2021

# Detecting Surface Interactions via a Wearable Microphone to Improve Augmented Reality Text Entry

R. Habibi
*Michigan Technological University*, rhabibi@mtu.edu

# DETECTING SURFACE INTERACTIONS VIA A WEARABLE MICROPHONE TO IMPROVE AUGMENTED REALITY TEXT ENTRY

By

Reza Habibi

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Computer Science

MICHIGAN TECHNOLOGICAL UNIVERSITY

2021

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Computer Science.

Department of Computer Science

| | |
|---|---|
| Thesis Co-advisor: | *Dr. Keith Vertanen* |
| Thesis Co-advisor: | *Dr. Scott Kuhl* |
| Committee Member: | *Dr. Guy Hembroff* |
| Department Chair: | *Dr. Linda Ott* |

# Dedication

## To my parents

I dedicate this thesis to my parents. Without their patience, understanding, and most of all love, I would neither be who I am nor would this work be what it is today. I'm forever grateful to my parents.

# Contents

# List of Figures

# List of Tables

# Acknowledgments

I would like to express my most sincere gratitude to my advisors, Dr. Keith Vertanen and Dr. Scott Kuhl, for their support and inspiration through the development of this research. I thank them for their two years of support, mentorship, and the opportunity to work with them as a student and provide me with funding opportunities. From day one, they trusted me and helped me pursue research that I was passionate about, and provided me with all the resources to conduct this dissertation.

I am very fortunate to have them as my advisors.

I greatly appreciated Dr. Guy Hembrof for being on my committee and for his help and support throughout my Master's program.

I am also indebted to my family and friends who always rooted for me. Also, I greatly appreciated Dr. Adrienne Minerick and Dr. Eugene Levin for their help and support during my graduate studies.

# List of Abbreviations

| | |
|---|---|
| AR | Augmented Reality |
| VR | Virtual Reality |
| MR | Mixed Reality |
| CONV | Convolution |
| CNN | Convolutional Neural Network |
| HMD | Head-mounted display |
| WPM | Words Per Minute |
| SD | Standard Deviation |

# Abstract

Augmented Reality (AR) and Mixed Reality (MR) enable us to build a new generation of human-computer interfaces. In the future, AR Head Head-mounted display (HMD) might replace mobile phones devices, and people use HMDs to enter text while they are on the go on any surface. Despite advances in sensors, cameras, and recognition systems in AR HMDs, accurately detecting when a tap occurs is difficult. A finger can be detected in a mid-air text entry system via visible light camera data, infrared camera, and artificial intelligence algorithms. However, executing mid-air taps is difficult without tactile feedback and determine precisely when a tap occurred is challenging. This thesis investigates whether we can detect and distinguish between surface interaction events such as tapping or swiping using a wearable mic from a surface. Also, what are the advantages of new text entry methods such as tapping with two fingers simultaneously to enter capital letters and punctuation? For this purpose, we conducted a remote study to collect audio and video of three different ways people might interact with a surface. We also built a CNN classifier to detect taps. Our results show that we can detect and distinguish between surface interaction events such as tap or swipe via a wearable mic on the user's head.

# Chapter 1

# Introduction

Interaction with real environments and physical objects is a critical aspect of augmented reality (AR) and mixed reality (MR). The development of new sensors and cameras brings new capabilities to the augmented reality world. However, little has been done to improve text entry and surface interaction accuracy on AR keyboards. This project aims to study whether audio captured using an HMD's microphone can bring more accuracy and functionality to AR text entry. We trained a model to detect when a tap or swipe has occurred. Additionally, we evaluated our model's ability to recognize various types of taps and swipes.

## 1.1 Background

Today, technology is one of society's most significant achievements, and it is an integral part of modern life. These technologies open up new possibilities and have evolved into a necessary component of human interaction and communication. People use computers and mobile devices to interact with data such as text, audio, and video. Hence, we need interaction techniques and tools such as keyboard and gesture input to bridge the gap between human and machine communication. The mouse and keyboard are heavily utilized in traditional interaction systems for selecting and clicking buttons and targets. The widespread availability of touchscreens, primarily in hand-held devices such as mobile phones, has displaced the mouse and keyboard.

There is a significant development of new technologies in augmented reality, which rely on similar interaction fundamentals for most parts. The primary focus of recent research is on different text entry and surface interactions, such as vision and speech-based interaction, hand-held controllers, and wearable devices such as gloves and rings—however, those studies have inherent drawbacks in recognizing inputs and texts. For instance, an optical-based camera needs a substantial amount of processing power, high-resolution data, and its accuracy depends on other factors, such as ambient light and noise. Different approaches, such as wearing gloves, wristbands, or hand-controllers, occupy hands with an external device and prevent them from

using two hands freely. Furthermore, speech recognition systems are ineffective in environments with a high level of background noise. Additionally, they face privacy and social concerns.

In the future, AR HMDs might replace mobile devices, and people might use HMDs to enter text while they are on-the-go. Midair text entry is one of the solutions for future AR text entry systems. However, using a mid-air text entry suffers from several limitations. For instance, typing on a mid-air keyboard and executing mid-air taps without tactile feedback is difficult. Also, we need to detect the position of fingers and determine when a tap has occurred. A finger can be tracked using a visible light or an infrared camera. However, determining precisely when a tap occurred is one of the challenges.

Tapping on everyday surfaces is another solution for future AR text entry systems. Different surfaces such as walls or tables are abundant and may be more comfortable to use. Moreover, the sound of tapping or swiping may aid in determining when the tap or swipe has occurred. For example, it may assist the visible light camera by utilizing a sensor fusion technique and synchronizing both audio and video data acquired during an action such as tapping.

The purpose of this study was to investigate augmented reality text entry through the use of tap and swipe input on a surface. We looked at both tapping and gestures because both can be used for typing, games, and multimedia applications. Finally,

we evaluated our system in a remote study using a wearable mic on the user's head to simulate AR HMD sensors in conjunction with an everyday surface. To begin, we devised three distinct techniques for entering capital letters and punctuation. Second, we used the study's audio data to train a classifier and detect tapping and swiping on a surface.

## 1.2 Thesis Structure

This thesis consists of five chapters. Chapter 2 is about the related work and existing research in text entry and gesture input recognition in AR. Chapter 3 discusses the main research questions, experimental design, and components of the experiment. Chapter 4 elaborates on the experiments' results, and Chapter 5 is the in-depth analysis and discussion about current limitations and possible future works.

# Chapter 2

# Related work

Recent emerging novel technologies such as augmented reality (AR), and new devices such as advanced Head-Mounted displays (e.g., Microsoft HoloLens and Magic Leap headset) bring new opportunities to daily human life. Interaction with the surface and text entry are core applications in augmented reality applications such as AR tabletop in education [57], rehabilitation [52], and multimedia [35]. In this chapter, we review the recent literature on surface interaction in AR. Prior work falls into two main areas. In the first section, we discuss touch interaction on surfaces and review several past studies related to this work. In the second section, we review text entry and surface interaction in AR.

## 2.1 Touch input on surfaces

### 2.1.1 Touch input on instrumented surfaces

Interaction on instrumented and not instrumented surfaces is not a new concept. We define not instrumented surfaces as a surface which do not have any sensing capabilities by themselves. However, instrumented surfaces contain sensing capabilities by themselves. Touchscreens are one of the categories in instrumented surfaces, and they measure the capacitance between the display and a user's finger. This method has a long history in the Human-Computer Interaction field [30].

SmartSkin [42], ThemetaDESK [53], and DiamondTouch [11] are other examples of instrumented surfaces. They employed surface-integrated capacitive sensors to recognize human hands and fingers. This system calculates the 2D location of the hand and finger via a grid-shaped sensor. This system can sense multi-finger interaction and provide visual feedback on the surface via a projector but they did not implement any text entry. Moreover, these efforts were limited by some factors, such as user interaction calibration, portability, and the size and scale of touchscreen technology.

### 2.1.2 Interaction on not instrumented surfaces

Prior work has been done on the idea of using a surface and not instrumented characterized based on vision sensing, acoustic sensing, or a hybrid approach. The recent emergence of different cameras has led to the widespread use of other methods to detect touch on surfaces, including LIDAR, RGB cameras, and depth cameras.

#### 2.1.2.1 Surface interaction based on vision

Paradiso et al. [38] proposed four different systems to detect near-surface gesture interaction, including a low-cost laser-based approach to track the polar coordinates of the hand above a large plane surface. However, three-dimensional object scanning on a large surface requires sufficient resolution and a powerful, accurate laser scanner. RGB cameras (e.g., [1, 32, 59]) allow touch sensing by analyzing images and videos of objects coming from a camera. Sugita et al. [50] described a camera approach involving image processing and the color pattern of a finger when it touches a surface. The system uses image processing techniques and several computer vision algorithms to determine the difference between the fingertip color patterns on the surface. However, this system is not practical for supporting accurate touch and multi-gesture-based input.

Some other optical techniques [15, 33, 43, 58] use a threshold method to detect an object on a different surface and a camera pointed toward a surface. Moreover, they use projectors to provide visual feedback on the surface.

Depth cameras calculate the spatial accuracy of items on a surface by calculating the distance between each point in the camera's field of view. (e.g., PlayAnywhere [60], DIRECT [63], Paradiso [38], and Wilson [61]). Typically, objects are recognized using this approach by distinguishing between the object's depth data and the depth map model from the background. The background model can be created in a variety of ways, including by capturing multiple frames and averaging them to create a background profile [62], doing real-time 3D reconstruction [23] or creating a unique model for each finger without creating a depth map from a background such as Flexpad [49].

Despite all of these efforts, vision touch sensing suffers from various limitations such as occlusions, noise, and delay. Also, determining the contact moment between the finger and the surface is difficult.

#### 2.1.2.2 Surface interaction based on acoustics

There has been a significant amount of research done related to acoustic sensing. One of the most popular approaches is passive acoustic systems, which rely on the sound produced by touching or dragging a finger on top of the surface. Scratch input [16]

is one example of a passive acoustic system based on the unique high frequencies produced by dragging fingernails on different surfaces such as wood, fabric, or walls. The system classified different gestures based on their amplitude profiles captured by a stethoscope attached to a microphone. However, this method of sensing is limited to using a stethoscope attached to a surface.

RapTapBath [51] is a system that uses piezoelectric sensors to analyze sounds produced by a tap on the edge of a bathtub. A piezoelectric sensor detects and converts changes in pressure, force, and strain to an electrical charge. In their approach, the location of the tap is determined by calculating the difference in signal arrival times at the piezoelectric sensors. However, since the system has no way to determine when the tap occurred, it can only look at the relative differences between when each mic hears the sound. This approach is similar to the time difference of arrival (TDOA), which has been described as early as 1985 [2, 4, 5, 31]. This paper's method can also identify different types of taps, such as knuckle, pad, and tip. They also provided visual and audio feedback via a projector and speaker. However, this paper's approach is limited to the surface attachment of an array of sensors, and thus is not applicable to future augmented reality applications that require text entry via everyday surfaces.

Toffee [64] used a similar system for detecting taps on a surface. This system relies on piezoelectric sensors and the traditional TDOA method for detecting the location of the tap. This system added an array of piezoelectric sensors attached to the bottom

of devices such as smartphones and laptops to detect taps on the surfaces around the device. For instance, a cell phone may be placed on a table and the area around it used to detect taps. These efforts were limited to the space around the device, the number of sensors, and other objects on top of the surface. Expressive Touch [39] proposed tapping force as a new modality for interacting with devices such as mobile phones, tablets. They showed it was possible to measure force-sensitive tapping by studying the sound wave produced by tapping on the surface. The study used four contact microphones to detect the highest point of the amplitude value in the sound wave.

Another study [17] presented a similar approach to identifying the type of object being used for the input. This system uses a microphone connected to a stethoscope and a support vector machine to classify the acoustic signatures of different objects, such as different parts of the finger. This system achieved 95% accuracy for identifying four input types. Using wearable sensors is another area of exploration that has explored surface interaction. Tapskin [67] used the skin around a watch as a surface and used the watch microphone, gyroscope, and accelerometer to identify three gestures set on the skin with 97.32% accuracy. They showed that hitting a surface with different parts of the body causes a specific sound. For instance, the position close to the knuckle with more bone structure has more energy at lower frequencies.

Most of the mentioned studies in this section are limited to utilizing a surface with

an array of sensors. In our detection system, we used a wearable mic on the user's head to detect tapping and swiping without any sensors attached to surfaces.

## 2.2 Text entry and gesture-based input in AR and MR

MRTouch [65] is another example of vision-based touch interaction in MR HMDs. This system uses data coming from the Microsoft HoloLens v1 depth and an infrared camera to detect and track fingers and surface planes in real-time. MRTouch locates each known surface and touchpoint over it in a depth frame. This paper showed 95% button input accuracy with an average positional error of 5.4 mm. However, the Microsoft HoloLens depth camera suffers from very high latency. Moreover, this paper reported a high rate of missed touches of 3.5% and extra touches of 19% related to hover sensing.

ARKB [29] proposed a system consisting of a vision-based finger tracking attached to an HMD and an AR keyboard. Moreover, they provide audio feedback when fingers touch keys on the virtual keyboard. This system uses 3D position information obtained from fingers and a virtual keyboard. Moreover, it determines the collision between the finger point clouds and the keyboard plane in order to detect a keyboard tap. However, they did not conduct a text entry experiment.

Typing on Glasses [14] investigated the smartglasses touchpad for gesture-based input and text entry by introducing Swipeboard. Swipeboard is a smart eyewear text entry technique that uses two directional gestures to select a subgroup of keys and a specific key in the second step. This paper proposed a new technique called SwipeZone, which uses a touchpad on the side of a smartglasses for entering text and gesture-based input. A text entry study reported an 8.73 WPM entry rate, which is 15.2% faster than the default swipeboard in smart glasses. PalmType [56] used the palm as a base for the keyboard and Google Glass as a smart wearable to display the keyboard. Also, they used a wrist-worn sensor to enable typing without visual attention to the hand. This system mapped a QWERTY layout to the user's hand and showed this approach is 39% faster than the other touchpad-based QWERTY keyboards. However, this system is limited to the wrist-worn sensor which occupies hands with an external device.

BISHARE [68] studied the interaction between smartphones and AR HMDs. This paper relied on a framework for supporting both smartphones and HMDs. They introduced several design principles for interaction between a smartphone and an HMD. For example, a cross-platform interaction technique uses hand gestures and local touch on the phone as an input event platform or the ability to extend display space for 2D and 3D content. However, this paper does not provide a text entry experiment.

Reifinger et al. [41] describe infrared-based hand-gesture recognition for augmented reality applications. This system tracks the position and orientation of each finger based on the user's thumbs and index fingers. They showed that the proposed system reduces the average task duration time by a third compared to the mouse and keyboard. However, this system restricts the user by wearing hardware. VISAR [12] is another mid-air text entry approach for AR HMDs. This system provides an error-tolerant text prediction system that uses a statistical decoder. There is also a supportive mechanism to modify the auto-correction process. They used a Microsoft HoloLens v1 to provide a mid-air virtual keyboard and a hand-tracking system for tracking one hand. The study showed VISAR is 17.4% faster than Microsoft HoloLens default gaze-direct cursor system. The second experiment showed probabilistic auto-correcting text entry and literal text with reduced character error rates (CERs). Moreover, they reported a mean entry rate of 16.76 words-per-minute via their refined design, a 19.6% increase compared to the baseline.

MyoKey [28] utilizes surface Electromyography (sEMG) and a forearm wearable sensor that captures arm motion information. This system uses arm motion information to build an interactive system and identify five gestures in real-time. The system uses a 1-line horizontal text entry layout with 27 characters. They used American Sign Language as their gesture-based input. For instance, gesture 1 moves the cursor to the left and gesture 2 moves it to the right. They also used arm motion information to make a cursor work with a 1-line keyboard layout. This paper showed 91% accuracy

in the five gestures. However, they did not conduct a traditional full-size keyboard text-entry between different keyboard types.

# Chapter 3

# Research Methodology and Experimental Setup

## 3.1 Motivation

A surface typing system is composed of three primary components. We need to display a virtual keyboard and determine the time, type, and location of a surface interaction event such as a tap or swipe. This thesis aim was to focus on the time and type of action and detect tap and swipe on a surface using acoustic data captured via a wearable mic on the user's head. We also conducted a text entry experiment and explored three different input methods to enter capital letters and punctuation

which might be helpful for a future AR text entry system.

As we mentioned earlier (Chapter 2), it appears that current systems still have several challenges to resolve. Toffee [64] used an array of piezoelectric sensors attached to the device. However, the number of sensors, the difficulty of attaching sensors to the device, the lack of text entry experiments, and the difficulty of classifying various types of tap and gesture input are all drawbacks of this system. MRTouch [65] and VISAR [12] used Microsoft Hololens v1 for tracking hands. According to these studies, the Microsoft Hololens v1 depth camera suffers from very high latency and these systems are limited to the performance of the Microsoft Hololens v1 depth and infrared sensors. However, we need to detect the position of fingers and determine when a tap has occurred. A finger can be tracked via an HMD. However, it may be a challenge to determine precisely when a tap has occurred. For instance, users are adept at determining when they have come into contact with a surface, and their fingers may approach the surface without touching it.

All mentioned limitations demonstrate that we need a system to detect tap and swipe that does not need an array of piezoelectric sensors attached to the surfaces or devices. While it is a problem for Hololens v1 sensors to recognize when a tap has occurred, we intended to use the sound of a tap and swipe to recognize when a tap has occurred. We also ran a text entry experiment to see whether the way of entering capital letters and punctuation for an AR text entry system is the best. We investigated text entry

input using the shift key on the keyboard, tapping with two fingers at the same time, and swiping up with one finger to enter capital letters and punctuation. All of the factors mentioned above prompted us to pose the following questions.

1. **Question 1:** Can we detect surface interaction events such as tapping and swiping via a wearable mic on the user's head? Does acoustic data capture enough acoustic features to distinguish between tapping with two fingers at the same time, single tap, and swipe? Our primary hypothesis was that it might be possible to detect when a surface event occurs and it might be possible to differentiate between different types of surface events. However, it would be challenging.

2. **Question 2:** Is there a detectable difference between tapping and swiping on a sheet of paper versus off the sheet of paper? Exploring the impact of different surfaces is a potentially interesting topic and might be helpful for future AR text entry systems. We used a paper keyboard attached to the wall to simulate the experience of using a virtual keyboard displayed on future augmented reality glasses. Our primary hypothesis was that using data from a paper keyboard surrogate for what would probably be displayed by an AR headset would be close enough to detect surface event interactions on a surface without paper.

3. **Question 3:** Which way of entering capital letters and punctuation is more accurate and efficient? For instance, we compared the entry rate of a single tap

on the keyboard versus swipe up on the keyboard and the shift key to enter capital letters and punctuation. We hypothesize that swiping up and tapping with two fingers at the same time provides a faster alternative for entering capital letters and punctuation.

## 3.2 Setup

In an ideal world, we would use an augmented reality head-mounted display (HMD) to display a virtual keyboard on the wall in the lab environment. However, due to the COVID-19 pandemic, running in-person lab studies became difficult. So instead, we designed a study that could be completed remotely via a mobile phone simulating an AR HMD, utilizing a sheet of paper on the wall as a virtual keyboard. We developed an Android application written in Java to conduct this study remotely (Figure 3.6). Moreover, a Python-based web application for participant registration and distribution of all necessary files and materials for conducting our experiments.

We considered over 20 variables when developing the mobile application to ensure that data collection was consistent and accurate during the remote experiment. We considered factors such as audio and video resolution, as well as optimizing recorded files to ensure a manageable upload size. We used the mobile application to upload all files to our server. In our web application, we included a section for downloading

mobile applications (APKs) via a tiny URL or QR code, as well as keyboard printouts, an experiment checklist, and a reference sheet. Additionally, we used our web application to provide instructions for conducting our experiments and filling out our questionnaires.

We used the front-facing camera on the mobile phone to achieve a wide field of view, similar to that of AR HMD cameras, and the microphone to record audio from taps and gestures. Throughout the recording, participants held their phones in front of their forehead Figure (3.2). As a result, they were unable to view the phone's screen during the recording. We defined two gesture functions and controlled the recording procedure via the phone's touchscreen. They were instructed to swipe left or right to advance to the next sentence and up or down to delete and repeat the previously recorded sentence. Additionally, we provided online support in the form of a one-hour zoom session for each participant to assist them throughout the experiment—in addition to the interactive assistance provided by the app and website.

## 3.3 Experiment Procedure

1. Participants downloaded and installed a mobile application on their phones. We asked participants to print reference sheets and a keyboard layout.

2. Before starting the experiment, we described how to measure their distance to

**Figure 3.1:** A participant measuring their distance to the wall.

the wall (Figure 3.1). We asked them to use one straight arm as a measuring procedure to find their distance to the wall and then take one short step forward and align the bottom of the keyboard printout with their hand. The main reason behind this measurement was to align the keyboard layout location on the wall with the participant's height. Then we asked them to attach the keyboard printout and reference sheet to the wall.

3. Following that, we requested that participants log into the mobile application using their assigned ID. Throughout the experiment, we demonstrated how to use the application's gesture function to delete and repeat a sentence, as

well as navigate to the next sentences. To control the recording, we included two primary gesture functions. Swipe up and down to delete and repeat the sentence, respectively, and swipe left or right to go to the next sentence.

4. We described how to hold the phone. For instance, it should come into contact with the participant's forehead (Figure 3.2). Also, they should avoid covering the front face camera or bringing down the phone during the experiment. The application announced all necessary instructions via audio. For instance, it told the participants about finishing the conditions or the number of the sentence they were being asked to enter.

5. Prior to each condition, we explained how to practice it and the steps required to complete it. Additionally, we discussed how to upload videos following the conclusion of the experiment.

6. We asked them to complete a questionnaire on the experiment website after they completed the two experiments.

## 3.4 Experiment 1: Swipe-up, Dual-finger, Single

This experiment's main goal was to answer our third question about which way of entering capital letters and punctuation is more accurate and efficient? Additionally,

**Figure 3.2:** A participant entering text on a paper keyboard.

we used audio data acquired in this experiment to create our classifier in experiment 2.

### 3.4.1 Condition 1: Swipe-up

We focused on gesture interaction and selected swipe up as a common interaction primitive on touchscreens, which is easy to learn. Today, gesture interaction is one of the common interaction methods in user interfaces and text entry systems [46, 54]. Hence, keyboards in AR can take advantage of regular keyboard gesture functions such as SWIPE-UP (Figure 3.3) to enter capital letters and punctuation. Moreover, SWIPE-UP requires fewer keys than the shift key to enter capital letters and punctuation. From an acoustic and sensing perspective, gesture interaction such as swipe may have advantages, and it may have a distinctive acoustic signature compared to a single tap. In this condition, participants were instructed to type lowercase letters using

simple taps and to swipe up to enter capital letters and punctuation. (Figure 3.3). For example, in order to enter "G" and "!" in sentence "Good day!", participants must swipe up.

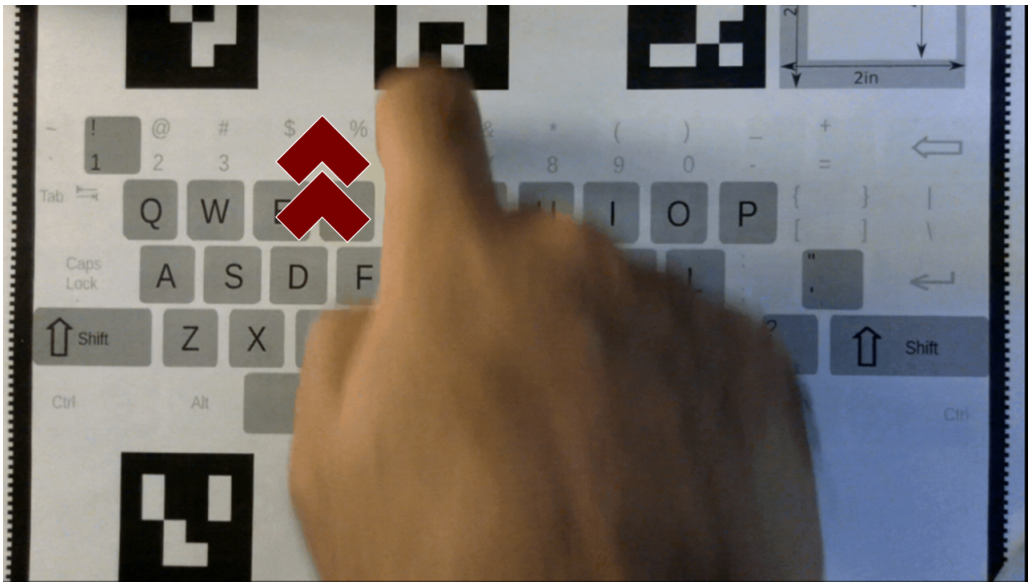



**Figure 3.3:** A participant swiping up to enter a capital letter.

### 3.4.2 Condition 2: Dual-finger

We defined DUAL-FINGER as a tapping (Figure 3.4) on the surface consisting of two fingers tapping at the same time and it was different from double-tap. Double-tap is a term that refers to two consecutive taps, similar to double-clicking a mouse. From an acoustic standpoint, it may be difficult for an audio classifier to detect double-tap because detecting rapid, consecutive taps might be hard and possibly confused with single taps.

As a result, we proposed DUAL-FINGER, which may be more detectable. It may be easier to detect the acoustic signature of a DUAL-FINGER rather than a double-tap. In this condition, we asked participants to type lowercase using simple taps on the keyboard layout and tap with two fingers at the same time to enter capital letters and punctuation. For instance, in order to enter "G" and "!" in sentence "Good day!", participants must tap with two fingers at the same time.

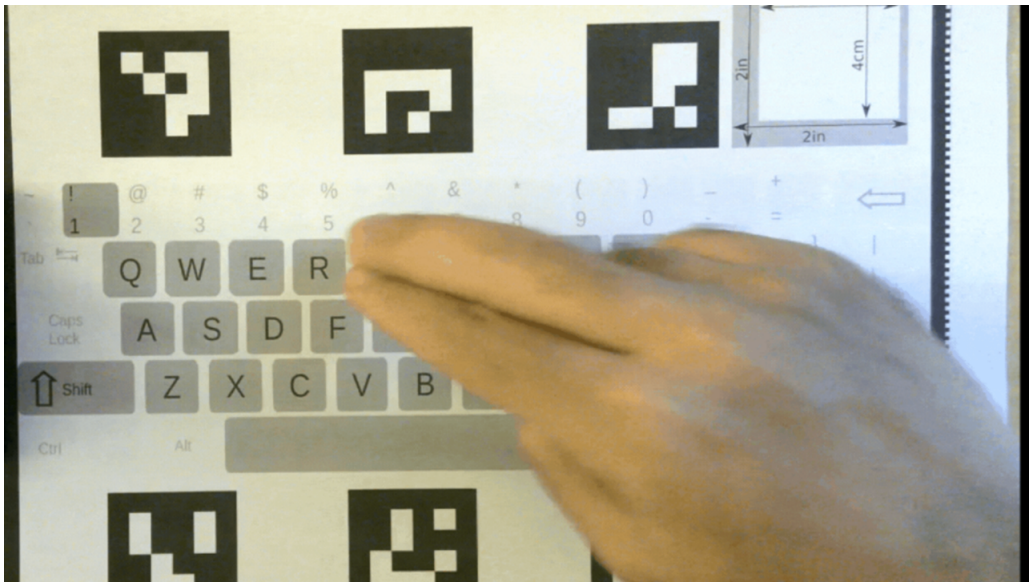



**Figure 3.4:** A participant using DUAL-FINGER to enter a capital letter.

### 3.4.3 Condition 3: Single

This condition is similar to a touchscreen keyboard where a user tap the shift key which causes the next key to be shifted. Most people are familiar with touchscreen keyboards these days because they employ a shift key to enter capital letters and

punctuation. In the single condition, participants entered capital letters by tapping on the shift key and then on the letter. For instance, in order to enter "G" and "!" in sentence "Good day!", participants required to tap on the shift key. This condition closely mimics entering text on a traditional keyboard. We call this condition SINGLE because of the single tap the user makes on the shift key. This condition's primary motivation was to compare standard text entry performance and single tap acoustic signature to the other mentioned conditions in this experiment (Figure 3.5).

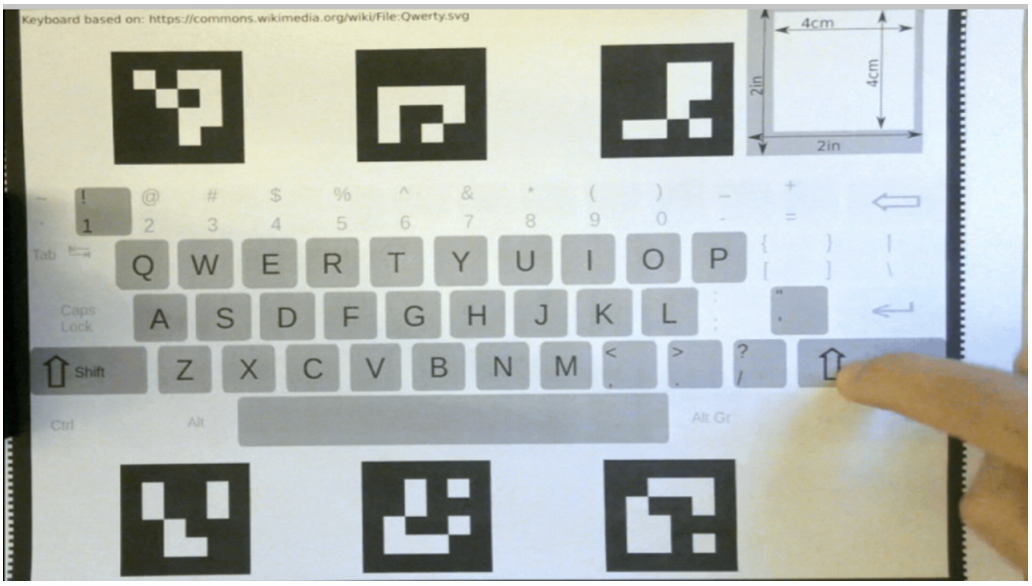



**Figure 3.5:** A participant using shift key to enter a capital letter.

## 3.5 Experiment 2: Off-paper/On-paper

Participants who participated in Experiment 1 (Section 3.4) completed experiment 2 and used the same keyboard printout attached (Figure 3.7) to the wall. The primary objective of this experiment was to address our first and second questions about the difference between tapping and swiping on and off a sheet of paper. In a future scenario involving augmented reality, the user interacts with a simulated keyboard projected onto the wall or another similar surface. As a result, it is critical to investigate various taps and gestures on various surfaces. Using a paper keyboard surrogate for what would probably be displayed by an AR headset would be close enough to tap on a surface without paper. This would allow us to collect lots of data using a phone and a sheet of paper rather than data from people using an actual AR headset.

In this experiment, we asked participants to perform the Swipe-up, Dual-finger, and Swipe-up conditions five times on and off the sheet of paper. In the first task, participants were instructed to tap five times in the center of the keyboard printout. In the second task, participants were instructed to tap five times on the wall above the keyboard printout. We asked them to swipe up five times in the middle of the keyboard printout for the third task and for the fourth one, repeat the swipe up but on the wall above the keyboard printout. We asked them to repeat the Dual-finger five times in the middle of the keyboard printout for the fifth task, and five times on

the wall above the paper printout for the final task.

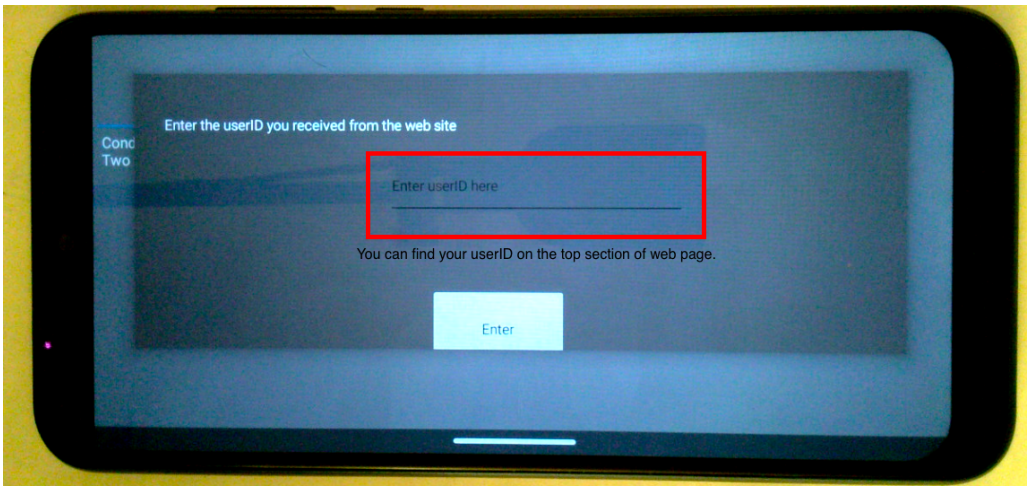



**Figure 3.6:** The login page of the experiment's mobile application. We assigned a unique ID to each participant.

### 3.5.1 Keyboard layout

We used a 104-key US keyboard layout [9] and customized it based on our conditions and experiment. We kept some keys and removed some parts of the keyboard (Figure 3.7) to provide a printable keyboard layout. For instance, we kept the letters, numbers, space bar, shift key, as well as punctuation such as comma, exclamation mark, question mark, and quotation marks. We replaced backspace with a function on our mobile application to delete the possible mistakes during the text entry experiment because we were not able to detect the location of keys in this experiment in real time.

There was no visual or audio feedback for the text entry experiment. Hence, we decided to remove backspace and replaced it with a gesture in our mobile application. Building a dataset that included audio and video data from the text entry experiment was one of the goals of this study. We added six fiducial markers to the top and bottom of the keyboard layout. We did not used these fiducial markers to detect finger location in this study but It will help future vision experiments find where the keyboard printout is and how it is oriented relative to the camera in the video feed. We provided a measuring box on the keyboard layout's top right corner to ensure the keyboard is the correct size. We asked participants to measure this box after printing the keyboard.

### 3.5.2 Reference Sheet Layout

For the text entry experiment, we provided a letter-size reference sheet (Figure 3.8). Participants read sentences from this reference sheet and typed them on the keyboard printout. We required sentences with more than three capitalized letters, so we used both the Twitter [55] and the Enron mobile [55] dataset. We selected 45 sentences from each dataset with capital letters at the beginning, middle, and end of sentences. Moreover, sentences with more than or fewer than three capital letters were excluded from the reference sheet dataset. Experiment 1 required participants to type six sentences in three different conditions. The order of these conditions was

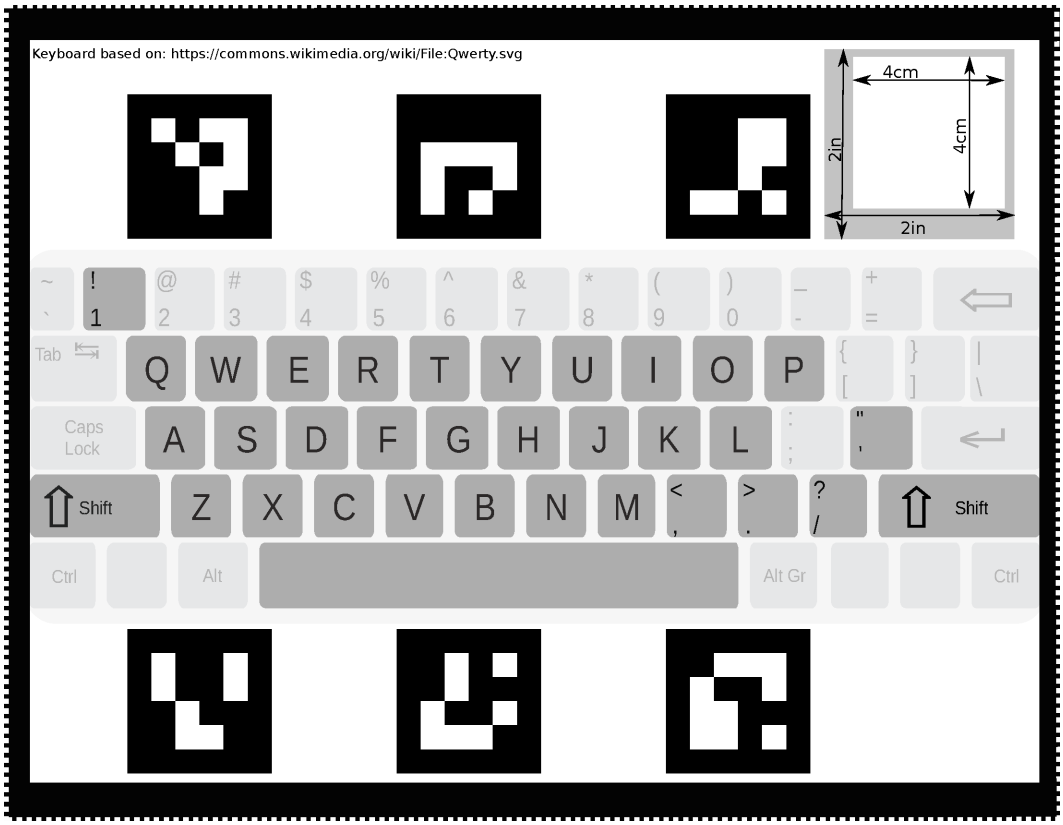



**Figure 3.7:** We asked participants to print paper keyboard and attach it to the wall before starting the experiment.

counterbalanced. The final section of the reference sheet is for Experiment 2, in which interaction events such as tapping and swiping on and off a sheet of paper.

## 3.6 Participants

We recruited 18 participants to run the application on their phones for this study. The participants' ages ranged from 18 to 50 (mean 24.5), 11 were female and seven were male. Four participants were left-handed, thirteen participants were right-handed,

| Sentence | Condition: Swipe-up - To enter capital letters and punctuation, swipe up starting from your desired key with your finger on the keyboard printout |
|---|---|
| 1 | Lynn, got to the office OK. |
| 2 | I miss Nikkon and Cannon. |
| 3 | I like Nick and Everett. |
| 4 | But I hope to see you at Kari's house. |
| 5 | Anything by new author Angela Marsons. |
| 6 | Hope everything is going OK at home. |

| Sentence | Condition: Dual-finger - To enter capital letters and punctuation, tap once with two-finger at the same time on the desire letter or punctuation |
|---|---|
| 1 | LOL you mean their nose? |
| 2 | Work, crossfit, YouTube, and sleep. |
| 3 | RIP to the Brazil Soccer team. |
| 4 | I saw Xabi Alonso in person today. |
| 5 | CarpComms have been at it again Boss. |
| 6 | Mark Ellenberg is off for Passover. |

| Sentence | Condition: Single - To enter capital letters and punctuation, tap on the shift key, then tap on the desire letter or punctuation |
|---|---|
| 1 | Please send to Gary Smith. |
| 2 | He also wasn't under FBI investigation. |
| 3 | I think that Pam Butler called him. |
| 4 | Yes we need to get it in ASAP. |
| 5 | Or at least DM me? |
| 6 | But I think they're all DVC. |

| Command | Condition: Off-paper/On-paper - Complete each of the following tasks |
|---|---|
| 1 | Tap 5 times with your index finger on the middle of the keyboard printout |
| 2 | Tap 5 times with your index finger on the wall, above the keyboard printout |
| 3 | Swipe up 5 times on the middle of the keyboard printout |
| 4 | Swipe up 5 times on the wall above the keyboard printout |
| 5 | Tap 5 times at the same time with two fingers on the middle of the keyboard printout |
| 6 | Tap 5 times at the same time with two fingers on the wall, above the keyboard printout |

**Figure 3.8:** Reference sheet layout printed by participants. The first three counterbalanced conditions were for Experiment 1. The last condition was for Experiment 2.

and one participant reported equal dominance. Seventeen participants reported they frequently enter text on a desktop keyboard, and nine participants look at the keys when typing on a desktop keyboard. Seventeen participants often use a mobile keyboard, and 12 participants look at the keys when typing on a mobile phone.

# Chapter 4

# Experiment Results

## 4.1 Experiment 1: Swipe-up, Dual-finger, Single

In this experiment, first, we calculated the words per minute (WPM) for each sentence entered by a participant and then averaged the results for each condition for each participant.

We calculated words per minute by dividing the total number of characters, including spaces, by five and multiplying by the time interval between the user's first and final taps on the keyboard printout. We viewed each recorded video and determined the time of the initial and final tap.

### 4.1.1 Text entry rate

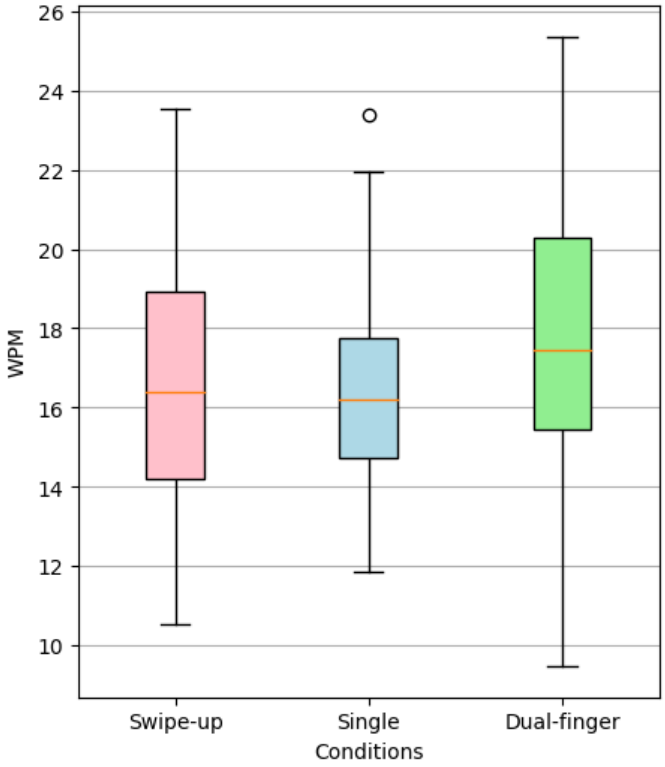



**Figure 4.1:** Words-per-minute (WPM) in Experiment 1.

We analyzed text entry performance and words-per-minute (WPM) for all three conditions. Figure 4.1 shows participants were slightly faster in Dual-finger (17.8 WPM) than Swipe-up (16.4 WPM). We also saw a lower words-per-minute entry rate in Single (16.3 WPM). However, as Table 4.1 shows, these differences were not statistically significant.

**Table 4.1**
Experiment 1 results for words-per-minute (WPM). The top section shows the overall average ± SD [min, max]. The bottom section shows one-way ANOVA.

| Condition | Words-per-minute (WPM) |
|---|---|
| Dual-finger | 17.8 ± 3.9 [9.5, 25.4] |
| Single | 16.3 ± 3.2 [11.9, 23.4] |
| Swipe-up | 16.5 ± 3.7 [10.5, 23.5] |
| ANOVA | $F_{2,34} = 2.02$, $p = 0.14$ |

### 4.1.2 Questionnaire

Following the second experiment, participants completed a questionnaire. We asked them which condition they preferred the most and least, and why. Additionally, we questioned them about fatigue throughout the study. Eight participants preferred the Dual-finger condition. These participants answered that they preferred Dual-finger because it was faster, required no additional movement, felt natural, and was simple to use. Three participants indicated that the Dual-finger condition was their least preferred, primarily because it may be difficult for people with larger finger sizes and they were concerned about hitting the wrong key.

Three participants preferred the Single condition, owing to their familiarity with it. In comparison, eleven participants rated Single as their least preferred condition because pressing multiple keys in different locations on the keyboard slowed down the action or caused them to lose their typing flow when switching between the shift and

**Table 4.2**
A selection of positive (+) and negative (−) comments from participants about each condition in Experiment 1

| Dual-finger |
|---|
| + "I felt like I could type faster and not worry about different commands. It felt natural." |
| + "Using the Dual-finger was more like regular typing; there was no extra movement involved." |
| + "It was less steps to capitalize by just taping with 2 fingers." |
| − "My fingers are different sizes so it was a little hard to hit one key with two fingers." |
| − "I did not prefer the Dual-finger method as much as the others due to the size of the keyboard." |
| − "Felt like I was pressing two buttons." |

| Swipe-up |
|---|
| + "Swiping was easy and could be done without much added effort." |
| + "It took me less time and energy compared to having to use another key or finger." |
| + "Feel just a tiny bit more easier." |
| − "It was the most unfamiliar to me and sometimes I was not sure whether to swipe up or not." |
| − "Took longer time." |
| − "Swiping up is confusing because we in phones we swipe up for other kind of actions." |

| Single |
|---|
| + "More used to it." |
| + "Feels like real keyboards." |
| + "I am much more familiar with using the shift key." |
| − "Having to tap another "key" took extra time and slowed the type entry." |
| − "Felt like an extra step. Slowest." |
| −"Prefer not to use multiple keys." |

letter keys. Seven participants indicated their preferred Swipe-up condition because it eliminated the need for finger adjustments, as with condition Dual-finger, or for additional keys, as with condition Single. Four participants stated that they

disliked SWIPE-UP conditions because it took longer to use or it was confusing to use. In terms of fatigue, 11 participants reported experiencing some level of fatigue throughout the experiment. Table 4.2 gives a list of selected positive and negative comments about the condition.

### 4.1.3 Open Comments

Additionally, the questionnaire asked participants to imagine themselves wearing a pair of future augmented reality smart glasses. They were told these future AR smart glasses could project a keyboard onto any surface in their environment and could detect their interactions with the projected keyboard. We asked what locations and surfaces they would type, such as walls or tables. Most of the participants preferred a table or desk and some mentioned their body or a vertical wall. Additionally, we inquired about any changes they would make to the keyboard's size or orientation. Some users expressed a desire for a larger keyboard to facilitate typing with two hands. However, the majority chose to remain at their current size. Finally, we asked how they would envision interacting with the projected keyboard. Would they use one hand or both? How are they going to input symbols? The majority of participants preferred to type with both hands and use a separate row for punctuation or a combination of keys and punctuation, switching between them via conditional functions such as swipe. A list of selected comments is listed in Table 4.3.

**Table 4.3**
Selected answers to questions about the future AR keyboard.

| What locations and surfaces do you think you would type on? |
|---|
| "Floor, desk, wall, at home, inside table." |
| "I prefer typing on a table." |
| "Table, Wall, and on my body (e.g., forearms)." |

| What would you change about the keyboard's size or orientation? |
|---|
| "Put all punctuation in a separate line above or below letters." |
| "I prefer typing on a table." |
| "I think the size was fine." |
| "I would make it bigger for augmented reality as it seems small if I wanted to use two hands to type." |

| Would you use one or both hands and How would you input symbols? |
|---|
| "I would use both hands and use it the same way as a computer keyboard." |
| "I would use both hands. I envision it being like typing on a keyboard that is connected to my computer but in AR" |
| "I would use both hands. I would also use the shift key to input the symbols." |

We asked participants to comment on how they thought this study could be made easier in the final section of the questionnaire. The majority of them found the study simple. Also, we asked about the most confusing part of the study and any other comments about it. Some found our experiments confusing when measuring their distance to the wall.

Overall, we were unable to confirm our hypothesis. Our hypothesis was that the Swipe-up and Dual-finger interaction methods would be faster than using a conventional shift key.

## 4.2 Experiment 2: Detecting and classifying taps and gestures

A classification algorithm's primary goal is to discover and learn patterns between distinct groups of data, and by generalizing these differences well, it may predict unseen data. In other words, try to minimize the errors associated with unobserved data. In recent years, the Convolution Neural Network (CNN) has gained attention in the audio research field because of recent audio classification and speech recognition successes [18, 40, 44]. Moreover, prior work on environmental sound classification found that time-frequency representations are highly beneficial as learning characteristics [13, 19, 37].

Generally, we describe a waveform as a representation of the signal sample value intensity varying over time. However, this is not a good representation of the information in a signal for a CNN. In order to extract the information embedded in a signal, we need to use time-frequency representation. Time-frequency represents signal information over both time and frequency (Figure 4.2).

Spectrograms are one type of time-frequency representation used to describe an audio signal. It is made of pixels that represent the intensity of a range of frequencies at a particular time step. For example, brighter pixels have a greater amount of energy

for that specific frequency.

The Mel-spectrogram (Figure 4.3) is a subcategory of spectrograms that converts audio frequency values into a scale that matches human hearing perception model (mel scale). Also, it makes it ideal for applications like speech recognition and audio categorization that require a human hearing perception model. Moreover, because of the 2-dimension representation of the mel-spectrogram, it is a good candidate for our CNN. Finally, they have lately been successfully employed to classify sounds [21].

Overall, we used time-frequency data (Figure 4.3) and CNN to create an audio classifier to classify and detect surface interaction events. The following section will go over the specifics of our CNN classifier.

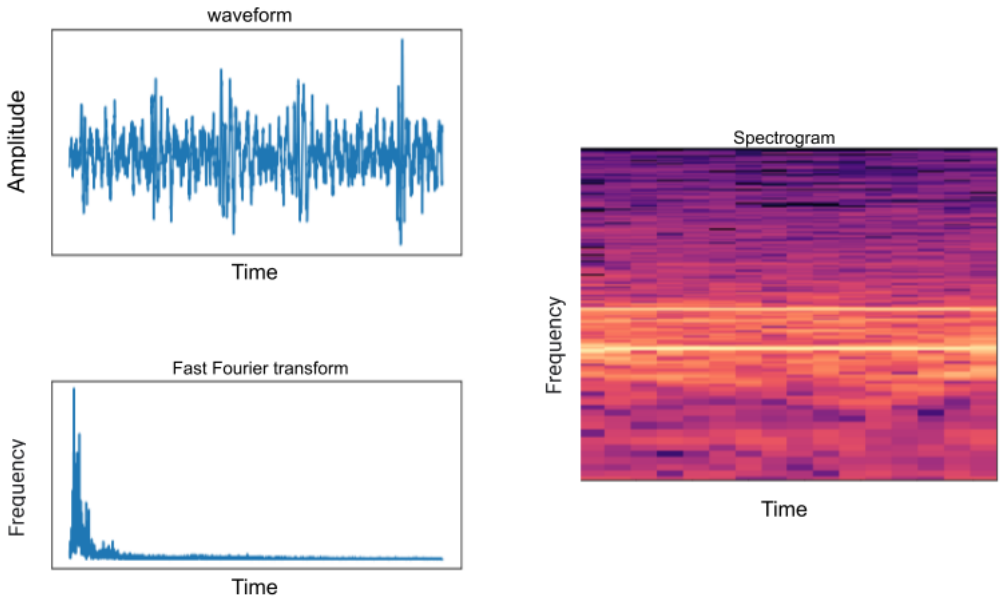


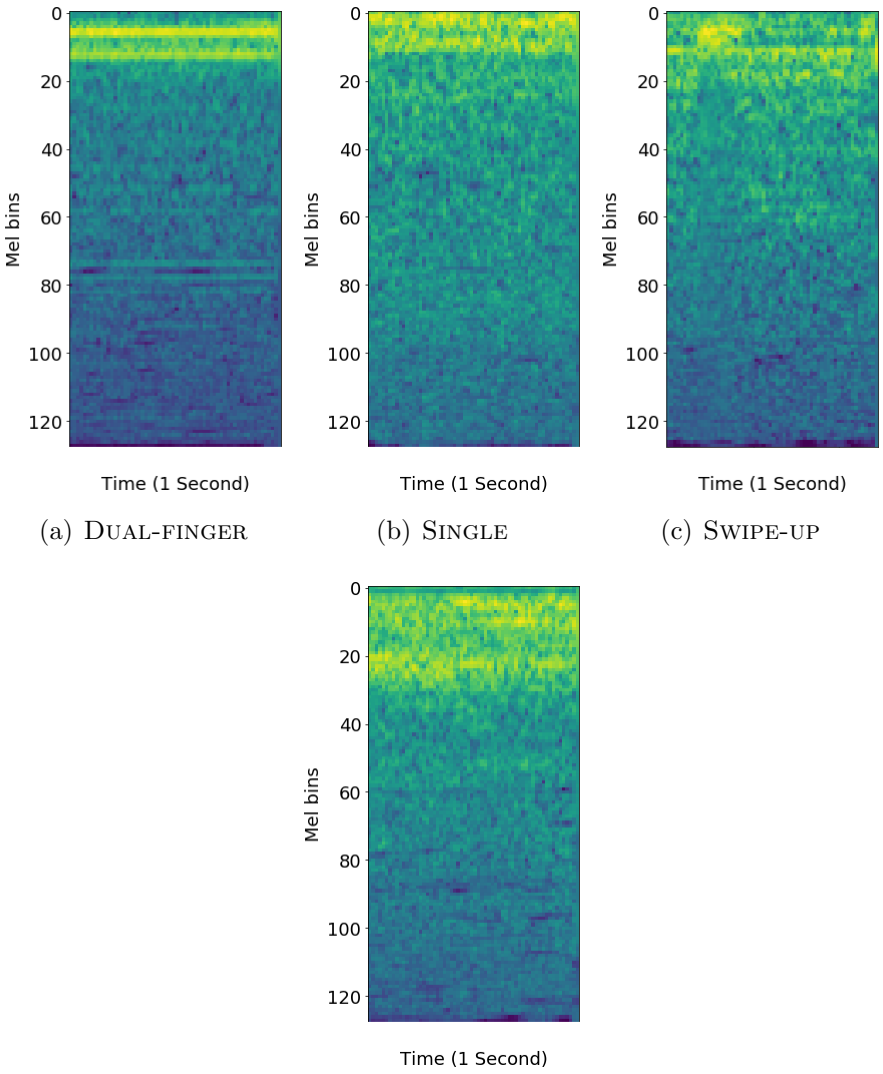

**Figure 4.2:** The waveform (top) and time-frequency (bottom and right) representations of a single audio sample from the DUAL-FINGER class.

### 4.2.1 Pre-processing

Our two experiments resulted in a total of 432 recorded videos. We excluded 16 sentences because of corrupted files and duplicated sentences. This problem at most affects two videos of the same participant in a given condition. We began by watching all the videos and identifying the sequence of video frames that constituted the different actions Swipe-up, Dual-finger, and Single. Moreover, we extracted frames without any events called NoTap. We named our classes based on these extracted events (Swipe-up, Dual-finger, Single, NoTap). As a result, we extracted 1,840 audio clips (460 for each class) which each contains a single event. For instance, there were no two Swipe-up actions in a single clip. Since we need the same size input for our CNN, all audio samples were standardized by clipping and padding them to a 1-second duration.

### 4.2.2 Architecture

A convolution neural network usually consists of multiple different layers stacked on top of each other (Figure 4.4). We define convolution as an operation where we have an input and a kernel sliding over the input data to create a feature map. The feature map shows which features were detected in the input.




(a) Dual-finger (b) Single (c) Swipe-up

(d) NoTap

**Figure 4.3:** Mel-spectrograms for 10 milliseconds of audio data from each of our four classes. (a) Dual-finger (b) Single (c) Swipe-up (d) NoTap

We constructed our convolutional neural network using time distributed 1D convolution and time-frequency data (mel-spectrogram) as its input data. Time distributed 1D convolution was a good fit for our application because it learned local patterns in the frequency spectrum over time, which aligned with our input data dimension. Moreover, 1D convolution performed well in applications with small datasets similar

to our dataset [25, 26].

Time distributed 1D convolution looks at each time step where each time set is a set of mel features. This method helped us to organize some data in a sequence without considering their order. Hence, a pattern learned at one position can be recognized at another position afterward. We also experimented with a variety of other networks, including Long short-term memory (LSTM) [20] and very deep convolutional networks for large-scale image recognition (VGG) [48]. However, 1D convolution achieved better performance and provided better accuracy on the unseen data.

The first layer of our network was extracted mel features from our audio dataset. The batch normalization layer [22] was the second layer, and it helped our model learn and generalize new data more effectively. Our classifier's basic concept was to start with general features in the first layers and then move deeper and learn more patterns in the final layers by increasing the number of parameters. It helped CNN to learn more abstract representations of the input data as we went through layers. We employed five 1D convolution layers with a kernel size of four and began with a filter size of eight, progressively increasing the filter size to 16, 32, 64, and 128 for the last four 1D convolution layers. For all 1D convolution layers, we employed the rectified linear unit (ReLU) as an activation function.

Each 1D convolution layer is followed by a max pooling 2D layer. The primary reason for implementing max pooling is to minimize the number of feature-map coefficients

needed to be processed. Large feature maps also lead to overfitting. We used global max pooling after the final 1D convolution layer to highlight the most present features in the last layer, allowing us to add one dense layer to the model and perform classification. After global max pooling, we applied dropout to prevent overfitting by randomly dropping out some output features of the layers during training. Moreover, an L2 weight regularization was added to reduce training dataset overfitting and improve model generalization. Finally, the value produced by the last layer was converted and normalized into a probability distribution using a softmax layer as an output layer.

### 4.2.3 Experiment setup

We used Librosa [34] to extract audio features and Keras [8] to implement our 1D convolution CNN. We prepared a train and test dataset based on the Leave-one-out cross-validation. We held out one participant from our dataset and each training set was made up of all participants except the one held out. The goal was to know the expected accuracy of the system on a participant who had just used the training model based on a collected set of other participants' data. To keep balance, we need to repeat the training experiment 18 times with each participant being held out one time. This approach may perform better on a particular set because the held out test participant is more or less like the data in the 17 trained participants' data.

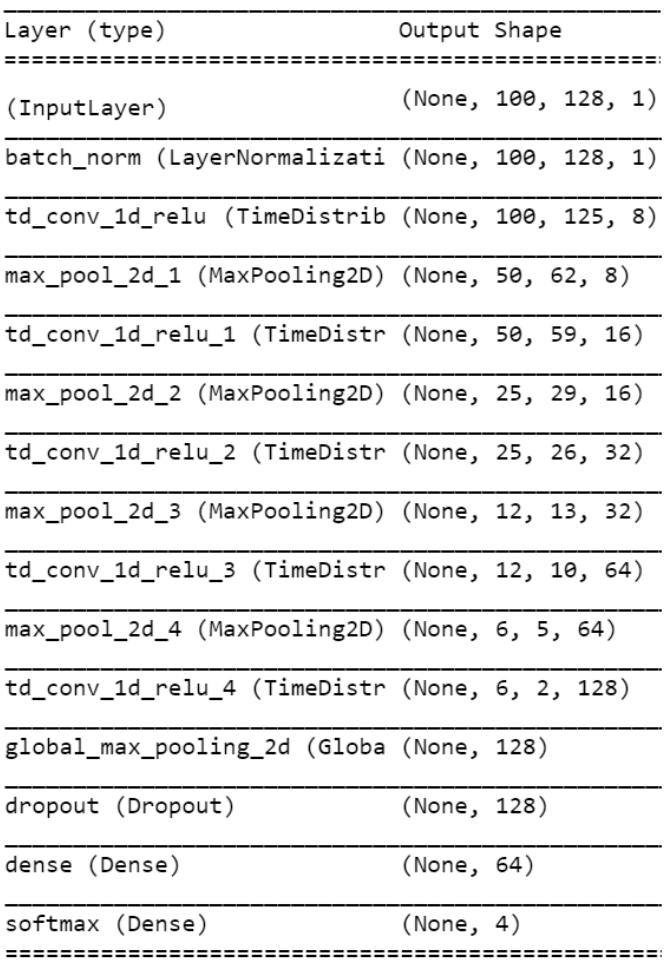



**Figure 4.4:** Model architecture for the 1D convolution classifier.

We experimented with a variety of settings and found that the Adam optimizer [24] with 30 epochs, a batch size of 16, a learning rate of 0.001, and a momentum term [3] of 0.9 gave the best results. Moreover, after our last global max pooling, we had 0.001 L2 weight decay and a 0.5 dropout probability. On an RTX 2080 Ti GPU, training and testing took four hours for all 18 training runs.

### 4.2.4 Results

T Our results summarized in Table 4.4. We also provided normalized confusion matrices (Figure 4.5) for On-paper and Off-paper conditions. The confusion matrix is an additional tool for analyzing our classifier's output. The rows represent the actual classes, whereas the columns represent the predicted classes. As we described earlier (Section 3.5), we analyzed tapping and swiping on a sheet of paper versus off the sheet of paper. The confusion matrices demonstrate that several classes have been classified incorrectly. In both matrices, the classifier was unable to reliably classify audio snippets for the Swipe-up, Single, and Dual-finger classes.

In Off-paper data, Dual-finger 35% and Single 20% performed somewhat worse than Dual-finger 40% and Single 23% in On-paper data but slightly better in Swipe-up 25% versus Swipe-up 23% in On-paper. The NoTap class did better than other classes with 83% predicted classes in both matrices.

**Table 4.4**
The result of classification evaluation. The overall accuracy was then determined by averaging accuracy across all 18 training run.

| **Experiment No** | **Accuracy** |
|---|---|
| 1 | 52% |
| 2 | 56% |
| 3 | 42% |
| 4 | 68% |
| 5 | 45% |
| 6 | 24% |
| 7 | 34% |
| 8 | 48% |
| 9 | 56% |
| 10 | 45% |
| 11 | 34% |
| 12 | 33% |
| 13 | 47% |
| 14 | 25% |
| 15 | 55% |
| 16 | 65% |
| 17 | 33% |
| 18 | 42% |
| **Overall average** | **45**% |

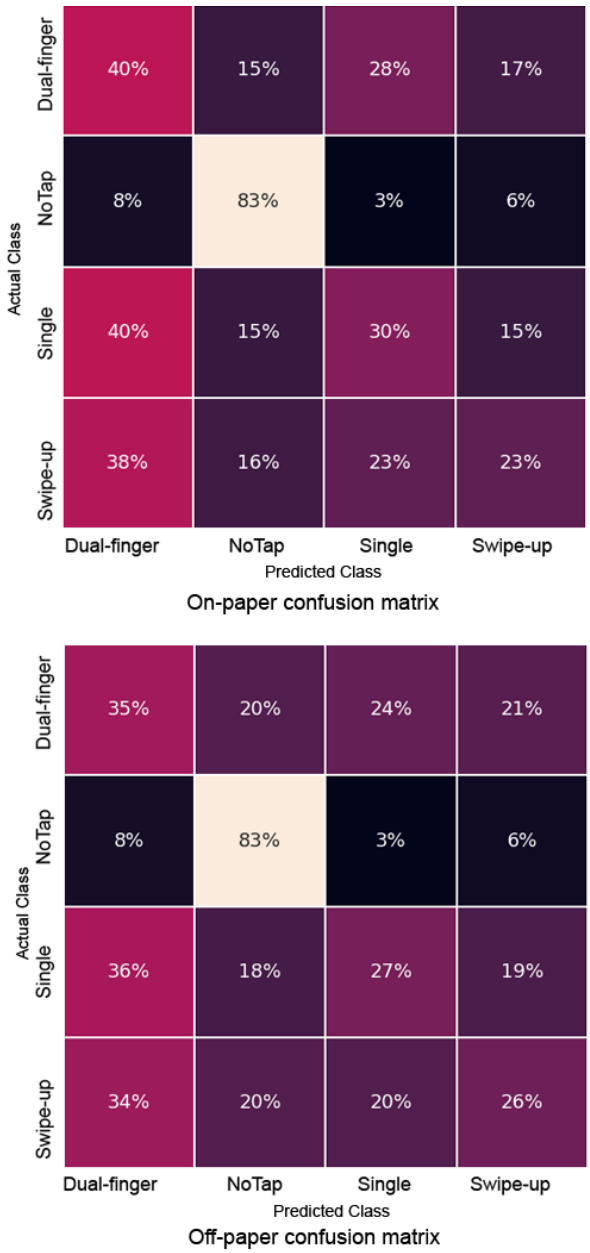



**Figure 4.5:** Confusion matrices for On-paper data (top) and Off-paper data (bottom) classes. SWIPE-UP was the most challenging class to predict in both on and off paper audio data because it was frequently mistaken with DUAL-FINGER. NOTAP with the highest proportion of predicated classes was also the network's best prediction.

# Chapter 5

# Discussion and Future Work

## 5.1 Discussion

This thesis investigated augmented reality surface interaction by leveraging tap and swipe input on a surface for future AR applications where people use head-mounted displays to enter text while they are on-the-go on any surface. Despite advances in sensors, cameras, and recognition systems, augmented reality HMDs face significant problems determining when a tap or swipe has occurred. A finger can be detected via visible light or infrared camera. However, it might be challenging to determine precisely when a tap has occurred since determining surface contact is likely hard.

We examined tapping and swiping to address the mentioned issue, since both interaction techniques may be used for typing, games, and multimedia applications. For this purpose, we conducted two experiments. In an ideal scenario, we'd employ an augmented reality head-mounted display (HMD) in a lab setting to project a virtual keyboard on the wall. However, due to the COVID-19 pandemic, we evaluated our system in a remote study where we utilized a sheet of paper as a keyboard and a mobile phone to simulate AR HMD sensors in conjunction with a surface. First, we developed three methods for entering capital letters and punctuation and compared them using words-per-minute metric. Second, we trained a classifier to detect surface interaction events and distinguish between tapping with two fingers at the same time, single tap, and swipes using recorded audio from the study.

The first experiment explored the difference between text entry input using the shift key on the keyboard (Single condition), tapping with two fingers at the same time (Dual-finger condition), and swiping up with one finger (Swipe-up condition) to enter capital letters and punctuation. Although we hypothesize that swiping up and tapping with two fingers at the same time provides a faster alternative for entering capital letters and punctuation, the results showed that participants were only slightly faster in Dual-finger (17.8 WPM) than Swipe-up (16.4 WPM) and Single (16.3 WPM), but this difference was not statistically significant. It may be interesting to see if the lack of a difference continues if more data was collected.

The second experiment investigated surface interaction events such as tapping and swiping on and off a sheet of paper. The paper keyboard is intended to simulate the experience of using a virtual keyboard displayed in future augmented reality glasses. Hence, it is essential to investigate different taps and gestures on different surfaces. From the mel-spectrogram derived from the audio data, we constructed a 1D convolutional CNN audio classifier. The classifier does a much better job of recognizing the NoTap class and accurately predicted 83% of the provided test data. However, it performed relatively poorly in other classes, such as Dual-finger, Swipe-up, and Single. For instance, in On-paper data, the classifier predicted 40% Dual-finger, 23% Swipe-up and 30% Single, while in Off-paper data, the classifier predicted only 35% Dual-finger, 26% Swipe-up and 27% Single. Our primary hypothesis was that it might be possible to detect when a surface event occurs and it might be possible to differentiate between different types of surface events. The results showed that we can detect surface events and it is possible to differentiate between different types of surface events.

Overall, The results showed that we can detect surface events and it is possible to differentiate between different types of surface events. We can determine whether a tap or swipe occurred on the surface versus no event occurring. There was some ability to discriminate within the four classes (Dual-finger, Single, Swipe-up, and NoTap). We also proved that we can use data from a surface with a paper keyboard to detect taps on a surface without paper. Moreover, we can use our system

on a participant who just used the training model based on a collected set of other participants' data. In other words, we can train a system to detect surface interaction events on any data that hasn't been in our training dataset.

As previously stated, it is extremely beneficial because detecting surface impact with head-mounted vision sensors can be difficult. For example, users are adept at determining when they have come into contact with a surface, and their fingers may approach the surface without touching it. As a result, using vision-based systems may not be sufficient. However, our approach is capable of resolving this issue. Additionally, we gathered our audio dataset from a variety of places and under a variety of conditions. As a result, our model may be a good illustration of a real-world situation in which we must detect a tap or swipe varying surfaces.

## 5.2 Limitations

Our classifier is capable of detecting when a user taps or swipes the surface. However, it is unable to ascertain the timestamp of an surface interaction event within an audio sequence. We intend to address this limitation by enhancing our detection system to accurately determine the timestamp of audio events inside an audio file.

Another limitation is the dataset's size. In general, there are no definitive answers

for determining the optimal size of a CNN dataset and it depends on a variety of factors, including the problem's scale and dataset parameters. Hence, we need to investigate more about the optimum size for our dataset. We intend to address these limitations in the future. We can determine the optimal dataset size for our problem by analyzing the prior work dataset and the performance of our network.

We chose mel-spectrogram as the input data to our classifier mainly because of recent successful use of mel-spectrogram in sound classification. Moreover, due to the fact that mel-spectrogram data is two-dimensional, it is an excellent candidate for our CNN. However, it may be worthwhile to experiment with alternative audio representations in the future, such as waveform, Linear-STFT, or CQT, and compare the results to the current ones.

The system we used was not designed to detect rapid, consecutive taps. Using DualFinger eliminates the need to detect when rapid, consecutive taps occur in the audio data and might be faster and/or easier for users to perform than double-tap. Nonetheless, future work may be needed in the area to detect when taps occur when a person is entering text with two hands.

We conducted our experiments remotely due to COVID-19 restrictions, which precluded us from using an AR HMD and conducting an in-person experiment. Hence, we asked participants to use their phones and they were required to hold the phones at their foreheads and type sentences with another hand; they were unable to enter

text with two hands. Additionally, we faced several obstacles, including developing a mobile application compatible with various mobile device versions. We made every effort to provide straightforward instructions, a simple user interface for the mobile application, and an easy experiment setup. However, conducting a remote study affected our experiment and confused some participants. It was challenging to provide good instructions, helping people remotely when they had trouble—for example, adjusting the keyboard layout on the wall to accommodate the participant's height and installing the application on older versions of Android.

Finally, Holding the phone at the forehead with one hand and reading sentences from the reference sheet below the keyboard was another limitation of this experiment, resulting in fatigue and errors such as incorrect sentence entry during the experiment. We anticipated this issue and attempted to make the reference sheet as clear as possible, but we still noticed some errors.

## 5.3 Future work

Due to the fact that work on augmented reality surface interaction has only begun recently, there are many possibilities for future work. We make the following recommendations for future work based on the experiment results and mentioned limitations:

1. A critical component of our strategy is the use of alternative sensing methods, such as vision. We have already collected video clips from our experiment, which will aid in the extension of our current model and detect surface event's location on a surface. Additionally, it will assist us in classifying the various types of events.

2. Finding the optimal dataset size for our problem by analyzing the prior work dataset and the performance of our network.

3. Extend our text entry by detecting when taps occur when a person is entering text with two hands.

4. Investigate alternative audio representations in the future, such as waveform, Linear-STFT, or CQT, and compare the results to the current ones.

5. We intend to make the audio/video dataset from collected data available for other public use.

6. We intend to expand the current dataset by acquiring additional data. The first advantage of having more data is that it improves performance and avoids overfitting with a small dataset.

7. Conducting in-person experiments with better control over the experiment process may result in a higher-quality dataset.

8. Extend our audio detection system to include the detection of timestamp events

in audio data.

9. Additionally, we intend to expand our research by examining different surface materials and horizontal surfaces.

## 5.4 Conclusion

This thesis investigated augmented reality text entry by leveraging tap and swipe input on a surface for future applications. Despite advances in sensors, cameras, and recognition systems, augmented reality users face significant problems determining when a tap or swipe has occurred. We investigated the ability of a wearable mic on the user's head to capture acoustic data from surface interaction events such as taps or swipes, and made an audio classifier to detect those events. Additionally, we investigated the benefits of novel text entry methods such as tapping with two fingers simultaneously, swiping up, and single tapping to enter capital letters and punctuation.

Despite the challenges, our system demonstrated new potential for future augmented reality text entry during the experiment. There was some ability to detect tapping and swiping. Moreover, discriminate within the four classes (Dual-finger, Single, Swipe-up, NoTap) primarily between the NoTap and other classes, which is quite useful for future augmented reality text entry systems.

# References


[1] Agarwal, A., Izadi, S., Chandraker, M., and Blake, A. High precision multi-touch sensing on surfaces using overhead cameras. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)* (2007), IEEE, pp. 197–200.

[2] Bancroft, S. An algebraic solution of the gps equations. *IEEE Transactions on Aerospace and Electronic Systems*, 1 (1985), 56–59.

[3] Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R. Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), IEEE, pp. 8624–8628.

[4] Blanchard, W. Air navigation systems chapter 4. hyperbolic airborne radio navigation aids–a navigator's view of their history and development. *The Journal of Navigation 44*, 3 (1991), 285–315.

[5] Caffery, J. J., and Stuber, G. L. Overview of radiolocation in cdma cellular systems. *IEEE Communications Magazine 36*, 4 (1998), 38–45.

[6] Chen, X., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: exploring joint interactions on a smart phone and a smart watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2014), pp. 159–168.

[7] Choi, K., Joo, D., and Kim, J. Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras. *arXiv preprint arXiv:1706.05781* (2017).

[8] Chollet, F., et al. Keras. `https://keras.io`, 2015.

[9] Commons, W. File:qwerty.svg — wikimedia commons, the free media repository, 2021. [Online; accessed 21-February-2021].

[10] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (2009), Ieee, pp. 248–255.

[11] Dietz, P., and Leigh, D. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (2001), pp. 219–226.

[12] Dudley, J. J., Vertanen, K., and Kristensson, P. O. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI) 25*, 6 (2018), 1–40.

[13] Gong, Y., Chung, Y.-A., and Glass, J. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778* (2021).

[14] Grossman, T., Chen, X. A., and Fitzmaurice, G. Typing on glasses: Adapting text entry to smart eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (2015), pp. 144–152.

[15] Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (2005), pp. 115–118.

[16] Harrison, C., and Hudson, S. E. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (2008), pp. 205–208.

[17] Harrison, C., Schwarz, J., and Hudson, S. E. Tapsense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), pp. 627–636.

[18] Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al. Cnn architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2017), IEEE, pp. 131–135.

[19] Hertel, L., Phan, H., and Mertins, A. Comparing time and frequency domain for audio event recognition using deep learning. In *2016 International Joint Conference on Neural Networks (Ijcnn)* (2016), IEEE, pp. 3407–3411.

[20] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9* (12 1997), 1735–80.

[21] Huzaifah, M. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *arXiv preprint arXiv:1706.07156* (2017).

[22] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (2015), PMLR, pp. 448–456.

[23] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth

camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (2011), pp. 559–568.

[24] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., and Inman, D. J. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing 151* (2021), 107398.

[26] Kiranyaz, S., Ince, T., Hamila, R., and Gabbouj, M. Convolutional neural networks for patient-specific ecg classification. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2015), IEEE, pp. 2608–2611.

[27] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25* (2012), 1097–1105.

[28] Kwon, Y. D., Shatilov, K. A., Lee, L.-H., Kumyol, S., Lam, K.-Y., Yau, Y.-P., and Hui, P. Myokey: surface electromyography and inertial motion sensing-based text entry in ar. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (2020), IEEE, pp. 1–4.

[29] Lee, M., and Woo, W. Arkb: 3d vision-based augmented reality keyboard. In *ICAT* (2003).

[30] Lee, S., Buxton, W., and Smith, K. A multi-touch three dimensional touch-sensitive tablet. *Acm Sigchi Bulletin 16*, 4 (1985), 21–25.

[31] Leo, C. K. *Contact and free-gesture tracking for large interactive surfaces.* PhD thesis, Massachusetts Institute of Technology, 2002.

[32] Letessier, J., and Bérard, F. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (2004), pp. 119–122.

[33] Matsushita, N., and Rekimoto, J. Holowall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th annual ACM symposium on User interface software and technology* (1997), pp. 209–210.

[34] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (2015), vol. 8, Citeseer, pp. 18–25.

[35] Molero, D., Schez-Sobrino, S., Vallejo, D., Glez-Morcillo, C., and Albusac, J. A novel approach to learning music and piano based on mixed reality and gamification. *Multimedia Tools and Applications* (2020), 1–22.

[36] Nirjon, S., Gummeson, J., Gelb, D., and Kim, K.-H. Typingring: A wearable ring platform for text input. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services* (2015), pp. 227–239.

[37] Orr, M. C., Pham, D. S., Lithgow, B., and Mahony, R. Speech perception based algorithm for the separation of overlapping speech signal. In *The Seventh Australian and New Zealand Intelligent Information Systems Conference, 2001* (2001), IEEE, pp. 341–344.

[38] Paradiso, J. A., Hsiao, K.-y., Strickon, J., Lifton, J., and Adler, A. Sensor systems for interactive surfaces. *IBM Systems Journal 39*, 3.4 (2000), 892–914.

[39] Pedersen, E. W., and Hornbæk, K. Expressive touch: Studying tapping force on tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2014), pp. 421–430.

[40] Piczak, K. J. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)* (2015), IEEE, pp. 1–6.

[41] Reifinger, S., Wallhoff, F., Ablassmeier, M., Poitschke, T., and Rigoll, G. Static and dynamic hand-gesture recognition for augmented reality applications. In *International Conference on Human-Computer Interaction*

(2007), Springer, pp. 728–737.

[42] Rekimoto, J. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2002), pp. 113–120.

[43] Rekimoto, J., and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (1999), pp. 378–385.

[44] Salamon, J., and Bello, J. P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters 24*, 3 (2017), 279–283.

[45] Salamon, J., Jacoby, C., and Bello, J. P. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia* (2014), pp. 1041–1044.

[46] Sharma, R. P., and Verma, G. K. Human computer interaction using hand gesture. *Procedia Computer Science 54* (2015), 721–727.

[47] Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems* (2015), pp. 3633–3642.

[48] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[49] Steimle, J., Jordt, A., and Maes, P. Flexpad: highly flexible bending interactions for projected handheld displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), pp. 237–246.

[50] Sugita, N., Iwai, D., and Sato, K. Touch sensing by image analysis of fingernail. In *2008 SICE Annual Conference* (2008), IEEE, pp. 1520–1525.

[51] Sumida, T., Hirai, S., Ito, D., and Kawakatsu, R. Raptapbath: User interface system by tapping on a bathtub edge utilizing embedded acoustic sensors. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (2017), pp. 181–190.

[52] Tabbakh, S. K., Habibi, R., and Vafadar, S. Design and implementation of a framework based on augmented reality for phobia treatment applications. In *2015 International Congress on Technology, Communication and Knowledge (ICTCK)* (2015), IEEE, pp. 366–370.

[53] Ullmer, B., and Ishii, H. The metadesk: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology* (1997), pp. 223–232.

[54] Vertanen, K., Gaines, D., Fletcher, C., Stanage, A. M., Watling, R., and Kristensson, P. O. Velociwatch: designing and evaluating a virtual

keyboard for the input of challenging text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–14.

[55] Vertanen, K., and Kristensson, P. O. Dataset 2: Recommended language models" (2019). mobile text dataset and language models. In *Dataset 2: Recommended Language Models" (2019). Mobile Text Dataset and Language Models* (2019), p. 2.

[56] Wang, C.-Y., Chu, W.-C., Chiu, P.-T., Hsiu, M.-C., Chiang, Y.-H., and Chen, M. Y. Palmtype: Using palms as keyboards for smart glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (2015), pp. 153–160.

[57] Wei, D., Zhou, S. Z., and Xie, D. Mtmr: A conceptual interior design framework integrating mixed reality with the multi-touch tabletop interface. In *2010 IEEE International Symposium on Mixed and Augmented Reality* (2010), IEEE, pp. 279–280.

[58] Wellner, P. The digitaldesk calculator: tangible manipulation on a desk top display. In *Proceedings of the 4th annual ACM symposium on User interface software and technology* (1991), pp. 27–33.

[59] Wilson, A. D. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces* (2004), pp. 69–76.

[60] Wilson, A. D. Playanywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM symposium on User interface software and technology* (2005), pp. 83–92.

[61] Wilson, A. D. Using a depth camera as a touch sensor. In *ACM international conference on interactive tabletops and surfaces* (2010), pp. 69–72.

[62] Xiao, R., Harrison, C., and Hudson, S. E. Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2013), pp. 879–888.

[63] Xiao, R., Hudson, S., and Harrison, C. Direct: Making touch tracking on ordinary surfaces practical with hybrid depth-infrared sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (2016), pp. 85–94.

[64] Xiao, R., Lew, G., Marsanico, J., Hariharan, D., Hudson, S., and Harrison, C. Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services* (2014), pp. 67–76.

[65] Xiao, R., Schwarz, J., Throm, N., Wilson, A. D., and Benko, H. Mrtouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics 24*, 4 (2018), 1653–1660.

[66] Zhai, S., and Kristensson, P. O. The word-gesture keyboard: reimagining keyboard interaction. *Communications of the ACM 55*, 9 (2012), 91–101.

[67] Zhang, C., Bedri, A., Reyes, G., Bercik, B., Inan, O. T., Starner, T. E., and Abowd, G. D. Tapskin: Recognizing on-skin input for smartwatches. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (2016), pp. 13–22.

[68] Zhu, F., and Grossman, T. Bishare: Exploring bidirectional interactions between smartphones and head-mounted augmented reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020), pp. 1–14.