Handwriting Recognition System Leveraging Vibration Signal on Smartphones

Dian Ding, Lanqing Yang, Yi-Chao Chen, *Member, IEEE*, and Guangtao Xue, *Member, IEEE*

Abstract—The efficiency of human-computer interaction is greatly hindered by the small size of the touch screens on mobile devices, such as smart phones and watches. This has prompted widespread interest in handwriting recognition systems, which can be divided into active and passive systems. Active systems require additional hardware devices to perceive movements of handwriting or the tracking accuracy is not adequate for handwriting recognition. Passive methods use the acoustic signal of pen rubbing and are susceptible to environmental noise (above 60dB). This paper presents a novel handwriting recognition system based on vibration signals detected by the built-in accelerometer of smartphones. The proposed scheme is implemented in three stages: signal segmentation, signal recognition, and word suggestion. *VibWriter* is highly resistant to interferences since the normal environmental noise (below 70dB) will not cause the vibration of the accelerometer. Extensive experiments demonstrated the efficacy of the system in terms of accuracy in letter recognition (75.3%), word recognition (86.4%) and number recognition (79%) in a variety of writing positions under a variety of environmental conditions.

Index Terms—Vibration signal, handwriting recognition

1 Introduction

The portability and high performance of smart phones have brought convenience to people's daily study and work. Existing interaction methods based on smart phones (buttons, touch screens, etc.) have problems such as easy accidental touch and low input efficiency. Researches on alternative interaction systems have focused on speech recognition [1] and handwriting recognition [2], [3]. In the public places such as offices, speech-based methods inevitably disturb others, therefore the handwriting-based methods are a better approach [3].

Most existing handwriting recognition methods can be categorized as localization-based and scratch-based methods. Localization-based methods detect the movement of the user's hand or pen via inertial sensors [4] or wireless signals, such as acoustic signal [5], [6], [7], WiFi signal [2], and magnetic signal [8]. Methods based on WiFi signal [2] or magnetic signal [8] have limitations for experimental scenarios. The acoustic-based tracking methods [5], [6], [7] achieve millimeter-level tracking accuracy. Since the medium size of letters in handwriting is $2.5-3.5mm$ according the researches in graphology [9], these methods can still impair the recognition accuracy. Scratch-based methods [3], [10], [11] involve the detection of acoustic signals generated by dragging a pen or finger across a surface, but these methods are highly susceptible to environmental noise [3], [10], e.g., the ambient noise in a cafe is about $61dB$ [11].

In this paper, we seek to overcome the shortcomings of existing handwriting recognition schemes by developing a system that uses the built-in accelerometer of the smart phone to detect the vibration signals generated by a pen writing on the desk. *VibWriter* has no dependence on additional hardware and is more resistant to interference from environment noise and vibrations. The system also demonstrates outstanding recognition performance under different conditions, such as different smart phones, different desks and different writing regions.

The development of *VibWriter* imposes a number of challenges:

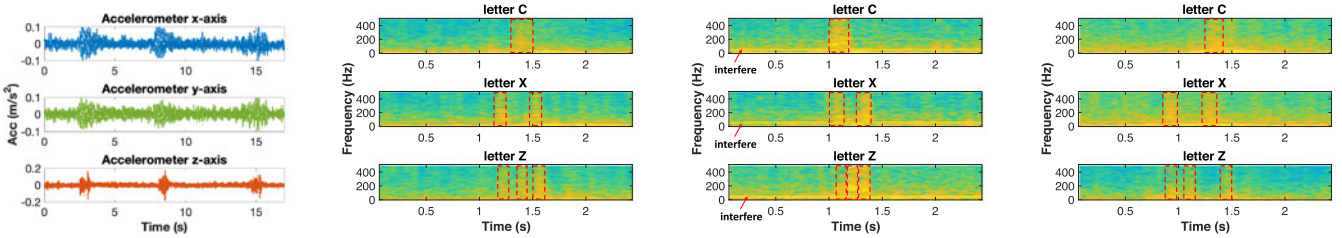
1) The sampling rate of the built-in accelerometer tends to be low and lacking in stability. This imposes daunting challenges in reconstructing and processing vibration signals from an input with limited bandwidth.
2) The fact that the vibration signal indicating the start of a new letter is usually generated by a tap or swipe makes it difficult to differentiate between letters. Real-world writing scenarios also present numerous unexpected situations prompting the user to write more quickly or more slowly. Finally, a small time interval between letters can lead to signal overlap, whereas a large time interval can hinder signal separation.
3) The removal of noise from the signal can be hindered by variations in noise characteristics over time.

VibWriter addresses these issues using the corresponding solutions listed below:

1) Data missing from the vibration signal is reconstructed using the spline interpolation algorithm. The $Xception$ module is used to extract deep features for the residual architecture and depth-wise separable convolution layers.

• *The authors are with the Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {dingdian94, yanglanqing, yichao, gt_xue}@sjtu.edu.cn.*

Manuscript received 11 June 2021; revised 21 Jan. 2022; accepted 28 Jan. 2022. Date of publication 4 Feb. 2022; date of current version 5 June 2023.
This work was supported in part by NSFC under Grants 61936015, U1736207, and 62072306, in part by the Startup Fund for Youngman Research at SJTU, and in part Program of Shanghai Academic Research Leader under Grant 20XD1402100.
(Corresponding author: Guangtao Xue.)
Digital Object Identifier no. 10.1109/TMC.2022.3148172



(a) Vibration signal of volunteer 1. (b) Frequency spectrum of volunteer 1. (c) Frequency spectrum of volunteer 1 with interferences. (d) Frequency spectra of volunteer 2.

Fig. 1. Preliminary experiments with Samsung S7 (495Hz): Figs. 1a and 1b Vibration signal and frequency spectrum generated by writing the letters "C", "X" and "Z" of volunteer 1; Fig. 1c Frequency spectrum of volunteer 1 with different interferences; Fig. 1d Frequency spectrum generated by writing the letters of volunteer 2.

2) A mean window is used to detect signal segments that are characteristic of handwriting. The problems of signal overlap and signal separation are dealt with by combining information in the time and frequency domains and selecting appropriate time for signal splitting and merging based on changes in signal strength.
3) We develop a dynamic denoising algorithm, which uses the noise signal generated during idle periods as a reference.

To the best of our knowledge, this is the first vibration-based handwriting recognition system. The main contributions are summarized as follows:

1) We demonstrate that the built-in accelerometer of the smart phone provides the sensitivity and resolution required for the detection of vibration signals generated by handwriting.
2) We develop the signal processing techniques required to deal with these vibration signals, including signal construction, feature extraction, and feature classification. We also resolve the problems of signal overlap and signal separation.
3) We build a light-weighted recognition network based on knowledge distillation to further enhance the effectiveness of the system.
4) We implement *VibWriter* on an Android smart phone. In experiments, the system achieves accuracy of 75.3% in letter recognition, 86.4% in word recognition and 79% in number recognition.

The remainder of this paper is organized as follows. In Section 2, we outline the preliminary experiments used to verify the feasibility of recognizing handwriting via vibration signals. Section 3 outlines the design of the proposed system. Section 4 outlines the underlying theory and technical details of each component in the system. Experiment parameters and results are detailed in Section 5. Limitations are addressed in Section 6. Related works are discussed in Section 7. Conclusions are drawn in Section 8.

2 Background

VibWriter uses the built-in accelerometer of a Samsung S7 to detect vibration signals generated by the desk when in contact with a pen. This section outlines preliminary experiments aimed at answering the following fundamental questions:

i) Do the Vibration Signals Generated by the Desk Produce Characteristics of Different Letters?

In the first experiment, we seek to determine whether the vibration signals generated by the desk produce characteristics of different letters [12]. The accelerometer of smart phone can achieve the sampling rate of approximately $500Hz$ [13], and even a small strokes of $0.1s$ can generate 50 samples. Therefore, we try to recognize different handwriting letters with the vibration signal. One volunteer is tasked with writing the letters "C", "X", and "Z". As shown in Fig. 1a, the exceedingly weak amplitude of the vibration signals make it difficult to differentiate between the three letters directly. Besides, different letters comprise different numbers of strokes, as indicated by the spectrum in which the letter "Z" comprises three strokes, the letter "X" comprises two , and the letter "C" comprises only one stroke (see Fig. 1b).

ii) Do the Different Environments and Users Affect the Vibration Signal?

In the second experiment, we first test the vibration signals in different environments. When the volunteer is writing, we add different vibration disturbances such as arm movements and the fan. As shown in Fig. 1c, the vibration caused by the fan and the movements of the user's arm is concentrated in the lower frequency band (below $200Hz$), and the high frequency part of the vibration signal can still distinguish the strokes written by the volunteer. However, the uncertainty of vibration interference distribution puts forward the requirements for signal denoising.

Then, we invite another volunteer to write the letters as shown in Fig. 1d. We can also distinguish the strokes of the user from the spectrum. However, due to differences in pauses, stroke order and strength in the writing process, the differences in vibration signals make it difficult to popularize signal recognition.

Preliminary experiments prove that based on the vibration signal, the user's strokes can be recognized to distinguish handwriting in different environments. Nonetheless, it would be difficult to differentiate between all of the letters based solely on the number of strokes. When writing quickly, many letters would be indistinguishable from others with the same number of strokes (e.g., "D"and "P" or "C" and "O"). Besides, the stroke length is affected by the writing size, making it difficult to directly extract the individual stroke length and the length relationship between the strokes. A feature extraction scheme is required for letter recognition.

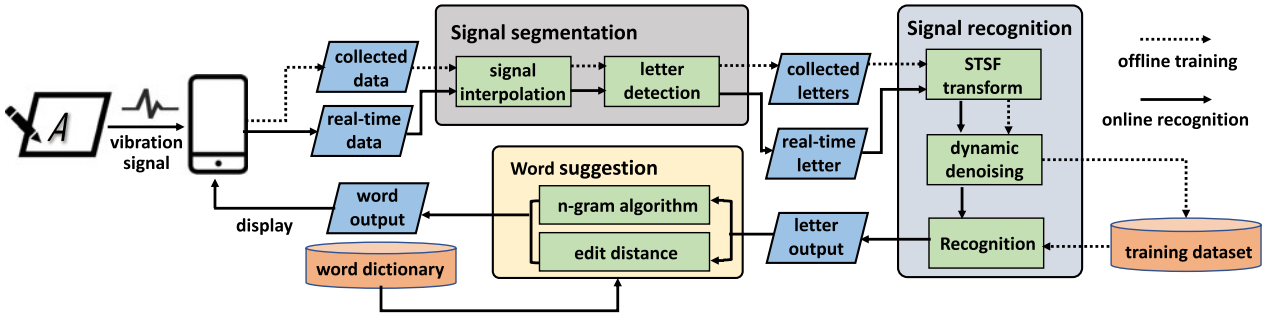


Fig. 2. Overview of *VibWriter*.

3 System Overview

As shown in Fig. 2, *VibWriter* comprises three modules: signal segmentation, signal recognition, and word suggestion. Vibration signals detected by the built-in accelerometer are first sent to the signal segmentation module to be divided into discrete segments. The signal recognition module assembles the segments into letters and numbers. Finally, the word suggestion module combines the letters into words to be displayed on the smart phone. The three functions are examined in greater detail below.

Signal Segmentation. Data missing from the vibration signal is reconstructed using the spline interpolation algorithm. A peak detection algorithm based on mean window is then used to filter out the extraneous signals. The problems of signal overlap and signal separation are dealt with by combining information in the time and frequency domain and selecting appropriate points for signal splitting and merging based on changes in signal strength. The resulting signal is then sent to the signal recognition module.

Signal Recognition. A dynamic denoising algorithm removes noise to facilitate the extraction of STFT features, which are then sent to the classification module for the extraction of deeper information using the concept of focal loss to deal with difficult samples. The letters with the highest probability of matching the extracted features are then output to be combined with previous candidate letters until a word interval is detected. The resulting combination of letter candidates is then forwarded to the word suggestion module.

Word Suggestion. The n-gram and edit distance algorithms are used to deduce words with various length. *VibWriter* improve the recognition performance at word level and display the results to the user.

4 System

This section details the overall system, including the letter segmentation, letter recognition, and word suggestion modules.

4.1 Signal Segmentation

As shown in Fig. 1 the amplitude of the vibration signal significantly differs from that of accelerometer-related noise. Thus, our first objective is to compare the amplitude of the signal with that of noise. Unfortunately, data acquisition in real-world situations can lead to a number of issues, such as inconsistent accelerometer sampling intervals, incomplete data segmentation, letter concatenation, and interference from other vibration sources. The proposed segmentation algorithm deals with these issues in two stages: interpolation and detection.

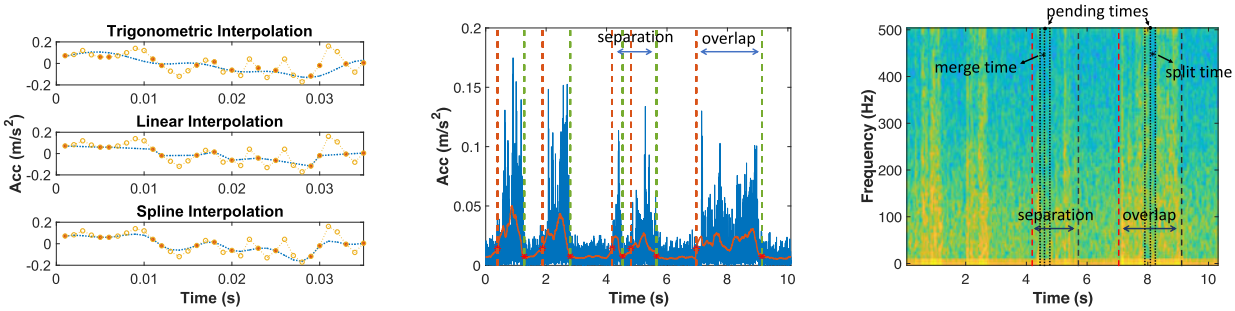
Interpolation. Obtaining the highest sampling rate from the built-in accelerometer precludes the stable sampling rate of raw data [13]. In most situations, more than half of the vibration signals are missing, such that the actual number of samples collected per second is roughly 490.

The accuracy of timestamps is $1ms$. Therefore, the ideal approach would involve upsampling the raw data to $1000Hz$. This linear interpolation approach has previously been used to stabilize the sampling rate [13]. However, when the time interval exceeds $4ms$, the complete cycle of the signal (above $250Hz$) is missing and cannot be recovered via linear interpolation.

Our efforts to resolve the problem of sampling instability led us to compare a variety of methods including those based on interpolation [14] and reconstruction [15]. Methods based on reconstruction infer the composition of the signal within a certain frequency band in accordance with known points under the constraints of the Nyquist Rate. However, the high frequency part of the vibration signal lasts only a few ms , which means that reconstruction-based methods are only able to recover signals that fall within an extremely narrow band due to the limited number of sampling points. Methods based on interpolation use local sampling points to recover the missing data, and are therefore not limited by the length of the signal, resulting in superior real-time performance.

We compare a variety of interpolation algorithms, including spline interpolation, trigonometric interpolation and linear interpolation, as shown in Fig. 3a. Real vibration data were obtained from the accelerometer (Grove ADXL356C) in order to calculate the error corresponding to the three interpolation algorithms. Spline interpolation generates the corresponding quadratic function curve from the derivatives of the sampling points. Therefore, spline interpolation shows more effective than linear interpolation in the recovery of lost data over extended time intervals, and outperforms trigonometric interpolation in terms of how well the recovered signal fits the original data. Furthermore, the mean squared errors of the interpolation algorithms are 0.00468, 0.00585 and 0.00331 respectively.

Spline interpolation uses low-degree polynomials in each interval, and selects polynomial pieces in a manner that ensures a smooth fit when combined. For a unstable sampling signal $q(t)$ and known points $(x_1, y_1), (x_2, y_2)$, the third-order polynomial can be written as follows:



(a) Interpolations for missing data. (b) Detection of letters. (c) Detection of letters.

Fig. 3. Letter segmentation metrics: Fig. 3a Results of various interpolation methods: the red dots are the vibration signals (hollow dots are stable samples, solid dots are unstable samples), the blue lines are the interpolation results; Fig. 3b Vibration signal (blue) and corresponding weighted mean signal (red) showing different writing conditions: normal fast writing, an interruption during fast writing (with an interval during one letter) and continuous writing (without interval between letters). The dotted line indicates the results of segmentation based on amplitude; Fig. 3c Proposed solution to deal with signal separation and overlap caused by interruption and continuous writing.

$$q(t) = (1 - t(x))y_1 + t(x)y_2 + t(x)(1 - t(x))((1 - t(x))a + t(x)b) \tag{1}$$

where

$$t(x) = \frac{x - x_1}{x_2 - x_1}$$

$$a = k_1(x_2 - x_1) - (y_2 - y_1)$$

$$b = -k_2(x_2 - x_1) + (y_2 - y_1)$$

$$k_1 = q^{'}(x_1)$$

$$k_2 = q^{'}(x_2)$$

Detection. Generally, the tap of a pen on the desk surface produces a distinctive vibration pattern indicating the beginning of writing. However, in some situations where the user seeks to write quietly, such as a meeting room, the writing process begins with a swipe. This situation makes it difficult to identify the start of writing. The signal produced by a tap presents an abrupt change in amplitude, whereas the amplitude of the signal produced by a swiping motion grows gradually. The common approach to segmentation often fails to identify vibration signals that begin with a swipe [3], [10]. We calculate the mean value $M(t)$ of the vibration signal $S(t)$ with the sliding window $t_w = 100ms$.

Letter detection is based largely on three time thresholds T_1, T_2 and T_3, and three amplitude thresholds A_1, A_2 and A_3. T_1 and T_2 indicate the minimum and maximum lengths of the letters, whereas T_3 indicates the time interval between words. A_1 and A_2 indicate the maximum and minimum absolute values of $M(t)$, whereas A_3 indicates the minimum absolute value of interference. We use the time threshold to constrain the signal length of letters and words, and the amplitude threshold to judge the begin and end of the signal.

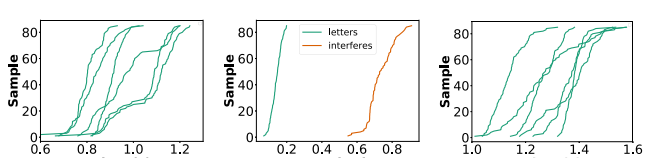
Peak selection is based on the amplitude threshold, where the start threshold is $M_{start} = 0.2 \times A_1 + 0.8 \times A_2$ and the end threshold is $M_{end} = 0.1 \times A_1 + 0.9 \times A_2$.

In instances where the amplitude of $M(t_0)$ exceeds M_{start}, timestamp t_0 indicates the start of a writing segment. As long as the user is writing in a normal manner, it is possible to identify the end of a writing segment based on M_{end}, as shown in Fig. 3b.

As shown in Fig. 4a, preliminary experiments show that the handwriting time remains stable for most users. Therefore, under normal circumstances, it can be assumed that the users write in the block-letter style. However, we observe a number of special situations in which the signal is difficult to segment. In cases where the time interval between letters is short, the vibration signals of different letters can overlap in the time domain, due to the vibration signal lingering for a few milliseconds after writing ceases. Signal separation can also be hindered when the writing process is interrupted and will cause the incomplete segmentation. Besides, there are vibration interferences such as finger tapping on the desk, which can also affect the signal detection.

First, We set $t_{segment}$ as the length of the segment. If $t_{segment} > T_2$, the segment is identified as a combination of two letter signals. T_2 represents the maximum length of a single letter according to our experiment in Fig. 4a. We can locate a candidate split location, based on $Min\{M(t)\}$ in the time domain. As shown in Fig. 1, the high frequency components of the vibration signal are mainly concentrated at the beginning of the signal. Combined with changes in signal strength in the spectrum, we can define the point with the weakest signal strength as the split point, as shown in Fig. 3c.

If $t_{segment} < T_1$, then it is designated a stroke of a letter. T_1 represents the minimum length of a single letter in Fig. 4a. Due to the remaining effect, the simple stitching of two segments is not good choice. Based on the observation of the spectrum above. We can define the point with the weakest signal strength as the merge point, so as to remove the remaining effect of the segment, as shown in Fig. 3c.



(a) Length of letters. (b) Amplitude of signals. (c) Length of intervals.

Fig. 4. Experiments on normal writing patterns in the time and amplitude domains: Fig. 4a Time elapsed while writing letters of different users; Fig. 4b Amplitudes of target signals and interference; Fig. 4c Intervals between words of different users.

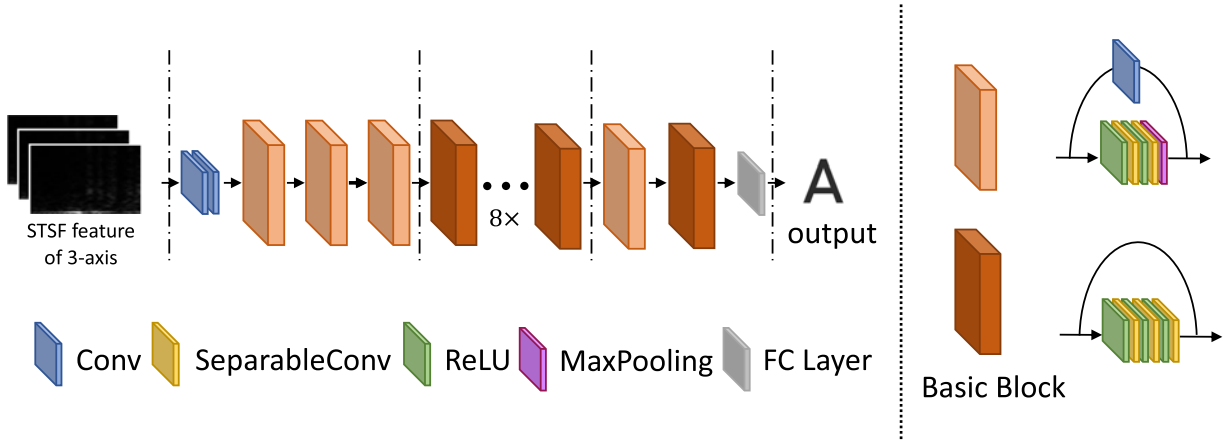


Fig. 5. Architecture of the Xception model.

Then, we set $a_{segment}$ as the maximum amplitude of the segment. If $a_{segment} > A_3$, then it is designated the vibration interferences. A_3 represents the distinguishing threshold between handwriting signal and vibration interference. Since the amplitude of segments differ considerably from the interference according to preliminary experiments, as shown in Fig. 4b. We will discard vibration segments with amplitudes above the threshold. Although this may cause a loss of handwriting signal, this type of interference will not occur frequently. In addition to large vibration disturbances such as knocking on the desk, minor disturbances such as common fans and keyboards on the desk will also affect the system. We can attenuate this type of interference by spectral subtraction. We further analyze these interferences in Section 5.3.

Finally, as shown in Fig. 4c, the length of intervals between words tends to be uniform under normal writing conditions. Thus, intervals exceeding T_3 are designated as the end of a word, and T_3 represents the distinguishing threshold between letters and words.

4.2 Signal Recognition

Preprocessing. We adopt Short-time Fourier Transform (STFT) to generate features in the frequency domain. The vibration signals of the three axes are converted into a STFT matrix representing the magnitude and phase of each frame and frequency, as follows:

$$STFT\{x[t]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{+\infty} x[n]\omega[n-m]e^{-j\omega n} \quad (2)$$

where ω represents the frequency of window function, and m represents the scale of window function.

The sampling rate of the built-in accelerometer ($1kHz$) is far lower than the acoustic signal of handwriting [3], [10], [11], [16], and the spectral distribution of signals and noise is similar. As shown in Fig. 1, the amplitude of noise signals below $100Hz$ far exceeds that of higher frequency signals. Furthermore, signals associated with ambient noise do not remain stable throughout the writing process. Thus, noise removal should be a dynamic process implemented only at specific time points. We develop a dynamic denoising algorithm, which identifies noise based on a reference signal collected during idle periods. We begin by establishing a noise sample $\hat{S}_{noise} = [s_1, s_2, ..., s_l]$, and then update the sample as:

$$\hat{S}_{noise} = \frac{1}{N}\sum_{i=1}^{N} S_{noise_i} \quad (3)$$

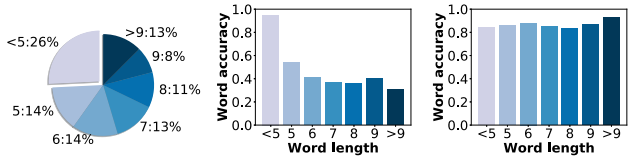
where l indicates the length of the noise sample according to different handwriting segments. S_{noise} preserves the noise signal between letters and words, and N represents the number of samples in S_{noise}. Then, we can denoise the signal with the spectrum subtraction [17]:

$$\|Y(k)\|^2 = \|S_{signal}(k)\|^2 - \|\hat{S}_{noise}(k)\|^2 \quad (4)$$

where k represents the frequency range of the signal, $S_{signal}(k)$ and $\hat{S}_{noise}(k)$ represent the handwriting sample and the noise sample respectively. For each signal, we use the latest noise signal to update the noise sample.

Classification. Convolutional neural network (CNN) have proven highly effective in spectrum classification [3], [10], [18]. The spectral width of vibration signals is far narrower than acoustic signals. Therefore, the module have to extract handwriting features at various scales, (e.g., single taps, single strokes, and entire letters). As shown in Fig. 5, the *Xception* model [19] takes advantages of ResNet [20] and Inception [21]. As the model gets deeper, problems such as gradient disappearance arise. The residual structure in *Xception* (the arcs in the Basic Block) effectively solves this problem and enables features of different depths in the model to be fused. Second, the deepening of the model inevitably increases the computational burden on the hardware. the *Xception* model uses an Separable Convolution (the yellow layer in the Basic Block), which splits the normal convolution into two parts: Channel-wise Convolution and Point-wise Convolution. Channel-wise Convolution extracts features separately for individual channels in the feature, and Point-wise Convolution aggregates the feature points in different channels by 1×1 convolution. Thus, the $n \times n \times m$ parameters (m represents the number of channels) required for ordinary convolution are reduced to $n \times n + m$.

To further improve the accuracy of the model, we employ Focal Loss to facilitate learning using difficult samples as follows [22]:



(a) Word distribution. (b) N-gram algorithm. (c) Edit distance algorithm.

Fig. 6. Word suggestion results: Fig. 6a Distribution of words of various lengths among the 5000 most common words in COCA; Figs. 6b and 6c Accuracy in word identification respectively using N-gram and Edit Distance algorithms.

$$FL(p_t) = -\alpha(1-p_t)^{\gamma} log(p_t) \tag{5}$$

where p_t represents the output of the model, α and γ are correlation coefficients. $\alpha(1-p_t)^{\gamma}$ reverses with the difficulty of sample, so as to strengthen the difficult samples.

Light-Weighted Model. Even though $Xception$ reduces the computational burden of the model by using the volume machine layer, further compression of the model is needed to achieve higher effectiveness. We propose the $Xception\_light$ model based on the $Xception$ model.

To achieve a lightweight classification model while maintaining generalization capabilities, $Xception\_light$ utilizes the idea of knowledge refinement to transfer knowledge from a large and cumbersome model to a small model [23]. In *VibWriter*, the classification network in Fig. 5 can be seen as a trained teacher network that imparts features of different handwriting contents to a small student network, as shown in Fig. 7. Compared with the teacher network, the student network compresses the eight basic modules in the middle of the $Xception$ model into one. As a result, the computational effort of the student network is reduced by more than half compared to the teacher network.

Specifically, the training of the student network consists of two phases. In the first training phase, the teacher network is trained and the features output by the teacher network are extracted, F_t. Then, the student network is trained with the features generated by the teacher network as the target, and the features generated by the student network are written F_s. The loss function in the first phase is the cross entropy of the output features of the two models:

$$L(F_s) = -\sum_i F_t \times log(F_s) \tag{6}$$

In the second training phase, the student network uses handwriting samples to refine the parameters of the network. The loss function still uses Focal Loss.

4.3 Word Suggestion

We notice the fact that users often write a word rather than a single letter. Therefore, we develop a word suggestion algorithm to enhance handwriting recognition performance at the word level.

N-gram Algorithm. Language models are widely used in natural language processing (NLP), such as text categorization[24] and machine translation[25]. We employ the N-gram to determine the probability distribution of letters in words. The chain rule of letters is defined as follows:

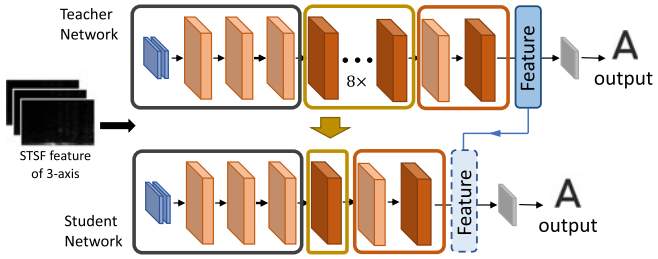


Fig. 7. Architecture of knowledge distillation.

$$\begin{aligned} &P(\omega_1, \omega_2, \ldots, \omega_n) \\ &= P(\omega_1)P(\omega_2|\omega_1)\cdots P(\omega_n|\omega_1, \ldots, \omega_{n-1}) \end{aligned} \tag{7}$$

where $\omega_i, i \in [1, n]$ represents the letter in the word. The conditional probability of each letter occurrence is calculated in terms of maximum likelihood, as follows:

$$P(\omega_i|\omega_1, \ldots, \omega_{i-1}) = \frac{C(\omega_1, \omega_2 ..., \omega_i)}{\sum_{\omega} C(\omega_1, \omega_2 ..., \omega_i, \omega)} \tag{8}$$

where $C(\cdot)$ represents the number of times a string appears in the dataset. Obviously, it would be unrealistic to directly calculate $P(\omega_i|\omega_1, \ldots, \omega_{i-1})$ based directly on maximum likelihood estimation. Assuming that the probability of current letter occurring depends only on the the first $n-1$ letters, we obtain the following result:

$$P(\omega_i|\omega_1, \ldots, \omega_{i-1}) = P(\omega_i|\omega_{i-n+1}, \ldots, \omega_{i-1}) \tag{9}$$

Based on the above formula, the 3-gram language model is defined as follows:

$$P(\omega_i|\omega_1, \ldots, \omega_n) = \prod_{i=1}^{n} P(\omega_i|\omega_{i-1}, \omega_{i-2}) \tag{10}$$

Edit Distance. It can be noted that accuracy in correcting misspelled words is closely related to the lengths of the words. As shown in Fig. 6b, when the length exceeds five letters, the accuracy of word suggestion schemes decreases significantly. Thus, we analyzed the length distribution of the 5000 most commonly used words in the Corpus of Contemporary American English (COCA) in Fig. 6a. The words exceeding 6 letters make up more than half of the total; therefore, we focus on longer words using the edit distance algorithm.

Edit distance refers to the minimum number of editing operations required to change from one string to another [26]. Permitted editing operations include replacing one character with another, inserting one character, and deleting one character. The shortest edit distance between the first i characters of string a and the first j characters of string b can be written as $Lev_{a,b}(i,j)$. The recursive formula used to determine the edit distance between two strings is as follows:

$$Lev_{a,b}(i,j) = \begin{cases} max(i,j) if min(i,j) = 0 \\ min \begin{bmatrix} Lev_{a,b}(i-1,j)+1 \\ Lev_{a,b}(i,j-1)+1 \\ Lev_{a,b}(i-1,j-1) \end{bmatrix} & a_i = b_j \\ min \begin{bmatrix} Lev_{a,b}(i-1,j)+1 \\ Lev_{a,b}(i,j-1)+1 \\ Lev_{a,b}(i-1,j-1)+1 \end{bmatrix} & a_i \neq b_j \end{cases} \tag{11}$$

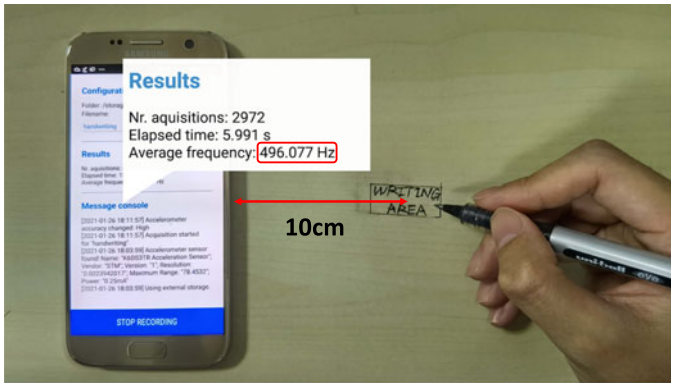


Fig. 8. Experimental Setup.

As shown in Fig. 6c, the edit distance greatly improve accuracy in correcting spelling errors in long words. Thus, we employ the N-gram algorithm for words of less than five letters and edit distance for longer words.

5 Evaluation

5.1 Experimental Setup

Hardware. *VibWriter* is implemented on a Samsung S7 and a MacBook Pro (Intel Core i9 CPU@2.3GHz and 16GB RAM) is implemented as the server. Based on the built-in accelerometer[1], we can achieve a sampling rate of about $490Hz$[13], [27]. We conduct our experiments in the normal laboratory. As shown in Fig. 8, we collect the training set and test set on a wooden desk. The smart phone is placed in the center of the desk, perpendicular to the lower edge of the desk. The writing region is located $10cm$ to the right of the smart phone.

Letters and Numbers. We first invited 10 volunteers to write a sample of 26 uppercase letters and 10 numbers with a gel pen to create the dataset. Two of the volunteers wrote the letters and numbers 80 times each, while the remaining volunteers wrote the letters and numbers 40 times each. All volunteers wrote directly on the table at their own speed, strength and in any order they wished. For each volunteer, we selected 20 samples from the collected letter and number datasets as the test set, and the remaining samples were used as the training set for cross-validation to test the system's letter and number recognition ability.

Words. Volunteers were asked to write a random article (about 100 words) from New Concept English to create a test set of word to test the accuracy of the system's word suggestion.

Parameter. For segmentation, we set the minimum and maximum length of letters $T_1 = 0.4s$, $T_2 = 1.5s$, minimum time of the word interval $T_3 = 1s$ and the minimum absolute value of interferences $A_3 = 0.4$ according to our experimental observation in Fig. 4. We introduce the parameters in details in Section 4.1. For letter recognition, we set the segment and the overlap of STFT at 128 and 120. Finally, we set the Focal Loss coefficients $\alpha = 0.2$ and $\gamma = 3$ [22].

Training. For the teacher network: $Xception$, we set the batch size at 32 for 40 epochs with Adam algorithm (learning rate of 0.0008). For the student network: $Xception\_light$, we set the batch size at 32 for 20 epochs with Adam algorithm (learning rate of 0.0008) for knowledge distillation, and set the batch size at 16 for 20 epochs with Adam algorithm (learning rate of 0.0004) for classification. For each volunteer, we build a handwriting recognition model. We trained the models in three steps. First, we trained the base model using a training set of volunteers who write letters and numbers 60 times. Then, we fine-tuned the base model using the training set of each volunteer to build the teacher model for each volunteer. Finally, we built the student model for each volunteer based on the teacher model using knowledge distillation. Besides, we build the recognition models for letters and numbers separately.

5.2 Micro Benchmarks

In this section, we evaluate the performance of three main components of *VibWriter*.

5.2.1 Signal Segmentation

First, we evaluate the accuracy of the system in terms of letter segmentation, as shown in Fig. 9a. The segmentation algorithm outlined in Section 4.1 prove highly effective in dealing with signal overlap and signal separation. However, there are some cases that fluctuations in the vibration signals are too weak to detect. Those situations are deemed segmentation failures. The average accuracy results in the segmentation of letters, numbers and words were 98.07%, 98.3% and 97.5%, respectively. Overall, this degree of accuracy should suffice for most practical applications.

5.2.2 Signal Recognition

We use the top-1 output of the network as the recognition result. As shown in Figs. 9e and 9f, the average accuracy in letter and number recognition is 75.3% and 79%. Analysis of misclassification reveals that around 20% of the letters "K" and "N" are misidentified as "R" and "V", respectively. Clearly, a word suggestion algorithm is required to achieve reasonable recognition performance.

$Xception\_light$ is compared with other classification methods, including LeNet, AlexNet, ResNet and Inception. As shown in Fig. 9c, the accuracy of $Xception\_light$ is far higher than that of the other classification algorithms. We also seek to reduce the number of training sets via model fine-tuning of the teacher network: $Xception$, as shown in Fig. 9d.

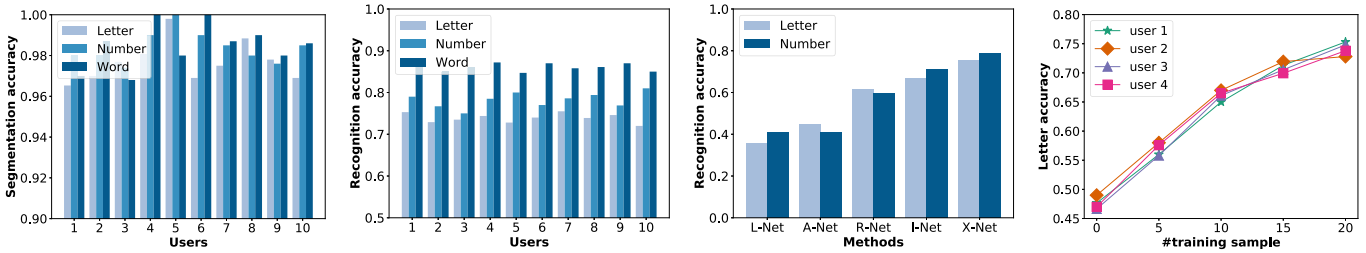
5.2.3 Word Suggestion

The performance of the VibWriter system using the N-gram algorithm for short words and the edit distance algorithms for longer words is verified by counting the number of correct words suggestions. As shown in Fig. 9g, the proposed algorithms achieve overall accuracy of 86.4% for words of various lengths. The inter-user accuracy of the system is shown in Fig. 9b. As shown in Fig. 9h, without the word suggestion algorithms, the average overall accuracy in word recognition is only 12.7%.

5.3 Macro Benchmarks

In this section, we evaluate the performance of *VibWriter* under a variety of conditions. In each experiment, we vary

1. We use a third-party application AccDataRec for diaplay.

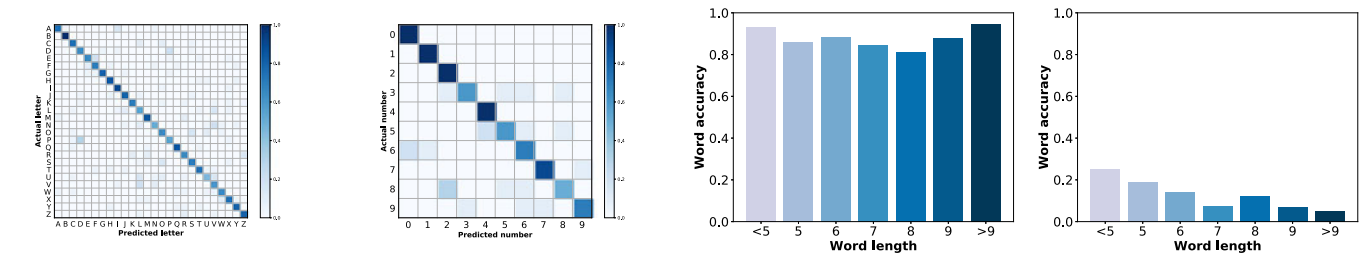


(a) Segmentation accuracy of different users.

(b) Recognition accuracy of different users.

(c) Letter accuracy of different methods.

(d) The effect of #training sample.



(e) Confusion matrix of letter recognition.

(f) Confusion matrix of number recognition.

(g) Accuracy with word suggestion.

(h) Accuracy without word suggestion.

Fig. 9. Accuracy of the VibWriter system.

only one variable, such as the writing distance, writing angle, writing angle, etc.

5.3.1 *Writing Location*

The distance between the smart phone and the handwriting region is experimented by moving the phone in a horizontal direction, in a range of $5cm$ to $120cm$ from the handwriting region. During this process, the angle of the phone is not changed. Then, the phone is then placed back in its original position ($10cm$ to the left of the handwriting region) and the angle of the phone is changed. The volunteers are tasked with writing the same words as test set. Overall, *VibWriter* achieves high accuracy in terms of handwriting recognition regardless of the distance and angles between the writing position and the smart phone, as shown in Figs. 11a and 11b.

5.3.2 *Vibration Interference*

Unlike the interference discussed in Section 4.1, the disturbances of minor vibrations could potentially interfere with *VibWriter*, such as the vibration of the desktop fan, people walking around, tapping on the keyboard, etc. We evaluate each of these Interference separately. The desktop fan is placed on the desk at a distance of $5cm$ from directly above the smart phone to simulate interference from electronic devices. Besides, two volunteers are asked to walk around the desk or tap the keyboard on the desk to simulate the other two interferences. The keyboard is placed $20cm$ to the right of the writing region. Then, the other volunteers are tasked with writing the same words as test set.

As shown in Fig. 11c, vibration interference have an impact on the vibration signal. Nonetheless, the dynamic denoising algorithm (described in Section 4.2) is able to maintain number recognition accuracy (above 69%) and word recognition accuracy (above 75%). Clearly, *VibWriter* is robust to most of the vibration-related interference commonly encountered in the real environments.

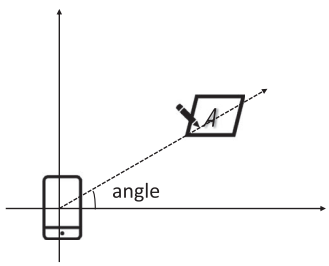


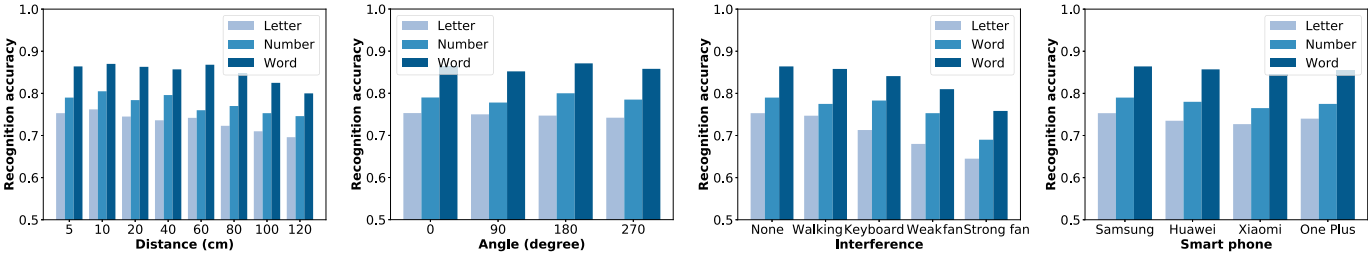
Fig. 10. Angle of the smart phone.

5.3.3 *Different Phones*

Since the size and weight of the smart phone could conceivably affect the vibration signal. Furthermore, sensors vary in terms of sampling rates. We verify the performance of the system application on different phones while keeping the phone position, writing position and other conditions constant. The volunteers are tasked with writing the same words as test set. As shown in Fig. 11d, *VibWriter* work well on a wide variety of smart phones.

5.3.4 *Different Desks*

Different properties of the task may affect the vibration signal, such as the roughness, thickness and size of the desk. Thus, we conduct an experiment in which volunteers are tasked with writing the same words as test set on four smooth and four rough desks. Besides, for the desks with the same roughness, different sizes and thicknesses are chosen for the experiment. We place the smart phone in the center of the different desks, with the writing position and other conditions remaining the same. The results in Fig. 12a indicate that *VibWriter* is applicable to a wide range of desks. The horizontal vibration information of a smooth desk is weak, but they can achieve the number recognition accuracy (above 63.3%) and word recognition accuracy (above 71%).



(a) Influence of different distances. (b) Influence of different angles. (c) Influence of different vibration interferences. (d) Influence of different phones.

Fig. 11. Evaluation of VibWriter under different conditions.

5.3.5 Different Vibration Sources

The vibration signal generated by writing is closely related to the vibration source, such as different pens (include gel pen, pencil and stylus), different medium (include a piece of A4 paper and notebook) and desk material (include wooden, glass and metal). The volunteers are tasked with writing the same word as the test set under different conditions. We verified different pens, medium and desk materials separately. We also kept the phone position, writing distance and other conditions constant.

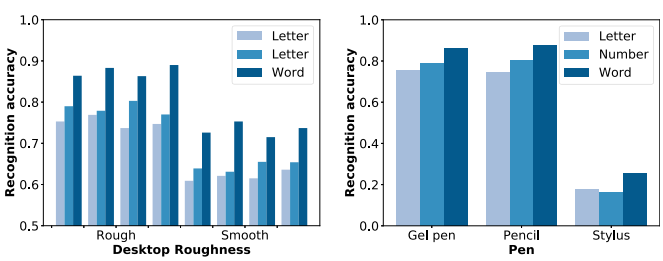
As shown in Fig. 12b, the accuracy of the stylus is significantly lower than that of hard pens, because the vibration signal generated by the softer tip is weak. Therefore, we do not recommend writing with stylus.

Fig. 13a gives the results of different medium, the results show that notebook has worst accuracy of 12% (number) and 14.5% (word). Since the medium between the pen tip and the desk will seriously affect the propagation of the vibration signal, especially when the contact between the medium and the desk is loose or spaced, the vibration signal may be completely isolated.

The different desk materials also affect the recognition accuracy, as shown in Fig. 13b. Wooden desks are usually rough, whereas glass and metal desks are smoother and produce vibration signals of lower amplitude than wooden desks.

5.3.6 Vibration Damping

Placing objects between the writing region and the smart phone inevitably affects the transmission of vibration signals. Thus, we experimentally evaluate the effect of desktop loading on the propagation of vibration signals. We place objects of various weights (but similar contact area) between the writing region and the phone to simulate the different loads on the desktop, such as a pack of tissues ($0.25kg$) and a stack of books ($5kg$). As shown in Fig. 13c, *VibWriter* is basically unaffected by the damping effects of the lightweight objects, and Heavy objects (e.g., books) can effectively block the transmission of signals. Therefore, when interacting with *VibWriter*, users can protect their privacy by placing a heavy object near the writing position.



(a) Influence of different degrees of roughness. (b) Influence of different pens.

Fig. 12. Evaluation of VibWriter under different conditions.

5.3.7 Environment Noise

Acoustic noise in the surrounding environment is a type of vibration signal, which could conceivably affect the signals detected by the accelerometer [27], [28]. Thus, we evaluate the robustness of *VibWriter* to ambient noise. The results in Fig. 13d demonstrate that *VibWriter* is largely unaffected by environmental noise (below $70dB$). Realistic environments such as cafes have a noise level of about $61.5dB$ [11], therefore *VibWriter* can be applied in different realistic environments.

5.4 System Evaluation

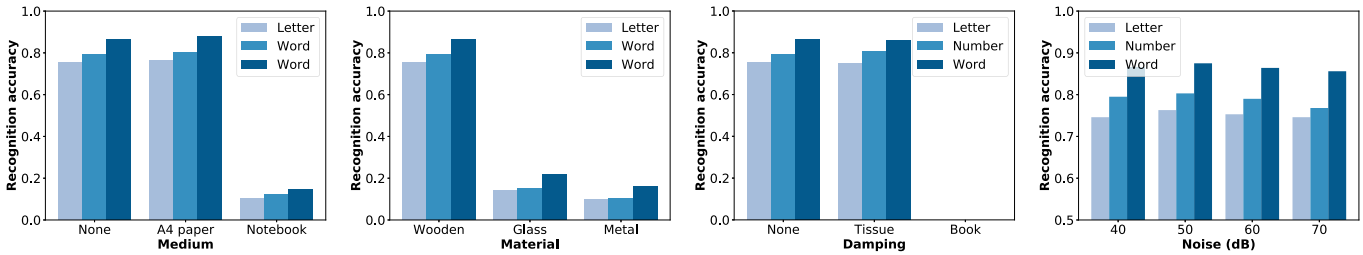
5.4.1 Responsiveness

Latency (delays in system response) is a crucial issue in real-time input systems. In assessing the responsiveness of the overall system, we measure the time that elapsed between receiving a signal and outputting a result. The average latency in recognizing letters and numbers is $54ms$. The average latency in recognizing different words is $239ms$. These results indicate that the responsiveness of *VibWriter* is sufficient for real-time operations.

6 Discussion

Implement on smart watches. Due to hardware limitations, the sampling rate of accelerometers in smart watches is approximately $100Hz$. Coarse-grained data cannot be used to identify the user's writing. While we believe that as smart watches continue to be updated, *VibWriter* can be applied to smart watches and other mobile devices.

Sampling Rate of Smart Phones. Taking an Android phone as an example, setting the highest sample rate ($SENSOR\_DELAY\_FASTEST$) [13] is able to reach a sampling rate of $400Hz$ to $500Hz$. However, high frequency sampling depends on the operating state of the system, and the sampling interval will increase when the system is busy, leading to the problem of unstable sampling rate. Besides, the second highest sampling rate ($SENSOR\_DELAY\_GAME$) has a delay of $20ms$, and the accelerometer has a sampling rate of $50Hz$ and a bandwidth of $25Hz$, which cannot be used to recognize the handwriting letters.



(a) Influence of different medium. (b) Influence of different materials. (c) Influence of vibration damping. (d) Influence of noise.

Fig. 13. Evaluation of VibWriter under different conditions.

7 Related Work

In this section, we discuss the existing representative works. According to different ways of obtaining handwriting information, the existing methods can be roughly divided into three categories:

7.1 Vision-Based Methods

Vision-based methods obtain handwriting inputs in the form of picture and then use machine learning (e.g., convolution neural networks) [29], [30] to perform recognition tasks. The main problem associated with this method is the need to interrupt the writing process to capture the image. *VibWriter* will not burden the user with other operations during the continuous collection process.

7.2 Localization-Based Methods

The main idea of localization-based method is to recover the user's writing trajectory by tracking hand or pen in the space during the writing process. The major approaches ever used are motion-based and wireless signal-based.

Motion-Based Methods. These methods usually need to adopt embedded devices with built-in sensors such as gyroscope and accelerometer. [31] utilized the gyroscope and accelerometer built in the smart watch to track the movement of the user's hand. GyroPen [32] treated smart phones as pens, and the built-in sensors are used to track the user's actions and recognize the handwriting letters. Pentelligence [4] integrated the microphone and accelerometer into an electronic pen, combining the sound of writing with the moving information of the pen to recognize the user's handwriting. DeWristified [33] verified the security of the handwriting recognition system based on the built-in motion sensor of the wearable device.

Wireless Signal-Based Methods. Wireless signal-based methods use wireless signals to sense the movements of the user's hand or pen, such as light, Wi-Fi and magnetic signal. WiReader [2] used Wi-Fi signal to sense the movement of user's hand based on Channel State Information. MagHacker [8] used the magnetic sensor built into smart phones to detect changes in the magnetic field of stylus during the writing process. Acoustic-based tracking methods [5], [6], [7], [34], [35], [36], [37] achieved millimetre-level tracking accuracy, the tracking error can increase as the writing distance increases. [7] showed that the error increases from $5mm$ to $15mm$ while the distance increases from $10cm$ to $40cm$. According to the researches in graphology [9], the medium size of handwriting letter is $2.5-3.5mm$. Therefore, these methods can still impair the recognition accuracy [18], [38]. As a comparison, *VibWriter* uses the built-in accelerometer of the smart phone. During the evaluation, the size of handwriting letters is around $5mm$ and the recognition accuracy remains similar across distances from $10cm$ to $60cm$.

7.3 Scratch-Based Methods

Scratch-based handwriting methods use the acoustic signal caused by the friction during handwriting process [3], [10], [11], [16], [39], [40], [41]. [39] first presented the handwriting recognition system based on the acoustic signals and achieved a recognition accuracy of about 80%. SoundWrite [16] implemented a handwriting recognition system based on acoustic signals on the mobile phone. WordRecorder [10] used the spectrum diagram of the acoustic signals of single letter. WritingRecorder [3] designed the Inception-LSTM module to extract deep local features and time-series relations between frames. Ipanel [11] found that the acoustic signals caused by finger sliding against the desk depend on different movements. However, the scratch-based methods are sensitive to ambient noise, and the recognition accuracy decreases significantly when the noise is above 60dB. Specifically, WordRecorder [10] showed the letter recognition accuracy is reduced by 37% from 79.8% to 50% with 60dB noise (while that of *VibWritter* is 75.3%); WritingRecorder [3] showed the word recognition accuracy is reduced by 19.8% from 92.8% to 74.4% with 65dB noise (while that of *VibWritter* is 86.4%). On the other hand, *VibWriter* is robust against both environmental sound noise and vibration noise.

7.4 Vibration-Based Application

Vibration signals are closely related to daily behaviors, such as walking [42], [43], talking [13], [27], [28] and authentication [44], [45], [46]. FootprintID [42], [43] used the vibration signal of the floor when walking to identify different users. (sp)iPhone [47] used the built-in accelerometer of the smart phone to recognize text entered on a nearby keyboard. Lamphone [28] used The weak vibration of acoustic signal on the bulb to establish the the acoustic signal. Spearphone [27] and paper [13] used the effect of the phone's built-in speaker on the built-in accelerometer to steal the acoustic signal through the vibration signal. SurfaceVibe [12] proposed a vibration-based interaction tracking system for multiple surface types. [44], [45] enabled user authentication by means of user characteristics sensed by vibration signals.

8 Conclusion

This paper introduces a novel handwriting recognition system based on vibration signals. The proposed VibWriter system is able to overcome instabilities in sampling rates and does not require external hardware devices. Extensive experiments demonstrated the efficacy of the system in terms of recognition accuracy in letter (75.3%), number (79%) and word (86.4%) in a variety of positions under a variety of environment conditions.

In future work, we will extend the system to include lowercase letters, and develop a recognition system that runs entirely on the smart phone. Additional methods will be included to improve the recognition accuracy, including sentence-based suggestion and the fusion of vibration signals with other sensors, such as acoustic signals and gyroscope signals. We will also explore the user-independent features in the vibration signal of handwriting to explore the feasibility of handwriting stealing based on the vibration signal.

References

[1] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *CoRR*, vol. abs/1003.4083, 2010. [Online]. Available: http://arxiv.org/abs/1003.4083

[2] Z. Guo, F. Xiao, B. Sheng, H. Fei, and S. Yu, "WiReader: Adaptive air handwriting recognition based on commercial wi-fi signal," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10483–10494, Oct. 2020.

[3] H. Yin, A. Zhou, G. Su, B. Chen, L. Liu, and H. Ma, "Learning to recognize handwriting input with acoustic features," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 2, pp. 1–26, Jun. 2020.

[4] M. Schrapel, M.-L. Stadler, and M. Rohs, "Pentelligence: Combining pen tip motion and writing sounds for handwritten digit recognition," Assoc. Comput. Mach., New York, NY, USA, pp. 1–11, 2018, doi: 10.1145/3173574.3173705.

[5] K. Wu, Q. Yang, B. Yuan, Y. Zou, R. Ruby, and M. Li, "EchoWrite: An acoustic-based finger input system without training," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1789–1803, May 2021.

[6] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl. Serv.*, 2017, pp. 15–28.

[7] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netwo.*, 2016, pp. 82–94.

[8] Y. Liu, K. Huang, X. Song, B. Yang, and W. Gao, "MagHacker: Eavesdropping on stylus pen writing via magnetic sensing from commodity mobile devices," in *Proc. 18th Int. Conf. Mobile Syst., Appl. Serv.*, 2020, pp. 148–160.

[9] Study of Handwriting: Size of Letters in Handwriting. [Online]. Available: https://www.handwriting-graphology.com/study-of-handwriting/

[10] H. Du, P. Li, H. Zhou, W. Gong, G. Luo, and P. Yang, "Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning," in *Proc. IEEE IEEE Conf. Comput. Commun.*, 2018, pp. 1448–1456.

[11] M. Chen *et al.*, "Your table can be an input panel: Acoustic-based device-free interaction recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, pp. 1–21, Mar. 2019.

[12] S. Pan *et al.*, "SurfaceVibe: Vibration-based tap swipe tracking on ubiquitous surfaces," in *Proc. 16th ACM/IEEE Int. Conf. Informat. Process. Sensor Netw.*, 2017, pp. 197–208.

[13] Z. Ba *et al.*, "Learning-based practical smartphone eavesdropping with built-in accelerometer," in *Proc. Annu. Netw. Distrib. Syst. Security Symp.*, 2020.

[14] P. Getreuer, "Linear methods for image interpolation," *Image Process. Line*, vol. 1, pp. 238–259, 2011.

[15] E. Margolis and Y. C. Eldar, "Nonuniform sampling of periodic bandlimited signals," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2728–2745, Jul. 2008.

[16] M. Zhang, P. Yang, C. Tian, L. Shi, S. Tang, and F. Xiao, "SoundWrite: Text input on surfaces through mobile acoustic sensing," in *Proc. 1st Int. Workshop Experiences Des. Implementation Smart Objects*, 2015, pp. 13–17.

[17] A. S. Rathore *et al.*, "SonicPrint: A generally adoptable and secure fingerprint biometrics in smart devices," in *Proc. 18th Int. Conf. Mobile Syst. Appl. Serv.*, 2020, pp. 121–134.

[18] Y. Zhang *et al.*, "Endophasia: Utilizing acoustic-based imaging for issuing contact-free silent speech commands," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–26, Mar. 2020.

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1800–1807.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.

[23] Distilling the knowledge in a neural network. 2015, *arXiv:1503.02531*.

[24] P. Nather, "N-gram based text categorization," Lomonosov Moscow State Univ., Citeseer, 2005.

[25] J. B. Mariño *et al.*, "N-gram-based machine translation," *Comput. Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[26] F. P. Miller, A. F. Vandome, and J. McBrewster, *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*, Alpha Press, 2009.

[27] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers," *CoRR*, vol. abs/1907.05972, 2019. [Online]. Available: http://arxiv.org/abs/1907.05972

[28] B. Nassi, Y. Pirutin, A. Shamir, Y. Elovici, and B. Zadov, "Lamphone: Real-time passive sound recovery from light bulb vibrations," Cryptology ePrint Archive, Tech. Rep. 2020/708, 2020, https://eprint.iacr.org/2020/708.

[29] Z. Li, Q. Wu, Y. Xiao, M. Jin, and H. Lu, "Deep matching network for handwritten chinese character recognition," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107471.

[30] Y. Zheng, B. K. Iwana, and S. Uchida, "Mining the displacement of max-pooling for text recognition," *Pattern Recognit.*, vol. 93, pp. 558–569, 2019.

[31] H. Jiang, "Motion eavesdropper: Smartwatch-based handwriting recognition using deep learning," in *Proc. Int. Conf. Multimodal Interact.*, 2019, pp. 145–153.

[32] T. Deselaers, D. Keysers, J. Hosang, and H. A. Rowley, "GyroPen: Gyroscopes for pen-input with mobile phones," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 2, pp. 263–271, Apr. 2015.

[33] R. Wijewickrama, A. Maiti, and M. Jadliwala, "Dewristified: Handwriting inference using wrist-based motion sensors revisited," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.*, 2019, pp. 49–59.

[34] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "FingerIO: Using active sonar for fine-grained finger tracking," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1515–1525.

[35] W. Mao, J. He, and L. Qiu, "Cat: High-precision acoustic motion tracking," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 69–81.

[36] W. Mao, M. Wang, W. Sun, L. Qiu, S. Pradhan, and Y.-C. Chen, "RNN-based room scale hand motion tracking," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–16.

[37] A. Wang and S. Gollakota, "MilliSonic: Pushing the limits of acoustic motion tracking," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–11.

[38] H. Pan, Y.-C. Chen, Q. Ye, and G. Xue, "Magicinput: Training-free multi-lingual finger input system using data augmentation based on mnists," in *Proc. 20th Int. Conf. Informat. Process. Sensor Netw.*, 2021, pp. 119–131.

[39] W. Li and T. Hammond, "Recognizing text through sound alone," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 1481–1486.

[40] T. Yu, H. Jin, and K. Nahrstedt, "WritingHacker: Audio based eavesdropping of handwriting via mobile devices," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, p. 463–473.

[41] H. Yin, A. Zhou, L. Liu, N. Wang, and H. Ma, "Ubiquitous writer: Robust text input for small mobile devices via acoustic sensing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5285–5296, Jun. 2019.

[42] S. Pan, N. Wang, Y. Qian, I. Velibeyoglu, H. Y. Noh, and P. Zhang, "Indoor person identification through footstep induced structural vibration," in *Proc. 16th Int. Workshop Mobile Comput. Syst. Appl.*, 2015, pp. 81–86.

[43] S. Pan *et al.*, "FootprintID: Indoor pedestrian identification through ambient structural vibration sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–31, Sep. 2017.
[44] X. Xu *et al.*, "TouchPass: Towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–13
[45] J. Liu, C. Wang, Y. Chen, and N. Saxena, "Vibwrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 73–87.
[46] C.-W. You *et al.*, "SoberComm: Using mobile phones to facilitate inter-family communication with alcohol-dependent patients," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2019, pp. 1–31.
[47] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, 2011, pp. 551–562.

Dian Ding received the MS degree from the Department of Automation Science and Electrical Engineering, Beihang University in 2019. He is currently working toward the PhD degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include wireless communication and sensing, cyber security, and human-computer interaction.

Lanqing Yang received the bachelor's degree from the Department of Software Engineering, University of Electronic Science and Technology of China in 2017. He is currently working toward the PhD degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include mobile computing and intelligent sensing.

Yi-Chao Chen (Member, IEEE) received the BS and MS degree from the Department of Computer Science and Information Engineering, National Taiwan University in 2004 and 2006, respectively, and the PhD degree in computer science from the University of Texas at Austin in 2015. In 2018, he joined Shanghai Jiao Tong University (SJTU), as a tenure-track assistant professor with the Department of Computer Science and Engineering. Prior to joining SJTU, he spent a year as a researcher with Huawei Future Network Theory Lab, Hong Kong, and then was a co-founder with Hauoli LLC. His research interests include networked systems and span the areas of wireless networking, network measurement and analytics, and mobile computing.

Guangtao Xue (Member, IEEE) received the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2004. He is currently a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include vehicular ad hoc networks, wireless networks, mobile computing, and distributed computing. He is a member of IEEE Communication Society.