VibWriter: Handwriting Recognition System based on Vibration Signal

***Abstract*—The efficiency of human-computer interactions is greatly hindered by the small size of the touchscreens on mobile devices, such as smart phones and watches. This has prompted widespread interest in handwriting input methods, which can be divided into active and passive methods. Active methods require additional hardware devices to perceive movements of handwriting, such as specific wristband, whereas 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 smart phones. *VibWriter* is highly resistant to interference 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 (76.15%) and word recognition (88.14%) when dealing with words of various lengths written by various users in a variety of writing positions under a variety of environmental conditions.**

***Index Terms*—vibration signal, handwriting recognition**

I. Introduction

The shortcomings of touchscreen input methods have become increasingly obvious with the advent of smart phones, smart watches, and other intelligent devices [1]. Much of the research on alternative input methods has focused on speech recognition [2] and handwriting recognition [3], [4]. Handwriting input is often the only option in cases where privacy is a concern.

Most existing handwriting input methods can be categorized as vision-based, localization-based, and scratch-based methods. The vision based methods [5], require access to examples of the user's writing under suitable lighting conditions, and are ill-suited to extensive writing tasks. Localization-based methods detect the movement of the user's hand or pen via inertial sensors [6], [1] or wireless signals [3], [7]. However, the adoption of these devices is limited by their reliance on external hardware devices. Scratch-based methods [8], [4] involve the detection of acoustic signals generated by dragging a pen across a surface, but these methods are highly susceptible to environmental noise (above 60dB).

In this study, we seek to overcome the shortcomings of existing handwriting recognition schemes by developing a system that uses the built-in accelerometer of smart phones to detect the vibration signals generated by a pen writing on the desk. In experiments, *VibWriter* prove highly robust to interference from environment noise and vibrations. The system also demonstrate outstanding recognition performance when implemented on a variety of phones and desks with the mobile device located at various distances from the writing area.

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

(1) The sampling rate of built-in accelerometers 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.

(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 develope 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:

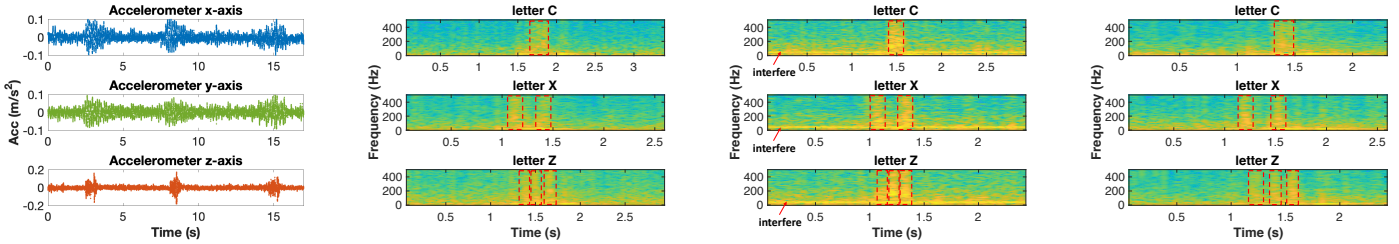
(1) We demonstrate that the accelerometer built into smart phones provides the sensitivity and resolution required for the detection of vibration signals generated by handwriting.

(2) We develope 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 implement *VibWriter* on an Android smart phone. In experiments, the system achieve accuracy of 76.15% in letter recognition and 88.14% in word recognition.

II. 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



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

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

experiments aimed at answering the following fundamental questions: i) Do the vibration signals generated by the desk produce characteristics of different letters? ii) Do the different environments and users affect the vibration signal?

In the first experiment, we seek to determine whether the vibration signals generated by the desk produce characteristics of different letters [9]. One volunteer is tasked with writing the letters "C", "X", and "Z". As shown in Fig.1(a), the exceedingly weak amplitude of the vibrations make it difficult to differentiate between the three letters directly. However, 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.1(b)).

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.1(c), 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 requirements for signal denoising.

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

Preliminary experiments have proved that based on the vibration signal, the user's strokes can be recognized to distinguish writing 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"). A feature extraction scheme of far greater sophistication is required for letter recognition.

III. System

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

A. Letter Segmentation

As shown in Fig.1, our first objective is to compare the amplitude of the signal with the 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 possible sampling rate from the built-in accelerometer precludes the sampling of raw data at a fixed interval [10]. In most situations, more than half of the vibration signals are missing, such that the actual number of sample points 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 [10]. However, when the time interval exceeds $4ms$, the entire signal cycle above $250Hz$ is missing and cannot be recovered via linear interpolation.

We compare a variety of interpolation algorithms [11], including spline interpolation, trigonometric interpolation and linear interpolation, as shown in Fig.4(a). Spline interpolation prove more effective than linear interpolation in the recovery of lost data over extended time intervals, and outperformed trigonometric interpolation in terms of the degree to which the recovered signal fits the raw data.

Spline interpolation uses low-degree polynomials in each interval, and selects polynomial pieces in a manner that ensures a smooth fit when combined. For known points

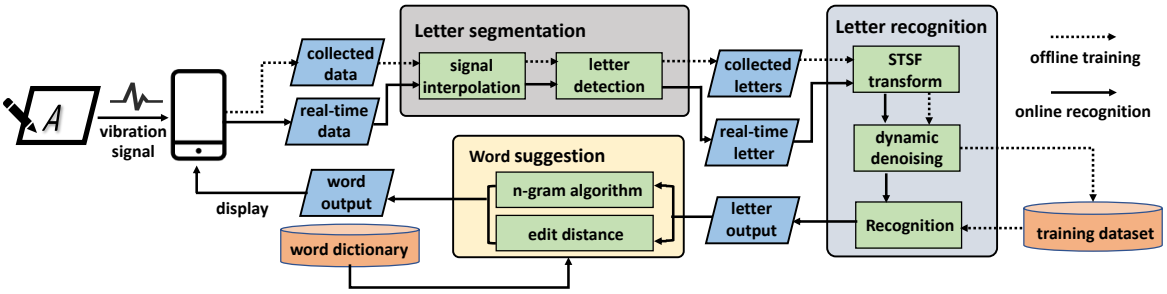


Fig. 2. Overview of *VibWriter*.

$(x_1, y_1), (x_2, y_2)$, the third-order polynomial can be written as follows:

$$q(t) = (1-t(x))y_1 + t(x)y_2 + t(x)(1-t(x))((1-t(x))a + t(x)b) \quad (1)$$

where

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

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

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

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

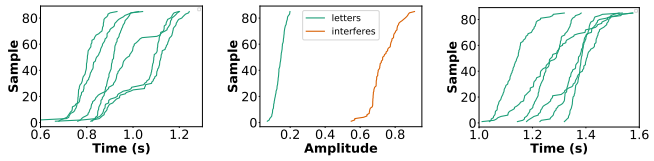
$$k_2 = q^{'}(x_2)$$

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. Those situations make 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 [8], [4]. We calculate the mean value of the vibration signal $S(t)$ from the area covered by 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.4(b).



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

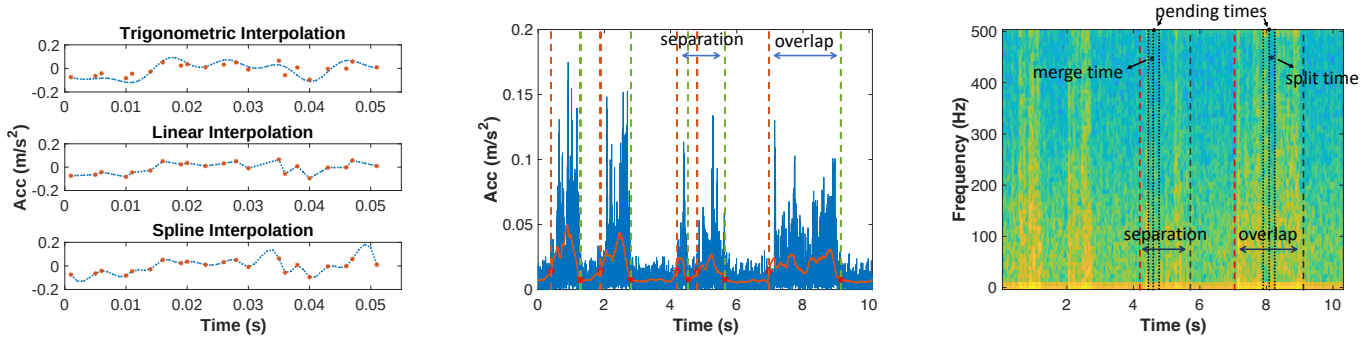
Fig. 3. Experiments on normal writing patterns in the time and amplitude domains: 3(a) Time elapsed while writing letters of different users; 3(b) Amplitudes of target signals and interference; 3(c) Intervals between words of different users.

As shown in Fig.3(a), preliminary experiments reveal that the writing speed of most users remains stable. 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 vibrations lingering for a few milliseconds after writing ceases. Signal separation can also be hindered when the writing process is interrupted and cause the incomplete segmentation. Besides, there are vibration interferes such as finger tapping on the desk, which can also affect 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.3(a). 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.4(c).

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.3(a). 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.4(c).

Then, we set $a_{segment}$ as the maximum amplitude of the



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

Fig. 4. Letter segmentation metrics: 4(a) Results of various interpolation methods; 4(b) 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; 4(c) Proposed solution to deal with signal separation and overlap caused by interruption and continuous writing.

segment. If $a_{segment} > A_3$, then it is designated the vibration interfere. 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.3(b). In addition to large vibration disturbances such as knocking on the desk, minor disturbances such as common fans and arm movement on the desk will also affect the system. We further analyze this type of interference in Section.IV-C.

Finally, as shown in Fig.3(c), 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.

B. Letter Recognition

Preprocessing: We adopt STSF to generate features in the frequency domain. The vibration signals of the three axes are converted into a STSF matrix representing the magnitude and phase of each frame and frequency, as follows:

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

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 [12], [8], [4], [13], and the spectral distribution of signals and noise is similar. As shown in Fig. 1(d), 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 develope 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:

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

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 [14]:

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

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 classifying the spectral characteristics of wireless signals [8], [4]. The spectral width of vibration signals is far narrower than that of acoustic signals and the signals collected by the built-in accelerometer are prone to serious data loss. 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 [15] takes advantages of ResNet [16] and Inception [17] to extract features of various depths. The depth-wise separable convolution layer decomposes the convolution layer into a channel-wise spatial convolution layer and a 1×1 convolution layer to reduce the computational burden associated with most deep neural networks. Besides, the residual structure can fuse features of different depths.

To further improve the accuracy of the model, we employ Focal Loss to facilitate learning using difficult samples as follows [18]:

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

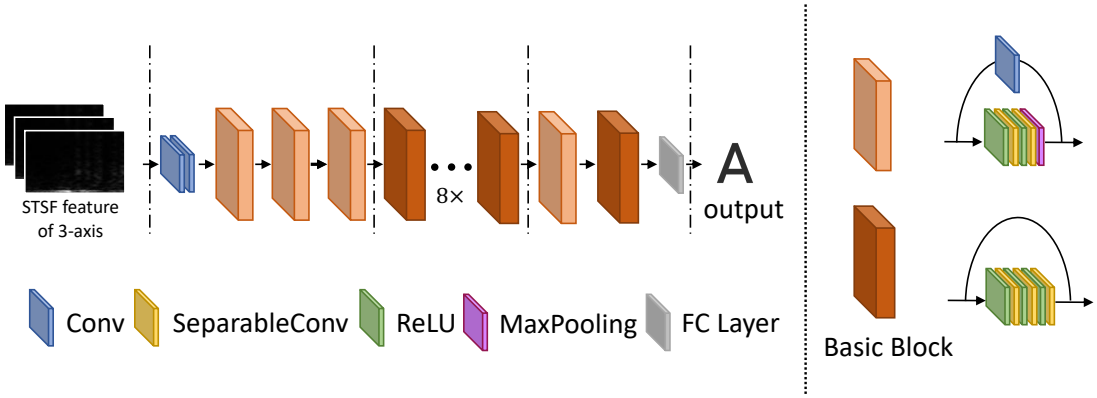


Fig. 5. Architecture of the Xception model.

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

C. Word Suggestion

We notice the fact that users often write words rather than single letters. 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 [19] and machine translation [20]. We employ the N-gram to determine the probability distribution of letters in words. The chain rule of letters is defined as follows:

$$P(\omega_1, \omega_2, ..., \omega_n) = P(\omega_1)P(\omega_2|\omega_1)\cdots P(\omega_n|\omega_1, ..., \omega_{n-1}) \tag{6}$$

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

$$P(\omega_i|\omega_1, ..., \omega_{i-1}) = \frac{C(\omega_1, \omega_2..., \omega_i)}{\sum_\omega C(\omega_1, \omega_2..., \omega_i, \omega)} \tag{7}$$

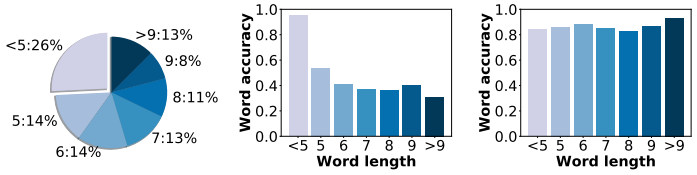
where $C(\cdot)$ represents the number of times a string appears in the dataset. Obviously, it would be unrealistic to directly calculate $P(\omega_i|\omega_1, ..., \omega_{i-1})$ based directly on maximum likelihood estimation. Assuming that the probability of current letter occurring depends only on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

$$P(\omega_i|\omega_1, ..., \omega_{i-1}) = P(\omega_i|\omega_{i-n+1}, ..., \omega_{i-1}) \tag{8}$$

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

$$P(\omega_i|\omega_1, ..., \omega_n) = \prod_{i=1}^{n} P(\omega_i|\omega_{i-1}, \omega_{i-2}) \tag{9}$$

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



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

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

Edit distance refers to the minimum number of editing operations required to change from one string to another. 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:

$$Lev_{a,b}(i,j) = \begin{cases} max(i,j) & if min(i,j) = 0 \\ min \begin{bmatrix} Lev_{a,b}(i-1,j)+1 \\ Lev_{a,b}(i,j-1)+1 \\ Lev_{a,b}(i-1,j-1) \end{bmatrix} & a_i = b_j \\ min \begin{bmatrix} Lev_{a,b}(i-1,j)+1 \\ Lev_{a,b}(i,j-1)+1 \\ Lev_{a,b}(i-1,j-1)+1 \end{bmatrix} & a_i \neq b_j \end{cases} \tag{10}$$

As shown in Fig.6(c), 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.

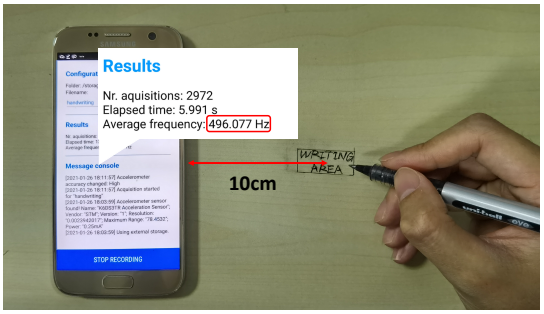


Fig. 7. Experimental Setup.

IV. Evaluation

A. 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 $490Hz$ [21], [10].

Training set. We first invite six volunteers to write samples of 24 uppercase letters with a gel pen as a training set. Two of the volunteers write the 24 letters 60 times each, whereas the rest of the volunteers write the letters 20 times each. All the volunteers write directly on the desk at their own speeds, strength and in any order they wished. As shown in Fig.7, during this phase, the mobile can be placed directly on the desk at any angle, and writing area is fixed at 10cm on the right side of the smart phone.

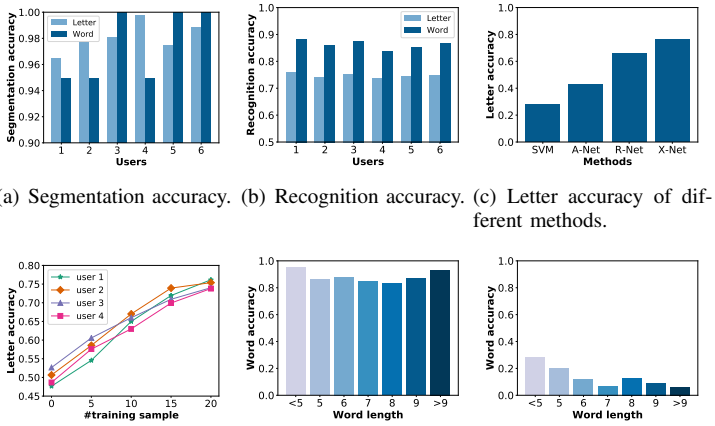
Testing set. The volunteers are then tasked with writing the top 20 words of each length in COCA to test the overall accuracy of the system. In this phase, for each user, the location of the mobile phone is consistent with the collection process of the training set.

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 interferes $A_3 = 0.4$ according to our experimental observation in Fig.3. We introduce the parameters in details in Section.III-A. For letter recognition, we set the segment and the overlap of STSF at 128 and 120. For Xception, we set the batch size at 32 for 40 epoches. We also used the Adam algorithm with a learning rate of 0.0008. Finally, we set the Focal Loss coefficients $\alpha = 0.2$ and $\gamma = 3$ [18].

B. Micro Benchmarks

In this section, we evaluate the performance of three main components of *VibWriter*. For each volunteer, we build a handwriting recognition model. For the volunteers who write letters for 60 times, we can build the model with their own handwriting samples. For the rest volunteers, we use the handwriting samples of two volunteers to build the basic model, and then fine-tune the model with their own samples.

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



(a) Segmentation accuracy. (b) Recognition accuracy. (c) Letter accuracy of different methods.

(d) The effect of #training sample. (e) Accuracy with word suggestion. (f) Accuracy without word suggestion.

Fig. 8. Accuracy of the VibWriter system.

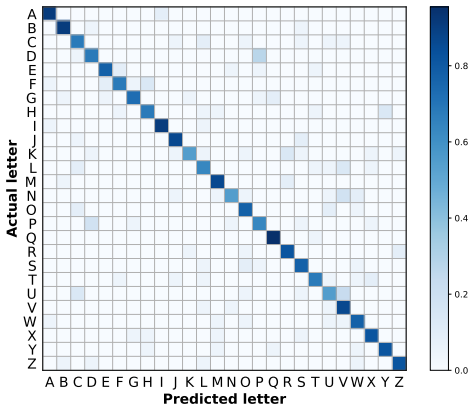
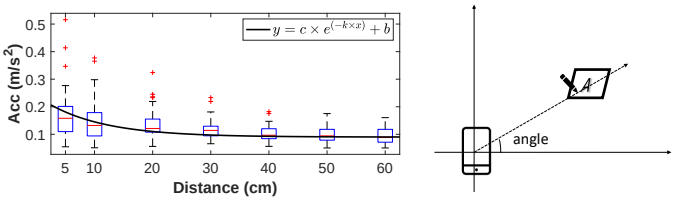


Fig. 9. Confusion matrix of letter recognition.

1) Letter Segmentation: First, we evaluate the accuracy of the system in terms of letter segmentation, as shown in Fig.8(a). The segmentation algorithm outlined in Section.III-A 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 and words were 98.07% and 97.5%, respectively. Overall, this degree of accuracy should suffice for most practical applications.

2) Letter Recognition: We use the top-1 output of the network as the recognition result. As shown in Fig.9, the average accuracy in letter recognition is 75.69%. Analysis of misclassifications reveal 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 is compared with other classification methods, including Support Vector Machine (SVM) and other CNN methods (AlexNet and ResNet18). As shown in Fig.8(c), the accuracy of Xception is far higher than that of the other classification algorithms. We also seek to reduce the number of training sets via model fine-tuning, as shown in Fig.8(d).



(a) Vibration attenuation as a function of the distance.

(b) Angle of the phone.

Fig. 10. Different phone locations.

TABLE I
INFLUENCE OF WRITING DISTANCE.

Distance	Letter Accuracy	Word Accuracy
5cm	76.15%	88.14%
10cm	76.92%	88.33%
20cm	74.03%	85.83%
40cm	74.61%	87.5%
60cm	75.19%	86.67%

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.8(e), the proposed algorithms achieved overall accuracy of 88.14% for words of various lengths. The inter-user accuracy of the system is shown in Fig.8(b). As shown in Fig.8(f), without the word suggestion algorithms, the average overall accuracy in word recognition is only 13.57%.

C. Macro Benchmarks

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

1) Writing Location: A preliminary experiment is conducted to model the signal attenuation imposed by the desk. As shown in Fig.10(a), 200 taps are respectively recorded at various distances ranging from $5cm$ to $60cm$. The attenuation of vibration signals as a function of distance is calculated as:

$$P(x + \Delta x) = P(x)e^{-\alpha(\omega)\Delta x} \tag{11}$$

where $\alpha(\omega) = \alpha_0\omega^\eta$ and ω refers to the frequency of the signals. The experiment results reveal that the attenuation of vibration signals is gradual over these distances.

The volunteers are also tasked with writing the same words as testing set at various distances ranging from $5cm$ to $60cm$, and different angles, as shown in Fig.10(b). Overall, *VibWriter* achieved high accuracy in terms of handwriting recognition regardless of the distance and angles between the writing position and mobile device, as shown in Tab.I and Tab.II.

2) Vibration Interference: Unlike the interference discussed in Section.III-A, the disturbances of minor vibrations could potentially interfere with *VibWriter*, such as the vibration of the desktop fan, the slight shaking of the human body and the movement around the table. We use desktop fans to simulate this interference. In the experiment, the fan is placed on the desk at a distance of $5cm$ from the smart phone. Volunteers are then tasked with writing the same words as testing set while the fan is operating at various speeds.

TABLE II
INFLUENCE OF WRITING ANGLE.

Angle	Letter Accuracy	Word Accuracy
0°	76.15%	88.14%
90°	74.5%	87.2%
180°	75%	86.43%
270°	74.42%	84.83%

TABLE III
INFLUENCE OF VIBRATION INTERFERENCE.

Fan State	SNR	Letter Accuracy	Word Accuracy
Close	13.24	76.15%	88.14%
Weak	10.91	69.23%	80.83%
Strong	6.63	64.04%	75.83%

As shown in Tab.III, increasing the power of the fan noticeably increase the signal to noise ratio (SNR). Nonetheless, the dynamic denoising algorithm (described in Section. III-B) is able to maintain recognition accuracy above 64%. Clearly, *VibWriter* is robust to most of the vibration-related interference commonly encountered in the work environment.

3) Different Phones: The volunteers are tasked with writing the same words as testing set with different smart phones. Since the size and weight of the phone could conceivably affect the vibration signal. Furthermore, sensors vary in terms of sampling rates. As shown in Tab.IV, *VibWriter* work well on a wide variety of smart phones.

4) Different Desks: Different properties of the tesk may affect the vibration signal, such as the roughness, thickness and size of the table. Thus, we conduct an experiment in which volunteers are tasked with writing the same words as testing set on four smooth and four rough desks. The results in Tab.V 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 word recognition accuracy above 70%.

5) Different writing conditions: The vibration signal generated by writing is closely related to the vibration source, such as different pens and different medium. The volunteers are tasked with writing the same word as the testing set under different conditions. Different pens include gel pen, pencil and stylus, while different medium include a piece of A4 paper and notebook.

As shown in Tab.VI, 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.

Tab.VII gives the results of different medium, the results show that notebook has worst accuracy of 14.16%. 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.

6) Environment Noise: Acoustic noise in the surrounding environment is a type of vibration signal, which could

TABLE IV
INFLUENCE OF PHONE TYPES.

Phone	Sampling Rate	Letter Accuracy	Word Accuracy
Samsung S7	495	76.15%	88.14%
HUAWEI P40	495	72.5%	85.83%
Xiaomi Max	475	73.84%	84.17%
One Plus 7pro	450	75%	87.5%

TABLE V
INFLUENCE OF DESKS.

Desk	Property	Letter Accuracy	Word Accuracy
Rough	Big and thick	76.15%	88.14%
Rough	Big and thin	77.69%	90%
Rough	Small and thick	74.23%	88.33%
Rough	Small and thin	75.77%	86.43%
Smooth	Big and thick	60.19%	71.66%
Smooth	Big and thin	62.11%	73.33%
Smooth	Small and thick	62.5%	75%
Smooth	Small and thin	63.46%	73.57%

conceivably affect the signals detected by the accelerometer [21], [22]. Thus, we evaluate the robustness of *VibWriter* to ambient noise by assessing handwriting recognition accuracy as a function of signal to noise ratio (SNR). The results in Tab.VIII demonstrate that *VibWriter* is largely unaffected by environmental noise.

D. System Evaluation

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 520 letters is $165ms$. The average latency in recognizing 120 words is $239ms$. These results indicate that the responsiveness of *VibWriter* is sufficient for real-time operations.

2) User Study: A survey is conducted to collect feedback from the volunteers in terms of accuracy, input speed, responsiveness, and security. Scores are based on satisfaction with 5 points ranging from very unsatisfied (1) to very satisfied (5).

As shown in Tab.IX, more than 85% of the volunteers express satisfaction with the system in terms of accuracy, input speed, and system security, whereas 75% are satisfied with the responsiveness of the system. Some of the volunteers comment that *VibWriter* is less susceptible to eavesdropping than conventional touchscreen input methods.

V. RELATED WORK

A. Vision-based Methods

Vision-based methods obtain handwriting inputs in the form of picture and then use machine learning (e.g., convolution neural networks) [5], [23] 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.

TABLE VI
INFLUENCE OF DIFFERENT PENS.

Pen	Letter Accuracy	Word Accuracy
Gel pen	76.15%	88.14%
Pencil	75.1%	87.5%
Stylus	15.38%	20.83%

TABLE VII
INFLUENCE OF DIFFERENT MEDIUM.

Medium	Letter Accuracy	Word Accuracy
Nothing	76.15%	88.14%
A4 paper	75.1%	87.5%
Notebook	11.53%	14.16%

B. 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. Paper [24] propose air-writing recognition system, which can obtain the user's action information from sensors in special hardware. Besides, paper [6] utilize the gyroscope and accelerometer built in the smart watch to track the movement of the user's hand. GyroPen [25] treats smart phones as pens, and the built-in sensors are used to track the user's actions and recognize the written contents. Pentelligence [1] integrates 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.

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 [3] uses Wi-Fi signal to sense the movement of user's hand based on Channel State Information(CSI). MagHacker [7] uses the magnetic sensor built into smart phones to detect changes in the magnetic field of stylus during the writing process. *VibWriter* uses the built-in accelerometer of the smart phone to sense the vibration of desk, without the need for additional hardware devices. Besides, *VibWriter* is also superior in accuracy to handwriting input methods based on smart watches, which is 71.9% in word recognition [6].

C. Scratch-based Methods

Scratch-based handwriting methods use the acoustic signal caused by the friction during handwriting process. WordRecorder [8] used the spectrum diagram of the acoustic signals of single letter, and designed a deep neural network based on LeNet and AlexNet. Paper [4] designed the Inception-LSTM module to extract deep local features and time-series relations between frames. Ipanel [12] found that the acoustic signals caused by finger sliding against the desk depend on different movements. However, all scratch-based recognition methods require a relatively quiet environment in order to

TABLE VIII
INFLUENCE OF ENVIRONMENT NOISE.

Noise	SNR	Letter Accuracy	Word Accuracy
40 dB	14.17	75%	89.16%
50 dB	14.36	76.53%	87.5%
60 dB	13.24	76.15%	88.14%
70 dB	13.61	75%	86.66%

TABLE IX
USER SATISFACTION OF *VibWriter*.

Satisfaction	Accuracy	Speed	Delay	Security
Very Satisfied	8	7	5	10
Satisfied	9	11	10	8
Normal	3	2	5	2
Unsatisfied	0	0	0	0
Very Unsatisfied	0	0	0	0

achieve suitable accuracy. When the noise decibel reaches 65dB, the accuracy of this type of method will be obviously drop to 74.4% in word recognition [4]. *VibWriter* uses the vibration information of the desk to recognize handwriting, so it has higher robustness to environmental noise.

VI. DISCUSSION AND 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 accuracy in letter detection (76.15%) and word detection (88.14%) when dealing with words of various lengths written by various users in a variety of positions under a variety of environment conditions.

In future work, we will extend the system to include lowercase letters. We are also developing a module aimed at reducing the amount of data required for training. Additional methods will be included to improve the recognition accuracy, including sentence-based suggestion and the fusion of vibration signals with information related to the movement of the user's wrist while writing.

REFERENCES

[1] M. Schrapel, M.-L. Stadler, and M. Rohs, "Pentelligence: Combining pen tip motion and writing sounds for handwritten digit recognition," 04 2018, pp. 1–11.

[2] 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

[3] Z. Guo, F. Xiao, B. Sheng, H. Fei, and S. Yu, "Wireader: Adaptive air handwriting recognition based on commercial wi-fi signal," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[4] 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, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3397334

[5] Z. Li, Q. Wu, Y. Xiao, M. Jin, and H. Lu, "Deep matching network for handwritten chinese character recognition," *Pattern Recognition*, vol. 107, p. 107471, 2020.

[6] H. Jiang, "Motion eavesdropper: Smartwatch-based handwriting recognition using deep learning," in *2019 International Conference on Multimodal Interaction*, ser. ICMI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 145–153.

[7] 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 *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 148–160.

[8] H. Du, P. Li, H. Zhou, W. Gong, G. Luo, and P. Yang, "Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1448–1456.

[9] S. Pan, C. G. Ramirez, M. Mirshekari, J. Fagert, A. J. Chung, C. C. Hu, J. P. Shen, H. Y. Noh, and P. Zhang, "Surfacevibe: Vibration-based tap swipe tracking on ubiquitous surfaces," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017, pp. 197–208.

[10] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-based practical smartphone eavesdropping with built-in accelerometer," 01 2020.

[11] P. Getreuer, "Linear Methods for Image Interpolation," *Image Processing On Line*, vol. 1, pp. 238–259, 2011.

[12] M. Chen, P. Yang, J. Xiong, M. Zhang, Y. Lee, C. Xiang, and C. Tian, "Your table can be an input panel: Acoustic-based device-free interaction recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3314390

[13] M. Zhang, P. Yang, C. Tian, L. Shi, S. Tang, and F. Xiao, "Soundwrite: Text input on surfaces through mobile acoustic sensing," in *Proceedings of the 1st International Workshop on Experiences with the Design and Implementation of Smart Objects*, ser. SmartObjects '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 13–17. [Online]. Available: https://doi.org/10.1145/2797044.2797045

[14] A. S. Rathore, W. Zhu, A. Daiyan, C. Xu, K. Wang, F. Lin, K. Ren, and W. Xu, "Sonicprint: A generally adoptable and secure fingerprint biometrics in smart devices," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 121–134.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[19] P. Nather, "N-gram based text categorization," 2005.

[20] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà, "N-gram-based machine translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[21] 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

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

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

[24] M. Chen, G. AlRegib, and B. Juang, "Air-writing recognition—part i: Modeling and recognition of characters, words, and connecting motions," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 403–413, 2016.

[25] T. Deselaers, D. Keysers, J. Hosang, and H. A. Rowley, "Gyropen: Gyroscopes for pen-input with mobile phones," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 2, pp. 263–271, 2015.