Lip-Interact: Improving Mobile Device Interaction with Silent Speech Commands

Ke Sun1,2, Chun Yu1,23†, Weinan Shi1,2, Lan Liu1, Yuanchun Shi1,2,3
1Department of Computer Science and Technology
2Key Laboratory of Pervasive Computing, Ministry of Education
3Global Innovation eXchange Institute, Tsinghua University, Beijing, 100084, China
{k-sun14, swn16, liulan17}@mails.tsinghua.edu.cn, {chunyu,shiyc}@tsinghua.edu.cn

ABSTRACT
We present Lip-Interact, an interaction technique that allows users to issue commands on their smartphone through silent speech. Lip-Interact repurposes the front camera to capture the user’s mouth movements and recognize the issued commands with an end-to-end deep learning model. Our system supports 44 commands for accessing both system-level functionalities (launching apps, changing system settings, and handling pop-up windows) and application-level functionalities (integrated operations for two apps). We verify the feasibility of Lip-Interact with three user experiments: evaluating the recognition accuracy, comparing with touch on input efficiency, and comparing with voiced commands with regards to personal privacy and social norms. We demonstrate that Lip-Interact can help users access functionality efficiently in one step, enable one-handed input when the other hand is occupied, and assist touch to make interactions more fluent.

CCS Concepts
•Human-centered computing → Interaction techniques;

Author Keywords
Lip interaction; silent speech; mobile interaction; touch-free; semantic gesture; vision-based recognition.

INTRODUCTION
Today’s smartphones enable a wide range of functionalities. At present, most of the devices rely on touch as the primary input approach. However, simply tapping on the screen with a fingertip lacks expressivity. Given the limited screen size and hierarchical UI layouts, reaching a desired item often requires multiple steps of sliding and clicking. In addition, interactions can be more inconvenient when the user’s hands are occupied (e.g. holding a cup of coffee, wearing gloves in winter, etc.).

† denotes the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST ’18, October 14–17, 2018, Berlin, Germany
© 2018 ACM. ISBN 978-1-4503-5948-1/18/10…$15.00
DOI: https://doi.org/10.1145/3242587.3242599

In order to augment the input ability on a mobile device, several approaches have been explored previously, such as on-screen gestures [35, 1], keyboard shortcuts [32, 2], and extending the input space to the edge [7], back [49, 5], and above area [20, 9] of the device. Unfortunately, these gesture-based approaches suffer from one or more of the following issues: gesture designing, scalability of the gesture set, and gesture learning and memorizing. On the other hand, voice input is an alternative channel that offers rich expressivity and enables always-available interaction. However, voice recognition can be easily affected by ambient noise and people do not use it in many circumstances due to the privacy concerns [40].

In this paper, we introduce Lip-Interact, which enables users to access functionality on the smartphone by issuing silent speech commands (Figure 1). Lip-Interact repurposes the device’s front camera to capture the user’s mouth, segment silently speaking sequences and recognize lip movements into
commands by a deep-learning-based image sequence recognition technology. For example, to take a selfie, a user can simply mouth the verbal command "open camera" to directly launch the app in one step instead of searching for it manually, and then mouth "capture" to trigger the shutter button rather than reaching it with the unstable thumb and causing the picture to blur. Also, a user can issue commands such as: "home", "back", "screenshot", "flashlight" and so on to easily control the device’s settings.

The goal of Lip-Interact is to provide a simple, yet powerful approach to issuing commands on the smartphone. Compared with search-and-click-based touch interaction, Lip-Interact is essentially a kind of gesture (where the user performs one of a set of defined actions to express the input intention), thus enabling efficient one-step functionality access. Meanwhile, because the "gestures" in Lip-Interact are based on the natural language, it also avoids many limitations of conventional gesture input, such as the goodness of gesture-command mapping, the scalability of gesture set and the learning cost. Finally, in contrast to general audible voice input, the main difference is that Lip-Interact commands are silent and recognized by the camera. Therefore, the interaction is immune to ambient noise and largely alleviates the issues surrounding personal privacy and social norms in public environment.

The concept of silent speech is not new. Previous work has studied silent speech interfaces, but mainly focusing on sensing technologies. Most of the solutions attached additional sensors, for example on a helmet [22], facemask [26], or neckband [16]. To our knowledge, none has studied to use the on-device front camera to combine silent speech with the smartphone as a command input approach. We deem this to be a promising setup for practical use. We also for the first time study the feasibility of Lip-Interact from three aspects:

1) Design and Implementation. We discuss the design principles and present a set of interface mechanisms that support novice learning, visual feedback and online model personalization. We describe algorithms that automatically segment the mouth image sequences and recognize the commands with an end-to-end deep learning model. We finally implement a working Lip-Interact system that allows users to operate the smartphone in real time. 2) Accuracy. We collect silent speech data of 44 commands from 22 participants. The commands cover both system-level functions (launching apps, system quick settings, and handling pop-up windows) and application-level ones (Notepad and WeChat). Results show that for a context-based set of around 20 commands, the mean recognition accuracy is 95.4% for user-independent setting and can reach 98.9% for user-dependent setting. 3) Evaluation. With two controlled experiments, we show that Lip-Interact provides a more efficient and stable way of accessing functionality on a smartphone than touch, especially for single-handed input. In comparison with audible voice input, participants report significantly higher levels of agreement for Lip-Interact on privacy and social acceptance.

RELATED WORK
Lip-Interact intersects with three key literatures: gesture interaction, voice input and silent speech interfaces.

Accessing Functionality with Gestures
One of the main challenges when designing interfaces for touch-based mobile devices is the limited size of the display. Touch on GUIs is straightforward but may require a lot of search and multiple operations to reach the desired item [13]. To enhance the smartphone’s input ability, researchers have explored access to functionalitities through gestures [33, 2, 31, 8, 34, 1]. Gestures can be very useful on small command sets [48]. However, with larger sets, the design and the scalability of the gesture set, the effort required to learn and memorize it limit its widespread use.

Multimodal Interfaces and Voice Interaction
Multimodal interfaces process two or more combined user inputs to create new interaction possibilities, such as Pen+Voice [15] and Gaze+Touch [38]. Multimodal interfaces can support more flexible, efficient and expressive means of human-computer interaction [14], especially in mobile interaction when mobility may cause temporary disability and the user is unable to use a particular input [37]. A successful example is voice assistant, which is being integrated into modern smartphones, such as Google’s Assistant [19] and Siri [3]. Users can instruct the assistant to open apps, tap buttons and access specific settings without touching the screen. Although voice interaction is accurate and efficient, in our pilot survey of 15 users, only one user occasionally uses voice assistant to set calendar and another two users use speech for text input. All of them concerned about privacy issues (i.e. "I don’t want my interactions to be observed by others in my immediate surroundings") and social norms (i.e. "my voice may disturb the surrounding environment"), especially in public environments [40]. Lip-Interact moves one step further by keeping the external form of voice interaction but not vocalizing any sound. Therefore, Lip-Interact is expected to be less noticeable and be able to largely increase users’ privacy.

Silent Speech Interface and Lip Reading
A silent speech interface (SSI) [16] is a system that allows speech communication without using the sound made when people vocalize their speech sounds. Researchers have used various sensors (e.g. magnetic [17], EEG [39], EMG [46, 26], ultrasound imaging [18, 22] and non-audible murmur (NAM) microphone [36, 21]) to detect tongue, facial, throat movements and recover the speech content. Brumberg et al.[6] used an intracortical microelectrode brain computer interface (BCI) to predict users’ intended speech information. Hueber et al.[22] adopted a customized helmet where an ultrasound transducer was locked at a fixed position beneath the chin while an optical camera was fixed in front of the mouth. Features extracted from the two image sources were used for recognition. Recently, Kapur et al. [26] presented a wearable interface: AlterEgo, on which five EMG sensors were carefully placed above the face to capture the neuromuscular signals. The key difference between AlterEgo and previous SSIs is that their system works even when the user does not open the mouth. However, these prior works all use an invasive setup, impeding the scalability of these solutions in real-world scenarios.

With the recent advance of computer vision technologies, lip-reading from vision has achieved significant improvement
With the development of computer vision technology, the front camera on mobile devices has enabled a variety of new features, such as selfie beautification, face-based AR animation and face unlock. Lip-Interact offers users an opportunity to use the camera to "actively" interact with the device. The user silently issues commands by mouthing the verbal commands but not vocalizing the sound. The camera recognizes the commands, and the system triggers corresponding functionalities. Before implementing the Lip-Interact system, we first describe several key design principles of the technique.

**LIP-INTERACT DESIGNS**

With the development of computer vision technology, the front camera on mobile devices has enabled a variety of new features, such as selfie beautification, face-based AR animation and face unlock. Lip-Interact offers users an opportunity to use the camera to "actively" interact with the device. The user silently issues commands by mouthing the verbal commands but not vocalizing the sound. The camera recognizes the commands, and the system triggers corresponding functionalities. Before implementing the Lip-Interact system, we first describe several key design principles of the technique.

**Compatible with Existing Touch Interaction**

Since touch is straightforward and performs well in many contexts on mobile devices, the goal of Lip-Interact is to offer users an alternative based on their current cognitive and motor conditions when touch is inferior. In addition, Lip-Interact can also work together with touch to enable a more efficient and fluent user input experience. For example, a user may prefer to type with fingers while using Lip-Interact for fast text editing (e.g. bold, highlight, undo, etc).

**High Level of Recognition Accuracy**

The commands silently spoken by the user must be recognized accurately and robustly. A false-positive error in mobile interaction can be annoying to users and requires heavy effort to undo. However, silent speech recognition has intrinsic limitations. There are two main reasons for this. First, several phonemes often look the same (from human eye or sensing signals) when produced in general speech. This leads to a low recognition accuracy in general sentence lip reading with only a 50.2% word error rate [11], which is unusable for an interactive input system. Second, in everyday conversation people do very little or even no mouth movement so that the amount of information presented is not sufficient for recognition. This exacerbates the problem of low recognition accuracy. In order to make silent speech truly usable on a smartphone, we add the following two design constraints:

- **Concise and Differentiable Commands**
  Lip-Interact only considers short and visually distinguishable commands rather than long sentences such as those in a conversational system. This also allows users to have quick access to functionalities since most of the functionalities on mobile devices are discrete.

- **Properly Exaggerating Lip Movements**
  Users are instructed to issue commands in a more "standard" way than when they are casually talking by exaggerating their lip movements. Please note that the required exaggeration is still within a normal speaking range. Users only need to mouth the command by thinking that their lip movement can present each syllable clearly.

**Context-Aware Functionality Support**

Figure 2 illustrates how Lip-Interact is integrated into smartphone usage. Lip-Interact is context aware: covering different aspects of functionality when using different applications. The system identifies the current state and uses the corresponding model to recognize users’ input. When on the homescreen or on the recent-applications interface, the user can open a desired app by issuing e.g., "open camera". Within a specific app, the supported functions are determined by the app itself. For example, after entering Camera, the user is allowed to issue "selfie" to reverse the camera, "video" to switch the shooting mode, and "capture" to trigger the shutter button, etc. When a message arrives or a dialog pops up, the user can easily process them by issuing "reply", "mark as read", "cancel", etc. Besides the features specially supported by the current context, the user is allowed at any time to access universal system functionalities such as "home", "back", and "screenshot".

Please note that in this work, we cannot cover all the functionalities on smartphones in Lip-Interact because of their immense quantity, especially the built-in features of various apps. To evaluate the feasibility of Lip-Interact, we selected 44 popular commands (including those for two representative apps) to collect data, implement system and conduct user studies. Such process and implementation can be easily generalized to other command sets. We will describe the details of the commands and their elicitation process in the next section.

**User Learning and System Feedback**

With Lip-Interact on a smartphone, the user should be clearly informed about commands that are currently available. When a feature is exposed to a user for the first time, a floating window is displayed around the icon with its command words on it (Figure 3.a). When the system detects that the user is issuing commands, a real-time visual indicator appears at the top of the screen (Figure 3.b). Recognition results will be displayed at the indicator's location for two seconds after the user finishes issuing a command. If the result is incorrect, the user is allowed to tap it and assign it to the correct class to improve the recognition model in subsequent use.
**Figure 3.** (a) For novice to Lip-Interact, red prompt windows are displayed to guide users to use the technique. (b) When a user is issuing command with Lip-Interact, a blue circle appears at the top of the screen as real-time visual feedback. This figure also shows part of the GUIs of the two selected applications in our evaluation: (a) WeChat, (b) Notepad.

**SYSTEM IMPLEMENTATION**

To implement the Lip-Interact prototype, we equipped a smartphone (Huawei P9 Plus with a 5.5 inch screen running Android 7.0) with a 720p/30-fps camera at the top (see Figure 5.a). The camera was connected to a computer which ran the image processing and neural network classifier. The computer and the smartphone communicated over local network to share interface information and send commands in real time. Although introducing some latency, this setup afforded us a flexible framework for faster prototyping. However, with full access to a cutting-edge device’s camera and neural processing unit, it seems likely that Lip-Interact could be realized on a real product with no additional computing resources.

**Command Set**

We used Standard Chinese as the experiment language to facilitate data collection and user evaluation locally. The command set was determined through a user-elicitation study with 15 participants. Each participant was asked to list his/her most frequently-used mobile apps, system settings, and actions to handle pop-up messages and dialogs. For each category, we calculated the average ranking of every item across participants and selected the top results into the final command set. We also chose two representative apps for evaluation: WeChat\(^1\) and Notepad. WeChat is a multi-purpose social media mobile app and the largest app by active users in China. WeChat has a wide range of functionalities including messaging, moments, audio, payment services, etc. To access frequently-used features, users often need to tap and swipe multiple times or choose from a hierarchical menu (3.a). We selected Notepad as the second app because text editing is a common mobile activity, and we hoped to learn how Lip-Interact can assist touch and improve the experience on some tasks where touch could be limited (e.g. undo/redo, adjust cursor position). We developed the Notepad app (3.b) by ourselves which supported basic text editing functionalities. For both apps, we elicited the most frequently-used functionalities as above. Figure 4 shows the final command set in Chinese and the corresponding English translation.

**Segmenting Mouth Sequence of Silent Speaking**

The frames from the front camera are processed with the DLib face detector [29]. In each frame, the method of Kazemi and Sullivan [27] is used to identify the facial landmarks, 20 of which describe the feature points of the mouth (Figure 5.b). The Opening Degree (OD) of the mouth is measured by the distance $h$ between the upper and lower lips divided by the distance $w$ between the left and the right corners of the mouth.

To segment the command issuing sequence, an online sliding window algorithm [28] is used to detect and transfer between the following four states: begin speaking, continue speaking, stop speaking and other (Figure 5.c). Within each sliding window, seven statistical features of OD are calculated: current value, sum, mean, max, standard deviation, number of frames between max and min, and absolute energy. If and only if all of these features satisfy certain thresholds (learned from training data with a C4.5 decision tree [41]), the algorithm decides whether or not to remain in the current state or move to the next state. For per frame between begin speaking and stop speaking, an affine transformation is applied based on the lip landmarks to extract a mouth crop image of size $H \times W = 100 \times 80$ pixels (Figure 5.b). The RGB channels of the mouth images over the speaking interval $T$ are normalized to have zero mean and unit variance, forming a $T \times H \times W \times 3$ input for the next step of command recognition.

**End-to-End Deep-Learning Command Recognition**

Our recognition method built on the latest research [4, 12] on lip reading in computer vision and was improved to adapt to

---

\(^1\)https://en.wikipedia.org/wiki/WeChat
Lip-Interact in mobile interactions. Figure 6 illustrates our deep-learning architecture, combining the steps of spatial transformation, representation learning and dynamic modelling into a single end-to-end model.

Spatial Transformer

The architecture starts with a Spatial Transformer network [24] on each input frame whose dimension is $H \times W \times 3$. The Spatial Transformer (ST) is a learnable module which has the ability to spatially transform feature maps, allowing the model to learn invariance to rotation. Unlike conventional lip-reading tasks [47, 4, 12] where the mouth position is relatively fixed, for Lip-Interact in smartphone use, the orientation between the user and the camera may vary each time a command is issued, and may even change while the user issues a single command. Our spatial transformer module uses bilinear sampling and a 16-point thin plate spline transformation (TPS) with a regular grid of control points as the transformation function to solve relative mouth positioning toward the camera. The localization network inside the module is a two-layer CNN (32 and 64 filters respectively) + one fully connected layer (512 units).

Representation Learning and Dynamic Modelling

The output frames of each Spatial Transformer module are concatenated along the time dimension, forming a $T \times H \times W \times 3$ input for the following $3\times(3$D Convolutions, 3D max-pooling, spatial dropout) layers. 3D convolutional networks are more suitable for spatiotemporal feature learning than 2D convolutional networks [44, 25]. The features extracted by the convolutional nets are then flattened and passed into two bidirectional Gated Recurrent Units (GRU) [10] for the modelling of dynamics. Finally, the output at the last time-step of the second GRU is followed by a fully connected layer with a softmax function to predict the lip-command class.

Implementation and Training Details

Table 1 summarizes the hyperparameters of the recognition model. The number of frames $T$ is set to 70 ($\approx 2.33$s). Longer image sequences are truncated and shorter sequences are padded with zeros at the end. We use the ReLU activation function and batch-normalize [23] the outputs of each convolution layer. Dropout [43] of 0.5 is applied after each max-pooling layer. The models are trained end-to-end by the Adam optimizer [30] with the multinomial cross entropy loss and a batch size of 32. The learning rate is initially set to $10^{-3}$ and scheduled to decrease to $10^{-5}$ over time.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Size / Stride / Pad</th>
<th>Input dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-TPS</td>
<td>4x4 grid TPS</td>
<td>70x80x100x3</td>
</tr>
<tr>
<td>Conv3D</td>
<td>3x7x7/1,3,3/1,3</td>
<td>70x80x100x3</td>
</tr>
<tr>
<td>Pooling</td>
<td>1x2x2/1,2,2</td>
<td>70x27x34x32</td>
</tr>
<tr>
<td>Conv3D</td>
<td>3x5x5/1,1,1,1,2</td>
<td>70x13x17x32</td>
</tr>
<tr>
<td>Pooling</td>
<td>1x2x2/1,2,2</td>
<td>70x13x17x64</td>
</tr>
<tr>
<td>Conv3D</td>
<td>3x3x3/1,1,1,1,1</td>
<td>70x6x8x64</td>
</tr>
<tr>
<td>Pooling</td>
<td>1x2x2/1,2,2</td>
<td>70x6x8x96</td>
</tr>
<tr>
<td>Bi-GRU</td>
<td>96</td>
<td>70x1152</td>
</tr>
<tr>
<td>Bi-GRU</td>
<td>96</td>
<td>70x192</td>
</tr>
<tr>
<td>FC-Softmax</td>
<td>N</td>
<td>192</td>
</tr>
</tbody>
</table>

Table 1. Model hyperparameters. $N$ is the size of the command set.

Post- Vocal Check

After the recognizer returns the result, a vocal check is applied to distinguish Lip-Interact from users’ normal speaking. For example, talking to other people while using the device will also result in mouth movement but should not be detected as issuing a Lip-Interact command. As a counter-example, Figure 7 shows the analog audio signal from the device’s microphone when a user speaks "open WeChat" in a dining room.

We first apply a band-pass filter on the audio signal to remove part of the background noise, and then calculate the peaks (syllables) of the signal. If the number of the syllables does match the command words of the recognition result, it is decided that the user is not using Lip-Interact.

Device Control and Feedback Implementation

Our implementation on the smartphone uses Android Accessibility Service². The service runs in the background and retrieves window content in real-time. It provides a layout

²https://developer.android.com/training/accessibility/service.html
We recruited 22 participants (12M/10F, P1-P22), aged between 20 and 28 years \((M = 24.5)\). All were full-time undergraduate and graduate students on campus. Participants rated their level of previous experience with speech interaction at 2.36 \((SD = 0.79)\) on a 1 to 5 scale (no experience to very experienced).

**Participants**

We recruited 22 participants (12M/10F, P1-P22), aged between 20 and 28 years \((M = 24.5)\). All were full-time undergraduate and graduate students on campus. Participants rated their level of previous experience with speech interaction at 2.36 \((SD = 0.79)\) on a 1 to 5 scale (no experience to very experienced).

**Experimental Setup and Design**

We conducted the experiment in a laboratory space with participants sitting or standing comfortably (Figure 5.a) near a window. We recorded participants’ mouth image sequences of issuing silent speech commands using the system described above. Each participant completed 44 blocks. Each block corresponded to one of the 44 commands listed in Figure 4. During the whole data collection process, participants experienced varied lighting conditions: bright and cloudy daylight, natural indoor light and lamplight in the evening, etc. We were trying to make the data diverse and the result representative.

**Procedure**

Participants were first given a short introduction to our task. At the beginning of each block, the participant tapped the START button on the device’s screen to start the task. Then the participant repeated the command 28 times: the first three times in normal audible speaking manner and the last 25 times in the Lip-Interact manner. A circular view on the device’s screen showed the progress: red indicated that the participant was issuing a lip command with significant mouth movements, while green indicated the other states. During the experiment, the participant was asked to hold the phone as he or she normally would in everyday use. To increase the data diversity, the participant was also instructed to randomly change the orientation between the face and the device within a certain range each time before issuing a command. A one-minute break was scheduled between blocks. The whole experiment took around 90 minutes.

**Results and Analysis**

**Quantifying the Exaggeration of Lip Movements**

We quantified the exaggeration of lip movements by comparing the Maximum Mouth Opening Degree \((MMOD)\) of the "exaggerated" silent speech vs. normal audible speech when the participants were issuing the same commands. On average, \(MMOD\) increased by only 7% when participants were issuing silent speech commands. All participants understood the "exaggerating lip movements" requirement after we explained the reason and felt comfortable throughout the experiment.

**Data Augmentation**

Before feeding the data to our model, we augmented the dataset by applying a horizontally mirrored transformation on the mouth crop images. In total, there were 48,335 command samples, with at least 1,092 samples for a single command.

**Recognition Group**

For training and evaluation, we divided the 44 commands into four groups based on the principle of \([system\ functionality + current\ context]\). The four groups were \(SH\) (20), \(SW\) (19), \(SN\) (19), and \(SP\) (16), where \(S\) denotes system’s global functions, \(H\) denotes homescreen app-launching, \(W/N\) denotes WeChat / Notepad’s built-in features and \(P\) denotes actions to respond to pop-up messages and dialogs (see Figure 4). The numbers in parentheses indicate the number of commands in the group. We performed model training and evaluation on each group.

**Leave-One-Subject-Out Cross-Validation**

Of the 22 participants, the samples of one participant were retained as the validation data for evaluating the model, and the samples of the other 21 participants were used as the training data. The cross-validation process was repeated 22 times, and the accuracies were averaged to produce the final result. This experiment is a predictor of the real-world performance when a user uses Lip-Interact for the first time.

Table 2 summarizes the overall classification accuracies of the four groups. On average, the accuracy of leave-one-subject-out was 95.464%, indicating that our system could accurately identify even a new user’s commands. Looking into the \(\sim 4.54\%\) misclassified samples, we found that these errors were not evenly distributed in different classes. Instead, the common

<table>
<thead>
<tr>
<th>Group</th>
<th>(SH)</th>
<th>(SW)</th>
<th>(SN)</th>
<th>(SP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>96.18%</td>
<td>95.47%</td>
<td>94.21%</td>
<td>95.99%</td>
</tr>
</tbody>
</table>

Table 2. Classification accuracies of four command groups with leave-one-subject cross-validation.
situation for a participant was that the recognition of a command was either almost entirely correct or almost entirely wrong. For example, the classification accuracy of SW of P4 was 96.625%. Among the 19 commands, 18 could be accurately recognized. But for the command “Like”, 25 of the 47 samples were misclassified into “Bluetooth”.

Progress over Time
This section evaluates recognition accuracy across time to simulate real-world scenarios when the pre-trained model improves as the number of uses increases. For each participant, we denoted the pre-trained leave-her/him-out model as $M_0$ and all of her/his samples as $S_0$. For each time $t$, a sample was randomly selected from each class in $S_t$. The parameters of $M_t$ were updated with the selected samples, forming the new model $M_{t+1}$, which was then evaluated by the remaining samples $S_{t+1}$ of the participant. The models were updated with the batch size of $2 \times$ the number of classes and with the epoch of 30. The other settings were the same as those in the implementation section.

Figure 8 shows the results. The classification accuracies of all four groups improved significantly as the number of updates increased, reaching 99.22%, 99.21%, 98.37% and 99.11% respectively at the 6th update. The accuracies increased the most at the 1st update with an average improvement of 1.74%. For P4, the classification accuracy of SW raised to 98.78% after the 1st update and finally reached 99.40%. The results showed that Lip-Interact could quickly learn and personalized the model as users made more use of it.

Summary
The above results verify the feasibility of Lip-Interact from two aspects. (1) On a limited set of mobile interactions, the commands issued by users through silent speech can be distinguished accurately by vision-only methods. Our deep learning model has sufficient generalization ability for new users with a recognition accuracy of over 95%. (2) The model is expected to have a strong learning ability in real-world scenarios, since smartphones are usually used by only a single user.

STUDY 2: EVALUATE LIP-INTERACT VS. TOUCH
Current smartphones use buttons, icons and menus to trigger functionality because they are easy to perceive and use. This experiment examines the interaction performance of Lip-Interact on a smartphone, with a focus on efficiency in comparison with general touch. Research questions include:

1. What is the advantage of Lip-Interact compared with touch interaction?
2. What is the behavior of users using Lip-Interact, and how does it change over time?

Participants
We recruited 10 right-handed participants (P1-P10, 5M/5F), aged between 22 and 30. None had participated in the previous data-collection experiment.

Experimental Design
The experiment was conducted in a laboratory space with participants sitting comfortably. Each participant was asked to complete a series of interaction tasks on the smartphone under three input conditions: Lip-Interact, touch operation with two hands, and touch operation with one hand. Figure 9 shows the whole interaction task. The task simulated daily smartphone usage, including 22 of the 44 functionalities supported by our Lip-Interact system. The order of the three conditions were counter-balanced across participants.

Task and Procedure
Each participant was first exposed to a practice session. The participant was given two minutes to get familiar with the GUI and touch interactions on the device. Next, the participant was instructed to traverse all the functionality of Lip-Interact. The prompt floating window was displayed for learning. For each function the participant met, he or she was asked to issue the lip command once. This data was used later to update the pre-trained models for personalization. Finally, the participant was asked to remember the task contents.
The participant was then exposed to the experimental session. For each input condition, the participant consecutively completed three times (blocks) of the interaction task. In each block, the experimenter advanced the task by controlling a sounder. The “Di” sound from the experimenter was used as the starting signal for the current subtask. Upon hearing it, the participant was asked to complete the subtask as quickly as possible and then wait for the next sound. In the experimental session, the prompt window was not displayed.

**Results and Analysis**

A total of 735 valid instances were collected. For the segmentation of the silent speech sequence, the recall was 99.18%, and the precision was 99.59%. For the command recognition, the mean accuracy was 97.9%.

**Measures**

We measured input time in seconds to quantify the efficiency. For each functionality access, input time was defined as the time interval from the starting signal (or the end of the last operation) to the time when the device began to respond to the user input. For analysis, we divided the tasks into six types based on their touch interaction properties. These types were:

- **Type I**: Accessing between multiple pages at the same level of the hierarchy (e.g., on the home screen, scroll left and right to locate and then tap to launch an app).
- **Type II**: Accessing between pages across different levels of the hierarchy. This often happens within an app (e.g., from the current page, perform several “back” actions, then navigate to enter a new page, and finally access the desired functionality).
- **Type III**: Opening a menu and selecting a command.
- **Type IV**: Accessing a functionality on the Quick Settings Drop-down panel (e.g., flashlight).
- **Type V**: Single tap within the comfortable finger range (e.g., tap the home button).
- **Type VI**: Single tap out of the comfortable finger range (e.g., tap the back button in the upper left corner).
- **Type VII**: Precise pointing which is difficult for touch because of the “fat finger” problem [42] (e.g., adjust a cursor in text).

The definition of each input type was determined based on the participant’s actual operation, which was recorded on video and labelled manually later.

**Input Time**

We collected a total of 90 blocks of task (10 participants × 3 input conditions × 3 replications). For each task type, we ran a one-way repeated-measures ANOVA for input condition.

Table 3 shows the mean input time of accessing the seven types of functionality under three input conditions. For each interaction type, the input time was significantly affected by input condition (Type I: \( F_{1,29} = 4.9, p < .05 \); Type II: \( F_{1,29} = 41.9, p < .0001 \); Type III: \( F_{1,29} = 85.6, p < .0001 \); Type IV: \( F_{1,29} = 30.3, p < .0001 \); Type V: \( F_{1,29} = 9.34, p < .01 \); Type VI: \( F_{1,29} = 13.7, p < .001 \)). On average, participants spent less time with Lip-Interact on all interaction types except for Type V. Lip-Interact increased input speed the most for Type II, Type III and Type IV, with an improvement of 26.42% ~ 54.05% compared to touch with two hands and 46.81% ~ 59.25% compared to touch with one hand. These tasks generally require multiple steps of touch operations to complete.

Lip-Interact also had the smallest standard deviation of input time on all interaction types, showing that Lip-Interact provided a more stable and dependable way to access functionality. The efficiency of common touch interaction was strongly influenced by the target location and user familiarity.

**Input Easiness**

Table 4 lists participants’ quantitative rating of overall input easiness for the three input conditions, from strongly disagree 1 to strongly agree 5. An analysis using Friedman test showed that for our tasks participants reported significantly stronger agreement for Lip-Interact than the two touch conditions on input easiness (\( p = .00015, \chi^2 = 17.64 \)).

**User Behavior and Feedback with Lip-Interact**

All participants expressed that Lip-Interact was easy to understand and learn. P3 commented that “I could remember all the commands right after the practice session. These commands are concise and intuitive.” Six participants expressed surprise at the accuracy of the recognition. P9 said “I had never thought these things could be identified. When I first silently spoke the command and saw the correct response, It felt really good.”

In particular, participants enjoyed input with two channels: touch + Lip-Interact together in the Notepad app. P2 said “It felt like reading incantations. I did not have to select the command from the menu every time so that I could concentrate on the text itself”. Two of the participants (P2, P4) reported that they took advantage of the parallelism of the two channels to speed up interaction pairs. P4 commented: “For the task of selecting a word and then highlight/bold it, I initially performed the two steps in sequence in the first block. But since using Lip-Interact always took some time, in the next blocks, I tried to start issuing the command before I finished the selection. It is faster.” We calculated the whole input time from the beginning of the word selection to the end of marking it. The cases using Lip-Interact in parallel (2.37s) were 22.8% faster than using Lip-Interact sequentially (3.06s).

The definition of each input type was determined based on the participant’s actual operation, which was recorded on video and labelled manually later.

**Table 3. Mean input time (SD) in seconds of different types of tasks in three input conditions in Study 2.**

<table>
<thead>
<tr>
<th>Type</th>
<th>Lip-Interact</th>
<th>Touch (two hands)</th>
<th>Touch (one hand)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1.757 (0.178)</td>
<td>1.845 (0.423)</td>
<td>2.216 (0.748)</td>
</tr>
<tr>
<td>II</td>
<td>1.782 (0.382)</td>
<td>3.265 (0.586)</td>
<td>3.837 (1.270)</td>
</tr>
<tr>
<td>III</td>
<td>1.749 (0.172)</td>
<td>3.277 (0.507)</td>
<td>3.288 (0.426)</td>
</tr>
<tr>
<td>IV</td>
<td>1.541 (0.160)</td>
<td>3.353 (1.195)</td>
<td>3.782 (1.242)</td>
</tr>
<tr>
<td>V</td>
<td>1.387 (0.098)</td>
<td>0.934 (0.268)</td>
<td>1.318 (0.442)</td>
</tr>
<tr>
<td>VI</td>
<td>1.366 (0.197)</td>
<td>1.416 (0.410)</td>
<td>2.066 (0.643)</td>
</tr>
<tr>
<td>VII</td>
<td>1.557 (0.078)</td>
<td>1.866 (1.186)</td>
<td>2.207 (1.130)</td>
</tr>
</tbody>
</table>

**Table 4. Interaction Easiness in three input conditions in Study 2.**

<table>
<thead>
<tr>
<th>Easiness (SD)</th>
<th>Lip-Interact</th>
<th>Touch (two hands)</th>
<th>Touch (one hand)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 (0.44)</td>
<td>4.3 (0.46)</td>
<td>2.8 (0.40)</td>
<td></td>
</tr>
</tbody>
</table>
With each technique, we chose the subway as our experimental environment. When the ten participants were divided into five groups. Each group simulated a stranger nearby. Participants issued a total of 367 Lip-Interact commands. For the segmentation of the silent speaking sequence, the recall was 97.8%, and the precision was 97.3%. For command recognition, the mean accuracy was 97.5%.

An analysis using the Friedman test showed that participants reported significantly higher levels of satisfaction for Lip-Interact than for voice input on all three measures ($p = .00157, \chi^2_r = 10$). The results are listed in Table 5.

For Privacy, P4 commented that "If there are other people around me, there is no privacy when using the voice interaction. People will know what I am doing on my device. Using the silent speech can largely protect my interaction privacy, unless others are looking at me very carefully."

For Comfort as user, participants expressed two concerns about voice control. The first concern was that "making sound may disturb people around me" (P6). The second concern was self-embarrassment. P2 said "Every time I spoke out a command, I felt like someone else was watching me. It was strange and sometimes awkward." With Lip-Interact, however, "it was much less noticeable".

As for Comfort as people in the surrounding environment, most participants rated 3 (neutral) on voice input. P5 commented "The feeling of being a passerby was related to the current environment. It did not matter in most cases, but sometimes I could not help but wonder what he or she was doing ". But for Lip-Interact, "I rarely detected a noticeable difference."

MORE EXAMPLE APPLICATIONS

Finally, we encouraged participants to suggest more applications for Lip-Interact. We list several representative examples.

Camera

After "open camera", the use can quickly switch shooting mode and settings by issuing "photo", "video", "panorama", "selfie", "reverse camera", "flash on/off" lip commands instead of recalling where the buttons are. When the user has posed for a selfie with one hand holding the device, he or she can simply mouth "capture" rather than awkwardly adjusting the hand and reaching it with the thumb.

Video Player

When watching videos on a smartphone, the user’s hands are not always available. For example, holding the device with one hand while having meals, or washing the phone that is fixed on a stand while lying down. In these cases, Lip-Interact should allow users to operate the interface by issuing "play", "stop", "fast-forward", "rewind", "louder" and "softer" commands.

Shared Bike

This example application comes from a real-life experience of one of our participants. "Days ago, I came out of the lab and..."
and intended to rent a smart shared bike. I was wearing gloves. I had to take off my gloves, scroll several times to locate the app, then tap the unlock button to scan the QR code, and finally put on my gloves again.” Lip-Interact can greatly reduce such temporary disability with two simple commands: "open Mobike" and "unlock".

**Smart Watch and Head-worn Device**

This work focuses on the implementation and evaluation of Lip-Interact for smartphone, but we also foresee the potential of Lip-Interact on other platforms. Smartwatch has an ultra-small screen on which UI layouts and touch interaction are more limited. Lip-Interact is expected to allow hands-free interaction on smartwatches with a built-in camera sensor on the front panel. Head-worn devices (AR/VR glasses) separate the space of displaying and the interaction interface. For current mainstream solutions, mid-air gestures suffer from limited size of the command set and arm fatigue, while voice input has the inherent issue about privacy and social norms. We envision that a tiny camera is integrated into the microphone, capturing users’ mouth movement and enabling Lip-Interact.

**DISCUSSION AND LIMITATIONS**

We have described Lip-Interact, a novel interaction technique that repurposes the front camera of smartphones to detect lip movement and recognize it into commands. Lip-Interact provides an alternative input method on smartphones with both simplicity and expressivity. We verify the feasibility of Lip-Interact by implementing a real-working prototype and conducting three experiments. We demonstrate that the model of a [spatial transformer network + CNN + RNN] architecture can accurately recognize around 20 silent speech commands with training data collected from dozens of users. We incorporate voice check and design visual feedback to offer an accurate, robust and friendly user experience. We also research on-line personalization, which confirms the potential and benefit of adapting the machine learning model to end users.

Although in our study, Lip-Interact outperformed touch-based input on most of the investigated tasks with a shorter completion time and higher input easiness, our purpose is not to substitute touch with Lip-Interact but to provide an alternative input approach. This is because touch-based input is still faster and more direct for many tasks (e.g. scrolling a list) and enables more complex interactions (e.g. typing). The use of Lip-Interact depends on the current task as well as users’ cognitive and motor skills. The advantages of Lip-Interact can be summarized as: (1) allowing for access to functionality efficiently in one step, (2) providing necessary interaction ability when hands are not available, and (3) assisting touch to make device use more efficient and fluent. In addition, Lip-Interact also has potential to be applied to other interaction platforms (e.g. accessibility, smartwatch, head-worn devices), which deserves to be further studied.

Lastly, we discuss the main limitations of our work, which also points to the directions of future work.

**Improve Recognition Accuracy of Silent Speech Commands**

In order to raise the recognition accuracy of silent speech to a practically usable level in a mobile input system, we add two constraints: (1) limited command set based on current context and (2) properly exaggerating the lip movements in the interaction designs as well as (3) incorporate Spatial Transformer Networks in the recognition model. The performance outperformed prior research which handled even smaller command sets [26, 12]. Although the size of the command set is still limited in our work, Lip-Interact can be easy to be generalized to support other applications. Our empirical experience indicates that as long as the mouth image sequence of the commands can be distinguished by human eyes, the model can accurately recognize them. Future work should be done to try Lip-Interact on more applications. For the "exaggeration" requirement, our participants understood the necessity, and did not report any discomfort during the studies. At the same time, we admit that this is an interaction limitation on users.

**System Robustness**

Some commercial smartphones like the iPhone X are already equipped with special hardwares to ensure the robustness of face detection. Therefore, the robustness of the Lip-Interact system mainly relies on two factors: the robustness of recognizing the user’s intention to use silent speech and the robustness of recognizing the issued commands.

For the intention recognition, at present we solve it by detecting the changes of the mouth opening degree and the audio signal. It worked well under our experimental conditions. But we do not rule out that the performance will be affected by more noise in real-life use cases. The algorithms need to be improved and tested in more various environments. To address this, we can also consider some explicit mode-switch methods, such as pressing a button or squeezing the phone’s edge. Similar methods are often used in HCI research and commercial products. For command recognition, the bottleneck lies in the expressivity of silent speech - the information conveyed by the mouth movements. Our research shows high accuracy for recognizing about 20 commands. Future work should be done to improve the algorithm and explore the theoretical upper bound.

**Other Limitations**

Our implementation uses an additional mounted camera and host server for computing. We are trying to integrate Lip-Interact into a self-contained smartphone, which will reduce the response time and improve the user experience further. Our experiment results reflect the performance of Lip-Interact in a controlled setting. We compared Lip-Interact with touch input on only one device. Different device sizes and UI designs can lead to different performance. This is also why we classified each input into different types and analyzed respectively. Besides, we have only tested the Chinese language. But we think Lip-Interact can be easily applied to other languages as long as the mouth movements of commands are differentiable.

**ACKNOWLEDGEMENT**

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61672314 and No. 61572276, Tsinghua University Research Funding No. 20151080408, and also by Beijing Key Lab of Networked Multimedia.
REFERENCES


