RECOGNITION OF HUMAN COMPUTER OPERATIONS BASED ON KEYSTROKE SENSING BY SMARTPHONE MICROPHONE

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

EARNEST GEORGE

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

RECOGNITION OF HUMAN COMPUTER OPERATIONS BASED ON KEYSTROKE SENSING BY SMARTPHONE MICROPHONE

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

EARNEST GEORGE

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Recognition Of Human Computer Operations Based On Keystroke Sensing By Smartphone Microphone** *submitted by* **Mr. EARNEST GEORGE**, *Reg.No.***MAC15CS026** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

....	....	....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head of the Department***

Date: Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the God Almighty for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

Human computer operations such as writing documents and playing games have become popular in our daily lives. These activities can be used to facilitate context-aware services. The recognization of human computer operations is done through keystroke sensing with a smartphone . First the microphone embedded in a smartphone is utilized to sense the input audio from a computer keyboard. Keystrokes is identified using fingerprint identification techniques. The determined keystrokes are then corrected with a word recognition procedure, which utilizes the relations of adjacent letters in a word. By fusing both semantic and acoustic features, a classification model is constructed to recognize four typical human computer operations chatting , coding, writing documents and playing games. Evaluation of the proposed approach was done from multiple aspects in realistic environments. Experimental results validated the effectiveness of this approach.

Contents

List of Figures

LIST OF ABBREVIATION

ZCR	Zero Crossing Rate
NRNG	N gram distance
NLP	Natural Language Processing
FFT	Fast Fourier Transform
DV	Dice Coefficient vector
SVM	Support Vector Machine
NB	NaiveBayes

Introduction

Human computer operations such as document process- ing, spreadsheets, programming , and digital games, have become increasingly involved in our everyday life. These activities reflect interactions between the user and the computer, which can be exploited to facilitate context-aware services. In particular, human computer operations may reflect ones real time status as well as specific desire or requirements. For example, if a user is working on a document or coding, it is very likely that she does not want to be disturbed, and the mode of her phone should be set to the silent mode. In contrast, when a user is chatting or playing games, she may want to be in a relaxed atmosphere, and light music may be recommended to her. Based on the users operations,we can adapt the environment to different context, such as adjusting the volume of the phone ringing, or the light intensity in the room. Further, we can characterize the user with more information by using her operations on computer, e.g., a computer game enthusiast or a hardworking person, which is also helpful to provide personalized services. In addition,humans are usually absorbed in one computer operation and do not notice the time. To keep a healthy lifestyle, sensing human computer operation can help to set a personal reminder for taking a break or a new task. Moreover, recognizing these operations can be used to generate a personalized activity schedule and properly manage time for the user. Therefore, aiming at enabling all these context-aware services in a pervasive and non-intrusive manner, we propose to recognize human computer operations based on keystroke sensing with smartphones.

In general, human computer operations can be recognized by installing a keylogger on the computer. However, pro- viding personalized services via a computer (e.g., adjusting the light intensity) is not very convenient compared with using a smartphone or other wearable devices. Specifically, prior work has extensively studied human activity recognition and context-aware sensing by utilizing smartphones, ambient sensors, and wearable devices.For example, by using ambient sensors or wearable devices, prior research that mainly focused on characterizing a users physical activities and body movements, was able to recognize the refrigerator opening,eating, etc. With multiple sensors embedded, a smartphone can also be used to recognize a users activities, e.g., brushing teeth, running, and computer keystrokes. Moreover, a smartphone can be a control and monitoring system in smart homes.In this work, we aim to recognize human computer operations by using a smartphone. In particular, we do not adopt the technologies of ambient sensors or wearable devices mainly due to the following factors. The deployment of ambient sensors can be labor intensive and costly, and wearable devices are not always convenient or available. In addition, there exists some work related to recognition of human computer operations by using smartphones. For instance, keystrokes can be identified using the audio features of each keystroke on the keyboard. However, we cannot determine the type of operation only based on individual keys. Building on the existing studies, we further explore the feasibility of using smartphones for human computer operation recognition.

In this work, we capture a users keyboard inputs and recognize human computer operations by utilizing a single microphone in smartphones. We face several interesting chal- lenges. First, there are various kinds of human computer operations (e.g., writing documents or coding) that are rich in semantics, but the data collected using smartphone sensors is quite limited and may not show much difference between different operations. Therefore, obtaining the semantic information of human computer operation based on sensor data is essential yet challenging. Second, distinct features that can be used for recognition of human computer operations are still to be explored.

To tackle these challenges, we sense the content of a users keyboard input, and recognize four human computer operations, i.e., coding, writing documents, chatting, and playing games, which are commonly seen at work and entertainment. The effective recognition inspires our future exploration taking more human computer operations into consideration. This paper extends our previous work by optimizing the relations of keys for word correction, applying different similarity metrics to distinguish human computer operations, comparing the effectiveness with the existing work, conducting more experiments and evaluating our method from multiple aspects.In addition, we discuss the availability of the method. In summary, the contributions of our work are three fold.

- We capture the audio signals from keyboard input by utilizing the single microphone embedded in most commod- ity smartphone, and recognize the input words. Specifically, the keystrokes are first identified by the extracted acoustic features in frequency domain. To further correct the identified words, we then apply the methods in natural language processing (NLP) to obtain N-Gram based candidate word set, and select the right word from it by proposing an adjacent similarity matrix algorithm.

- We extract the features on semantics and audio signal, and further employ the AdaBoost algorithm for users com- puter operation recognition, i.e., chatting, coding, writing documents, and playing games. In particular, based on the corrected words, we propose lexicon preference and semantic rationality to characterize each human computer operation in semantics. Meanwhile, based on the input audio signals, input rate and audio energy are also used for human computer operation recognition.

- We conduct experiments in realistic environments to vali- date the effectiveness of the proposed method. Specifical- ly, we consider two types of keyboards and input modes.

Systematic analysis has been conducted and presented on the performance of the proposed features and algorithms.

Existing methods

Our work is broadly related to several areas of research:human computer operation recognition, human activity identification using smartphones, and keystroke recognition.

2.1 Human computer operation recognition

Human computer operations record what a user has done on the computer, which has been explored by existing studies and software. For instance, data from computer games and psychomotor measurements associated with keyboard entries and mouse movement can be monitored to infer a users cognitive performance. Meanwhile, discovering usage pattern from Web data is also commonly used to understand and better serve the needs of Web-based applications. These studies have mainly focused on the understanding of one type of human computer operations, whereas there is not a comprehensive study for users various activities on the computer. In addition, ActivTrak1 monitors peoples online activities to ensure greater productivity and safety. It workson a computer and collects application title bars, page titles,URLs, and screen shots. This type of application cannot provide more context-aware services in a pervasive and non-intrusive manner for the users, e.g., adjusting the mode of the phone ringing or the light intensity of the room. As many powerful sensors are now embedded in the smartphone, we want to monitor a users various computer operations by utilizing the smartphone, which can facilitate the development of personalized services, e.g., personal reminder, phone mode setting, indoor light adjustment.

2.2 Human activity identification using smartphones

The fast development of sensor-enriched smartphones provides an opportunity that human activity can be sensed by smartphones, ranging from personal movements to crowd behaviors. For example, accelerometer embedded in smartphones is capable of characterizing humans movements, e.g., standing, walking, running.Meanwhile, the microphone has been widely used to sense a users activity. For instance, by collecting the audio from the phones microphone, it is possible to recognize a users activity, e.g., listening to music, speaking, sleeping. By using accelerometer and microphone in a smartphone, a system can monitor running rhythm to help users better understand their running process. By using the built-in microphone on the smartphone, SymDetector detects the sound-related respiratory symptoms of a user, such as sneeze or cough. These studies offer related techniques and algorithms on smartphone sensing, which support and inspire our work. In our prior work, we proposed a method for human computer operation recognition by using the microphone of smartphones. In this paper, we address the limitations in our prior method by refining the relations of keys for word correction.Meanwhile, we use two similarity metrics, i.e., cosine similarity and Tonimoto coefficient, to measure the distance of different human computer operations. More importantly, we conduct realistic experiments and evaluate the method from multiple aspects.

2.3 Keystroke recognition

Keyboard input recognition is the basis of our proposed human computer operation recognition. There are some studies on keystroke identification by using sound-wave range measurement and wireless signals. In particular,the method of sound-wave range measurement requires two or three smartphones, and the technique of wireless signal sensing has stringent requirements for the environment. We hence do not use any of the two methods in our work.

In addition, the acoustic signal generated by each keystroke is unique, hence can be regarded as fingerprint features.By utilizing these observations, keystrokes were recognized with a supervised algorithm. Trading off training needs for accuracy through a dictionary-based approach was investigated. Moreover, combined with the acoustic model of Multipath Fading, the keystroke hit in a solid surface was also recognized . Using the insights gained from these studies,the keyboard input can be effectively recognized by utilizing a microphone built in a smartphone based on the fingerprint features of the keystroke audio. Different from existing studies,we extract 400 features in frequency domain for the keystroke identification. Also, prior research only studied the keystroke recognition, whereas we aim to recognize human computer operations, which cannot be achieved only by using separate keystrokes identified.

Proposed method

With the single microphone in a commodity smartphone, we record the acoustic signal from keystrokes and recognize the users computer operations. The proposed framework consists of three major components: keystroke identification, word correction, and human computer operation recognition, as shown in Figure.

We first pre-process the acoustic signals from keystrokes by denoising and key separation. We then identify the keystrokes by extracting features in frequency domain and correct the input words. Specifically, based on the N-Gram model in NLP, we propose non-repeating word segmentation based N-Gram distance to select the candidate word set. Utilizing the characteristics of audio signals, we design an algorithm based on adjacent similarity matrix to recognize the right word as shown in figure 3.1. In particular, the position relations of the letters in a word are redefined to obtain the constraint set, and then the constraint subsets method is applied to eliminate the effects of error conditions. After that, we extract the characteristics on semantics and audio signals, and further acquire the activity segment. The AdaBoost algorithm is then applied to recognize the users computer operations.

AdaBoost: AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire, who won the 2003 Gdel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost[1] is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Every learning algorithm tends to suit some problem types better than others, and typically has many different parameters and configurations to adjust before it achieves optimal

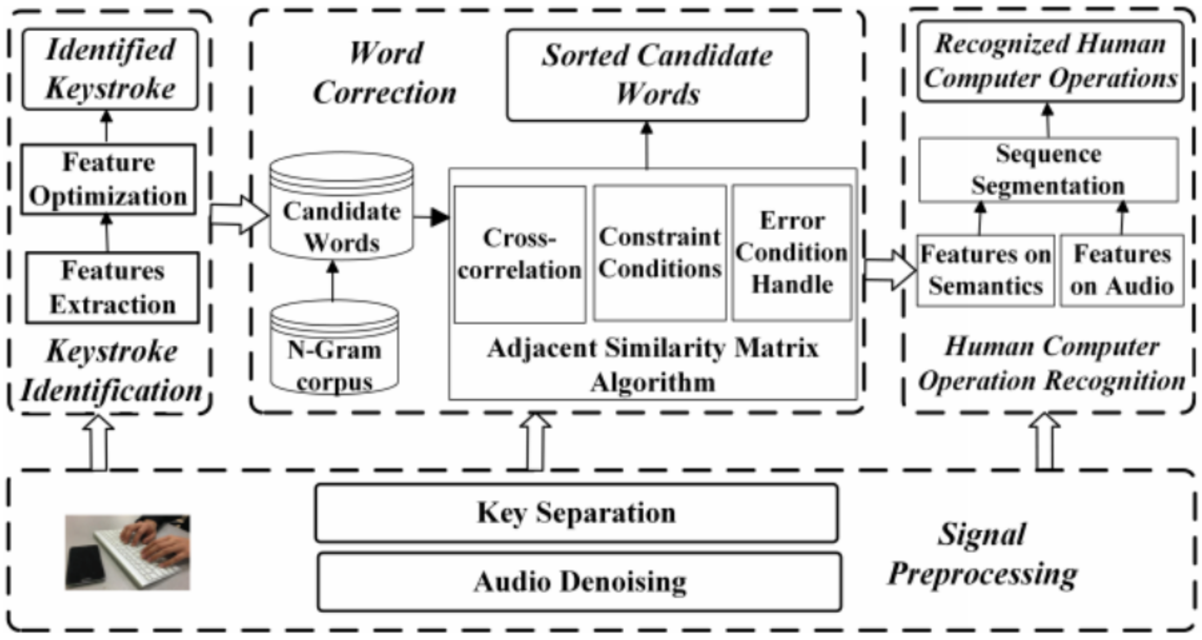


Fig. 3.1: Overview of the proposed approach

performance on a dataset, AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.Problems in machine learning often suffer from the curse of dimensionality each sample may consist of a huge number of potential features (for instance, there can be 162,336 Haar features, as used by the ViolaJones object detection framework, in a 2424 pixel image window), and evaluating every feature can reduce not only the speed of classifier training and execution, but in fact reduce predictive power, per the Hughes Effect. Unlike neural networks and SVMs, the AdaBoost training process selects only those features known to improve the predictive power of the model, reducing dimensionality and potentially improving execution time as irrelevant features need not be computed. A technique for speeding up processing of boosted classifiers, early termination refers to only testing each potential object with as many layers of the final classifier necessary to meet some confidence threshold, speeding up computation for cases where the class of the object can easily be determined. One such scheme is the object detection framework introduced by Viola and Jones:[2] in an application with significantly more negative samples than positive, a cascade of separate boost classifiers is trained, the output of each stage biased such that some acceptably small fraction of positive samples is mislabeled as negative, and all samples marked as negative after each stage are discarded. If 50 percentage of negative samples are filtered out by each stage, only a very small number of objects would pass through the entire classifier, reducing computation effort. This method has since been generalized, with a formula provided for choosing optimal thresholds at each stage to achieve some desired false positive and false negative rate.

In the field of statistics, where AdaBoost is more commonly applied to problems of moderate dimensionality, early stopping is used as a strategy to reduce overfitting.[3] A validation set of samples is separated from the training set, performance of the classifier on the samples used for training is compared to performance on the validation samples, and training is terminated if performance on the validation sample is seen to decrease even as performance on the training set continues to improve.

3.1 Keystroke identification

When we type different keys on the keyboard, distinct audio signals are generated. For example, the sound of typing the space bar can be easily distinguished from other keystrokes. Generally, these differences are caused by various reasons, e.g., the unique structures of some keys, the surrounding context of the keys in the keyboard (i.e., the structures of ambient keys), and the distance between the key and human ears. Therefore, using fingerprint identification, we can utilize acoustic features to differentiate different keystrokes. A keystroke consists of two stages: press and release. If the release of a key overlaps with the press of the next key, it is difficult to separate them. Therefore, to better address keystroke identification, we make two assumptions.First, there is no input from key combinations, e.g., ctrl+O. Second, a key is pressed only when the previous one has been released.

In this paper, we identify keystrokes by utilizing the single microphone embedded in smartphones. The audio waveform of the keystroke is short with bursts. When a key is tapped continuously, there are frequent fluctuations in the waveform, resulting in multiple peaks in zero-crossing rate (ZCR)[4]. However, the effects of noises are obvious that also generate peaks in ZCR. Also, the sound vibration of a keystroke in push and release stages are evident. As shown in Fig. 3.2, there is a silence period between the push and release.Similarly, there are also two stages when we press a key and do not release it, i.e., touch and hit. When we touch the surface of the key and then press down, it shows two peaks with a small silence period in the audio waveform of the push. We will exploit the characteristics of the audio waveforms in different stages to recognize keystrokes.

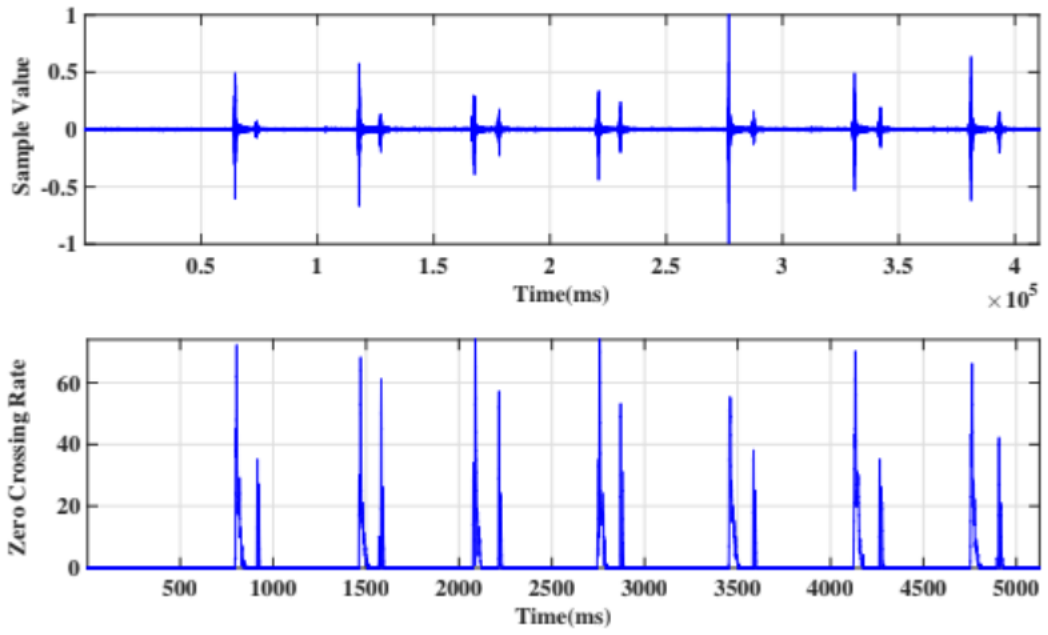


Fig. 3.2: Audio waveform and zero crossing rate when ’V’ is pressed several times

ZCR: The zero-crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back.This feature has been used heavily in both speech recognition and music information retrieval, being a key feature to classify percussive sounds. In the context of discrete-time signals, a zero crossing is said to occur if successive samples have different algebraic signs. The rate at which zero crossings occur is a simple measure of the frequency content of a signal. Zero-crossing rate is a measure of number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero. Speech signals are broadband signals and interpretation of

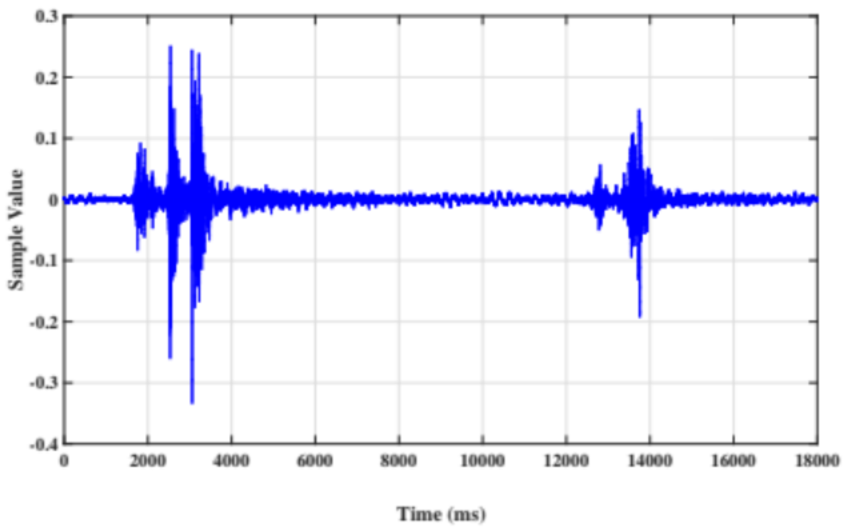


Fig. 3.3: Audio waveform of one keystroke

average zero-crossing rate is therefore much less precise However, rough estimates of spectral properties can be obtained using a representation based on the shorttime average zero-crossing rate . Audio waveform of one single keystroke is given in fig 3.3.

3.1.1 Data processing

To eliminate the effects of noises on ZCR and obtain clear features ofthe keystrokes, we first filter the original audio signals by utilizing the Wiener filtering algo-rithm.This is clearly explained in the figure 3.4. We then adopt endpoint detection to split each keystroke event, a widely used approachin speech detection. In particular, the audio of a keystroke usually has a short duration with highvibration frequency and large amplitude, leading to clear peaks in shortterm energy and short-time zero-crossing rate. Therefore, we use short-term energy and short-time zero-crossing rateas thresholds to achieve double-threshold endpoint detection.

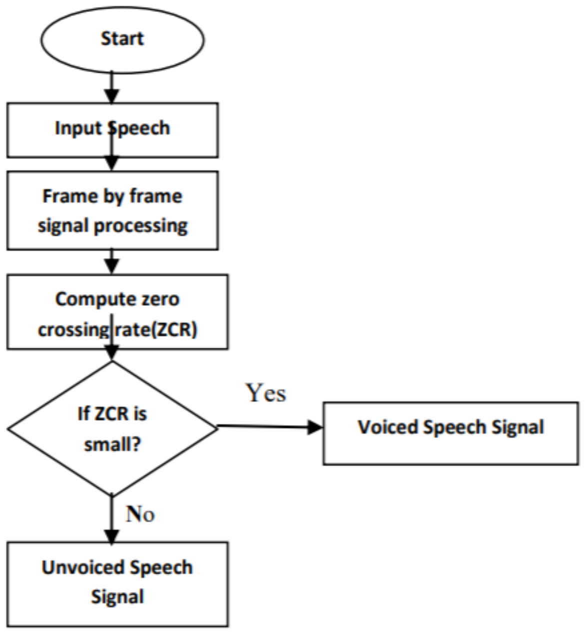


Fig. 3.4: The analysis for classifying the voiced/unvoiced parts of speech

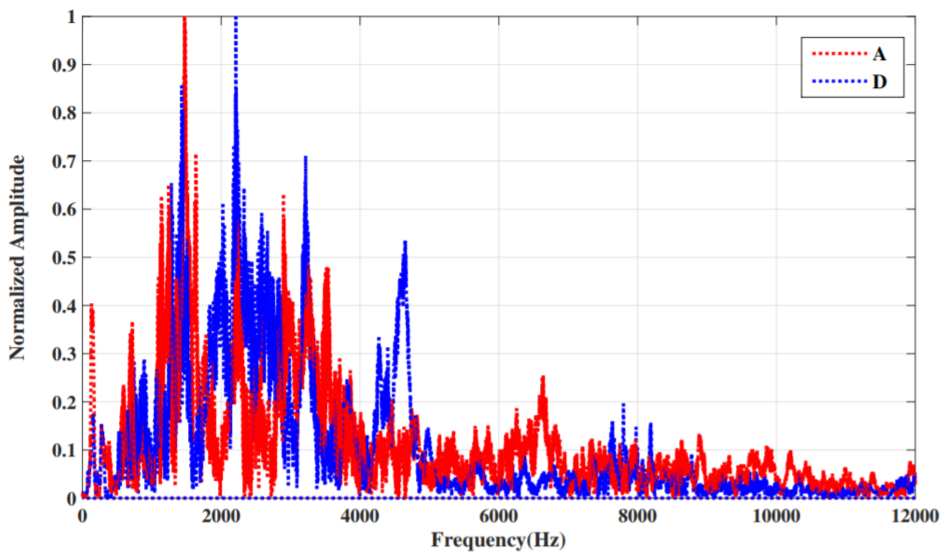


Fig. 3.5: Spectrum of keystrokes ’A’and’D’

3.1.2 Keystroke identification

Audio waveform spectrums corresponding to different keystrokes are distinct. Fig 3.5 presents the spectrums of keystrokes A and D, with a frequency range of 0-12000 Hz and the normalized energy. We observe that the frequency bands where the peaks of energy locate are distinct for A and D between 0 and 8000 Hz. Moreover, when the frequency is than 8000Hz, the energy is very low and the difference between energy peaks is not obvious. In other words, there is less information in the high frequency area. Therefore, we utilize the features of energy peaks in the range of 0-8000Hz to recognize keystrokes.

We first obtain the spectrum of each keystroke audio by applying Fast Fourier transform (FFT)[5], where the frequency range is set to 0-8000Hz. In particular, a 2ms Hamming window[6] is applied in FFT. Afterwards, a frequency window of 20 Hz is used to analyze the energy in different frequency bands. Specifically, the spectrum is divided into 400 segments, and we compute the average energy as the feature in each segment. In addition, the features are extracted in different stages of a keystroke respectively, i.e., the push and the release. Using the 400 features in frequency domain, we apply a decision tree algorithm to recognize which key is pressed for a specific keystroke audio.

3.2 Word correction

There may exist errors in keystroke identification, e.g., a letter in a word is not correctly identified, which may lead to accumulated errors for determining the input series of human computer operations. Therefore, we further refine the identified input series via word correction. Generally, words can be segmented by the space bar or Enter key, which can be accurately identified. We then map each separated letter sequence to the most appropriate word in a candidate word set to improve the results of keystroke identification.

N-gram In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the items are words, n-grams may also be called shingles.An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (n 1)order Markov model. n-gram models are now widely used in probability, communication theory, computational linguistics (for instance, statistical natural language processing), computational biology (for instance, biological sequence analysis), and data compression. Two benefits of n-gram models (and algorithms that use them) are simplicity and scalability with larger n, a model can store more context with a well-understood spacetime tradeoff, enabling small experiments to scale up efficiently.Syntactic n-grams are n-grams defined by paths in syntactic dependency or constituent trees rather than the linear structure of the text.For example, the sentence ”economic news has little effect on financial markets” can be transformed to syntactic n-grams following the tree structure of its dependency relations: news-economic, effect-little, effect-on-markets-financial.

Syntactic n-grams are intended to reflect syntactic structure more faithfully than linear n-grams, and have many of the same applications, especially as features in a Vector Space Model. Syntactic n-grams for certain tasks gives better results than the use of standard n-grams, for example, for authorship attribution. Another type of syntactic n-grams are part-of-speech n-grams, defined as fixed-length contiguous overlapping subsequences that are extracted from part-of-speech sequences of text. Part-of-speech n-grams have several applications, most commonly in information retrieval.Using Latin numerical prefixes, an n-gram of size 1 is referred to as a ”unigram”; size 2 is a ”bigram” (or, less commonly, a ”digram”); size 3 is a ”trigram”. English cardinal numbers are sometimes used, e.g., ”four-gram”, ”five-gram”, and so on. In computational biology, a polymer or oligomer of a known size is called a k-mer instead of an n-gram, with specific names using Greek numerical prefixes such as ”monomer”, ”dimer”, ”trimer”, ”tetramer”, ”pentamer”, etc., or English cardinal numbers, ”one-mer”, ”two-mer”, ”three-mer”, etc.

3.2.1 N-gram based candidate word set determination

With the identified keystrokes, the candidate word set is determined based on fuzzy matching, depending on the distance between the identified letter sequence and the word in corpus. Hence, the metric for word distance has great impact on the matching accuracy.

Popular metrics include Cosine similarity, edit distance, etc. In this paper, we utilize the N-Gram distance in N-Gram model and propose NRNGDist distance as the metric.

N-Gram model is widely used to predict or evaluate the validity of a sentence based on some corpus in natural language processing. It can also evaluate the difference of two strings, which is a popular method in fuzzy matching. For a word s of length M, N-Gram model splits it into a list of substrings, each of which has N letters and is called a Gram. With an offset of one letter, M N 1 Grams can be generated. The N-Gram distance between two words is defined as the number of same Grams. This method can effectively measure the distance of two words that have the same lengths. However, for words with different lengths, it is not satisfactory. For example, the N-Gram distance of boy and boyfriend is equal to the number of Grams that boy has, but we cannot conclude that the two words are the same.

To measure the distance of words with different lengths, we propose non-repeating word segmentation based N-Gram Distance, named NRNGDist. It not only considers the number of same Grams, but also takes into consideration the difference of lengths. A higher similarity of two words corresponds to a smaller NRNGDist. If the value of NRNGDist is zero, the two words are the same. For the input word or letter sequence iw, if the NRNGDist between iw and the word cw in corpus is smaller than a distance threshold , then cw is selected as a candidate word. Note that we put the original input letter sequence in the candidate set , as iw can be a new word that are not in the corpus.

$$NRNGDist = |GN(s)| + |GN(t)|2|GN(s)GN(t)| \qquad \text{(Equ:3.1)}$$

3.2.2 Word recognition by using adjacent similarity matrix algorithm

Word is recognized using adjacent similarity matrix algorithm. As the N-Gram based candidate set can be quite large, we further introduce several constraints to reduce the set. We observe that the audio of a keystroke is closely related to the keys physical position and the audio waves are similar when pressing the same or physically close keys. Therefore, by utilizing the input audio, we can obtain the position similarity of letters in a word, and use such position similarities as the constraint. We first use cross-correlation to describe the relations between any two letters in a word w, which is widely adopted in acoustic distance measurement. Based on the time series of audio from the input word that contains N letters, a correlation matrix has obtained.However, the correlation is hard to reach 1, even if the same key is hit twice. Therefore, it is difficult to determine the relation of keys from cross correlation. What exacerbates the problem is that an error condition may lead to incorrect recognition. To address these problems, we propose an adjacent similarity matrix algorithm.

```
Input: Candidate word set $\Psi = l_1, l_2, ..., l_z$, and audio signal
    of word $l = k_1 k_2 ... k_n$
Output: The word $l_{goal}$ satisfying maximum constraint sub-
    sets
 1: for each $k_i \in l$ do
 2:     $Rank(i, j) \Leftarrow Xcorr$;
 3: end for
 4: $\Pi \Leftarrow Rank(i, j)$;
 5: set $p$ and $m$;
 6: $\Gamma \Leftarrow \Lambda, \Pi$;
 7: $W \Leftarrow \Psi, \Pi$;
 8: $\Omega = (\omega_1, \omega_2, ..., \omega_k) = Transform(\Gamma \times W)$;
 9: $maxm = 0$;
10: for $\omega_i$ $in$ $\Omega$ do
11:     if $||\omega_i||_0 > maxm$ then
12:         $maxm = ||\omega_i||_0$;
13:         $goal = i$;
14:     end if
15: end for
16: return $l_{goal}$
```

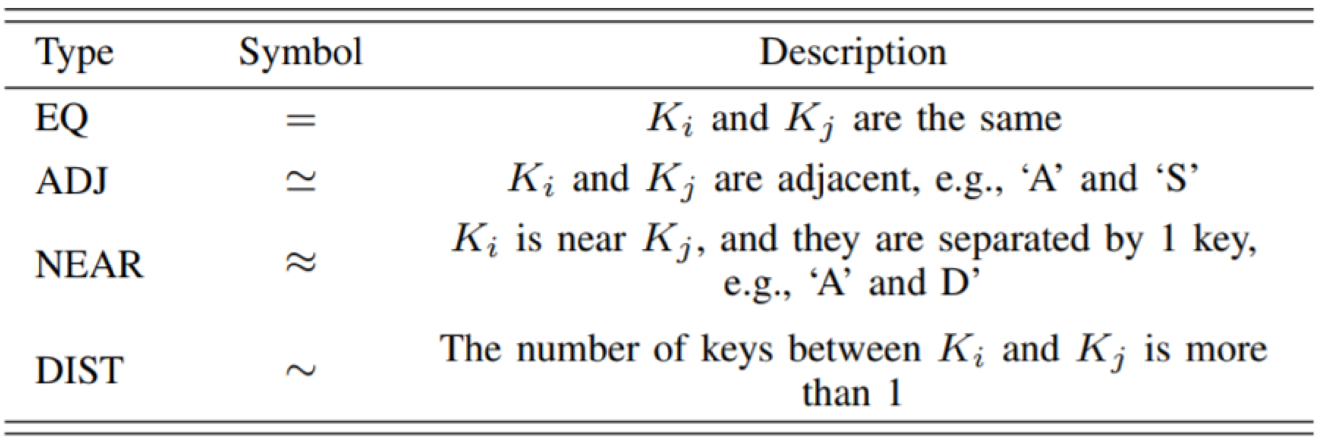
Specifically, based on the correlation matrix Xcorr, we redefine the relation of two keys as follows. The diagonal values Simii representing autocorrelation are set to 0 to avoid interference. For key Ki , we first sort the similarity between Ki and other keys in descending order. Rank(i, j) denotes the ranking of Simi,j for Ki . If Rank(i, j) is higher for Ki and Rank(j, i) is higher for Kj , i.e., Ki and Kj are very similar, then there is a high possibility that Ki is the same as Kj . Utilizing these rankings, we redefine the relations between two keys using. The descriptions about the relation symbols Table. The two keys are closer when the sum of Rank(i, j) and Rank(j, i) is smaller. This definition is not directly affected by the correlation coefficient, so it can effectively reduce errors. In fact, the word that satisfies all the constraints is usually not unique. And the word nose and help also satisfy this set. This is why we first determine the candidate word set based on N-Gram.

$$Rank(i,j) + Rank(j,i) 3 Ki = Kj \qquad \text{(Equ:3.2)}$$

$$Rank(i,j) + Rank(j,i) = 4Ki'Kj \qquad \text{(Equ:3.3)}$$

$$Rank(i,j) + Rank(j,i) = 5KiKj \qquad \text{(Equ:3.4)}$$

$$Rank(i,j) + Rank(j,i) 6KiKj \qquad \text{(Equ:3.5)}$$



Type	Symbol	Description
EQ	$=$	K_i and K_j are the same
ADJ	$\simeq$	K_i and K_j are adjacent, e.g., ‘A’ and ‘S’
NEAR	$\approx$	K_i is near K_j, and they are separated by 1 key, e.g., ‘A’ and D’
DIST	$\sim$	The number of keys between K_i and K_j is more than 1

For a word consisting of N letters, the number of constraints and constraint subsets are

$$N(N1)/2 \qquad (3.1)$$

We refine the word constraint via sampling all constraint subsets m times. Following the equal probability principle, a constraint subset will be selected by one sampling and not being put back. After m sampling, m subsets are obtained, each of which has

$$N(N1)/2p \qquad \text{(Equ:3.6)}$$

-conditions. The values of p and m are empirically determined. Using the obtained m condition subsets, we analyze the times that the same word is selected, i.e., the number of subsets that the word satisfies. The more frequently selected word is more likely to be the right one. By combining the constraint matrix with the constraint-word matrix W, we can obtain the relations between m constraint subsets and the candidate words, which is defined as a matrix.With this matrix, we can derive the selected times of the same word, which corresponds to the number of the value 1 in each column. Finally, the one with the most selected times is regarded as the correct word.

3.3 Human computer operation recognition

Based on the corrected word series, we characterize and recognize a users computer operations. During a period of time, the audio of a sequence of keystrokes can be from various operations, e.g., chatting, writing documents, and playing games. The inputs of these human computer operations have distinct features in semantics. For instance, U represents you in chatting, while this abbreviation is uncommon in documents. The audio wave are also distinguishable. For example, the intensity of pressing keys is stronger when playing games. Therefore, we utilize these characteristics in audio wave and semantics to recognize four common human computer operation, i.e., chatting, coding, writing documents, and playing games.

3.3.1 Semantic featuers

1) Lexicon Preference: Different human computer operations have distinct lexicons, e.g., the keywords are more common in coding, and the operation keys such as Q and W are widely used when playing games. Based on the differences of the input words, we extract the lexicon preference to characterize different human computer operations. In this work, we first define the identification set for each human computer operation respectively, based on the

Corpus of Contemporary American English2 . Specifically, the identification set of chatting is formed by the abbreviations commonly used in chatting (represented as 1), the identification set of writing documents is composed of the most frequently used 50 words (represented as 2), the identification set of playing games (represented as 3) contains the operation keys commonly used in games, such as Q, W, E, and R, and the identification set of coding consists of the keywords and reserved words (represented as 4). Afterwards, the Dice coefficient is used to measure the distance between the inputseries S and the specific identification set of each human computer operation, as defined in . Thus, the preferences of S for each operation can be obtained, i.e., D1, D2, D3, and D4. 2) Semantic Rationality: In general, the inputs of documents are more rational than those of coding in semantics. Similarly, the inputs when playing games seems random, while the words for chatting are more casual than documents [7]. Therefore, we propose semantic rationality to describe human computer operations. In particular, we apply N-Gram model to evaluate the semantic rationality. In natural language processing, a sentence S is composed of any N words in any orders, and the occurrence probability of the sentence, P(S), is then regarded as the semantic rationality by this equation:

$$P(w_i|w_1, w_2, ..., w_{i-1}) = P(w_i|w_{i-N+1}, w_{i-N+2}, ..., w_{i-1})$$

$$P(w_1, w_2, ..., w_m) = \prod_{i=1}^{m} P(w_i|w_{i-1}) = \prod_{i=1}^{m} \frac{P(w_i w_{i-1})}{P(w_{i-1})}$$

3.3.2 Acoustic features

Acoustic features are significant for the recognition of human computer operation, which can be extracted from two aspects. First, the input rate is effective for recognition, e.g., compared with programming, chatting is more casual and often done spontaneously, hence more likely to have a higher inputrate. Second, the energy of the audio, which can represent the power of pressing keys, is another piece of useful information. For example, when the users are playing games, they tend to press keys with more power. Therefore, we adopt the input rate and the audio energy to characterize the audio signals. m is the number of letters in the input series S, and t is the time length of S.

$$v = E/T \tag{Equ:3.7}$$

3.3.3 Human computer operation recognition

An input sequence may consist of multiple human computer operations, i.e., the entire time period can be divided into multiple non-overlapping time segments and each segment is for one operation. Therefore, to accurately recognize users computer operations, we need to segment the input series and recognize each sub-operation. We recognize human computer operations based on the operation segmentation.

We first use a sliding window for time division. Specifically, to avoid breaking an input sentence, we regard each word as a point. In such a way, the audio segment st in each time window is composed of multiple words. Based on the four features we can use a feature vector f t = (DV, P, v, E) to represent st. If two segments belong to the same human computer operation in consecutive time slots, then their features are more similar than those of other segments. Therefore, human computer operations can be differentiated based on the similarity of adjacent feature vectors[8].

To obtain each operation segment, we propose a slidingwindow slicing algorithm based on the feature vector difference. In case that the time window is 3, the input series

$$X = w1, w2...wn \tag{Equ:3.8}$$

can be represented as a set of triples and each triple corresponds to a feature vector f ti = (DV, P, v, E). Therefore, by setting the sliding step as 1, X can be denoted using the feature vector series f t1f t2...f tn2. In other words, the segmentation of input series can be achieved by slicing the feature vector series. As we have illustrated, if the feature vectors in consecutive time belong to the same human computer operation, the distance between them would be small. The distance of adjacent feature vectors can be measured by utilizing cosine similarity and Tonimoto coefficient[9] Based on the distance between feature vectors, we can obtain the operation segment

$$SimC_i(ft_i, ft_{i+1}) = \frac{ft_i \cdot ft_{i+1}}{\|ft_i\| \|ft_{i+1}\|}$$

$$SimT_i(ft_i, ft_{i+1}) = \frac{ft_i \cdot ft_{i+1}}{\|ft_i\|^2 + \|ft_{i+1}\|^2 - ft_i \cdot ft_{i+1}}$$

of the input series. An operation segment with a length of n will have n2 f t, and the average of f ti is defined as the feature of this operation segment to be used for human computer operation classification. In this paper, we select three classification models (i.e., AdaBoost, SVM, and ZeroR) to infer the type of each segment of human computer operation. AdaBoost is an adaptive boosting machine learning algorithm, which converges to a stronger learner by learning from the results of weak learners. SVM models the examples as points in space, and the examples of the separate categories are divided by a clear gap that is as wide as possible. ZeroR regards the majority classes as the classification results based on statistical rules of historical data. We will compare the performance of these three inference models in the experiments.

3.4 Perfomance evaluation

3.4.1 Experimental setup

Two types of keyboards are used in our experiments, i.e., the mechanical keyboard 87 Key Dareu, and the membrane keyboard Rapoo E1050. We utilize an iPhone 5s and use the single microphone to record the keyboard inputs. The sampling rate of microphone is 44.1 KHz. In the experiments, the phone and the keyboard are placed side by side as shown in figure.3.6.

Experimental results for keystroke identification are discussed now.1) Performance of

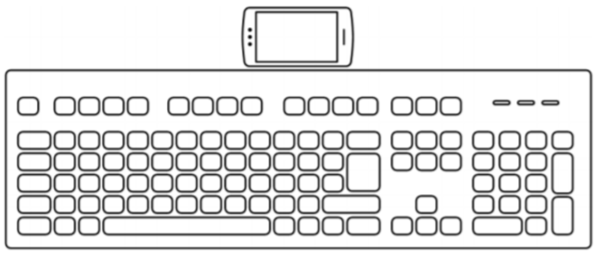


Fig. 3.6: Equations for distance of feature vector

Data Preprocessing: The Wiener filter can effectively remove the noise of the silence periods, which makes the featdouble-threshold detection method to detect the ending points of keystrokes, and the used parameters.. The uses of keystrokes more clear. As aforementioned, we adopt the results illustrate that double threshold detection method cannot only obtain the start and end points of each keystroke, but also label the two stages, i.e., push and release, without being affected by the peaks of Touch-Hit.

affected by the peaks of Touch-Hit. 2) Performance of Keystroke Identification: In this work, keystrokes are identified by utilizing 400 features in frequency domain, which are distributed in 0-8000Hz. The effectiveness of these features are first evaluated in the experiments. We extracted features from 25 keystrokes to validate their effectiveness.Features from the same keystrokes are similar, and different keystrokes have distinct features. Therefore, the extracted frequency features are able to effectively distinguish keystrokes. The identification method is evaluated by using two types of keyboards, mechanical keyboard and membrane keyboard. We recorded the audio waveform of 20 keystrokes from each letter and also commonly used symbol keys. Three classifiers, i.e., J48, NaiveBayes[9], and SVM, are applied by utilizing the frequency features from only the push, the release, or both the push and release stages. Experimental results indicate that for each classification algorithm, the features in push have better performance than those in release, indicating that features during the push stage carry more information than that of the release stage. The reason might be that the intensity of push is stronger, and the audio of the release is not obvious enough. Specifically, by using features from both stages, it achieves the best performance. Therefore, we conclude that different keystrokes have distinct frequency bands where the peaks appear. In particular, the results of SVM are similar when using the features from the push-release and the push stage. J48 algorithm achieves an accuracy of 95percentage, and outperforms other methods .

Further, for the mechanical keyboard and membrane keyboard, we use J48 to identify keystrokes in different stages. .Compared with the membrane keyboard, features in the push-release stage and only release stage obtain much higher accuracies when using the mechanical keyboard, while using features from the push stage, the results are similar for two keyboards. The reason might be that the structure of the mechanical keyboard is more complex than membrane keyboard. In particular, the feedback for a key release is obvious when using the mechanical keyboard, whereas membrane keyboard does not have the effects.Sample results of filter is shown in fig 3.7.

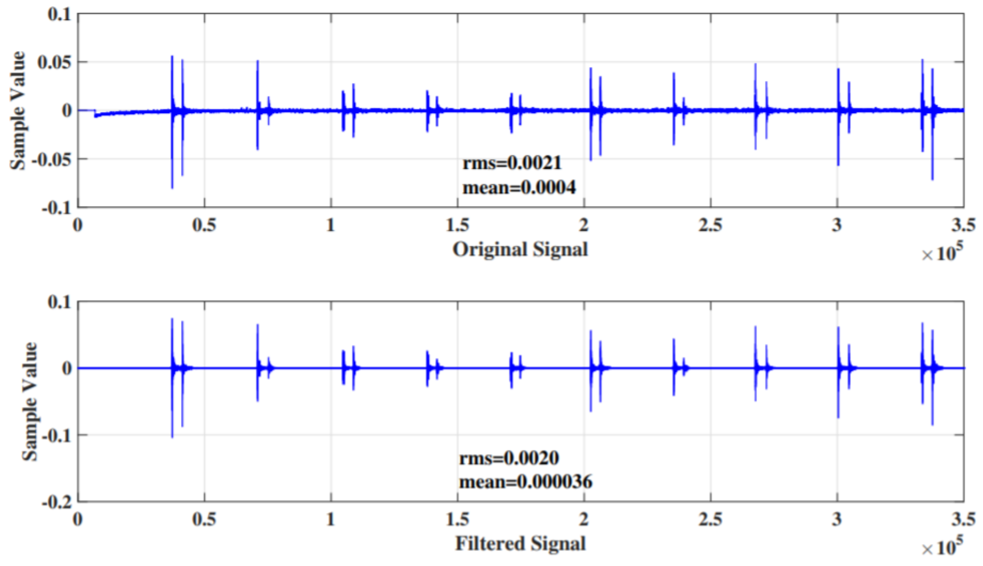


Fig. 3.7: Sample results of filter: original audio waveform and audio waveform after filtering

experimental results for word correction is discussed now.1) N-Gram based Candidate Word Set Determination: The candidate word set is first selected from a corpus based on theNRNGDist distance. We use the most frequent 5000 words in the Corpus of Contemporary American English as our corpus. For words with more than 2 letters, we first compute their Grams based on 2-Gram and 3-Gram. Short words with only one or two letters are not

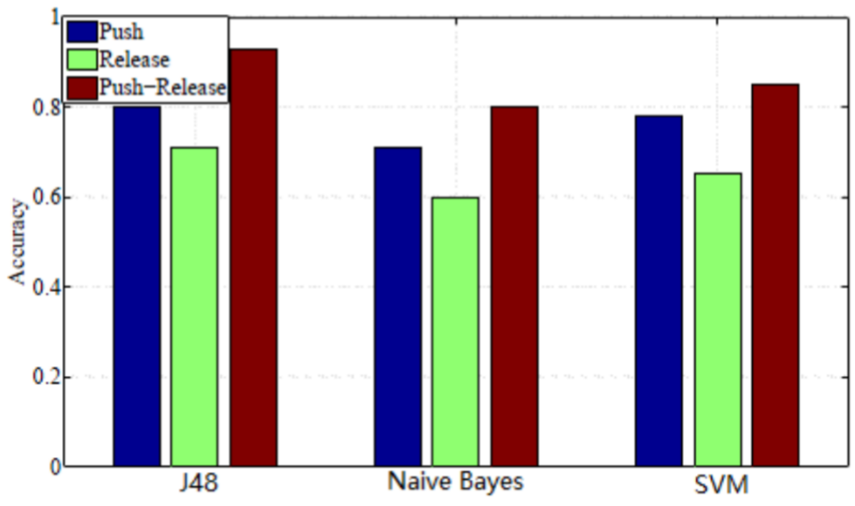


Fig. 3.8: Keystroke identification results by utilizing the features from different stages

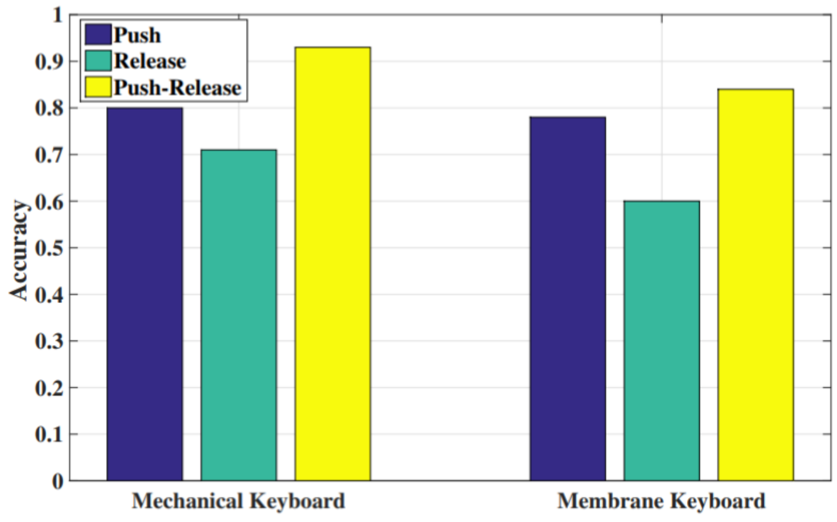


Fig. 3.9: Keystroke identification results by using two types of keyboards.

considered for two reasons. First, the words with one letter are difficult to correct, as there is less related information. Second, the words with two letters will generate only one constraint, thus if the constraint is inaccurate, the correction can not be achieved. Moreover, these short words can not greatly impact the final performance of human computer operation recognition. For words in the input series, they can be segmented based on the space bar, Enter key, and punctuations. The space bar and Enter key are recognized with 100 percent accuracy, while the punctuations, such as ,, ., and ;, can not always be accurately identified. Therefore, we slightly revise the Grams to eliminate the effects of punctuations, called R-Gram. Specifically, if the punctuation is in the middle of the word, we add two Grams; if it is the first or the last letter, the Grams remain unchanged.

2) Results of Word Recognition using Adjacent Similarity Matrix Algorithm: To obtain the constraints of the input words, we first compute the cross-correlation. In the experiments of keystroke identification , it has been demonstrated that the audio in push-release has abundant information. Thus the cross-correlation of the letters in a word is also obtained from both stages, i.e., the push and the release. Specifically, the audio waveform of each letter is divided into two stages, and then we compute the cross-correlation of any two letters in push and release respectively as shown in figure 3.8 and figure 3.9.

Experimental results for human computer operation recognition is dicussed now.To evaluate the performance of human computer operation recognition, we collected data of 15 users with different input modes, i.e., fingering input and single finger input. Fingering input as a common mode has the following characteristics. The input rate is faster, and touch-typing can be done with fingering skills and good keyboard habits. The fixed keys are usually pressed by the same finger and intensity. Single finger input refers to the mode used by people who are not good at typing, which can present the following characteristics. The keyboard habits are worse, and the keystrokes are mostly performed by the forefinger. Also, the input rate and intensity are distinct for the keys.

Five volunteers are students and five are office workers, who can use fingering input. For them, coding, chatting, writing documents, and playing online games are typical activities in their daily lives. Meanwhile, other volunteers are people who can only type with single finger input. They prefer to chat and play games when they are using computers. In the experiment, for each user, the same human computer operation occupies several time segments and is not completed in consecutive time, i.e., four human computer operations are mixed in each input series. We collected 200 time series of each operation in each input mode, i.e., 800 samples from fingering input and 800 samples from single finger input in total. We evaluated the performance of the recognition method as well as the proposed features, i.e., lexicon preference, semantic rationality, input rate, and audio energy.

1) Evaluation of Features: A human computer operation is characterized by four features, i.e., lexicon preference (denoted as L P), semantic rationality (denoted as S R), input rate (denoted as I R), and audio energy (denoted as A E). By using the collected data of four operations, we first illustrate the features distribution by computing the average, as shown in Table IV. In specific, lexicon preference is indicated by a four-dimensional vector, each dimension of which represents the distance between the input audio and one type of human computer operations. Semantic rationality is computed by the probability that the input words occur simultaneously in asentence. It is observed that there is significant difference among the four operations. We then use a J48 algorithm to evaluate the performance of these features under different input modes using 10-fold cross validation. Experimental results indicate that when the input mode changes, the four features have distinct classification performance. For the fingering input, the performance of input rate outperforms other three features, achieving a 74percentage accuracy. In case of the single finger input, the highest accuracy (73percent) is achieved with lexicon preference. To sum up, for the two types of input modes, the performance of input rate is quite different, while those of the other features are relatively stable.

We further analyze the performance of the features in more detail. By using one of the features respectively, the confusion matrices of the classification results are obtained.. The item in the table presents the results of two input modes, e.g., 54percent

55 percent means 54 percent samples are recognized when using fingering input, and 55 percentage samples are recognized when using single finger input. Based on the results, playing games can be easily recognized by utilizing any of the features. Writing documents is hard to be distinguished from chatting. In addition, the differences between two types of

human computer operations (i.e., writing documents and chatting) and other two types of human computer operations (i.e., coding and playing games) are quite significant. As for the performance of each feature, lexicon preference has outstanding results for coding characterization, and semantic rationality makes playing games distinguishable. The performance of input rate is similar for each human computer operation. Furthermore, from the aspect of the input mode, the features on semantics perform better in the experiments of single finger input, while the features on audio signals achieve more satisfactory classification results in the fingering input mode.

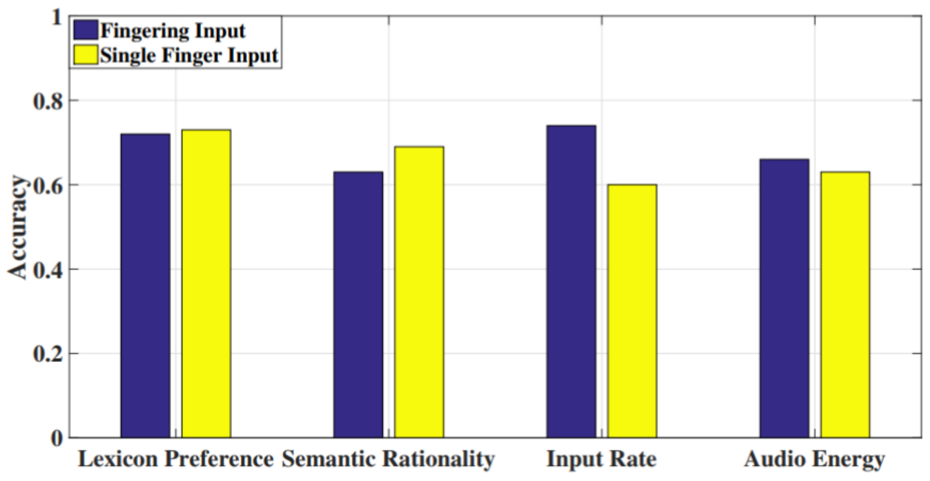


Fig. 3.10: Recognition results of human computer operation by using different features

2) Results of Human Computer Operation Recognition: We first obtain segments of human computer operations based on the feature vector similarity between adjacent segments.Cosine similarity and T onimoto coefficient are used to measure the distance of features. With the data of four human computer operations from single finger input and fingering input, we evaluate the segmentation results of the two metrics. If the error of the human computer operation segmentation is less than 4 words, the segmentation is considered correct. The results are illustrated in figure 3.10 where 200 segments are used for each human computer operation. Overall, Cosine similarity outperforms the Tonimoto coefficient. Based on the results of operation segmentation, we recognize human computer operations by applying three classification

models, including J48 based AdaBoost algorithm, ZeroR, and SVM. According to the experimental results shown in Fig. 12, by using ZeroR algorithm or SVM, the recognition results of fingering input are similar to those of single finger input.

Conclusion

This paper achieves human computer operation recognition by utilizing the mobile microphone. Specifically, we record the acoustic signals from users keyboard by using the microphone in the smartphone, and the keystrokes are first identified by utilizing the features in frequency domain. We propose an adjacent similarity matrix algorithm to correct the words. Finally, by extracting features on semantics and audio signals, users computer operations are determined, including chatting, coding, writing documents, and playing games. Experiments are conducted by using different keyboards and input modes, and the results validate the effectiveness of the proposed approach.

In the future, we plan to improve the generality of the algorithm. In addition, more realistic factors, such as the input of combination keys, will be considered. Meanwhile, we plan to utilize more available sensors for context-awareness and improving the accuracy of human computer operation recognition.

References

[1] K. Marshall, Working with computers, Perspectives on Labour and Income, vol. 22, no. 5, 2001.

[2] E. A. Boyle, T. M. Connolly, T. Hainey, and J. M. Boyle, Engagement in digital entertainment games: A systematic review, Computers in Human Behavior, vol. 28, no. 3, pp. 771780, 2012.

[3] B. Schilit, N. Adams, and R. Want, Context-aware computing applica- tions, in Mobile Computing Systems and Applications, 1994. WMCSA

1994. First Workshop on. IEEE, 1994, pp. 8590.

[4] M. Griffiths, Does internet and computer addiction exist? some case study evidence, CyberPsychology and Behavior, vol. 3, no. 2, pp. 211 218, 2000.

[5] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, Sensor- based activity recognition, IEEE Transactions on Systems, Man, and

Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 790 808, 2012.

[6] B. Guo, Z. Yu, L. Chen, X. Zhou, and X. Ma, Mobigroup: Enabling lifecycle support to social activity organization and suggestion with mobile crowd sensing, IEEE Transactions on Human-Machine Systems, vol. 46, no. 3, pp. 390402, 2016.

[7] R. Piyare, Internet of things: ubiquitous home control and monitoring system using android based smart phone, International Journal of Internet of Things, vol. 2, no. 1, pp. 511, 2013.

[8] A. Mehrotra, R. Hendley, and M. Musolesi, Prefminer: mining users

preferences for intelligent mobile notification management, in UBI- COMP16. ACM, 2016, pp. 12231234.

[9]Z. Yu, H. Xu, Z. Yang, and B. Guo, Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints, IEEE Transactions on Human-Machine Systems, vol. 46, no. 1, pp. 151158, 2016.