sept. 2014), 590–605. https://doi.org/10.1016/j.imavis.2014.06.004

[121] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2021. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* 109, 1 (Jan. 2021), 43–76. https://doi.org/10.1109/JPROC.2020.3004555 Conference Name: Proceedings of the IEEE.

A EXPERIMENTAL DATASET

This appendix lists the phrases chosen from the Enron Mobile Email corpus [110] for evaluating the proposed silent speech model.

(1) are you going to join us for lunch
(2) thanks for the quick turnaround
(3) please call tomorrow if possible
(4) are you getting all the information you need
(5) she has absolutely everything
(6) we can have wine and catch up
(7) i agree since i am at the bank right now
(8) i wanted to go drinking with you
(9) both of us are still here
(10) we need to talk about this month
(11) this seems fine to me
(12) is this the only time available
(13) do you want to fax it to my hotel
(14) i hope he is having a fantastic time
(15) can you help get this cleared up
(16) i would be glad to participate
(17) i worked on the grade level promotion
(18) that would likely be an expensive option
(19) we are waiting on the cold front
(20) you have a nice holiday too
(21) what is the cost issue
(22) i changed that in one prior draft
(23) we must be consistent
(24) we just need a sitter
(25) thanks for your concern
(26) has anyone else heard anything
(27) take what you can get
(28) call me to give me a heads up
(29) they are more efficiently pooled
(30) i am out of town on business tonight